Moving toward the intra-protocol de-ossification of TCP in mobile networks: Start-up and mobility

Author: Eneko Atxutegi
Supervisor: Fidel Liberal

January 2018

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Moving toward the intra-protocol de-ossification of TCP in mobile networks: Start-up and mobility

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Telecommunications Engineering

by

Eneko Atxutegi Narbona

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

University of the Basque Country (UPV/ EHU)

Jan 2018

Abstract

Mobile Broadband (MBB) usage has raised significantly over the last years and is expected to continue growing due to the inclusion of future 5G capabilities with unprecedented peak rates and low end-to-end latencies. However the achievement of such capacities and low delays is upperbounded by the management and performance of transport protocols. Thus, inefficient transport protocols may prevent communications from taking full advantage of the available radio resources. In this regard, TCP is still the most used transport protocol and its different congestion control algorithms (CCAs) carry the performance responsibility. While originally these CCAs have been implemented to face distinct use-cases in fixed networks, none of them has been designed to face the variability of mobile networks in terms of delay and throughput under a variety of network circumstances in an easily deployable way. Since the analysis of TCP over MBB is complex due to the different factors affecting the performance, we focus on two significant and generalized use-cases in MBB: mobility of users as a representation of the main feature of MBB in comparison to fixed networks and the performance of Start-up phase of TCP due to the presence of a majority of short flows in current communications.

Several works have addressed the importance of gaining more flexibility in the transport layer, building transport services on top of substrate TCP or UDP. However, these proposals have encountered the same limitations caused by the architectural dependencies of substrate protocols (e.g. do not allow any change once the transmission has started) and therefore, experiencing an "ossified" transport layer. This PhD work emerges as a response to address the ossification problem and in order to demonstrate that even within the family of TCP (intra-protocol) there is room for improvement proposals that could get partially rid of the performance limitations, introducing a framework to solve this issue by selecting the most appropriate CCA for each condition. To that end, we evaluate and decide the main impact factors in regards to the performance of different CCA in mobility circumstances and during the Start-up phase of TCP under 4G deployments and low latency deployments that move towards the potential delays in forthcoming 5G networks. These impact factors constitute the main heuristics of the proposed framework so as to decide the most suitable CCA. Finally, we validate our proposal in mobility scenarios taking into account that the selection is carried out either at the beginning of the transmission (limited flexibility of the transport layer) or real-time throughout the transmission (with flexible transport layer). It is concluded that the proposal could bring important performance improvements by selecting the best CCA candidate that considers both network status and requirements from the application layer.

Resumen

El uso de las redes móviles de banda ancha ha aumentado significativamente los últimos años y se espera un crecimiento aún mayor con la inclusión de las futuras capacidades 5G. 5G proporcionará unas velocidades de transmisión y reducidos retardos nunca antes vistos. Sin embargo, la posibilidad de alcanzar las mencionadas cuotas está limitada por la gestión y rendimiento de los protocolos de transporte. A este respecto, TCP sigue siendo el protocolo de transporte imperante y sus diferentes algoritmos de control de congestión (CCA) los responsables finales del rendimiento obtenido. Mientras que originalmente los distintos CCAs han sido implementados para hacer frente a diferentes casos de uso en redes fijas, ninguno de los CCAs ha sido diseñado para poder gestionar la variabilidad de throughput y retardos de diferentes condiciones de red redes móviles de una manera fácilmente implantable. Dado que el análisis de TCP sobre redes móviles es complejo debido a los múltiples factores de impacto, nuestro trabajo se centra en dos casos de uso generalizados que resultan significativos en cuanto a afección del rendimiento: movimiento de los usuarios como representación de la característica principal de las redes móviles frente a las redes fijas y el rendimiento de la fase de Start-up de TCP debido a la presencia mayoritaria de flujos cortos en Internet.

Diferentes trabajos han sugerido la importancia de una mayor flexibilidad en la capa de transporte, creando servicios de transporte sobre TCP o UDP. Sin embargo, estas propuestas han encontrado limitaciones relativas a las dependencias arquitecturales de los protocolos utilizados como sustrato (p.ej. imposibilidad de cambiar la configuración de la capa de transporte una vez la transmisión a comenzado), experimentando una capa de transporte "osificada". Esta tesis surge como respuesta a fin de abordar la citada limitación y demostrando que existen posibilidades de mejora dentro de la familia de TCP (intra-protocolar), proponiendo un marco para solventar parcialmente la restricción a través de la selección dinámica del CCA más apropiado. Para ello, se evalúan y seleccionan los mayores puntos de impacto en el rendimiento de los casos de uso seleccionados en despliegues de red 4G y en despliegues de baja latencia que emulan las potenciales latencias en las futuras capacidades 5G. Estos puntos de impacto sirven como heurísticas para decidir el CCA más apropiado en el propuesto marco. Por último, se valida la propuesta en entornos de movilidad con dos posibilidades de selección: al comienzo de la transmisión (limitada flexibilidad de la capa de transporte) y dinámicamente durante la transmisión (con una capa de transporte flexible). Se concluye que la propuesta puede acarrear importantes mejoras de rendimiento al seleccionar el CCA más apropiado teniendo en cuenta la situación de red y los requerimientos de la capa de aplicación.

Laburpena

Azken urtetan, banda zabaleko sare mugikorren erabilera igoera nabarmena jasan du eta hurrengo urtetan, 5G kapazitateen barneratzearekin batera, igoera joera jarraipena izango du. Kapazitate berri hauen artean, aurrekaririk gabeko transmisio abiadurak eta atzerapen motzak aurkitzen dira batik bat. Hala ere, garraio protokoloen kudeaketa eta errendimendua, aipatutako kuotetara heltzeko aukera mugatzen dute. Gaur egun, TCP da erabilitako garrai-protokolo nagusia eta bere kongestio kontrol algoritmoak (CCAk) errendimenduaren oinarrizko arduradunak. Jatorrian, CCA desberdinak diseinatu ziren sare finkoetako erabilera kasu ezberdinei aurre egiteko. Alabaina, CCA-etako inor izan da diseinatua era erraz batean ezartzeko eta sare mugikorren throughput eta atzerapenen aldakortasunei aurre egiteko. Kontutan izanda TCP-aren errendimenduan faktore anitz erlazionatzen direla, gure lana bi erabilera kasu adierazgarrietan bideratzen da: erabiltzaileen mugikortasunnean sare mugikorren ezaugarri nagusi moduan eta TCP-aren Start-up fasearen errendimenduan Interneten fluxu motzek duten presentzia nabarmenagatik.

Lan desberdinek azpimarratu dute garraio-geruzako malgutasunaren beharra, garraio-zerbitzuak TCP-ren eta UDP-ren gainean sortuz. Hala ere, proposamen berri hauek mugaturik aurkitu dute errendimendua substratu bezala erabilitako protokoloen arkitektura mendekotasunagatik (adibidez garraio-geruzako konfigurazioa aldatzeko posibilitatea ezezten behin transmisioa hasita dagoela), "hezurtutako" garraio-geruza esperimentatuz. Tesi honek, aipatutako mugei aurre egiteko eta TCP familiaren barruan (intra-protokoloa) hobekuntzak aurkitzea posible dela frogatzeko, marko bat proposatzen du non dinamikoki sare baldintzei aurre egiteko CCA egokiena aukeratzen den. Helburua lortzeko, aukeratutako erabilera kasuetan errendimendua gehien suntsitzen dituzten faktoreak ebaluatu eta aukeratzen dira. Ebaluazioa eta aukeraketa 4G sareetan eta potentzialki 5G sareetan aurkituko diren atzerapen motzeko baldintzetan gauzatzen dira. Faktore hauek heuristika bezala erabiliko dira gure markoan CCA egokiena aukeratzeko. Azkenik, proposamena mugimendu baldintzetan balioztatzen da bi CCA aukeraketa kontutan izanda: transmisioaren hasieran (garraio-geruza malgutasun mugatuarekin) eta transmisioan zehar dinamikoki (garraio-geruza malguarekin). Ondorioztatzen da, proposamenak hobekuntza nabarmena ekar ditzakeela errendimenduan CCA aukeraketa egokiena egitean non sare egoera eta aplikazio-geruzaren eskakizunak islatzen diren.

Resumen ejecutivo

Los últimos años han sido testigo del aumento en el uso de redes móviles de banda ancha, prácticamente alcanzando entre todas las redes inalámbricas los datos de tráfico transcurrido por las redes fijas. Los pronósticos de crecimiento son igualmente positivos para las futuras redes móviles de quinta generación (5G) cuyos datos auguran una mayor proyección que la actual cuarta generación (4G). Este aumento de las redes móviles y de los servicios ofrecidos está intrínsecamente relacionada con una mayor demanda de calidad de servicio (QoS) por parte del usuario final en términos de mayor velocidad y menor latencia entre otros requerimientos. Originalmente, en redes fijas, la mejora de la capa de transporte (generalmente TCP) posibilitaba una mejor adaptación al medio y obtención de las cuotas de rendimiento fijadas.

Sin embargo, los mecanismos y características inherentes a las redes móviles de banda ancha dificultan la consecución de dicha mejora. Primeramente, es importante destacar que dependiendo de la posición del usuario en la celda provisionadora del servicio móvil, las condiciones del canal radio variaran drásticamente debido a la propagación y fading de señal, estableciéndose unos límites máximos de velocidad. En segundo lugar, los recursos radio de una celda son limitados y compartidos por todos los usuarios finales adheridos a dicha estación base, limitando nuevamente la velocidad alcanzable y creando fluctuaciones de los recursos asignados a cada usuario debido a las políticas de planificación. Esta fluctuación en la capacidad disponible por usuario, dificulta el rendimiento de TCP debido principalmente al alto tiempo de respuesta frente a la variabilidad de las redes móviles.

Además de la variabilidad como característica intrínseca a las redes móviles de banda ancha, el rendimiento de TCP sobre dichas redes se ve afectado por otras dos circunstancias: situaciones de red y TCP críticas para el rendimiento y convivencia de redes móviles y TCP y limitaciones propias de la capa de transporte.

Respecto a las situaciones críticas desde el punto de vista de las redes móviles, se encuentran aquellas situaciones de alta movilidad en la que el canal radio fluctúa aún más por la combinación del cambio de posición en la celda, variación de la propagación y modificación del patrón de fading de señal. Por la parte de TCP, su fase de inicialización o Start-up puede suponer una posible fuente de rendimiento inadecuado debido a la falta de conocimiento del medio que tiene TCP durante esa fase. Los efectos comentados se consideran críticos por el incremento de la variabilidad por una parte y por la incertidumbre de rendimiento por otra.

Respecto a las limitaciones de la capa de transporte, es importante comentar que debido al éxito de TCP, el protocolo es utilizado tanto como protocolo de transporte por defecto o como sustrato para habilitar servicios de transporte. Estos servicios de transporte son mecanismos construidos entre la capa de transporte y de aplicación y que dotan de mayor flexibilidad y libertad en la implementación de nuevas funcionalidades por su desarrollo en el espacio de usuario del sistema operativo. Sin embargo, la propia selección de TCP como sustrato acarrea que la toda la transmisión permanezca adherida a las dependencias infraestructurales de TCP. Este efecto se hace llamar "ossification" (osificación) y conlleva tres grandes limitaciones: 1) Cierra toda oportunidad de selección o modificación de la capa de transporte bien al comienzo de la transmisión y con el flujo en tránsito. 2) Limita la innovación de protocolos de transporte. 3) Provee de una limitada o nula flexibilidad de comunicación a través de una application programming interface (API) entre la capa de aplicación y transporte. Estas limitaciones hacen que la capa de transporte no pueda representar las singularidades que se reflejan desde el nivel de aplicación. Con todo ello y teniendo en cuenta además la variabilidad del acceso radio, TCP actualmente rinde muy por debajo de las capacidades ofrecidas en las redes móviles de banda ancha actuales debido a la imposibilidad tanto de reflejar los requerimientos del nivel de aplicación como de adaptarse a las fluctuaciones de la red celular.

A pesar del uso generalizado del algoritmo de control de congestión (CCA) por defecto en TCP, actualmente existen más de 20 implementaciones diferentes. A lo largo de los años, cada implementación nueva ha tratado de combatir los nuevos casos de uso en entornos de red, cubriendo entre todas las implementaciones una muy diversa casuistica. Cada implementación gestiona de manera diferente las pérdidas a nivel de paquete, los retardos o el tráfico cruzado entre otras características. De esta forma, cada CCA es más proclive a obtener mejor rendimiento bajo ciertas condiciones de red y comportamiento del flujo de nivel de aplicación. Por tanto, nuestro estudio y propuesta busca encontrar los mejores candidatos de TCP para gestionar la transmisión en las situaciones críticas anteriormente mencionadas de alta movilidad y rendimiento durante la fase de Start-up. Nuestro trabajo ayuda por tanto en las tareas de de-osificación de la capa de transporte mediante la apropiada selección de la implementación de TCP capaz de gestionar de mejor manera el rendimiento sobre las condiciones de contorno (red y requerimientos de aplicación) de cada momento. Dado que la selección de realiza entre las diferentes implementaciones de TCP, se dice que la de-osificación es intra-protocolar. Para llevar a cabo el análisis de TCP en entornos de alta movilidad y durante la fase de Start-up y seleccionar y validar la selección del CCA más apropiado, la tesis se divide en varios capítulos que se desgranarán a continuación.

El **Capítulo 1** (Introducción) explica el contexto de la tesis y las motivaciones para realizarla. Además, identifica el nicho de investigación y define los objetivos primarios para llevar a cabo el análisis y la proposición inherentes a la tesis.

El **Capítulo 2** (Antecedentes) revisa el estado actual de la las redes móviles de banda ancha e identifica las versiones o publicaciones del estándar 3GPPP (3rd Generation Partnership Project) en las que se basa la actual tesis (versiones 8 y 9 de LTE), así como sus capacidades en comparación a versiones anteriores y posteriores. El capítulo revisa además los conceptos básicos de TCP para ayudar en las posteriores interpretaciones de resultados. Por último, se revisan y clasifican las diferentes implementaciones y modificaciones de TCP durante las tres últimas décadas para una mejor comprensión de los antecedentes de TCP y las motivaciones detrás de la creación de las distintas implementaciones.

El **Capítulo 3** (Estado del Arte) revisa y explica las proposiciones, estudios, trabajos y sugerencias más relevantes en el campo de los protocolos y servicios de transporte, los principales problemas sobre las redes móviles de banda ancha y los trabajos que tratan de proponer y promover alternativas tecnológicas para la comunicación y conocimiento de la capa de transporte y la parte radio. El capítulo es responsable de ordenar y clasificar aquellas áreas de actuación que requieren de mayor estudio (clasificadas como "aspectos de rendimiento") para un mejor entendimiento de los problemas en la interacción de distintos CCAs bajo condiciones críticas, así como ser capaz de elegir el mejor candidato de TCP para afrontar dichas circunstancias. Entre las áreas de actuación o que requieren de mayor estudio se encuentran las siguientes:

- Estudio de los casos de uso señalados para el análisis de TCP en redes móviles sobre distintos despliegues de LTE para confirmar los resultados obtenidos y desechar las dependencias relativas a un único despliegue.

- Se identifica un problema relativo al exceso de paquetes en colas intermedias que provoca tanto un aumento de los retardos punto-a-punto así como un incremento de la probabilidad de pérdida de paquete. Dichos efectos tienen su origen en una mala gestión de los propios CCAs y de los mecanismos propios de planificación de la red de acceso móvil que agrupan los paquetes y los sirven en ráfagas, alterando la interacción punto-a-punto y dificultando la interpretación de la situación de red por parte del servidor. Para ello, se identifica una necesidad de análisis respecto a las implicaciones de diferentes CCAs en distintas situaciones de red (entre otras situaciones se encuentran las de: diferencia de los CCAs con y sin tráfico cruzado, utilizando distintos tráficos y cambios abruptos en el ancho de banda disponible) para entender los comportamientos de TCP y crear una primera selección sencilla de los CCAs más apropiados. Este análisis constituye el primer aspecto de rendimiento (**Performance aspect 1**).

- Se identifica la necesidad de un profundo estudio de las diferentes implementaciones de la fase de Start-up que utilizan distintos CCAs para poder aclarar qué tipo de mecanismo es más adecuado ante qué situaciones de red. Este análisis constituye el segundo aspecto de rendimiento (**Performance aspect 2**).

- Se localiza un área de investigación referente a la adaptación de TCP (distintos CCAs) frente a diferentes variabilidades de la propia red móvil y la variabilidad provocada por el propio usuario en situaciones de alta movilidad que requiere de una mayor atención y estudio. Este análisis constituye el tercer aspecto de rendimiento (**Performance aspect 3**).

- A efectos de cubrir con la presente tesis las redes móviles actuales y futuras, se detecta una necesidad de análisis comparativo de las actuales redes 4G y los despliegues de baja latencia que comandarán la futura quinta generación (5G) de redes móviles de banda ancha. Este análisis constituye el cuarto aspecto de rendimiento (**Performance aspect 4**).

El **Capítulo 4** (Proposición de un marco para la de-osificación intra-protocolar) propone un marco de de-osificación intra-protocolar para conseguir evitar la osificación de la capa de transporte a través de la selección dinámica del CCA más apropiado para ser utilizado bajo ciertas condiciones de contorno (red y requerimientos de aplicación) y posibilidad de coexistencia entre diferentes CCAs. En resumidas cuentas, el marco es el siguiente:
El análisis de los aspectos de rendimiento anteriormente comentados aportará con

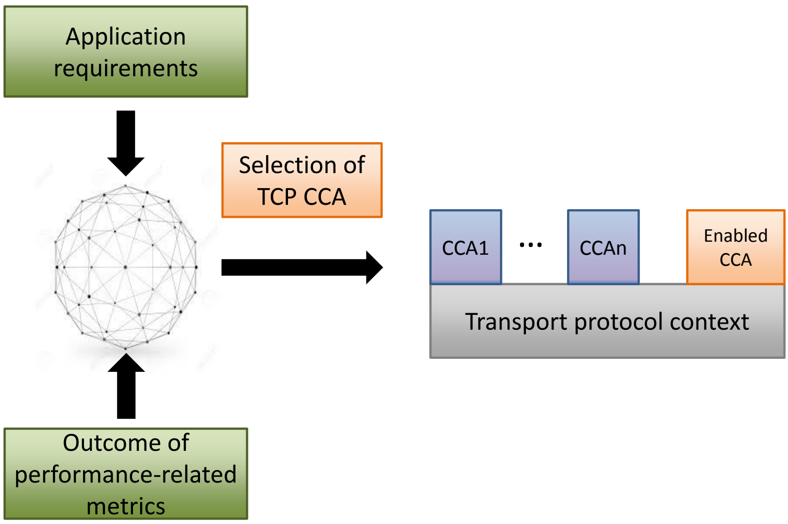


Figure 1: Marco de de-osificación intra-protocolar

heurísticas que se catalogarán como métricas de rendimiento que junto a los requerimientos desde el nivel de aplicación, seleccionarán el CCA más adecuado. Para ello, un único CCA estará habilitado en cada momento y el resto de CCAs disponibles se mantendrán en segundo plano compartiendo un contexto unificado de protocolo de transporte. Para que de esta manera, todos los CCAs tengan acceso a un modelo simplificado de la situación de red en base a la ventana de congestión imperante y la latencia.

El capítulo selecciona además los CCAs más idóneos para su análisis. A este respecto, se seleccionan CCAs con diferentes características: CUBIC y NewReno como CCAs basadas en pérdidas, CDG como el CCA basado en retardos, Illinois como un CCA combinado basado en perdidas y retardos, Westwood+ como un candidato combinado basado en pérdidas y retardos y con la capacidad de estimar el ancho de banda disponible y BBR como un CCA que gestiona el protocolo basado en un modelo que tiene en cuenta los retardos y la estimación del ancho de banda.

El capítulo presenta además los despliegues de LTE utilizados en la tesis y cruza los datos de capacidades de cada uno de ellos con los requerimientos técnicos del análisis de cada uno de los aspectos de rendimiento para así clasificar y decidir sobre qué despliegue lleva a cabo cada parte del análisis.

Por último, el capítulo describe la metodología de análisis para cada uno de los aspectos de rendimiento, siendo en resumen los siguientes:

- **Aspecto de rendimiento 1**: Se decide dividir el análisis en dos partes. Por una parte se analizan las limitaciones técnicas (bien por el propio CCA, la red, los nodos finales, etc.) que pueden llevar a cierto CCA a no ser capaz de alcanzar las capacidades de la red, para así ser capaz de entender las limitaciones de nuestra propuesta y sobre que escenarios es más proclive a tener éxito. En segundo lugar, una vez estudiadas las limitaciones, se analiza el comportamiento de diferentes CCAs ante situaciones de red de diferente calado: diferencia de los CCAs con y sin tráfico cruzado, utilizando distintos tráficos y cambios abruptos en el ancho de banda disponible.

- **Aspecto de rendimiento 2**: El análisis también se divide en dos partes. Por una el estudio de diferentes fases de Start-up disponibles en distintos CCAs sobre los despliegues LTE disponibles en esta tesis. Así se puede entender el rendimiento de cada uno de ellos y ver la afección que el propio despliegue tiene en el rendimiento final. Por otra, el estudio exhaustivo de las diferentes fases de Start-up sobre un despliegue real con diferentes operadores.

- **Aspecto de rendimiento 3**: Esta parte del análisis está dividida en cuatro partes debido al gran número de fuentes de distorsión en los resultados y la necesidad de analizarlos de forma independiente. 1) Se analizan diferentes patrones de movilidad para ver el efecto que éstos causan en el rendimiento de TCP. En la búsqueda de patrones de movilidad simplificados pero suficientemente significativos para el análisis, se deciden analizar dos patrones relativos a un usuario moviéndose en línea recta desde la estación base a posiciones lejanas (forward movement o decreasing quality movement) y vice-versa (backward movement o increasing quality movement). 2) Se analiza el efecto de la velocidad en TCP viendo cómo reacciona al cubrir un área de red más rápido bajo condiciones similares. 3) Se estudia el rendimiento y respuesta de TCP bajo condiciones muy variables de canal en escenarios de alta velocidad. 4) Por último, se recogen los características generales de las diferentes CCAs bajo circunstancias de alta movilidad en comparación a las características detectadas en el aspecto de rendimiento 1.

- **Aspecto de rendimiento 4**: Para el análisis de tanto redes actuales 4G como de despliegues de baja latencia, y frente a la falta de despliegues propios de 5G, se decide hacer uso de los despliegues 4G con retardo añadido para modelar el comportamiento real sobre redes 4G y con retardos minimizados para modelar las potenciales latencias en las futuras redes 5G.

El **Capítulo 5** (Análisis) es responsable de analizar cada aspecto del rendimiento por separado, presentando en ocasiones resúmenes de análisis pormenorizados y presentes en su totalidad en el **Anexo A**. Partiendo del análisis de los cuatro aspectos de rendimiento, el capítulo detecta las heurísticas que más afectan al rendimiento de TCP y que por tanto el marco de de-osificación intra-protocolar tiene en cuenta para la elección del CCA. Las heurísticas son las siguientes:

1. Se indica la necesidad de consideración del parámetro de Window Scale (WS) como representación del tamaño máximo de buffer que los nodos finales soportan. Este tamaño sirve para estimar si el límite es inferior y el CCA seleccionado

puede ser de una agresividad baja-media o si el límite es superior y a priori la limitación está en los nodos intermedios y un CCA más agresivo es necesario para su descubrimiento.

2. Se sugiere la consideración del tamaño del buffer o cola del cuello de botella a fin de despejar la incógnita del límite alcanzable (con un WS no limitante) y actuar en consonancia en la selección del CCA más apropiado.

3. Se demuestra que como norma general las limitaciones de rendimiento se dan debido al comportamiento de los CCA en vez de a una capacitación deficiente del "path" de red entre nodos. Este descubrimiento refleja la importancia y posible impacto que pudiese causar en el rendimiento general nuestra propuesta de la selección del CCA más apropiado para caso de uso.

4. Se descubren los rangos generales de cada CCA y las implicaciones de los mismos bajo diferentes circunstancias de red. Se demuestra que incluso los CCA basados en retardos, son capaces de hacer frente a una reducción drástica del ancho de banda disponible. Sin embargo, estas mismas implementaciones en contraste con las implementaciones con mecanismos basados en pérdidas, son incapaces de afrontar con solvencia un incremento repentino y sustancial del ancho de banda disponible.

5. Se descubre y demuestra el significativo impacto negativo que tiene la utilización de la fase de Start-up de CUBIC (CCA por defecto en Linux) llamada Hybrid Slow Start. Se comprueba que bajo la mayoría de situaciones de variabilidad de canal, el mecanismo causa un descenso considerable del rendimiento de TCP, teniendo un mayor impacto en las conexiones de corta duración. Por tanto, este descubrimiento esclarece que Hybrid Slow Start y los CCAs que hagan uso del mismo, no serán candidatos apropiados para la selección en flujos cortos o en fases de inicialización de flujos más largos. A este respecto, se añade además que las medidas de RTT de las que se valen tanto Hybrid Slow Start como algunos CCAs basados en retardos, no son suficientes para una gestión consistente sobre redes móviles debido a la alta variabilidad y creación de fluctuaciones abruptas en las medidas de RTT tomadas.

6. Se descubre que a pesar de encontrar diferencias en el rendimiento teniendo en cuenta las capacidades del operador utilizado y el tipo de flujo estudiado, como regla generalizada TCP BBR alcanza un mejor rendimiento que el resto de CCAs y su fase Start-up hace lo mismo en comparación al resto de implementaciones. Se añade que, para alcanzar estos resultados, BBR precisa de una cola en el operador suficientemente grande como para soportar las ráfagas periódicas del mismo. En caso de cumplirse la premisa, BBR puede aportar grandes avances en el rendimiento de TCP en general y la fase de Start-up en particular.

7. Se descubre que en entornos de media-alta velocidad, dependiendo del posicionamiento del usuario en la celda, la zona en la que está localizado el usuario puede acarrear grandes diferencias y dependencias para el rendimiento de TCP relativas a la variabilidad y al ancho de banda máximo alcanzable. En este caso, cada escenario de movilidad posibilita su división en tres grandes fases o zonas, separando de esta forma la lógica de selección en el marco de de-osificación

intra-protocolar. Para el patrón de movimiento forward (constante empeoramiento de la calidad de señal), se divide en 3 fases: Fase de Start-up, "sustentador de goodput" y área de calidad limitada. Para el patrón de movimiento backward (constante mejora de la calidad de señal), se encuentran otras tres fases: Fase de Start-up, área de calidad limitada y zona de escalado.

8. En condiciones de baja variabilidad: 1) En despliegues de baja latencia, puesto que el rendimiento de los CCA seleccionados es prácticamente igual en términos de goodput alcanzado, se recomiendo el uso de CCAs poco agresivos para evitar repentinas ráfagas de inyección de paquetes causadas por una alta agresividad del algoritmo de control. 2) En despliegues 4G, NewReno es capaz de alcanzar velocidades de transmisión próximas al ancho de banda disponible en escenarios backward (constante mejora de las condiciones de canal). Sin embargo, en el patrón de movimiento forward (constante empeoramiento de las condiciones de canal) CUBIC capaz de obtener mejores resultados.

9. En condiciones de alta variabilidad: 1) En despliegues de baja latencia, el rendimiento de todos los CCAs es muy parejo, siendo complicado decidir un mejor candidato en base a métricas simples de goodput alcanzado/sostenido. 2) En despliegues 4G, se detecta una relación entre la variabilidad del escenario de movilidad con la diferencia de rendimiento entre los dos patrones de velocidad, acusando NewReno más la fluctuación en las condiciones de red. A este respecto, CUBIC demuestra un comportamiento y rendimiento superior debido a la agresividad de la implementación.

El **Capítulo 6** (validación del marco de de-osificación intra-protocolar) es responsable de la validación del marco de de-osificación intra-protocolar propuesto. Para llevar a cabo esta validación se consideran entornos de media-alta movilidad. Los escenarios utilizados para la validación cubren dos patrones de movimiento (forward y backward), tres velocidades diferentes (a 60 km/h, a 120 km/h y a 300 km/h), tres requerimientos de nivel de aplicación (aplicaciones basadas en goodput, aplicaciones con necesidad de rendimiento balanceado de TCP medido por una métrica de rendimiento y aplicaciones sensibles a retardos) y la posibilidad de selección de cualquiera de los cinco CCAs bajo estudio (NewReno, CUBIC, Westwood+, Illinois y BBR).

La selección del CCA por parte del marco de de-osificación intra-protocolar sigue dos vertientes muy definidas. Estas dos vertientes tienen su mayor distinción en las capacidades de la capa de transporte, más concretamente de la flexibilidad de esta. A este respecto, la validación separa la opción de selección con limitada flexibilidad de la capa de transporte (Opción 1) y con capa de transporte flexible (Opción 2).

Respecto a la selección de CCA con limitada flexibilidad de la capa de transporte (Opción 1), recientes trabajos de investigación han demostrado que la capa de transporte actual posibilita la selección del protocolo de transporte antes de iniciar cualquier transmisión. De la misma manera, en vez de seleccionar entre diferentes protocolos de transporte, es posible seleccionar dentro del protocolo TCP un CCA en concreto dentro de los hábiles. Estos avances cuentan con mejoras en la API de comunicación entre el nivel de aplicación y de transporte para poder trasmitir y considerar los requerimientos de aplicación en el momento de la selección del protocolo de transporte. De la misma forma, se considera la posibilidad de añadir las condiciones de red a la lógica de selección. Teniendo en cuenta estos avances y que la flexibilidad actual de la capa de transporte solo permite la elección antes de comenzar cada transmisión, nuestra primera opción de selección (Opción 1) aborda esta casuística. La siguiente Tabla1 muestra los resultados obtenidos para todos los escenarios de movilidad, consideración de latencia punto-a-punto y métricas de requerimientos de aplicación. Cada celda muestra el CCA seleccionado y el asterisco muestra que varios CCAs son apropiados para su selección.

Contexto	60 km/h					
	Baja latencia			Latencias 4G		
	Goodput	Métrica	Retardo	Goodput	Métrica	Retardo
Forward	*	BBR	Westwood+	Illinois	*	NewReno
Backward	CUBIC	BBR	BBR	BBR	CUBIC	NewReno
Contexto	120 km/h					
	Baja latencia			Latencias 4G		
	Goodput	Métrica	Retardo	Goodput	Métrica	Retardo
Forward	*	BBR	*	BBR	*	NewReno
Backward	*	CUBIC	CUBIC	BBR	BBR	NewReno
Contexto	300 km/h					
	Baja latencia			Latencias 4G		
	Goodput	Métrica	Retardo	Goodput	Métrica	Retardo
Forward	*	BBR	BBR	CUBIC	CUBIC	*
Backward	NewReno	NewReno	BBR	BBR	CUBIC	*

Table 1: Opción 1: Selección del CCA más apropiado durante el establecimiento de la transmisión TCP.

Los resultados muestran 4 importantes detalles relativos a los CCAs y los contextos en los que han sido seleccionados:

1. BBR demuestra un buen rendimiento en terminos de goodput en entornos 4G. Sin embargo, no ofrece un rendimiento balanceado, ni apropiado para aplicaciones sensibles a retardos. En entornos de baja latencia, BBR ofrece un rendimiento más controlado (buenos registros de goodput a la vez que un bajo retardo y número de pérdidas de paquetes) y es seleccionado bajo la premisa de todas las métricas del nivel de aplicación.

2. Westwood+ representa una alternativa útil para los requerimientos de nivel de aplicación enfocados en el retardo, no maximizando el goodput pero preservando un retardo medio significativamente bajo.

3. Illinois demuestra ser una alternativa apropiada en situaciones combinadas de enfoque de maximización de goodput y grandes BDPs (entornos 4G), pero demostrando una agresividad inapropiada para el resto de los casos de uso.

4. NewReno provee una alternativa para aplicaciones sensibles a retardo en entornos 4G.

5. CUBIC demuestra ser un candidato idóneo para despliegues de tanto baja latencia como entornos 4G, adquiriendo un buen rendimiento enfocado en goodput y para aplicaciones que precisen de un comportamiento balanceado, pero no adecuado para aplicaciones sensibles a retardos.

Respecto a la opción con mayor flexibilidad de la capa de transporte (Opción 2), se considera que futuros despliegues y mejoras en la API de comunicación van a posibilitar el manejo y cambio de CCA "en caliente" mientras la transmisión se lleva a cabo. Esto es, si durante la transmisión se detecta que las condiciones de contorno han cambiado, se podría cambiar al CCA óptimo en vez de necesariamente tener que seguir con el seleccionado al comienzo del flujo. Buscando la validación de este mecanismo y teniendo en cuenta las diferentes fases o áreas detectadas en capítulos anteriores (con dependencias para el CCA distintas de una fase a otra), se decide evaluar la selección de otro candidato de TCP al comienzo de dichas fases.

Para la evaluación, selección de candidatos por fases y validación, varios pasos son necesarios: 1) Selección del CCA más apropiado por fases; 2) En caso de encontrarnos cambios de CCA de una fase a otra en el mismo escenario, el primer CCA deberá transmitir su información de contexto de transporte al segundo CCA. Para ello, se modela la ventana de congestión teniendo en cuenta el retardo del momento frente al retardo base; 3) Después, se ejecutan pruebas partiendo desde los puntos de intercambio de información y seleccionado el segundo CCA con los parámetros de iniciación modelados del primer CCA; 4) Evaluación y validación final de resultados y decisión sobre si el intercambio a beneficiado al rendimiento o por el contrario, es mejor seguir con un único CCA.

La siguiente Tabla2 muestra los resultados obtenidos para todos los escenarios de movilidad, consideración de latencia punto-a-punto, las fases o áreas detectadas y métricas de requerimientos de aplicación ("G" para goodput, "M" para las aplicaciones que precisen de un comportamiento balanceado y "D " para las aplicaciones sensibles a retardos). Cada celda muestra el CCA seleccionado. Los colores indican el resultado de todo el proceso de validación de intercambio de CCAs explicado: gris para los escenarios que no requieren de intercambio de CCAs, verde para los intercambio de CCA que benefician significativamente el rendimiento (más del 5%), amarillo para los resultados que igualan o mejoran ligeramente con el intercambio (igual resultado o mejora por debajo del 5%) y rojo para los casos en los que el intercambio supone la obtención de un peor rendimiento.

Los resultados obtenidos demuestran 3 detalles importantes:

1. Las mejoras significativas de rendimiento se dan por igual en ambos patrones de movilidad. La mayoría de estos escenarios demuestran que es altamente beneficioso hacer uso de un CCA para iniciar la transmisión y después de la fase de Start-up cambiar a otro CCA.

2. Se demuestra que en el patrón de movilidad backward (constante mejora de la calidad del canal), bajo entornos de reducida variabilidad (el escenario a 120 km/h demuestra ser menos variable que los otros dos) el CCA seleccionado desde el principio de la transmisión es suficiente para alcanzar el mejor rendimiento posible y por tanto, no se requiere de intercambios de CCA.

3. Se identifica que en el patrón de movilidad forward (constante empeoramiento de la calidad del canal), bajo condiciones de alta variabilidad de canal, la técnica de intercambio de CCA provee de una mejora en el rendimiento debido a la gestión de la fase o área de calidad limitada con un CCA diferente y más apropiado que el predominante en el resto del flujo.

/	Forward a 60 km/h			Backward a 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
G. - Baja latencia	NewReno	NewReno	CUBIC	NewReno	CUBIC	CUBIC
G. - Latencias 4G	NewReno	Illinois	CUBIC	BBR		
M - Baja latencia	BBR	BBR	Westwood+	NewReno	BBR	BBR
M - Latencias 4G	NewReno	NewReno	CUBIC	Illinois	CUBIC	CUBIC
D. - Baja latencia	Westwood+			BBR		
D. - Latencias 4G	BBR	NewReno	CUBIC	Illinois	Westwood+	NewReno
/	Forward a 120 km/h			Backward a 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
G. - Baja latencia	Illinois	Illinois	BBR	CUBIC		
G. - Latencias 4G	BBR	BBR	Illinois	BBR		
M - Baja latencia	NewReno	BBR	BBR	CUBIC		
M - Latencias 4G	CUBIC	NewReno	Illinois	Westwood+	BBR	BBR
D. - Baja latencia	NewReno	BBR	BBR	CUBIC		
D. - Latencias 4G	Westwood+	NewReno	NewReno	Westwood+		
/	Forward a 300 km/h			Backward a 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
G. - Baja latencia	Illinois			NewReno		-
G. - Latencias 4G	CUBIC	CUBIC	Westwood+	BBR	Illinois	-
M - Baja latencia	BBR	BBR	NewReno	BBR	NewReno	-
M - Latencias 4G	CUBIC			BBR	CUBIC	-
D. - Baja latencia	BBR	BBR	NewReno	Westwood+	BBR	-
D. - Latencias 4G	CUBIC	CUBIC	NewReno	NewReno	Westwood+	-

Table 2: Opción 2: Selección del CCA más apropiado en real-time a lo largo de la transmisión TCP.

4. Se detectan casos de uso en entornos 4G y enfocados en aplicaciones sensibles a retardo que el intercambio de CCAs empeora el rendimiento y por tanto, su utilización debe ser evitada.

De 36 casos de uso analizados, 11 han demostrado obtener el mejor resultado solo mediante la apropiada selección del CCA en el establecimiento de la transmisión (Opción 1 de selección intra-protocolar), 10 han reportado beneficios en el rendimiento de más del 5%, 13 han presentado mejoras del rendimiento por debajo del 5% y 2 casos han mostrado un empeoramiento del rendimiento al aplicar la técnica de la Opción de selección de CCAs basadas en una capa de transporte flexible. En general, la técnica de intercambio de CCAs demuestra una gran respuesta en los escenarios seleccionados y por tanto prueba que su aplicación es segura y beneficiosa en las redes actuales y más importante, del futuro.

Finalmente, el **Capítulo 7** (conclusiones) recoge las principales conclusiones del trabajo realizado y sitúa cada contribución en la publicación que ha sido recogida. Así mismo, se incluyen las principales limitaciones del trabajo alineándolas a su vez con las posibles líneas futuras de actuación.

Preface

Faith is taking the first step even when you don't see the whole staircase.

— Martin Luther King, Jr.

In my short but intense research life I have not come across a study or thesis that have been able to remain unaltered from the very first idea to the fulfillment of the objective. Research is a trip and so has been this thesis. This thesis happened due to a chain of coincidences and have been reminding me about the importance of small and "insignificant" events.All started years ago when I first discovered the book of Kurose thanks to Cristina. The networking seed was already planted. It was something different, I enjoyed reading it in a distinct way and I knew it was something special. A couple of years later, Joselu introduced me into the NQaS research group. It was a period of time that showed me the applicability of the concepts and knowledge that had been gathering throughout the degree with special focus on networking and the analysis of performance. I want to thank Edu for the welcome and side by side work. I also want to acknowledge Armando for his spirit and lessons. Thanks to the team in the LAB! You have made and you still make the environment easier in order to deal with the mental struggling. No names, all beloved!

After 2-3 years as a research assistant and synchronized with the end of my MSc, I started working as a Research fellow in the same department and research group. The more we discovered about networking the more thrilling I was about networking in general and transport protocols in particular. Thank you Eva for your help, support and guidance! With a clear attraction to transport protocols and being aware of stepping into a very crowded research field, I decided to focus my effort in different use-cases of mobile networks due to the lack of knowledge in this regard in comparison with fixed networks.

Networking understanding, analysis and deep study, led to a stay at Karlstad (KaU) in collaboration with Ericsson. DISCO research group is a recognized team with a long journey in the analysis of transport protocols and mobile networks performance. It was a wild decision. We only knew them due to their work. No personal relationship or even a conversation, I decided to move to Karlstad for three months and I am very proud of that decision. In fact, after two years, I went again to Karlstad and experienced the same good feeling that the group itself provides. Research, innebandy and fika. Perfect combination! Thank you all! In my second stay I shared office with

Andreas and Ali and the experience was amazing. All the best guys! Salam!

Special thanks for two people in KAU. Karl-Johan, very supportive colleague that demonstrates everyday dedication and willingness to keep the group synchronized. Anna, a truly gifted and resolutive researcher with dedication and intuition. Thank you both for the experience!

KAU was the destination of my first and third stays. Apart from that, my second stay took me to the breathtaking Ghent, Zwijnaarde more precisely, to collaborate in the LTE deployment of iMEC called w-iLab.t. I want to thank Dries and Vasilis for helping and supporting me throughout the measurement process with immense kindness. Besides, Belgium taught me that I knew nothing about beer.

Going back to our research group I want to express my gratitude to the two wise men of the tribe. Two brainy guys who an American would label "second to none". JOF, thank you for your resolutive and supportive advice. Fidel, thanks for the opportunity, dedication, time and accurate advice.

Quería agradecer a mi familia la dedicación, enseñanza y amor incondicional. Amama Mila, asko erraztu duzue bidea zuen laguntza eta bizitza bizitzeko moduagaz. Bizitasun paregabea eta burua erne! Amama Valen, tu catálogo interminable de frases cala en cualquiera. La lección más importante todavía me cuesta: "Oir, ver y callar".

Quiero agradecer a Rebeca todo lo aportado a mi vida y los consejos durante esta tesis. Eres mi otro lado, la persona que consigue que saque más de mi parte más amable, mi mejor valuarte en términos mentales y emocionales. Me mantienes conectado a la realidad y me has ayudado a pasar por los momentos complicados. Eres una lección de humildad y robustez. Conocerte me ha enseñado a valorar las cosas que toda mi vida he dado por hechas. Un ejemplo como persona que forma parte de mi apuesta de futuro. De corazón, mil gracias ! Xenon, se te quiere a pesar de los lametazos a las 5 de la mañana. ¡Señoooooor!

Aitor, benetan lagundu nau etxean beste motibatu bat egon izana. Igandeko kafea eta ikerketa saioa faltan botatzen dut. Karrera bukatzeko pare bat urte falta zitzaizkidanean, zu ikusita birgogoratu nuen ni ere momenturen baten ilusioa izan nuela eta kapaz izan nintzan burua berreskuratzeko. Eskerrik asko ta segi berdin motel!

Aita y ama, un gracias queda soso y poco representativo, pero tras ver la dedicación, la plena disposición, la educación aportada, los valores transmitidos y lo fácil que me habeis hecho la vida, solo puedo decir ¡GRACIAS! ESKERRIK ASKO! Solo cuando uno tiene una edad más cerca de tener hijos que de ser el hijo al que criar, te das cuenta de todo el esfuerzo que conlleva. Entrega máxima. Gracias por pelearos para dedicarme aún más tiempo. Eskerrik asko goiz guztietan edandako jukuengatik. Dena suertatzen da garrantzitsu, orain tesi hau defenditzeko. Gracias por negarme muchas cosas para valorar las que tenía. Eskerrik asko parkean, futbolean eta karaten pasandako ordu guztiengatik. Gracias por el apoyo ciego, dedicación exclusiva y protección. ¡Es difícil mereceros!

Contents

List of Tables

List of Figures

Abbreviations

3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
AIMD	Accumulative Increase Multiplicative Decrease
API	Application Programming Interface
AQM	Active Queue Management
BBR	Bottleneck Bandwidth and RTT
BDP	Bandwidth-delay product
BLER	Block Error Rate
CA	Congestion Avoidance
CCA	Congestion Control Algorithm
CDN	Content Delivery Network
CQI	Channel Quality Indicator
CRR	CQI Reporting Rate
CWND	Congestion Window
DNS	Domain Name System
DUPACK	Duplicated Acknowledgement
ECDF	Empirical Cumulative Distribution Function
ECN	Explicit Congestion Notification
eNodeB	Evolved Node B
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved UMTS Terrestrial Radio Access
FEC	Forward Error Correction
FQ	Fair Queueing
HARQ	Hybrid automatic repeat request
HSPA+	Evolved HSPA
HSPA	High-Speed Packet Access
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunication Union
IW	Initial Window
KPI	Key Performance Metric
LTE	Long-Term Evolution
MAC	Media Access Control

MCS	Modulation and Coding Scheme
MIMO	Multiple-Input, Multiple-Output
MME	Mobility Management Entity
NACK	Negative-Acknowledgement
NIC	Network Interface Controller
OFDMA	Orthogonal Frequency Division Multiple Access
PAx	Performance Aspect x
PDN	Packet Data Network
P-GW	PDN Gateway
PRB	Physical Resource Block
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RAN	Radio Access Network
RBG	Resource Block Group
RLC	Radio Link Control
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indication
RTO	Retransmission Timeout
RTT	Round-Trip Time
SACK	Selective Acknowledgement
SC-FDMA	Single Carrier Frequency Division Multiple Access
SDO	Standards Developing Organization
S-GW	Serving Gateway
SINR	Signal to Interference plus Noise Ratio
SISO	Single-Input, Single-Output
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
SSL	Secure Socket Layer
ssthres	Slow-Start Threshold
TbSize/TBS	Transport block Size
TCP	Transmission Control Protocol
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
UTMS	Universal Mobile Telecommunications System
WCDMA	Wideband Code Division Multiple Access
WS	Window Scaling

Table 3: List of abbreviations.

1 Introduction

1.1 The context and motivation of the work

Mobile Broadband (MBB) usage has raised 4,000-fold over the last decade and almost 400-million-fold over the past 15 years, growing more than 60% only in 2016 [4]. The close future is expected to be equally promising with an eight-fold increment in mobile data traffic between 2016 and 2021. In fact, according to these statistics, wireless and mobile Internet traffic will in two years exceed traffic from wired devices. In the near future, 5G will be the most utilized mobile technology even overcoming the traffic volumes and utilization of 4G due to the penetration of MBB in developing countries. Considering these figures, **it is clear that mobile networks are becoming increasingly important and the predominant mechanism to access Internet**.

The growth expectation is not only related to the traffic volume itself but also to the average speeds. The common **global MBB speed will exceed 20 Mbps on average by the end of 2021 [4] and is expected to surpass those numbers with the up-and-coming 5G developments, settling unprecedented target speeds and capabilities [1]**. In the same way that Internet access speeds are raised, so does the awareness of end-users in relation to their service level agreement (SLA) and to which extent the received service meets the contract (Quality of Service -QoS-), having a clear impact in their (dis)satisfaction [5] (Quality of Experience -QoE-). MBB access provides a variable environment that makes the performance playground very challenging for transport protocols. The sources of variability could be grouped in two:

- Client-related: depending on the position of the client in the cell and its mobility the conditions of the radio channel are altered by the propagation and fading pattern.

- Shared-resources-related: the mobile networks themselves dynamically assign radio resources to the clients in the same cell, resulting in fast variations in the available resources of a certain User Equipment (UE) over time.

In such a variable context, **inefficient transport protocols may prevent communications from taking full advantage of the available resources [6, 7]**. The performance of communications over MBB has been traditionally limited not only by the raw available radio capacity but by also limitations imposed by used transport protocols.

Since a large part of mobile Internet comprises TCP flows [8], **the performance of TCP over cellular networks has become an important and critical research topic**. However, TCP is not a single implementation but a family of implementations with different features and mechanisms that lead to distinct abilities depending on the network model. Each implementation forms a congestion control algorithm (CCA) reacting differently to losses, delays and cross-traffic and achieving different speeds. Primarily, different implementations of TCP aimed at improving the performance over fixed networks. In such scenarios, the achievable capacity of a certain path is fairly stable and, therefore, many TCP flavours assume that, once the achievable rate is reached, the CCA only needs to oscillate around such rate in order to take advantage of the entire capacity over time. However, in MBB these assumptions do not fit. The maximum capacities are constantly fluctuating, being modified by the schedulers' assignment in the base station. **Even though many different CCAs have been developed [9], still none of them have been implemented to face the variability of mobile networks in terms of delay and throughput under any circumstance of the network in an easily deployable way. The CCA's ability to adapt and their suitability for mobile networks rely on very different features and conditions compared to wired networks [6]**.

Since the analysis of TCP over MBB is complex due to the different factors affecting the performance and looking into the different situations that could bring greater performance differences due to its potential impact on the behaviour of CCAs and therefore on the achieved outcome, two are the selected general use-cases to analyze in the current thesis:

- A proper **understanding of the implications of the movement in the interaction between TCP and the radio part** is required. If the variability is the main feature in cellular networks even under stillness circumstances, when a UE is moving the assigned channel varies even more drastically over time. The fluctuation of the quality of the channel also depends on self-inflicted effects by the UE such as speed. This thesis covers the analysis of such scenarios.

- Taking into account that it has been proven that approximately 87% of the Internet flows are smaller than 1 kB, and only about 0.5% are bigger than 100 kB [10], **the flow Start-up performance of TCP over MBB is critical** and needs to be revisited in order to validate its behaviour.

In order to deal with complicated technical cases such as the interaction between the radio part of MBB and the transport layer, **some works have suggested the necessity to build mechanisms that allow greater flexibility and therefore, responsiveness. In this regard, transport services have emerged [11–13] building on top of TCP or UDP ad-hoc layers that work between the transport layer and the application layer. This way, they take advantage of the substrate transport protocol (mainly TCP and UDP) and gain some freedom and flexibility due to its development in the user space of the operating system (OS) and additional functionalities (i.e. congestion control of QUIC over UDP)**. However, the utilization of TCP and UDP forces the system to stick to the inherent characteristics of the selected substrate transport protocol from the beginning of the transmission until it is closed. This limitation has been named "ossification" and it has three main effects: 1) It reduces the opportunity to select and modify transport layer protocols both at the beginning of a certain transmissions and in-flight. 2) It leaves little room for transport protocol innovation. 3)

It provides limited or non-existent flexibility of the application programming interface (API) [12] between the application layer and the transport layer. **The existence of constraints in the communication between these two layers results in standardized behaviour of the transport protocol with no consideration of the requirements from the application layer or network status**.

The study of different TCP CCAs over distinct challenging MBB circumstances would help minimize the effect of ossification in the transport layer, providing with the most suitable TCP candidate. Since the choice is done within a selection of TCP candidates, **our work aims at de-ossifying the transport layer in an intra-protocol way**. The intra-protocol de-ossification process of TCP contributes in three major ways: 1) The analysis would prompt the strengths and weaknesses of each CCA under each circumstance. 2) The work would help detecting different patterns of phases in order to classify them due to their networking singularities. 3) The study would provide with the best TCP CCA solution for each situation based on not only the network status, but also the requirements from the application layer.

All in all, this thesis analyzes the performance of TCP, in MBB, under distinct mobility circumstances and also focuses on the flow start-up performance. The work also proposes a theoretical solutions, that is deployable as a transport service, based on the network status and the application requirements for an enhanced coexistence between distinct implementations of TCP and MBB and paves the way for further improvements.

1.2 Objectives

The current work aims studying and evaluating the interaction of TCP with MBB and detecting the most beneficial CCA selection depending on the network circumstances and the required transport layer target (i.e. focused on goodput, focused on reducing the delay and so forth). To this end, the following objectives are established:

- **Objective 1 - State of The Art analysis**: It is important to analyze the State of The Art situation in order to be able to identify the requirements and open issues related to the performance of TCP over MBB. In this regard, it should be reviewed the most relevant and recent proposals of transport protocols/services, the specific issues in MBB regarding TCP and the proposals that suggest the use of certain awareness between mobile networks and transport layer.
- **Objective 2 - Identification of relevant CCAs**: Considering the multiple proposals of TCP implementations throughout the last three decades, it is crucial to identify and select the CCAs that are more interesting to be analyzed. In order to select a wide range of CCAs and its appropriateness to face the presented challenging network conditions, it is important to first classify the CCAs by their main guiding feature and pick candidates from a variety of feature-based groups.
- **Objective 3 - Identification of major performance aspects**: It is essential to identify performance aspects in order to study the detected challenging conditions of Start-up and mobility and select the findings with major impact (i.e.

network itself, mobility, CCAs themselves and so forth). The objective could be divided in four subobjectives so as to easier guide the identification of major performance aspects.

– **Subobjective 3.1 - Technical and implementation limitations**: Our work is to detect the possible technical or implementation (CCA-related) limitations that could prevent one CCA or measurement realization from achieving its full potential. In this work, it is important to analyze the possible network constraints that could trigger flow control mechanisms, making the congestion control useless. Apart from that, it is vital to to detect the possible limitations that clients and servers may well cause.

– **Subobjective 3.2 - Delay variability in CCAs that consider RTT**: In mobile networks, due to scheduling policies, the packets are served in groups, leading to some alteration of the RTT measurements between consecutive RTT samples [14]. If we take into account that several CCAs (mainly delay-based and Hybrid proposals) consider the measurement of the RTT a predominant parameter to drive its behaviour, it is important to analyze the estimation of RTT in MBB and detect possible deficiencies in the performance of TCP for this reason. Related to this topic, it is also important to analyze the impact of different baseline RTTs for the same scenarios in order to compare the performance and responsiveness of TCP under distinct end-to-end delays.

– **Subobjective 3.3 - Start-up performance**: The beginning of any classic end-to-end transmission has a unique distinction: there is no knowledge of the bottleneck path to the receiver and the first rounds are dedicated to discover such information (i.e. the bandwidth of the bottleneck that limits the path). There is always a trade-off between the speed in the discovery phase plus the possible negative impact of massive packet loss events and a conservative approach that could underutilize the network capacity. In MBB, the so-called available bandwidth is in constant evolution. Therefore, it is crucial to determine the performance differences of distinct flow Start-up methods and define the most suitable network conditions for the application of each of them.

– **Subobjective 3.4 - Mobility**: It is important to analyze the performance and responsiveness of different CCAs under mobility in order to understand the implications of the movement in the interaction between the transport protocol and the MBB variability. A proper understanding of such interaction would benefit the analysis and subsequent selection of the candidate that best fits a precise network condition and requirements from the application layer.

- **Objective 4 - LTE testbed identification**: It is essential to determine which measurement deployment is more convenient to perform/test/validate each performance aspect. There exist different types of MBB deployments that range from a simulated environment to fully a real-world deployment. The challenge is to pick the best deployment based on its features and available possibilities so as to cover a certain analysis.

- **Objective 5 - Select the most suitable CCA**: Future MBB scenarios would allow not only a fast adaptation to the behaviour of transport protocols' algorithm to the variability of cellular access, but also selecting between different CCAs depending on the network conditions and application requirements. Following the same trend of cost-effectiveness regarding the creation of new protocols and the reusability of current ones, it is important to select the most adequate CCA among the selected ones depending on the network conditions and application requirements.

- **Objective 6 - Propose a selection heuristic sequence**: The selection of the most preferable CCA is based on a precise network situation and the application layer requirements. Regarding the flexibility of the transport layer, two options need to be taken into account: the selection of a certain CCA at the beginning of the transmission and the in-flight switch between CCAs to better fit the selection criteria. Even though different CCAs may well be appropriate for a certain network situation, the switch between flavours could prompt a completely different outcome than the results with single CCA experiments. Therefore, it is vital to propose and validate based on a heuristic sequence selection criteria the selected options and compare it with single CCA experiments.

1.3 Dissertation outline

The thesis dissertation is organized in six main chapters:

- **Chapter 1 - Introduction:**
  Chapter 1 explains the context and main motivations of the thesis and identifies the research niche. Besides, it briefly describes the tasks defined in order to fulfill the main objectives.

- **Chapter 2 - Background:**
  Chapter 2 is devoted to partially reviewing the current status of mobile network and pointing out the selected standardized eighth and ninth releases (R8 and R9) of 3GPPP (3rd Generation Partnership Project) for the study. Apart from that, the chapter revisits some basic concepts of TCP so that the further analytic chapter could be properly understood. Finally, some historic TCP implementations and modifications are reviewed in order to better understand the background of today's CCAs, the motivations and those CCA precursors of such ideas together with a classified bibliography regarding their features and performance.

- **Chapter 3 - State of The Art:**
  Chapter 3 covers the review and explanation of most relevant and recent proposals, approaches and studies in relation to transport protocols/services, mobile networks features and relation with transport layer performance. The chapter is devoted to ordering and classifying the research areas that require further study. It identifies the fields that are lacking deeper insights, and some guidelines according to the state of the art so as to better explain and settle the problems regarding the performance of CCAs to fully adapt to the challenging conditions that mobile networks propose. Besides, it identifies the performance aspects that need to be further analyzed in order to better understand the performance of distinct CCAs under challenging MBB conditions and be able to decide the best candidate for each circumstance.

- **Chapter 4 - Proposal of a intra-protocol de-ossification framework:**
  Chapter 4 is responsible for proposing a theoretical (deployable as a transport service) intra-protocol de-ossification framework in order to get partially rid of the ossification in the transport layer through the ad-hoc selection of best CCA candidate to face the combination of network status and application requirements. Besides, the chapter selects the CCA implementation that will be analyzed and defines the technical requirements of each performance aspect. In addition, it presents the available LTE deployments and their capabilities in order to decide the best testbed and strategy for the analysis of each performance aspect. Finally, the analytical methodology is described.

- **Chapter 5 - Analysis:**
  Chapter 5 targets the analysis of the detected performance aspects regarding challenging MBB conditions of flow Start-up and mobility. The chapter covers the analysis of the Start-up phase of TCP, together with the study of CCAs performance implications in stillness circumstances in order to detect the patterns in the behavior of distinct CCAs. Besides, the chapter deals with the analysis of TCP under mobility studying different TCP implementations under different end-to-end baseline latencies, under distinct speeds while the network conditions are similar, different movement patterns and different variable conditions.

- **Chapter 6 - Validation of the intra-protocol de-ossification framework:**
  Chapter 6 covers the application of criteria that consider the network status and requirements from the application layer while the transport layer a) only allows the CCA selection at the beginning of the transmission and b) allows the in-flight switch of CCAs. After selecting the best CCA implementation to face each situation, the chapter covers the actual experimentation of such combinations and evaluates the outcome.

- **Chapter 7 - Conclusions:**
  Chapter 7 summarizes the main conclusions of the dissertation and identifies the main limitations of the work aligning them with the description of future research lines.

2 Background

This chapter introduces the technology associated to mobile networks in order to explain the context and the evolution in terms of capabilities and architectural differences. The current chapter also overviews TCP and the different efforts dedicated to the enhancement of it over the last three decades.

Section 2.1 briefly reviews mobile networks so as to explain the technological and architectural differences, with special focus on maximum achievable rates.

Section 2.2 covers the evolution of TCP from its early stage, going through several improvements and algorithm proposals until the State of The Art situation. The Section also classifies the most significant congestion controls over time and based on the feature that prevails.

2.1 Review of mobile networks

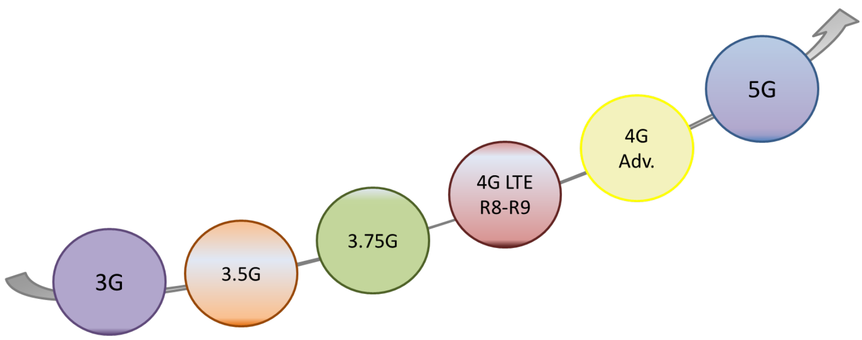
LTE (Long-Term Evolution) is the technology proposed by the 3GPP standardization body to achieve 4G specifications (although strictly speaking the technical requirements defined by ITU could be only fulfilled by WiMAX and the multiple releases of LTE Advanced). The basic and widespreadly used LTE only comprised the eighth and ninth standardized releases (R8 and R9) of 4G. Since manufacturers and providers equally sell and label LTE and 4G, they are commonly misinterpreted. When referring to LTE or 4G, this dissertation is pointing out to R8 and R9 of LTE [15,16].

Additionally 5G, targeting its earliest deployment around 2020, is expected to reach performance levels comparable to those in currently available fixed broadband technologies. In order to do so, it focuses on user experience, architectural evolution, enhanced services, improved performance, new business models and evolved management [1,17].

The section is divided in two parts: 1) The description of the evolution regarding technology capabilities. 2) The explanation of the quality reporting mechanism and the related resource assignment method.

2.1.1 Evolution of technology capabilities

One of the most important aspects of the evolution of mobile networks is the capabilities increment in terms of allowed transmission rates. In the end, every enhancement tries to provide the network with higher speeds by improving resource allocation, implementing evolved modulations and with better management of the infrastructure. Figure 2.1 summarizes the technology evolution of mobile networks from the third generation (3G) to the fifth one (5G).



Downlink max. peak rate	384 Kbps	14.4 Mbps	42 Mbps	300 Mbps	1 Gbps	10 Gbps
Uplink max. peak rate	128 Kbps	5.7 Mbps	11 Mbps	75 Mbps	500 Mbps	10 Gbps

Figure 2.1: Evolution of Mobile Networks technology

3G networks utilized Universal Mobile Telecommunications System (UMTS) with Wideband Code Division Multiple Access (WCDMA) air-interface technology to improve its performance in comparison with 2G networks. 3G was a very expensive deployment due to its novelty. However, it was the first mobile network that was ready to carry mobile TV, video chatting, navigational maps, web-based video and audio files. 3G used code-based multiplexing to transfer information to multiple clients on the same frequency band. It typically operated with 15MHz to 20 MHz channels. All these improvements allowed 3G to have a downlink and uplink speed of 384 Kbps and 128 Kbps respectively. Even though the peak rates do not seem impressive nowadays, 3G paved the way to further improvements and enabled the creation and adoption of smartphones.

The so-called 3.5G proposed an optimization of UMTS/WCDMA with High Speed Packet data Access (HSPA) in order to increase the spectral efficiency and reliability, and therefore the final performance. The transmissions in HSDPA specific channels (the downlink of HSPA) were shared among clients in the cell and are multi-code. They had better and adaptive modulation (Quadrature Phase-Shift Keying -QPSK- and 16 Quadrature Amplitude Modulation -16QAM-), shorter transmission time intervals (TTIs) and the introduction of hybrid automatic repeat request (HARQ) as a frame retransmission mechanism. The next proposal, 3.75G utilized an evolution of HSPA called Evolved HSPA (HSPA+). Three were the most significant improvements: 1) It enabled downlink and uplink modulations up to 64QAM and 16QAM respectively. 2) HSPA+ was the first technology introducing Dual Carriers (HSPA+ DC), allowing to simultaneously connect to two or three carriers when available carriers are not being utilized for other channels, achieving this way greater capacities. 3) HSPA+ used two or more different data streams over two or more antennas to be received by two or more reception antennas. To be able to do this and discern between data sources, different data pre-coding and channel pilots for each antenna were used. This technique is called MIMO (Multiple-Input, Multiple-Output) and enables higher data rates over the channel. All these enhancements allowed an important rate increment, achieving up to 42 Mbps in the downlink and 11 Mbps in the uplink.

4G emerged as the future of mobile communication with comparable speeds to the ones obtained in broadband wired connections. 4G networks allow high-speed mobile web access, better gaming due to the decrease in latency, high-definition mobile TV, high quality streaming videos and expanded multimedia services among other applications. In order to avoid interferences and increase the spectral efficiency, 4G uses channels of different bandwidths from 1.4 MHz to 20 MHz and works with OFDMA (Orthogonal Frequency Division Multiple Access) links in the downstream and SC-FDMA (Single Carrier Frequency Division Multiple Access) in the upstream. Such enhancements and new technology adoption report peak rates of 100 Mbps for downlink connections and 50 Mbps for the uplink in the case of SISO (Single-Input, Single-Output) systems and up to 300 Mbps in the downstream for MIMO systems.

The tenth release (R10) of LTE called LTE-Advanced merely focused on providing higher capacities. The essential aspect of LTE-Advanced is the Carrier Aggregation technique, which enables to download data from multiple network bands simultaneously. Carrier Aggregation is able to combine the signals from these different carriers, lengthening the channel bandwidth up to 100 MHz (maximum management of 5 simultaneous bands) and therefore the achievable data rates. In order to increase the transfer throughput even more, LTE-Advanced requires the use of MIMO systems. Taking into account the failure that LTE was in relation to the data rate expectations of 4G, LTE-Advanced fulfills such standardized 4G technology with maximum speeds of 1 Gbps.

In the forthcoming future, 5G networks will provide high capacities and rates (up to 10 Gbps), enabling high quality virtual reality, bulk transmissions between vehicular systems and ultra-high quality mobile TV among other uses. 5G will also allow ultra-low latencies with 1ms delays in device to device communications and a significant reduction of the end-to-end latency. The targets of low latency and massive throughput will require ultra-dense cellular networks with highly flexible Radio Access Network (RAN) architecture and topology and the ability to divide, place and distribute the tasks in a so-called functional split.

Apart from the technological advances of 5G, the latency reduction comes inherited from the architectural enhancements applied over certain 4G topologies. These evolved topologies are sustained by third parties as close to the edge of the cellular access as possible (i.e. Mobile Edge Computing -MEC-, service provisioning at the edge, RAN caching or fog computing among others [18, 19]), dramatically reducing the end-to-end latency.

Therefore, the evolution of the mobile networks technology not only indicates a clear tendency to higher throughputs, at least ten-folding the peak data rates of previous mobile network generation, but also presents a massive reduction of latencies in order to help minimize the time it takes data to transit the network.

Regardless the improvements in the radio part, [1] shows the actual experienced delays of the different technologies themselves (the RTT within the UE and the mobile core network gateway) without considering the time data takes to travel from the content source through Internet. In the case of LTE, with theoretical latencies in the radio part around 10ms, the reality prompts latencies between 68ms to 85ms in the third quarter of 2015 in USA [20]. Therefore, the end-to-end delays show values at least 5 times greater than theoretical values of the the radio technology. The reasons are manyfold: geographical distance, processing, queuing in intermediate buffers, scheduling techniques and a long etcetera. Those figures regarding end-to-end latencies make the performance playground even more challenging for transport protocols. The more time it takes a packet to be served and acknowledged to the source, the more time it will take the server to react and adjust to the mobile network conditions.

The time interval between each significant technology evolution has been about ten years. Each generation has brought important improvements comparing with the previous one and the evolution pace seems to be equally optimistic for further generations. Figure 2.2 depicts "*the timeline of technology generations, including past and future, showing initial deployment, the year of the peak number of subscribers, and decline. Each cellular generation spans multiple decades, with peak adoption occurring some 20 years after initial deployment*" [1]. So, when new technologies are proposed, the previous generation still has around 10 of the best years in terms of adoption and number of subscribers. Therefore, 4G is still relevant from the research point of view and the possibilities that still brings to end-users. Regarding 5G, taking into account the research and standardization momentum in which most of the work is yet to be fulfilled, based on applicability, realistic representation of real-world and availability of testbeds, we have decided to focus our study on 4G with a use-case of low latency that mimics forthcoming 5G deployments.

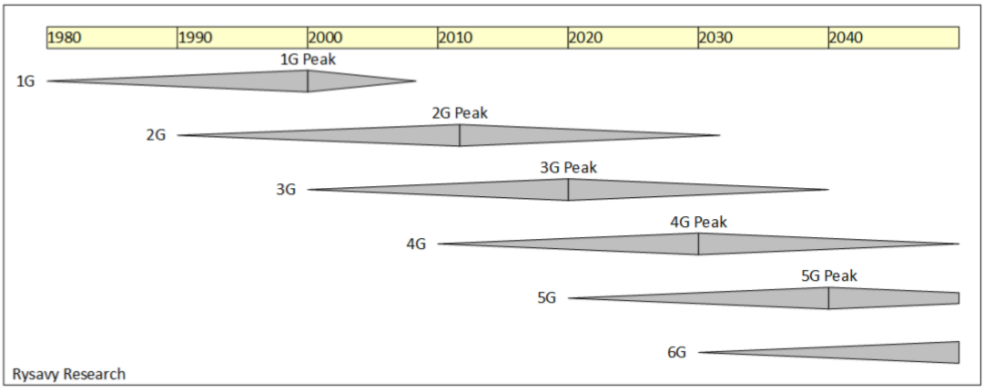


Figure 2.2: Timeline of Cellular Generations [1]

2.2 TCP protocol overview

Once that mobile networks have been reviewed and in order to explain the roadmap of TCP, this section describes the basic concepts of TCP as well as the early developments of TCP flavours.

2.2.1 Basic concepts of TCP

The brief explanation of networking basis are important in order to define a TCP background that would allow a better understanding of TCP and its implications, as well as familiarize with TCP's terminology and its different mechanisms and additional processes. Most of the management procedures in TCP are dedicated to handle the Congestion Window (CWND) variable that limits the amount of data that TCP is able to send at any time. The limitation of CWND will be always be clamped by the minimum between the calculated CWND by the CCA and the advertised Reception Window (RWIN) related to the flow control. This RWIN represents the amount of data that the receiver is able to receive at certain point. TCP also controls the packet flow by limiting the number of unacknowledged packets that could be in-flight. Every transmission has different phases and depending on them, the management mechanism of the CWND is totally different. The basic phases of TCP are defined in [21]:

- Slow Start: During this period the CWND typically increases by one packet for each Acknowledgement (ACK) reception until it reaches *ssthresh* value. After that period, the CWND enters a Congestion Avoidance (CA) phase. Another way of ending the ramp-up before the achievement of *ssthresh* is by detecting a congestion event or packet loss event with the reception of a triple duplicate ACK (3DUPACK) or a time-out. The DUPACK events are generated by the receiver when it receives out-of-order packets. After the 3DUPACK, the back-off policy is applied, halving the CWND and fixing the new *ssthresh* to previous CWND. However, if a time-out occurs the CWND will be decreased to one packet and the *ssthresh* to half of previous CWND so that it can enter congestion avoidance once it gets that value.

- Congestion Avoidance: A phase where the increment of the CWND is reduced to typically one packet per RTT period (standard synchronization with RTT or RTT-synchronized). The same loss event and time-out detection mechanisms and back-off are utilized. Due to the incremental and decremental pace during congestion avoidance phase, basic TCP CCAs are also known as AIMD (Additive Increase Multiplicative Decrease) mechanisms.

- Fast Retransmit: In TCP, the arrival of 3DUPACK is assumed as a sign of a lost packet. Therefore, TCP resets the transmission timer and retransmits the missing segment.

- Fast Recovery: The mechanism works together with Fast Retransmit. After receiving a 3DUPACK, establishing the *ssthresh* to half the value of CWND and retransmitting the lost packet, the CWND is set to *ssthresh*+3. This way, the CWND tries to reflect the arrival of packets that have not been lost and have caused the DUPACK. Afterwards, after each additional DUPACK for the same segment, the CWND is incremented in 1. When an ACK arrives to acknowledge previously unacknowledged data, TCP sets the CWND to *ssthresh* and follows the normal methodology of CA phase.

With the time being, many changes were introduced in TCP in order to solve detected deficiencies. The most important ones are still enabled by default in the off-the-shelf Linux equipment, the most usage operating system [22] with a 67% of the market share, and are the following:

- Window Scale / Window Scaling (WS) [23]: In order to better represent and manage the increasing buffer sizes in end-points, the WS field enables the expansion of TCP windows from a limiting management with 16 bits to 30 bits. Therefore, the option modifies the ability of TCP and allows working with buffer sizes greater than 64 KB up to 1 GB. The WS option is negotiated and sent between end-points in the TCP handshake and it would be, throughout the whole transmission, the representation of the maximum manageable capacity of end-points.

- Timestamps [23]: This option is symmetrically introduced in both data and ACK packets. It could be utilized to measure the timing of individual packets as well as whole-transmission-related timestamping. The Timestamps option is also used to measure the RTT and establish the Retransmission Time-Out (RTO) timer in accordance. "*Accurate and current RTT estimates are necessary to adapt to changing traffic conditions, while a conservative estimate of the RTO interval is necessary to minimize spurious RTOs.*" [23]. Besides, Timestamps as a synchronization enabler, help end-points realize about the segment misinformation of the other side.

- Selective ACK (SACK) [24]: Before the addition of SACK to TCP, TCP was only able to be informed about one lost packet per RTT. Therefore, the sender started to retransmit packets from the duplicated segment number on. However, some of those packets may well be already properly transmitted. To cope with this performance issue, SACK allows the reporting of segment number holes or lacking packets in order to make the retransmission easier and more accurate. Apart from that, the SACK field could also be used to inform between end-points about spurious retransmissions.

- Conservative SACK [25]: This mechanism enhances and replaces the loss recovery of basic TCP by using SACK. Therefore, the enabling and use of SACK is mandatory. Instead of modifying the CWND using Fast Recovery mechanism, the sender is able to know how much data is in-flight and decide whether it can transmit new packets or not comparing with the CWND.

- Forward ACK (FACK) [26]: In error recovery mechanisms a conservative procedure is to consider all the unacknowledged packets, segments in-flight. This is, if no information, either positive or negative, has been reported regarding a specific segment, it would be labelled as pending or in-flight packet. However, FACK method proposes a more aggressive approach by considering all the unacknowledged packets as lost segments that need to be retransmitted. The aggressiveness of this method has reported an overall important performance benefit.

- Eifel algorithm [27]: If the receiver sends a DUPACK with a Timestamp earlier than the one controlled by the sender, the DUPACK will be treated as misinformation of the receiver and the sender will continue sending new packets.

- Initial RTO of 1 sec [28]: Decision on reducing the default RTO from 3 seconds to 1. The establishment of 1 second fulfilled both requirements of being large enough to avoid spurious RTO and short enough to realize about a precise loss if no ACK arrives.

All the abovementioned improvements are present in the source code of Linux kernel and establish the base behaviour of currently available TCP flavours. Some of them are disabled because State of The Art solutions have prevailed over them. In the same way that classic Fast Recovery and Fast Retransmit phases were once replaced by improved solutions for loss recovery events such as SACK or FACK, FACK has been replaced as well in 2017 (explained in next Chapter that covers the description of the State of The Art situation). Even though some mechanisms are not currently being used, the brief explanation of them helps to better understand the background of classic TCP and the progress and development over the first years.

2.2.2 Early development of TCP flavours

Once the basic functionalities of TCP have been described, a brief temporal review of TCP flavours is compiled in this Section. Considering that most of the connections during early years of TCP were significantly stable, the CA phase and its enhancement used to be more meaningful for the overall performance than any other modification. Due to this, many efforts were dedicated towards the development of many different CCAs. Even though all CCAs aim to give a general solution for TCP, most of them assume certain networking problems and are therefore applicable for certain network conditions only. Therefore, each CCA reports improvements under a selection of network conditions (such as high delay, big bandwidths, targeting friendliness

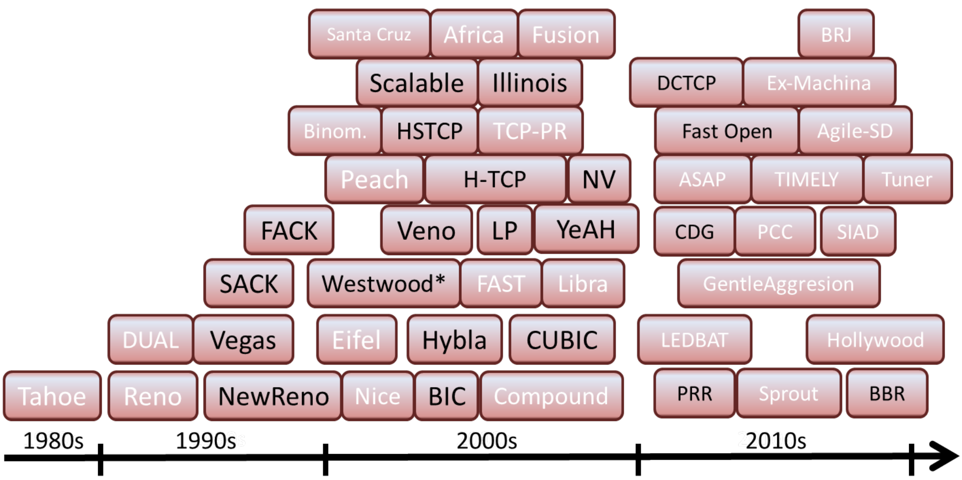


Figure 2.3: Evolution of TCP throughout the decades

among flows through the same bottleneck, with hybrid loss-based and delay-based methods and so forth [29]) but struggle in other environments.

Throughout last decades many CCAs have been proposed with declared benefits. Taking the original idea from [30], Figure 2.3 depicts distinct CCAs in a time axis. Two important things need to be clarified beforehand: 1) TCP mechanisms such as SACK and FACK appear as individual CCAs because in that period of time they were treated as completely new CCAs. 2) All the mechanisms and CCAs available in the off-the-shelf Linux kernel [22] have the name in black color. The rest, in white, are proposals that have had or still have notoriety but have not been included in the kernel. The subsection does not try to explain the different alternatives but raise the names and effort that have been dedicated to TCP during the last decades. Some of the most recent deployments will be accordingly covered in the next chapter in order to explain the State of The Art.

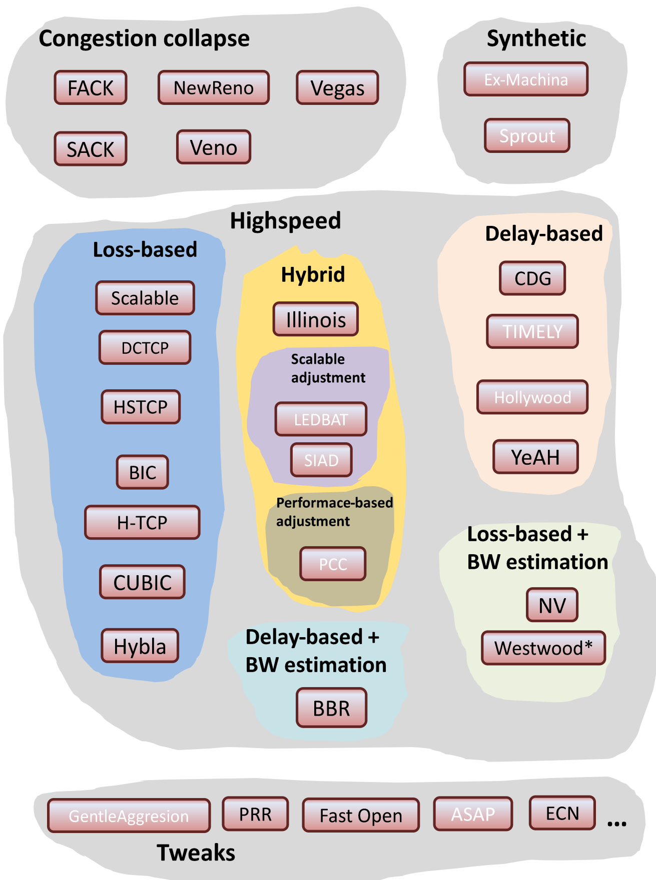


Figure 2.4: Classified bibliography of TCP. Original idea from [2].

Several studies [2, 9, 29, 31, 32] have highlighted the fact that the development focus has changed with the time being due to the evolution of network capabilities and therefore, the requirements. Starting from the basic idea of avoiding congestion events to the willingness to take full advantage of available network resources under different network conditions (fixes networks, wireless links, high delays, high speed and big bandwidth-delay product -BDP- and so forth). Thus, every CCA has its own networking goal and singular internal features. Even though the outcome of them is different, all CCAs could be grouped in conceptual characteristics. Figure 2.4 shows an adaptation of the CCA classification made by [2] regarding the features of TCP flavours and congestion control techniques.

In relation to mobile networks and considering the requirements that TCP needs to fill, three groups are identified:

1. The Loss-based CCA with bandwidth estimation that could be appropriate for wireless communications and, more precisely Westwood due to its presence in Linux and its benefits in air communications [33].

2. Hybrid CCA solutions that could not only react to losses but also taking into account the delay increment. These solutions could be useful due to the variability of bandwidth and delay in MBB and their potential detection of such variations. In this Hybrid group, Illinois is present in Linux.

3. Delay-based option with bandwidth estimation.

4. Common loss-based solutions that are aggressive enough to overcome the constrained features of cellular access.

If we take into account that the presented developments have been made merely based on the weaknesses of fixed networks in which TCP faces a much more stable network condition comparing with MBB, we would understand why the interaction between TCP and mobile networks is much more challenging. This Section has covered the evolution of TCP from its early stage and going through several improvements and CCA proposals until just before the State of The Art situation. Moreover, the Section has served as a brief tutorial of TCP evolution in order to allow a better understanding of next chapters.

3 State of The Art

This chapter covers the review of most relevant recent proposals of transport protocols/services, the specific issues in MBB and those improvements based on awareness between mobile networks and transport layer with higher impact.

The state of the art is divided in 4 sections which comprise the following topics:

- Section 3.1 briefly introduces several general purpose transport approaches. Each of them tries to give a solution for different networks or network conditions inside the same deployment by either proposing a completely new method, focusing in a very specific mechanism, targeting the avoidance of *bufferbloat* with queueing mechanisms, suggesting a computer-based synthetic CCA creation or building transport services on top of classic transport solutions.

- Section 3.2 explains the State of the Art of research works in the transport protocols over mobile networks and defines the specific issues.

- Section 3.3 provides several cross-layer proposals to improve the interaction, signalling and communication between mobile networks and transport protocols.

- Finally Section 3.4 presents the conclusions of the review of the State of the Art by analyzing current issues in MBB and the awareness efforts between cellular access and the transport layer, together with the capability of those efforts to solve the mentioned issues. In this regard, the Section also defines the issues that are still opened giving a research focus to advance in their avoidance.

3.1 Current general purpose transport approaches

Many proposal have tried to face challenges resulting from the evolution of data networks. Figure 3.1 classifies to **classify in which layer of the protocol stack they focused at**:

- Point (1): Some of them have completely **modified known TCP flavours**, suggesting other way to behave and react.
- Point (2): Others have focused their effort on **specific mechanisms or tweaks** that could be added to previous developments.
- Point (3): In the same way, there have appeared alternatives that defend **better Active Queue Management (AQM) techniques** in lower layers of end-nodes and intermediate routers, proxies and middleboxes.
- Point (4): Besides, different proposals have suggested the possibility of **TCP modelling and synthetically generating new CCAs**.
- Point (5): Finally, many research efforts have been dedicated toward the definition and first stage testing of **transport services**.

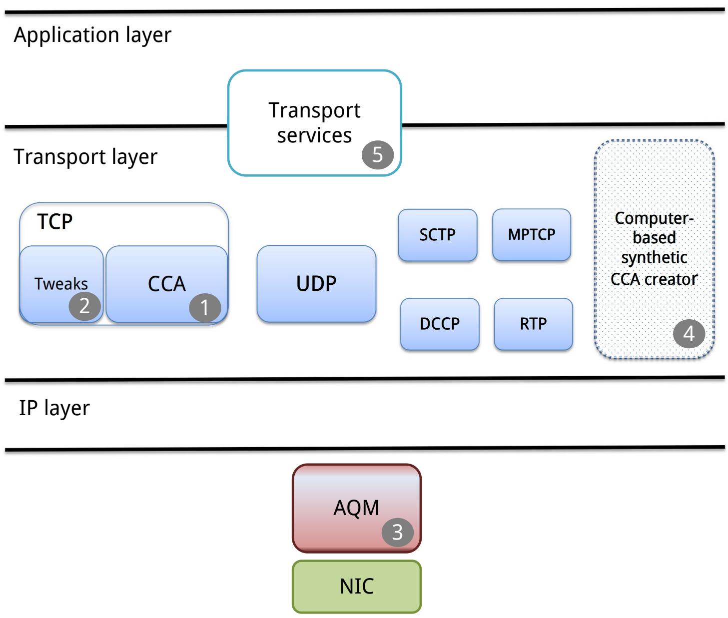


Figure 3.1: State of The Art advances regarding congestion control protocols/services

3.1.1 New CCA approaches

The aim of this subsection is to provide several examples of TCP CCA approaches

(Point 1 in Figure 3.1) so as to underline how different a CCA could perform to achieve a particular goal. The subsection also discusses the usage and adoption of those proposals as by-default alternatives. The following descriptions comprise the explanation of such TCP alternatives:

- Delay Gradients: Considering the bursty behaviour of RTT and in order to provide a less noisy congestion indicator based on RTT, CAIA delay gradients [34] were proposed. Looking for low delay and high throughput, the use of delay gradients has also been considered for datacenters in TCP TIMELY [35] as sufficient control signal to adjust the sending rate. Even though some datacenters may well improve their network performance with the use of delay gradients, there still exist doubts regarding the use of such congestion signal in wider and more complex deployments. Some high-speed TCP alternatives [36] use "smooth samples" of RTT as the second control parameter but do not let delay gradients entirely drive their CCA. Therefore, depending on the performance target, delay gradients could serve as a primary parameter (focused on delay) or in an hybrid CCA to provide with more accurate information about the end-to-end delay.

- Performance-oriented: Skipping from the classic TCP "hardwired mapping" which predefines the CWND behaviour after precise packet-level events, Performance-oriented Congestion Control (PCC) [37] proposes a evidence-based control mechanism. The key idea is to observe basic metrics for a proper performance (sending rate, latency, loss rate and so forth) and as a consequence decides the control response. Therefore, it is able to rapidly adapt to variable network conditions. However, it also suffers the impact of bad decisions or non-accurate measurement of parameters. All in all, it is an alternative that requires improvements but it is considered as a powerful alternative in future internet (different switchable congestion control options will be proposed [38] and the evolved version of PCC will be part of such possibilities [39]).

- Different-AIMD: Classic AIMD mechanism has shown that it struggles when it comes to rapid adaptation to highly variable environments. For that reason, various proposals have emerged to address the issue and try to cope with it:

    1. LEDBAT [40] is a standardized CCA that aims to utilize the available bandwidth but limiting the increase of the queueing delay. To that end, LEDBAT modifies both sender and receiver so as to enable the notification of one-way delay (OWD). The receiver measures the one-way delay considering its system time and timestamps in the received data packets. Afterwards, the receiver send its OWD assessment in the ACK packet. With the information of the OWD, the sender will decide whether it should increase its CWND proportionally to the relative difference between the current queueing delay and the target delay or decrease the CWND proportional to the difference between the target and the current queueing delay. Due to the delay-aware mechanisms, LEDBAT has shown more aggressiveness than the Standard TCP when the estimation of OWD is lowered. However, one of the weak points of LEDBAT may well be the requirement for receiver modifications, which it is not always straightforward due to the variety of devices.

2. TCP SIAD [41] also presents an OWD-based scalable increase and adaptive decrease alternative to solve this issue and be considered in future developments. As with LEDBAT, the strong requirement of modifications in the receiver could put aside these proposals.
3. A new utility-based CCA called Delay-Constrained Congestion Control (DCCC) algorithm [42] uses non-linear mapping between the measured delay and the control mechanism applied in the CWND update (same concept as LEDBAT and SIAD). It also combines delay and loss feedback information in a single term congestion signal, based on packet inter-arrival measurements. According to the primary results, DCCC helps mitigate unfairness among different base latency users within the same path. Nevertheless, this alternative seems to have limited adoption in the networking and research community.

- Datacenter TCP (DCTCP) [43] constitutes a congestion control for datacenters. Instead of detecting congestion with loss-based mechanisms, it takes advantage of the Explicit Congestion Notification (ECN) of intermediate routers to measure the amount of traffic that is suffering congestion. Considering the estimation of congestion, DCTCP manages and scales the CWND. The congestion detection method is capable of handling high packet bursts, low latency and high throughput even with the presence of short buffers, also known as shallow-buffers. These kind of buffers, reduce the maximum queueing delay delay in comparison with deep-buffered counterparts. However, they also tend to trigger high packet loss under bursty traffic conditions. TCP DCTCP is capable of reducing this harmful impact by ECN marking, detecting the packet congestion prior to the overflow and thus, reducing the overall delay and number of loss events. Currently DCTCP is applicable to deployments in controlled datacenters environments. However, DCTCP is not able to appropriately coexist with classic flavours of TCP. Therefore, for now, it should be avoided the deployment in wider and public Internet areas until it is safe to run alongside the rest of the TCP CCAs.

- TCP BBR (Bottleneck Bandwidth and RTT) [36, 44] widely differs in the management of the CWND comparing with the rest of the CCAs. It relies on the measured baseline RTT and the timing and rate of ACKs so as to infer if the packet injection rate is below (being able to raise the CWND even more) or over (starting to build-up the bottleneck queue with packets and increasing the delay [45]) the Bandwidth Delay Product (BDP) of the bottleneck. Another important modification of BBR is the inclusion of *pacing* in the sender. To this end, BBR needs the assistance of a packet scheduler that implements the feature of *pacing* (for instance Fair Queue -FQ-) or the fall-back and native *pacing* option of TCP layer itself. BBR has emerged with high deployability and promising primary results [36]. Due to these reasons and considering the possible future improvements, this dissertation considers BBR as one of the TCP alternatives.

- Tuning: Sometimes the development of something new in relation to TCP does not require a completely new method, but a different perspective. TCP Tuner [46] is the first GUI tools to tweak TCP CUBIC parameters on-the-fly enabling a deeper research and understanding of the protocol and related behaviour. Even though the proposal could not be part of off-the-shelf Linux implementation, it could serve to investigate the most adequate CCA configuration for a precise network area or circumstance and be able to adapt the CCA in future deployments.

All the abovementioned proposals and enhancements show that even after more than three decades from the first TCP version, there are plenty of ongoing efforts dedicated to the evolution of transport protocols. We have revisited a tuning alternative that despite the fact that is not widely deployable, it could at some point give troubleshooting ability to a certain provider over a selected network area. Besides, PCC has been explained, underlining the importance that the evolution of this alternative may well have in the future Internet. In addition, DCTCP, LEDBAT, SIAD and DCCC have been covered prompting deployability issues. All proposals are incomplete to be widely adopted either due to the lack of ability to coexist with other TCP flavours or due to the necessity to modify the receiver. Finally, this thesis considers BBR a good and novel TCP alternative that requires deeper analysis over distinct network circumstances.

3.1.2 Improvement of specific TCP mechanisms

Once the current State of the Art of CCAs have been explained, this subsection will cover the explanation and discussion over several specific TCP mechanisms (Point 2 in Figure 3.1). The improved mechanisms fall in five different groups: focused on connection establishment ad initialization, the Slow Start method, loss recovery and a brief explanation of ECN as congestion notification. Throughout the explanation, it will be highlighted whether the proposals are included in the Linux kernel used in the dissertation.

3.1.2.1 Connection establishment and initialization

The Initial Window (IW) value defines the number of segments in CWND utilized at the beginning of every transmission. It has been a topic of discussion since 1998 [47] when it was demonstrated that under some boundaries (less than 32), the higher the IW value, the faster completion time. However, later studies [48] found that "very large" IW may well have an impact on overshots, self-inflicted losses, the network load and therefore the delay induced to other flows. These efforts were pursuing the final standardization of ten segments (IW10) [49], leading to the default IW used nowadays. Even though the default value is 10, Dukkipati et al. suggested that "Future work should focus on eliminating the initial congestion window as a manifest constant to scale to even large network speeds and Web page sizes". In this regard, various proposals have been reported trying to allow: 16 segments in the IW [50], a variable IW [51] that could be applied in near and further future with different network features or a safe increase of the IW by spreading the packets (large IW and pacing) [52, 53]. All in all, it is something that still remains without a standardized solution that could fit in every network situation. In fact, every Content Delivery Network (CDN) is trying to build infrastructures closer to clients so as to reduce the RTT perceived by end-users. CDNs have its own and non-standard-compliant configuration [54] that seems to be very different between each other and they could have an unfair behaviour in the network among the rest of the users. Considering the difficulty to reason a change in the IW that could successfully cover most network cases, this thesis uses the default IW10 of Linux.

TCP Fast Open (TFO) [55] mechanism allows the transit of data in the three-way handshake (3WHS) and saves one RTT while establishing the transmission. However, TFO has to be accepted and compatible with both end-points. The key parameter of TFO is the use of a so-called cookie working as an authentication code. The first time a client establishes a connection with a certain server, the client requests a cookie. This cookie will be enabled in further TCP connections that target the same server. This dissertation uses a Linux kernel with TFO enabled by default. Even though the speed gain is clear during the connection establishment, the work does not aim to include this tweak in the analysis for two reasons:

1. The performance of different CCAs is expected to be measured at the full potential of the CCAs themselves but avoiding mechanisms that could make a difference between consecutive experiments or CCAs.
2. The purpose of this dissertation is to analyze the behaviour of CCAs in real-world. The connections between the same pair of client and server (due to the necessity of the TFO cookie) are limited and do not represent a wide reality.

3.1.2.2 Slow Start methods

Once the IW is established, the Slow Start mechanism starts to play its role in all TCP connections. At the last stage of the mechanism and about to enter CA phase, it is well-known that Slow Start mechanism has a suboptimal performance over high BDP networks, dramatically overshooting and resulting in thousands of dropped packets. In order to avoid this constraint, many proposals have claimed to have the solution:

a) Considering the impact that the lost of a large burst of packets may have on TCP behaviour, there is a possibility to limit the CWND [56] for connections that are expected to suffer from such an effect.

b) Another TCP modification takes into account the ability to communicate between the sender and the routers over the path so as to decide the appropriate sending rate (so-called Quick-Start [57]). Every tweak that has to pass through multiple actors of the scene, it is unlikely to be completely approved by every part. However, the approval could led to a boost in the mechanism.

c) Being aware of how complex is for Slow Start to perform according to the available rate while avoiding long series of losses, there is a proposal that analyses the behaviour of TCP without Startup phase (known as Jump-Start [58]).

All aforementioned proposals are good options to partially enhance the Slow Start mechanism, but rely on the same principle of losses detection and are not capable of measuring the impact of in-flight packets on queueing delay. Following this assumption, Hybrid Slow Start [59] was proposed in order to add to the Standard Slow Start the ability to assess delay (either by the length of ACK trains or by pure increase of delay that surpasses a certain threshold). It was a very successful option for high BDP networks and therefore, it was introduced as the Slow Start mechanism in CUBIC, which by that time was the default CCA in Linux kernel.

Considering the amount of different proposals in the specific field of the Slow Start phase, a comparative study of them was carried out [60]. According to the results, Jump-Start causes some unfairness to competing flows using Standard Slow Start. Quick-Start requires router support, which could be an unsolvable issue. More aggressive alternatives could cause harm in the performance of TCP. All those results could be aligned with the performance of Hybrid Slow Start in order to understand its adoption.

The recent proposal of BBR [36, 44] has brought a new Slow Start approach called Startup. During the Startup phase BBR ramp-ups as the Standard Slow Start method but it does not wait until a loss occurs. On the other hand it exits the increase when the achieved throughput gain is below the 25% throughout three consecutive RTTs. BBR waits three rounds to be sure that there is no limitation of the rate due to misperformance of the receive-window. Three rounds allow the receiver's receive-window autotuning mechanism to open the window for incoming traffic and the sender to capitalize the available bandwidth increment [44].

This work considers three different Slow Start alternatives due to their inclusion in off-the-shelf CCAs of the Linux kernel: The Standard Slow Start, the Hybrid Slow Start and the Startup phase of BBR. Other approaches propose different mechanisms but are based on the same loss-based principle of Standard Slow Start and are not worth-testing.

3.1.2.3 Loss recovery

Since TCP relies in the end-to-end principle, there exist challenging conditions such as the loss recovery events in which the synchronization of both end-points is perfect. For this reason, different improvements try to add heuristics to the loss and recovery treatment in order to cover the events in the best possible way, only recovering the packet that need to be recovered and wasting the least possible bandwidth.

Early Retransmit [61] method allows TCP to trigger fast retransmit mechanism faster when the CWND and the outstanding data are low. This way, TCP is able to recover rapidly comparing with the situation in which it would require to wait until time-out.

Proportional Rate Reduction (PRR) [62] does not use Fast Recovery and typical 0.5 back-off. PRR examines the amount of data per ACK that needs to be sent during recovery and avoids excessive window adjustment, finishing the recovery as close to *ssthresh* as possible.

Forward RTO-Recovery (F-RTO) [63] collaborates with Timestamp and SACK to face spurious retransmissions. Spurious retransmission events happen when a segment is correctly acknowledged by the receiver but the ACK does not reach the sender before the retransmission of the packet due to RTO. The mechanism stores the information of the first retransmitted unacknowledged packet that has been launched due to a time-out event. Helped by the following ACKs, F-RTO decides whether the previous time-out has been a legit time-out or a spurious one. In consequence, it continues retransmitting packets or starts sending new segments.

Tail Loss Probe (TLP) mechanism modifies TCP [64] allowing a quicker recovery of lost segments. When a sender does not receive any ACK for a precise period of time, TLP sends the last unacknowledged segment (loss probe). "In the event of a tail loss in the original transmissions, the acknowledgment from the loss probe triggers SACK/FACK based fast recovery. TLP effectively avoids long timeouts and thereby improves TCP performance" [64].

Recent ACK (RACK) method [65] is the newest algorithm that checks the timestamps of packets instead of counting DUPACKs. The idea is to retransmit the lost packets faster than waiting until a 3DUPACK event. The timestamps allow inferring whether the timestamps of some previously sent packets have exceeded a certain reordering threshold or not. If one packet is lost for RACK, RACK would mark that packet so as to be retransmitted. RACK uses TLP to send probe messages and "receive" information from the network and repair the lost segments as fast as possible while avoiding the recovery through RTOs. Since Early retransmit, PRR, F-RTO, RACK and TLP are enabled by default in the Linux kernel, this thesis keeps them working in order to fairly resemble the performance of each CCA.

In the last years other loss heuristics have been proposed. The main proposals are two: 1) RTO Restart method [66] was proposed in order to make faster the loss recovery in situations with small amount of packets in-flight for a precise connection. With the restart and assignment of a smaller timeout, "effective RTO becomes more aggressive in situations where fast retransmit cannot be used", improving short-lived TCP connections and rate-limited flows. A recent study was carried out [67] to evaluate to which extend RTOR and TLP reduce tail loss latency. Following a mixture of concepts, 2) TLPR was presented, applying the logic of RTOR to the TLP timer and was evaluated among the other options. According to the results RTOR reduced tail loss recovery duration in one RTT comparing with Standard TCP. TLP showed even larger gain compared with RTOR in many deployed scenarios. TLPR integrates the strengths of both mechanisms and performs the best in most cases. However, neither RTOR nor TLPR have shown successful and consistent enough results to be included in the Linux kernel. For the same reason, this work does not consider these alternatives.

3.1.2.4 Congestion notification

From the beginning of the definition of TCP, the protocol was structured to be end-to-end and only receive information of the network status sent by the receiver and encapsulated in ACKs. However, due to the complexity of technology and deployments and taking into account the differences between distinct network areas, the end-to-end interaction seems non-sufficient. In order to help TCP sender detect probable losses beforehand through a signal, the Explicit Congestion Notification (ECN) [68] was suggested. This way, network middleboxes, routers and end-points could notify that a loss is about to happen so as to adapt the sending rate through the CWND control mechanisms. However, ECN should be supported throughout the whole path, which does not seem to be the case right now. There is a deep problem regarding the deployment of such enhancement in Internet-Wide [69]. Even though ECN support is increasing among web-servers, there are still many limitations due to the existence of ECN-unfriendly middleboxes.

Even in a path with ECN support at every network hop, there are two main uncertainties: 1) There is no definition regarding how should the sender react to the reception of a packet with active ECN flag. There are many possibilities: each CCA should develop its ECN adaption, the adaptation is the same no matter the CCA under use, the adaptation is scalable depending on the rate of ECN receptions. 2) It is not clear how a ECN-enabled TCP flow and a non-enabled TCP would coexist. Even though ECN has a great potential, due to all abovementioned doubts and considering that ECN and ECN-friendly CCAs still need a long process to clear-out the uncertainties, this dissertation does not consider this feature.

3.1.3 Queueing mechanisms

One of the main deficiencies of TCP is the self-inflicted delay due to overshooting events that lead to packet losses. Looking for latency reduction and as a complement for end-to-end CCAs, AQM methods have to be seriously considered (Point 3 in Figure 3.1). In fact, a survey [70], which gathers the most significant efforts to this respect until 2013, categorizes AQM as a global, responsive and closed loop control inside congestion control schemes' taxonomy. Recent advances claim for AQM mechanism to reduce the effect of *bufferbloat* in the network [71]. The study presents some "best configuration practises" (offloading disabling, appropriate switch usage, all buffer limits' awareness and so on) that are important to consider by developers as well as the potential of a new hybrid scheduler/AQM mechanism called FQ-CoDel (Flow Queue -Controlled Delay) [72] for the reduction of latency and enabling a method to overcome *bufferbloat*.

It is interesting to notice that end-to-end CCAs and AQM scheduling algorithms work to get the same targets: low latency, high throughput, low losses and so on. However, the effort that an end-to-end CCA could be doing in terms of latency reduction may well be ruined by an AQM mechanism and vice-versa. Real-world measurements always hide this uncertainty, not being able to know whether the outcome is the best possible result or either the CCA or the AQM mechanism have impacted on the final performance. But in general, both proposals have their own and combinable behaviour with a great potential.

When referring to AQM, most of the time it is used in intermediate buffers such as routers, middleboxes and so forth is considered. However, the AQM policies could be also applied in the sender before the injection of the traffic to the Network Interface Controller (NIC). Regarding the injection ability, it has to be noted the established limitation with TCP small queue method [73] that controls the bytes that are transiting in lower than TCP layers and have not been submitted as outstanding data. Thus, TCP limits the data that is kept stored in internal buffers and reduces the overall induced delay, improving this way the overall performance as well as helping the task of AQM mechanisms.

In this thesis, there is no especial consideration of AQM mechanisms in intermediate nodes and the presence of them in non-controlled environments is seen as a positive fact. Regarding AQM scheduling algorithms in the sender, it is applied on-demand of the CCA. This is, all CCA before the existence of BBR do not use AQM methods. Nevertheless, BBR requires a *pacing* technique and the developers suggest the use of FQ scheduling algorithm in the sender in order to achieve the full potential of BBR. Therefore, only BBR works with the FQ AQM policer and the rest of the CCAs are let as they are.

3.1.4 Learnability of congestion control algorithms

A new approach in transport protocols considers the creation of CCAs by machines with little interaction with humans (Point 4 in Figure 3.1). Three main works are going to be commented.

[74] introduces Remy, a CCA generator for multi-user network and single bottleneck networks. Designers must specify assumptions or knowledge regarding the target network features in order the generator to define its objectives. One of the main conclusions of the results is that, when the actual network performance violates previous assumptions, the CCA generated by Remy suffers. However, in a more consistent network, the resultant CCA outperforms humanly created CCAs. In conclusion, this work opened a new branch in CCA creation field with surprisingly good results for stable networks.

[75] was dedicated toward the application of previous work over Cellular Network traces, which is known for its variability. Sprout was presented as the generated CCA for interactive applications that aims at having high throughput and low delay. Sprout's receiver observes the packet arrival times to deduce the dynamics of the network path and therefore to have the ability to "forecast" how many bytes should be sent by the sender without causing too long queueing delays (with a defined risk of not achieving this target in the 5% of cases). According to the results, Sprout works well over cellular networks when variability only comes from the available link, but performs poorly when cross-traffic is present, therefore a traditional TCP cross-traffic flow would have a substantial impact. To conclude, Sprout seems to be a promising trace-based responder with outstanding results, but with little ability to generalize its behaviour in a real-world environment.

[76] gathers all the experience from the two previous works and seeks for clarifying the suitability of learnability in CCAs and their machine-based generation. Even though it is stated that probably the conclusions cannot be fully applicable in order to solve the design problem of CCAs, human-designed protocols can learn from the provided insights. In this way, human CCA designers can take advantage of the factors that Remy has underlined as critical so as to focus on those influence factors and simplify the rest.

The presented vision is slightly different comparing with previous explanation. As mentioned before, it could provide a CCA designer or researcher with the influence factors of a certain network area or topology, but denotes deployability issues. For this reason, the current work does not consider these alternatives.

3.1.5 Transport services

The concept of transport services applies for inclusion of ad-hoc layers that work between the transport layer and the application layer (Point 5 in Figure 3.1), taking advantage of the substrate transport protocol (mainly TCP and UDP) and gaining some freedom due to its development in the user-space of the OS and additional functionalities (i.e. congestion control over UDP, improved communication between application and transport layer, multiplexing) [11–13]. Transport services have been built as a response to the ossification of the transport layer. The ossification comes due to:

1. The lack of the opportunity to select and modify transport layer protocols both at the beginning of a certain transmissions and in-flight.
2. The lack of room for transport protocol innovation.
3. The lack of flexibility of the API to communicate the application layer and the transport layer.

Currently, there are three different approaches that take advantage of transport services. On the one hand, those alternatives that, due to the success of TCP and UDP, use those transport layers as a substrate to build improved mechanisms on top of them. On the other, some modifications in the API between the application layer and the transport layer propose to select the most adequate transport protocols based on the application-layer requirements.

In order to better explain the three approaches of transport services, the subsection is divided in four: TCP as a substrate, UDP as a substrate and the selection of transport protocol based on application requirements.

3.1.5.1 TCP as a substrate

One the most notable modifications regarding the transport services over TCP is the proposal of SPDY by Google. The main target is to boost and improve the Web traffic [77], but suggests features that could be applicable for more contexts of use. The main changes try to solve uncovered weaknesses in the interaction of TCP and HTTP and are summarized in the following list.

- Pipelining: SPDY allows opening a single TCP connection to a domain and multiplexing different HTTP requests through the same connection. The use of a single connection reduces overheads (such as Secure Socket Layer -SSL-) and therefore helps achieve higher application-layer goodput.
- Header compression: SPDY supports HTTP header compression. It has been demonstrated that some headers are being continuously duplicated (e.g. information related to User-Agent) over the connection and that could be avoided.
- Server push: In order to reduce the latency and save some RTTs, SPDY has the ability to send HTTP objects even before they have been requested. Even though the feature could be useful under certain circumstances, there exist network situations in which it is not recommendable to push content beforehand without a pre-knowledge of the network status.

Considering the advance that could introduce in Web traffic and being aware of the drawbacks to overcome [78], SPDY became the foundation for the standardization of HTTP/2 [79] as an evolution of HTTP that takes advantage of transport services. With many companies putting their effort into HTTP/2 and its deployment spread, it looks as a strong candidate to the Internet that is about to come. At the present time, HTTP/2 has been conceived to work with TCP in the transport layer. Even though there exist efforts towards the adaptation of UDP-based transport services (e.g QUIC - explained in the next subsubsection) to HTTP/2 through an ad-hoc API, TCP remains as the reliable solution. Moreover, the multiplexing feature helps in low bandwidth and high delay situations with the avoidance of costly 3-way handshake comparing with HTTP/1.x.

3.1.5.2 UDP as a substrate

Regarding UDP-based transport services, it is important to mention QUIC. Developed by Google, it opens the possibility of having TCP-alike features as reliability and congestion control among other options on top of UDP. Its main contributions [11] are:

- **0-RTT connection establishment**: As UDP-based, QUIC does not require any 3WHS mechanism to start a communication. In order to avoid security issues, the first time that a QUIC client connects to a server, 1-RTT handshake has to be performed in order to acquire the information that later on will be used as a certificate (similar to the Fast Open cookie).
- **Novel Congestion Control**: QUIC uses a TCP CUBIC-alike congestion control but also considers other alternatives as Reno, PCC or BBR. However, regardless the congestion control under used, the mechanism shows some differences with the standard behaviour of CCAs. For instance, every QUIC packet has a different sequence number and explicitly carries the RTT parameter. Both mechanisms allow QUIC to distinguish between original and retransmissions, finishing this way with sequence number ambiguities.
- **Different multiplexing**: QUIC avoids TCP's Head-of-line (HOL) blocking. When multiplexing techniques are used with TCP, there is an issue with retransmission. The loss of a TCP packet leads to the block of all subsequent packets that need to wait until the retransmission is successfully accomplished. In QUIC, if a packet is lost, only the stream responsible for that lost will wait for the retransmission and the rest will continue their normal evolution.
- **QUIC uses Forward Error Correction (FEC)** packets to avoid retransmissions in loss recovery periods. If a packet in the group is lost, the contents of that packet can sometimes be recovered from the FEC packet and the rest of the packets in the group.

It is also fundamental to say that Google is working towards the adaptation of QUIC to the new network paradigm that is proposed with HTTP/2 [3] (see Figure 3.2). Even though HTTP/2 yet only relies in TCP, the mixture of both concepts has the potential to benefit future connections.

Related to resultant performance, a recent study [80] has shown some preliminary results. They found that QUIC showed higher goodput comparing with TCP CUBIC in the case of "under-buffered networks" and also in the presence of random losses but with an important increment of packet loss ratio. Even though the results are obtained in very specific network circumstances and the development is far from being widely deployed, the results report the ability of QUIC to perform better than standard TCP flavours under those network conditions.

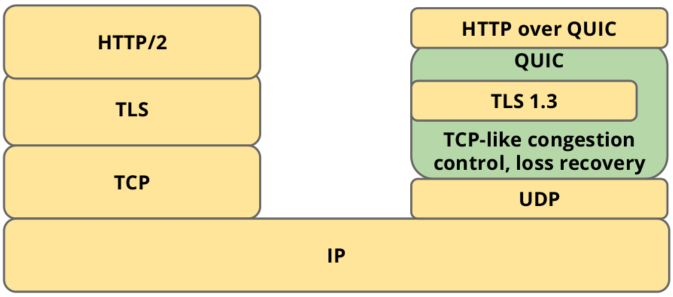


Figure 3.2: QUIC proposal by Hamilton et al. [3].

One important constraint that UDP-based protocols could have is the presence of blocking mechanisms in middleboxes, not allowing free end-to-end interaction. To cope with this problem and many others, an encapsulation mechanism called Substrate Protocol for User Datagrams (SPUD) [81] has been proposed. The idea is to provide an encapsulation for transport protocols that allows "minimal and selective exposure of transport semantics" and other transport information.

As mentioned before, the work in transport services over UDP requires time to evolve and successfully perform in current network topologies. Even if the performance enhancements are not widely deployable, QUIC may well be selected (see next Section 3.1.5.3) to be used in those network circumstances in which it could get performance benefits.

3.1.5.3 Selection of transport protocol based on application requirements

In last years, many efforts have been dedicated toward the de-ossification of transport layer. One of the main limitations of the transport layer is the lack of flexibility to adapt to the application-layer requirements. While some applications due to their features may require the use of TCP, others may well perform better with Real-Time Protocol (RTP) or Multipath TCP (MPTCP). One of the most cost-effective ways to resolve the ossification [82] is by providing an evolved API between application layer and transport layer.

The proposed solutions [12, 13, 83, 84] use protocol-independent mechanisms to set parameters based on the application requirements. Therefore, the application only describes the required service and the API abstracts that info in order to select the most appropriate transport protocols among the options. In a close future with the implementation of evolved transport services, the API would not only select the best transport protocol based on application requirements, but it would also consider network state. Even though this mechanism would require further signalling and interaction, recent advances [85] are evolving in this sense and could provide a more complex and complete API.

Inside the logic that decides which transport protocol is more appropriate, one problem arises. Sometimes, the sender does not know if the selected transport protocol is supported by the receivers. One way to solve this issues would be by the agreement of both end-nodes. The sender discovers the set of transport protocols available in the receiver and they agree on the best option among the available ones. However, this mechanism would consume a lot of time that would also impact the final performance of the transport layer. Other option is the to use a "test-and-select" approach that tries to start the communication with the most appropriate transport protocol. If the response is satisfactory, the transmission follows its normal behaviour. However, if the sender perceives that the receivers does not support the "proposed" protocol, it would try again but using the second most appropriate option in the list of transport protocols. This mechanism is called Happy Eyeballs and it was primarily use to select between IPv4 and IPv6, until [86] adapted it to work in transport protocol selection. Even though the method increases the initial session latency [86], it opens a framework for further improvements regarding the selection of most adequate transport protocols.

3.2 Specific issues in Mobile broadband

When analyzing how mobile networks' features have an effect onto the transport protocol behaviour, there are several characteristics that have to be mentioned.

It is clear that comparing mobile networks and fixed networks, the former have more variable channel conditions that could lead to throughput degradation [7, 87]. However, there are effects that, from a macroscopic point of view, are shared among the distinct networks. For instance, it has been proved [88,89] that even mobile networks suffer from the *bufferbloat* effect. Measurements over both 3G and 4G cellular networks of four U.S. providers and Swedish networks have concluded that *bufferbloat* represent a problem in MBB too. Therefore, the induced delay by different CCAs is of utmost importance to consider and understand.

Secondly, it has been demonstrated that there are differences among distinct mobile networks. A comparative work of 3.5G and 4G [90] showed that 4G networks are worse in regards to the TCP efficiency due to the superior throughput and variability. This is, the higher variability, the worse scenario for TCP due to the lack of rapid adaptability. However, the study also concluded that the proper usage and deployment of new AQM strategies in 4G were able to outperform 3.5G. The same differences apply in terms of variability for deployments of the same technology with different features that lead to distinct capabilities and fluctuations. It is important to detect the constraints of the deployment under use in order to assess its suitability to carry out certain experiments together with their intrinsic requirements.

Another important feature is that, in static positions, the variability mostly depends on the number of subscribers that are connected to the same base station and the cross-traffic that the users could generate. Therefore, in a situation with few UEs and under little cross-traffic, the dependence on TCP flavour represent similar results to the ones collected over fixed networks. For example, a work devoted to test different TCP mechanisms under the aforementioned conditions [91] stated that all TCP variants provide comparable goodputs but with a larger latency and number of retransmissions and timeouts for TCP CUBIC due to its aggressiveness. So, under very specific mobile network circumstances, the assumptions extrapolated from fixed network could be perfectly applied. In this regard, there is a need for understanding the implications while using one TCP implementation or other.

Besides, it is reasonable to gather all the works that have found new and determinant features over the interaction that mobile networks and the transport layer have. In this group, there are several features worth-mentioning.

- **Impact of variability**: Garcia et al. [92] conducted a measurement campaign in the cellular networks of four Swedish operators and analyzed the diurnal variation of TCP throughput and delay. Sudden increases in traffic load led to bandwidth variability and latency increment [6]. TCP happened to drastically reduce its throughput, and many times experienced timeouts. The timeout events were specially harmful due to the CWND reduction to one segment. In another study, Alfredsson et al. [93] proved that the variable modulation on the 4G link layer was contributing to retransmissions' increment and therefore higher delay and less throughput. In the following work [94] Huang et al. carried out a comprehensive study related to TCP throughput and latency estimation over a live LTE network. In their measurements, they found out similar timeout events. In regards to the impact of variability, there exists a need for evaluating how different variable circumstances affect the performance of different CCAs in order to better understand the implications of fluctuation in the channel quality and the following responsiveness of CCAs to adapt to it.

- **Uplink impact and bursty behaviour**: Regarding the evaluation of TCP throughput, it has been proved [95] that the uplink performance tends to degrade due to scheduling policies, severely impacting on ACK arrivals and therefore downlink injection ability. Those TCP flavours that merely depend its CWND upon the reception of ACKs are drastically affected by these ACKs reception, also called ACK-compression. Related to this issue, it has been demonstrated [96] that modern cellular networks' traffic has a tendency to become bursty. For this reason, there can be a large variation in the actual throughput during a short period of time (varying by up to two orders of magnitude within a 10-min interval). This variability could be even harder due to the fact that mobile ISPs often maintain a large and individual downlink buffer for each UE, provoking high latency instability. In relation to the impact of delay fluctuation the results in [92] showed that the performance of short flows was mainly dominated by the delay characteristics (i.e. the baseline RTT and the RTT variation). In this sense, we note that there is an understanding gap in relation to distinct bursty conditions and how do they affect the performance achieved by the CCA. Even though the whole transmission is affected by the delay variation, the flow start-up is known for suffering underperformance under such circumstances and thus, a thorough analysis is needed.

- **Impact of speed**: If mobile networks themselves suffer high variability, the channel conditions could be even more variable and challenging for TCP due to the movement of UEs. The movement leads to distinct propagations and fading patterns over time that, at the same time, impact on the assigned modulation to the UE, provoking "jumps" between consecutive channel quality reports. There are only few works that have considered the impact of speed on the performance of TCP in LTE networks. Merz et al. [97] carried out measurements in a live LTE network and studied speeds up to 200 km/h. The primary metrics in their study were spectral efficiency and the share and utilization of resource block among multiple users. Whereas Li et al. [98] focused on the frequency increment regarding RTT spikes, packet drops and network disconnections that were resulted from mobility cases. Even though there are works in relation to mobility in LTE, none of them has considered altogether different mobility patterns, together with different speeds and the performance of a wide selection of State-of-the-Art CCAs.

Taking into account the research and standardization momentum regarding 5G, in which most of the work is yet to be fulfilled, there are few works that have considered the performance of TCP in the future 5G MBB. Pedersen et al. [99] demonstrated the potential of using different TTIs in the eNodeB of 5G deployments depending on the metadata related to a certain channel. They showed that shorter TTIs were capable of allowing higher throughputs for short communications, whereas longer TTIs could overall benefit the performance of large transmissions. Sarret et al. [100] studied the forthcoming benefit of using full-duplex at the RLC in comparison with the current half-duplex implementation for an improved throughput and delay. Besides, they covered the possible configurations in ultra-dense 5G deployments that could limit the envisioned rates.

Even though several research studies and proposals have reported their concerns and findings regarding the effects between mobile networks and TCP under challenging conditions, none of them has considered the analysis of distinct CCAs of TCP during the Start-up phase and mobility with different speeds and movement patterns. Besides, we there is a research gap regarding the use of different end-to-end latencies and how would this difference impact on the performance of TCP not only in current MBB but also in low latency scenarios that move toward the overall potential delays in forthcoming 5G.

3.3 Awareness between mobile networks and transport layer

As stated before, a transport service that is able to not only take into consideration application requirements, but also the network status, requires further signalling and interaction with the network. This section presents the state of the art related to the interaction and signalling between the transport layer and mobile networks. The explanation will be split in two main parts. Firstly, some end-to-end cross-layer enhancement proposals are going to be presented. Finally and also related to cross-layer improvements, the works that suggest the introduction of a third party (such as proxies, middleboxes and so on) will be explained.

3.3.1 End-to-End improvement proposals

The aim of this subsection is to explain different end-to-end proposals to improve the interaction between mobile networks and the transport layer. Each category faces the problem from a different point of view.

- **Server-based**: This option is the most used one due to the fact that selected CCA's logic runs over the server. For this reason, most improvements have been proposed from the senders' point of view through the control of the CWND. One of most recent enhancements is Verus [101], a CCA that tries to learn from the network through the feedback received by RTT value and captures the relationship between such delay and in-flight packets so as to control the CWND.

- **Receiver-based**: An innovative option that is requiring more and more attention is the modification over receivers to improve transport layer performance. Jiang et al. [102] found out that some phone vendors were establishing upper bounds in the advertised windows in order to limit the increment of the CWND and avoid overshots. However, this limitation underutilizes the network in some cases. A dynamic adjustment of the receiver window was proposed with the potential to reduce the delay and increase the throughput. Following a similar idea Receiver-side TCP Adaptive Queue Control (RTAC) [103] has been proposed. According to TCP dynamics, RTAC estimates the appropriate number of in-flight packets and in consequence controls the transmission rate through the advertised window.

- **Both server and receiver**: This option follows the modification of both end nodes. One of the most representative solutions is the recent draft conducted by Johansson in IETF [104] to study and define adaptability and suitability concerns of congestion controls to mobile networks. In relation to the implementation, it proposes delay-aware modifications over CUBIC based on the OWD feedback gathered from the receiver.

Even though all aforementioned approaches have showed some improvements, the changes in the sender are always easier to be integrated due to the fact that there is no need for the costly and troublesome act of modifying clients' equipment, providing better and faster applicability. TCP's background has showed that it is clear that controlling the transmission rate through the CWND is the best option. However, considering the differences between uplink streams and downlink streams and the little trustworthiness of RTT measurements due to uplink scheduling, bursty behaviour and so forth, the end-to-end scheme of TCP may well need the intervention of an unbiased third-party that is capable of providing with the needed information regarding the network status. Next subsection covers the current State of the Art proposals in this regard.

3.3.2 Third-party intervention

Skipping from the end-to-end philosophy that have prevailed over the year, there are several proposals that suggest the need for a change in order to accelerate/adapt the transmissions over mobile networks through the intervention of a third-party. In this regard and according to the State of the Art in the field, three are the distinct approaches that are more reasonable to use.

- **Pure interventionism**: One of the options that have been suggested is the use of Performance Enhancing Proxies (PEP) [105] so as to guide the different flows to achieve a predefined target (keep delay low, maximize the throughput, fairness policies and so on). However, most of the mechanisms that could be applied in PEP constitute a change in end-to-end interaction and violate the architectural principles of the Internet (e.g. the packet acknowledgement and trustworthiness) and for that reason PEP is not recommendable for widespread use. Another proposal is to use similar principles but limiting the usage of them to certain transmission phases. In this way, a Miniproxy [106] has been suggested to boost 3WHS through a novel Early SYN Forwarding mechanism, achieving a shorter establishment and potentially impacting on the improvement of short-flows performance.

- **Transport-features awareness**: Ren et al. [107] propose a proxy that was able to dynamically manage the sending window based on a model that considers the dynamics of bandwidth and the previous knowledge of the limitation of the bottleneck queue. Although unrealistic due to the difficulty to have such information with the required time to react, they gave some hints to the improvement of such mechanisms. Similar idea was embraced by Liu et al. [108] by presenting a rate-based CCA to avoid packet losses and react to the rapid bandwidth fluctuation. However, as stated in PEPs problems, it was conceived to change end-to-end semantic and therefore, lacking deployability and applicability in current Internet or MBB. The insertion of PEPs and the impact of their packet intervention modifies the features in the path and misleads the servers, making them fail in their assessment regarding the end-to-end path and the consequent CWND management according to such measurements.

- **Radio-awareness**: The last option comprises the use of the radio information as input parameters for the adaptation of transport layer behaviour. One solution called CQUIC [109] aimed to achieve similar goals but with the difference that it leverages the irrespectively-sent CQI information reported by the UEs every TTI. Apart from the stated improvement, the most important thing is that the proposal suggests a change that requires as minimum modifications as possible. The main drawback of this method is the meaningfulness of the provided information. Even though the CQI represents the quality of a certain channel, it is far from being easily mapped to the final transport block size assigned to a precise UE. The CQI only reports the quality of a single UE but does not consider any other parameter of the cell (i.e. number of attached UEs, traffic enqueued, available resources). As recently stated, mobile content delivery optimization is possible through the throughput-based guidance of a third-party [110]. The idea is to expose the amount of bandwidth available for a certain UE to the data sources/servers helped by the guidance of a information-privileged party (such as throughput guidance manager following Mobile-edge Computing -MEC-principles). Even though it is a good idea, its applicability is under concern for the high dependence on the information provided by the eNodeB. Apart from that, since the sender merely depends upon the reception of information from the throughput guidance manager, the method is encapsulated with every ACK. This overhead could cause the performance being deteriorate. Therefore, the mechanism needs further improvements in the timing of the throughput guidance.

3.4 Conclusions on the State of the Art

This section summarizes the section and covers the most remarkable and applicable conclusions. The Section is devoted to wrap-up the specific issues in MBB and the awareness techniques between mobile networks and transport layer. In this regard, it is important to highlight the ability of the mentioned methods to partially or completely solve the aforementioned issues. Table 3.1 gathers such information in a simplified way. The first row represents the current issues in MBB, while the left column does the same for the explained awareness techniques. Inside the table the crosses depict a benefit of the selected technique for the specific issue, whereas the hyphen indicates no improvement of the issue using the precise awareness technique.

Tech. / Issues	Deployments	Bufferbloat effect	Bursty behaviour	Start-up performance	Variability	Mobility	End-to-end delay
Server-based	-	+	-	-	+	-	+
Receiver-based	-	+	-	-	-	-	+
Both	-	+	-	-	+	-	+
Interventionism	-	-	+	-	-	-	-
Proxy(transp.)	-	-	+	-	+	-	+
Radio-aware	-	+	-	+	+	+	-

Open-issues	Multi-testbed	Specific CCA features		Start-up performance	Adaptability of CCAs		4G vs. low latency deployments

Table 3.1: Specific issues in MBB and the awareness techniques between mobile networks and transport layer that lead to certain open-issues.

Every issue will be individually covered in the following paragraphs:

- Regarding mobile networks, different studies report their results under very specific frameworks or disciplines that do not necessarily have to reflect the constraints and behaviours that are present in the real-world. For that reason, the studies that have tested the actual reality, gain more credit. Nevertheless, it is not possible for every researcher to have access to such testing environment and due to this small-scale deployments, emulated testbeds and simulated environments play an important research-enabling role. However, there exist an explanation gap regarding the actual positive or negative impact that every testing environment has in comparison with others. Therefore, this thesis aims at evaluating in a multi-deployment way, matching and assigning the capabilities of each deployment with the requirements of each objective or subobjective.

- The bursty behaviour of the cellular network and the bufferbloat effect both damage the TCP performance by increasing the delay in the path and both are related in a sense. Even though the bufferbloat effect is created by the excessive packet injection of the CCA, both in the flow under test and as a cross-traffic, the bursty conditions also lead to have a less accurate notion of the radio and transport circumstances. Therefore, degrade even more the performance through a more abrupt bufferbloat. Both effects are affected by different parts of the end-to-end scenarios and thus, the way to solve them is distinct: a) A better management in the server, a more accurate notification of the receiver, both mechanisms working together or a radio-aware server could reduce the harmful impact of bufferbloat; b) Interventionism proxies with and without knowledge of the transport-layer parameters may well pace the traffic in order to avoid bursts in packet reception. However, none of the selected techniques are able to cope with both bursty behaviour and bufferbloat effects. In this regard the current dissertation stands that there is a clear need for a proper understanding of the impact of specific CCA features so as to boost better solutions that are capable of solving both issues.

- The Slow Start mechanisms have reported adaptability issues over mobile networks and even the delay-aware ones may result performing poorly for the bursty behaviour of ACKs and therefore the unreliability of RTT parameter. Radio-awareness could provide with the requested information in order to build a more adaptive Slow Start method. Regarding the signalling, even though they constitute good examples of how privileged information would benefit the decisions of a TCP senders, it has been stated that the selected proposals [109,110] are not mature enough to be widely deployed. Therefore, this thesis understands that a study over the most important Slow Start options would shed some light into the most appropriate method for precise situations in mobile networks and would therefore help future improvements or proposals.

- Following the studies that have been carried out in mobile networks, it has been detected an uncovered work in relation to two of the most critical situations for the transport performance for a certain UE; a) the intrinsic variability of the MBB and b) the performance of TCP during the self-induced issues while moving. Even though we detect different methods that could help reduce the impact of variability and the fluctuation created on the move, there is a research gap regarding the adaptability of a wide range of State-of-The-Art CCAs in challenging variable conditions. More precisely, there is a need for explaining the impact that different speeds and mobility patterns have on the adaptation capabilities of different CCAs.

- It has been detected a necessity of studying possible TCP solutions that are able to work equally well in 4G and low latency or low latency proximity service provisioning in forthcoming 5G deployments. Even though many proposed mechanism have the potential to help TCP in such circumstances, there still exists a lack of study that explains the effect of reducing the end-to-end latency. It is important to evaluate which are the capabilities of different CCAs in 4G and low latency scenarios to cope with similar network conditions. Thus, being able to propose CCAs due to their appropriateness or discard them due to the contrary.

Besides to the the detected research areas, the Chapter has detected mechanisms of the transport layer that are beneficial and others that could lead to misperformance. In this regard, three groups could be mentioned: 1) Even though the CCAs have not deliberately used AQM mechanisms in the sender, the recent BBR uses pacing and therefore requires FQ AQM scheduling enabled. Therefore the AQM scheduling algorithms in the sender should only apply the AQM method on-demand of the CCA. 2) It is important to be opened to complementary improvements such us HQ-CoDel to tackle *bufferbloat*, specific TCP features (e.g. ECN, Fast Open, RTO Restart, TLP or TLPR between others) and even reuse the previous advances over TCP. In this regard, the thesis would only consider the proposals that are finally accepted into the Linux kernel. 3) It is equally important to find and avoid those mechanisms that have showed and inefficient impact on TCP such as the intrusive middleboxes that violate end-to-end philosophy.

4 Proposal of a intra-protocol de-ossification framework

The current chapter introduces a proposal of an intra-protocol de-ossification framework that covers the description of such framework, the definition of technical requirements, the selected CCAs, the available LTE deployments and the analytical methodology. The chapter is organized as follows:

- Section 4.1 describes the theoretical (deployable as a transport service) intra-protocol de-ossification framework and defines the technical requirements in order to cope with the defined research open-issues in the previous Chapter.

- Section 4.2 describes the selected CCAs for the analysis, introducing their most important phases and features and reasoning the selection and inclusion of them in the candidates under test.

- Section 4.3 covers the measurement strategy that comprises four different LTE deployments: a simulated environment, an emulated testbed, a controlled deployment and a real-world deployment are described. Different and generic LTE setups are explained together with their strengths and weaknesses. Besides, each setup is linked with the actual experiments that the thesis has used to explain the rationale behind each experimentation approach. Finally, the different testbeds are compared and their scope and limitations are detected in order to be able to decide which LTE setup should be used for a precise research target. The Section is also devoted to explain the importance of real-world equipment in LTE setups.

- Section 4.4 comprises the explanation of the methodology in order to cope with each part of the analytical open-issues as a form of performance aspects to be analyzed.

4.1 Description of the framework and requirements definition

This chapter covers the brief conceptual description of the proposed intra-protocol de-ossification framework. On the basis of the detected open-issues in the previous Chapter, the framework will be fed with the outcome or results. That is, the evaluation of capabilities in distinct LTE deployments, the understanding of the implications of each CCA, the study of Start-up methods and performance, the analysis of CCAs' adaptability in highly variable network circumstances and the consideration of different end-to-end delays are the foundation for the de-ossification framework.

The idea is to build the framework upon the concept of transport services that allow more manageability of the transport layer due to the possibility of running transport-based services in the user space between the actual transport layer and the application layer. Considering the different network situations that are present in the real-world and the need for transport protocols to behave accordingly, it is fairly complicated to cover every network situation with a single congestion control.

To this end, the framework aims at giving the credit to some flavours regarding the ability to perform under certain network conditions (defined by the outcome while analyzing the open-issues). The technology evolution moves towards an almost fully softwarized environment together with the capability to make intelligent and rapid decisions. Therefore, future MBB scenarios would allow not only adapting fast the

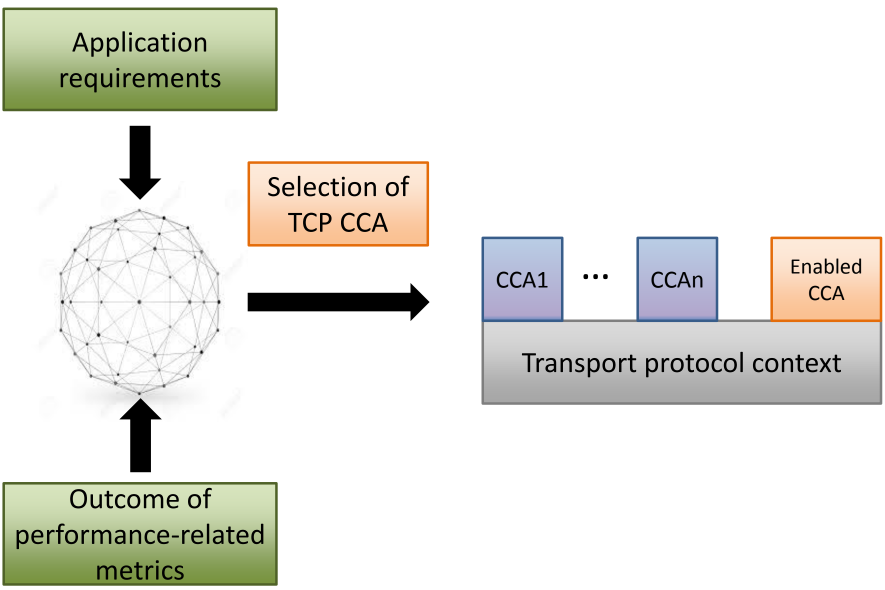


Figure 4.1: Intra-protocol de-ossification framework

behaviour of transport protocols' algorithm to the variability of cellular access, but also selecting between different CCAs depending on the network conditions and application requirements. Following the same trend of cost-effectiveness regarding the creation of new protocols and the reusability of current ones, the framework aims at appropriately selecting the CCA for the precise network circumstances (based on metrics of selected open-issues) and the application requirements (specific evaluation process).

Figure 4.1 illustrates the idea behind the framework where the selection of the TCP CCA is based on the outcome of performance-related metrics and application requirements (see Figure 4.1). The framework makes two important concepts coexist: 1) **Intra-protocol**: The selection of the most appropriate transport solution is carried out among the TCP family; 2) **De-ossification**: The framework provides with a solution (the mentioned intra-protocol selection) that gets rid of the constraints in the transport-layer in order to adapt to the combination of the network conditions and the application requirements. The following explanation will cover each concept individually, focusing on the reasoning, feasibility and suitability of such technique.

Intra-protocol

The main difference of the framework with previously mentioned works in transport services [12, 13, 83, 84] is that the candidate selection is done within the CCA family of TCP (intra-protocol). The reasons for the decision of this workaround are the following:

1. Even though there is an ongoing work regarding the adaptation of QUIC to HTTP/2, the selection of TCP as a substrate of HTTP/2 is clear. Therefore, the performance of TCP is of utmost importance and so is the necessity to analyze and propose solutions in this regard.

2. Building new transport services on top of UDP usually requires the creation and re-invention of TCP-alike services. Considering this feature, this thesis finds pure TCP more suitable for its internal features, evolved behaviour over the last decades and a wide range of CCA options. In this regard, a suitable selection of a certain CCA could not only benefit the performance of the services running on top of TCP, but also the better creation of TCP-alike congestion control selection mechanisms over UDP.

3. When referring to TCP, it is common to analyze the performance only from the perspective of the default CCA [85]. However, TCP comprises a whole set of candidates. The analysis over the default CCA could bring important findings regarding the general performance of TCP as is configured currently, but hides the potential of the rest of the CCAs.

4. The certainty that in any case a TCP candidate would be selected, avoids the costly effect of overhead and latency increment due to Happy Eyeballs technique.

Considering all the mentioned points, the framework relies in the potential and diversity of the CCA implementations in the TCP family in order to find the best solution for the precise conditions.

De-ossification

Regarding the de-ossification of the transport layer, we have previously introduced that currently there exist solutions [12, 13, 83, 84] that use protocol-independent mechanisms to set parameters on the transport layer based on the application requirements. The application only describes the required service or the constraints for an appropriate execution of the program running in the application layer (e.g. critical delay, maximum throughput, loss sensitive) and the API abstracts that info. The information is utilized to select the most appropriate transport protocols among the options. A more recent advance [85] has added the fact that the API not only considers application requirements but also network conditions. Even though this mechanism requires further signalling and interaction, the method could provide with a more complex and complete API.

Regarding the feasibility and the timing of the framework proposal, there are out of the scope of the thesis and the framework the improvement of the API that connects the transport layers with the application layer and the enhancement of signalling from the mobile network that allows including the knowledge from the network in the decisions on the API. Those works are ongoing efforts that could bring the implementation of evolved transport services. Regarding the signalling, even though it has been stated that the selected proposals [109, 110] are not mature enough to be widely deployed, they constitute good examples of how privileged information would benefit the decisions of TCP senders. Thus, the future inclusion of signalling metadata would only provide the framework with more valuable and meaningful representations of the network conditions and therefore, it would drastically benefit the achieved results.

Based on these advances and taking into account that currently the flexibility to select the most appropriate transport solution is limited to the beginning of the transmission, the framework considers the appropriate selection of CCAs prior to the establishment of a transmission. However, the framework also aims at covering future softwarized use-cases with a more complex in-flight CCA selection, adapting the transmission while in-progress and according to the changes in the "boundary conditions". This is, provided that in a close future, improvements over [12,13,83–85] would provide with a more flexible transport layer that would at the same time allow switching between similar transport protocols in-flight, the framework covers such possibility.

In order to deal with both applications of the CCA selection (at the beginning of the transmission and in-flight), the framework adds a common "transport protocol context" (see Figure 4.1). Such a common context stores the behaviour of the CCA under use ("enabled CCA" in Figure 4.1) so as to be able to input with initialization information to the following CCA. The stored parameters could be manyfold, but at least the minimum parameter or metric that all CCA share are the CWND and the delay. Therefore, it should be mandatory its inclusion. The following steps indicate the workflow of an inflight CCA switch as a transport service: 1) Selection of the most appropriate CCA at the beginning of the transmission taking into account the outcome of the performance-related metrics and the application requirements. 2) Detection of conditions change and decision regarding the appropriateness of a different CCA among the available ones. 3) Modelling of the initial CWND of the second CCA based on the CWND of the first CCA and the actual induced delay. 4) Switch of the transmission management to the second CCA that starts after the information input of the first CCA. In this simplistic case, only the initial CWND is tweaked. 5) Active wait of the transport service in order to decide whether the "Enabled CCA" is the most suitable one to face the current conditions. In order to avoid massive CCA switching events only smoothed samples of the outcome of the performance-related metrics are used.

Once the framework has been described, it is important to define the technical requirements that each open-issue (see Table 3.1) has in order to deal with its study and be able to later on decide which LTE deployment is more suitable taking into account the provided capabilities. The following descriptions briefly define the requirements of each open-issue:

- **Performance aspect 1 - Impact of specific CCA features**: In order to appropriately study the specific CCA features in the same network conditions, it is important to transpose the limiting factors of the real-world into a testbed with the ability to microscopically and individually assess every CCA and be capable of **comparing them**.
- **Performance aspect 2 - Start-up performance**: The start-up phases constitute a short but significant period of time in the performance of every TCP flavour due to the presence of many short-lived flows in the network. Therefore, it is essential to evaluate different Start-up proposals in environments with distinct variability circumstances in order to correctly understand the impact of network conditions on the performance and study the suitability of each method for each network momentum.
- **Performance aspect 3 - Adaptability of CCAs**: The adaptability analysis aims at evaluating the responsiveness of different CCAs in challenging network conditions. In this regard, this thesis covers the measurement of distinct mobility use-cases in which TCP struggles to make the most due to the fluctuation not only in the long terms but also sample by sample. In order to analyze the adaptability of CCAs in the aforementioned situations, the study requires different variability conditions so as to infer and confirm that the detected findings in regards to the behaviour of CCAs is applicable as a generalized pattern or is deployment-dependant.
- **Performance aspect 4 - 4G vs. low latency deployments**: The comparison study of 4G latencies and low latencies that move toward the potential delays in 5G deployments has two main requirements: 1) The utilization of real UEs in order to include all the LTE signalling and processing delay towards the eNodeB and the other way around. 2) The results with different end-to-end latencies should be comparable and therefore, the study needs the certainty that no cross-traffic is running through the same bottleneck, impacting this way the baseline delay.

This Section has described the intra-protocol de-ossification framework that tries to get rid of the constraints present in the transport layer and be able to adapt the selected transport solution to the network conditions as well as the application requirements. In this regard, the framework defends the idea of a selection within the TCP implementations due to the variety and diversity of options. Besides, the framework envisions the possibility of not only detecting the most appropriate TCP CCA at the beginning of the transmission, but also CCA switches in-flight when the "boundary conditions" change throughout the transmission. Finally, this Section gathers the technical requirements of the open-issues that would input the framework with the performance-related metrics.

4.2 Selected TCP CCAs

Once explained the intra-protocol de-ossification framework, it is important to describe the selected TCP CCAs as the eligible group of CCAs in the selection framework. The studied TCP variants fall into five categories with regard to their employed CCAs: loss-based, combined loss- and delay-based (with or without bandwidth estimation), and delay-based. A brief overview of the five picked TCP variants is individually given below.

TCP NewReno [111] uses four basic algorithms to decide its behaviour. 1) In Slow Start, the congestion window (CWND) value is increased by one packet for each acknowledgement (ACK) reception until a timeout period is consumed or a notification of a loss packet is received (with a triple duplicate ACK -3DUPACK-). 2 & 3) Depending on the event, NewReno would apply a different CWND reduction or backoff policy: halving the CWND in case of loss event (Fast Retransmit algorithm to recover the lost packet and Fast Recovery algorithm to manage the CWND reduction afterwards) and reducing the CWND to one segment with the timeout. 4) After the Slow Start mechanism, the CA phase increases the CWND by one packet per Round Trip Time (RTT) until another timeout of loss is detected. NewReno has been therefore chosen for our comparison due to its representation of the base TCP behaviour and its usefulness for comparison purposes.

TCP Westwood+ [33] modifies the sender to be capable of estimating the available bandwidth by measuring the receiving ACKs' stream. Afterwards, the estimation will be used to establish the new CWND after 3DUPACK episode as the product between estimated bandwidth and the minimum assessed Round Trip Time (RTTmin). Whereas, after a timeout the *ssthresh* is set to estimated bandwidth and the CWND is lowered to 1. TCP Westwood+ has been selected in this study for its hybrid behaviour using loss-based mechanisms together with delay-awareness. In fact, TCP Westwood+ was primarily developed to have a great response in wireless links in which the delay variability is common.

TCP Illinois [112] has an AIMD mechanism controlled by estimated queuing delay and buffer size. In a normal situation when no queuing delay is detected, the CWND is increased by 10 packets per RTT. If estimated delay starts increasing, the increment of CWND will be gradually lowering until the minimum value of 0.3 packets per RTT is reached. When the RTT is close to the maximum, the loss is considered as buffer overflow, whereas in low RTT the loss counts as packet corruption. Developed to perform efficiently within high speed networks, its loss-based and delay-awareness make a perfect candidate for our study.

TCP CUBIC [113] has different back-off and increment policies set according to a cubic equation. Three different stages are defined: Firstly, the CWND ramps up very quickly following a concave shape. Then, in the transition of achieving CWND value recorded before the last window reduction, CUBIC slows down having an increment close to zero. Finally, the CWND accelerates in a convex way seeking for more available bandwidth until a new loss event occurs. Besides, CUBIC uses Hybrid Slow Start [59] as a replacement for the classic Slow Start in order to reduce the number of loss packets in the last RTT. To that end, Hybrid Slow Start adds two exit conditions to the classic ramp-up period. In case one of them is met, the CWND will increase in accordance to the CA phase. The detection of such points is based on the measurements of ACK trains and RTT delay samples so as to detect whether the path's bottleneck is congested or the current value of the CWND has already achieved the available capacity. CUBIC is the default CCA in Linux kernels and therefore, a candidate to be selected in a comparison of different TCP implementations as the representation of by-default performance and widespread impact (67% of world-wide servers use Linux [22]).

TCP CDG [34] modifies the TCP sender in three main aspects. Firstly, CDG uses $\bar{g}_{min}$ and $\bar{g}_{max}$ delay gradients assessed during a measured RTT interval as a congestion indicator (less noisy than pure RTT). Secondly, the state of the bottleneck queue is estimated so that packet losses are treated as congestion signals only when the queue is full. Thirdly, CDG uses Hybrid Slow Start (like CUBIC). However, it is more strict than CUBIC since the application of the Hybrid mechanism starts from the very beginning (CUBIC only considers Hybrid Slow Start if $CWND > 16$ typically). The selection of TCP CDG has been based on its novel use of delay gradients in the AIMD mechanism and to evaluate the actual usefulness of such a different feature in mobile networks.

During the last quarter of 2016, an important TCP flavour was added to the TCP family. Due to the preliminary results [36, 114, 115] and the importance that may well have in the future, the thesis added in early 2017 TCP BBR [36, 44] to the selected CCAs group. **TCP BBR** is a model-based CCA that builds its network path model based on two parameters: the estimated bottleneck bandwidth and the RTT. The estimated bottleneck bandwidth is measured by calculating the timing and rate of incoming ACKs in the sender. Based on the model, BBR infers whether the packet injection rate is below or over (starting to build-up the bottleneck queue with packets and increasing the delay [45]) the Bandwidth Delay Product (BDP) of the bottleneck.

Another important modification of BBR is the inclusion of *pacing* in the sender. To this end, BBR needs the assistance of a packet scheduler that implements the feature of *pacing* (for instance Fair Queue -FQ-) or the fall-back and native *pacing* option of TCP layer itself. Besides, TCP BBR works under four states: 1) During the Startup state BBR ramp-ups as the classic Slow Start method but it does not wait until a loss occurs. Instead BBR exits the increase when the gain in throughput during three consecutive transmission rounds is below 25%. BBR waits three rounds to be sure that there is no limitation of the rate due to a underperformance of the reception-window. Three rounds allow the receiver's reception-window autotuning mechanism open the window for incoming traffic and the sender to capitalize the available bandwidth increment [44]. 2) During the draining state, BBR tries to keep the RTT as close to the measured baseline RTT as possible and gets rid of the excessive packets in the bottleneck. 3) Probing bandwidth state is formed by 8 stages in a cycle in which BBR cruises at assessed BDP in six of them if no change is detected. The other two phases are to probe for more bandwidth (established in a 25% increment of the pacing rate) and to drain the queue in case the previous phase build-ups the bottleneck queue. 4) Probing RTT state is a mechanism that seeks for the measurement of the baseline RTT if during a long period of time (10 seconds in the current configuration) no new baseline RTT samples are detected, reducing the CWND to four segments for at least 200ms to afterwards establish the last known and successful CWND value. The selection of BBR is based on the recent inclusion of it in the Linux kernel and the importance that may well have in the future according to the preliminary results [36, 114, 115].

As examples of loss-based CCAs, we study both NewReno [111] and CUBIC [113]. NewReno was selected due to its prevalence in research and its large implementation base, and CUBIC by the fact that it is the default CCA in Linux. TCP CAIA delay gradient (CDG) [34] represent a delay-based variant that drives the CCA only considering the delay gradient of the transmission. The Westwood+ [33] congestion control was taken as an CCA example of a combined loss- and delay-based with bandwidth estimation technique. In many ways, TCP Westwood and its successor TCP Westwood+ laid the foundation for the work on designing a CCA that is able to distinguish between congestion and non-congestion related packet losses in wireless networks without any support from the wireless MAC layer. Also, Illinois [112] was selected as an example of a combined loss- and delay-based CCA. In contrast to Westwood, Illinois primarily targets high-speed and long-delay networks. Finally, TCP BBR (Bottleneck Bandwidth and RTT) [36, 44] model-based CCA that drives the congestion avoidance management based on two parameters: measured baseline RTT (delay-based) and the timing and rate of ACK packets (bandwidth estimation).

4.3 LTE deployments

In order to evaluate the performance of different CCAs in different network situations and identify the main responsible for such performance (study of detected open-issues) we have designed a progressive experimentation plan, leading to the so-called ”experimentation stair”.

The four different experimentation approaches considered are cut off highlighting the strengths and weaknesses of them in terms of scalability and suitability to model real deployments. Moreover, each approach is linked with the deployment that has been used in this thesis. It has to be noted that the capabilities of each LTE deployments would serve to match the technical requirements that each open-issue depend upon (see Section 4.1).

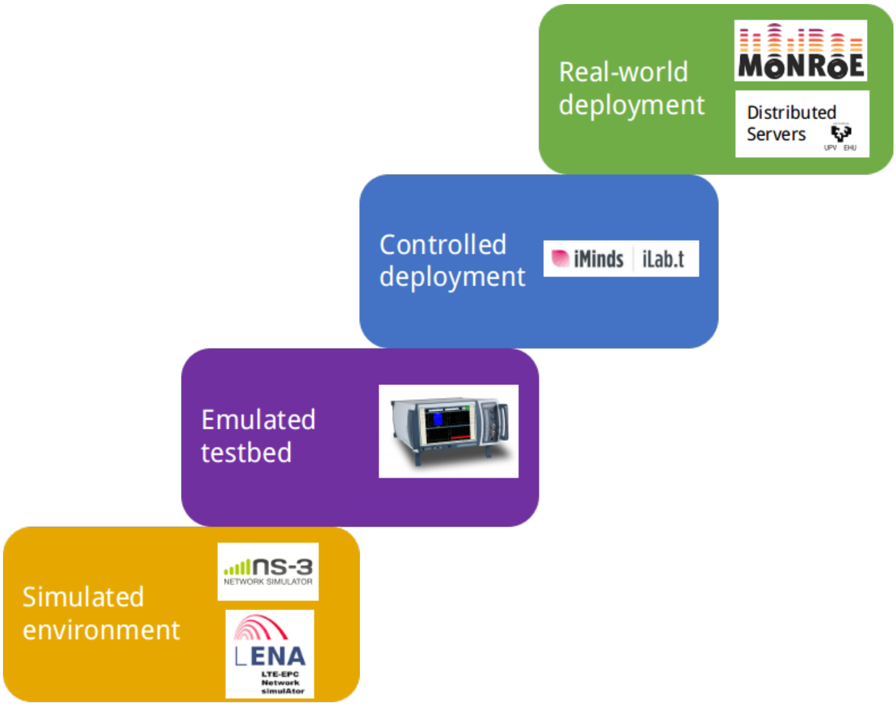


Figure 4.2: Experimentation stair

4.3.1 Simulated environment

In research, the simulated environment is one of the most used deployments. The resources that have to be spent are very little (for instance a single computer) and therefore it is an available experimentation way for almost everyone. Regarding the strengths, two are the most important: 1) the repeatability among different simulations and therefore the ability to fully compare among multiple realizations, and 2) the possibility to gather parameters and have them under control. In contrast, actual performance may differ from real world ones: i.e. applied fadings are synthetic and far from realism, the UEs are not real (in terms of physical effect but also protocols) and so is the reporting mechanism.

Considering all the abovementioned features and in order to cope with a simulated LTE environment, we decided to use a hybrid simulation/emulation setup based on the ns-3 network simulator (ns-3.26 more precisely). The selection of ns-3 was based on the wide range of possibilities that provides in terms of modelled technologies, the available thorough documentation and the technical support. Even though ns-3 allows performing TCP-based measurements with its natively modelled TCP flavours, the behaviour of these TCP implementations differ from real-world ones. Thus, we opted for using Direct Code Execution (DCE) instead (DCE 1.9 more precisely), allowing the execution of a full Linux network stack [116] and enabling the configuration and tweak of nowadays TCP/IP stack.

In order to cover and add the LTE side to ns-3, we employed the LTE/EPC add-on module developed in the LENA project. Trying to model a real deployment becomes a very tough task with many variables involved. In order to better understand the struggle to model the LENA module as close to the real-world as possible, the Section 4.3.1.1 covers a required non-straightforward configuration and tweaks that need to be applied.

4.3.1.1 Effect of CQI reporting mechanism

In ns-3, all processes are simulated and we have access to management parts of the mobile network that in the real-world are vendor-specific, manufacturer-specific or provider-specific. In this regard, the quality reporting and channel adaptation mechanism is entirely available. To understand how ns-3 manages this process, we will start from a certain SINR measurement. For instance, the Figure 4.3 shows a variable SINR shape.

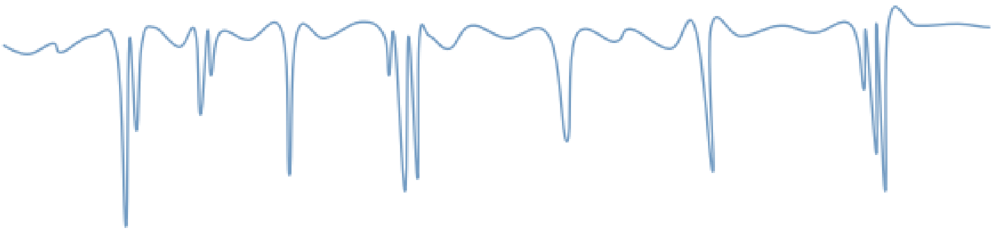


Figure 4.3: SINR trace with fading.

Inside the CQI reporting mechanism two methods are differentiated: Wideband reporting and Sub-band reporting. The Wideband reporting uses the average CQI value among the CQI values of all assigned PRBs, therefore reporting a single average CQI value per TTI for the whole frequency band. In contrast, Sub-band reporting method takes the average CQI values per each PRB group.

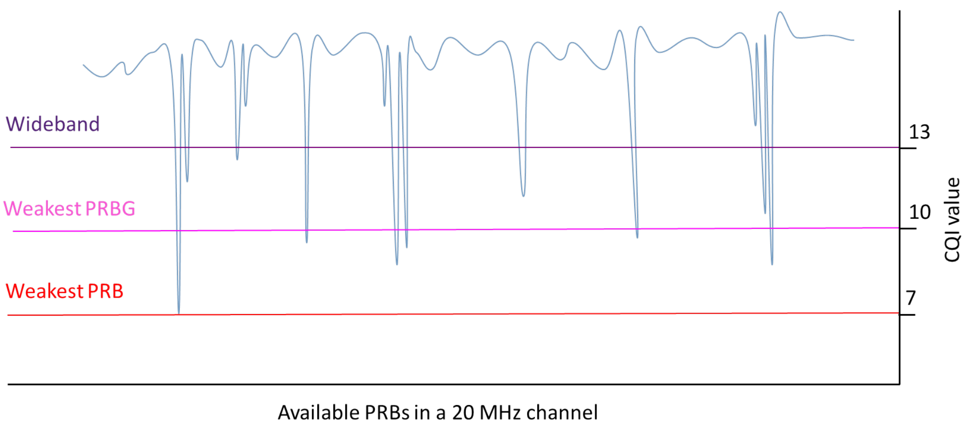


Figure 4.4: SINR trace with fading and assigned CQIs.

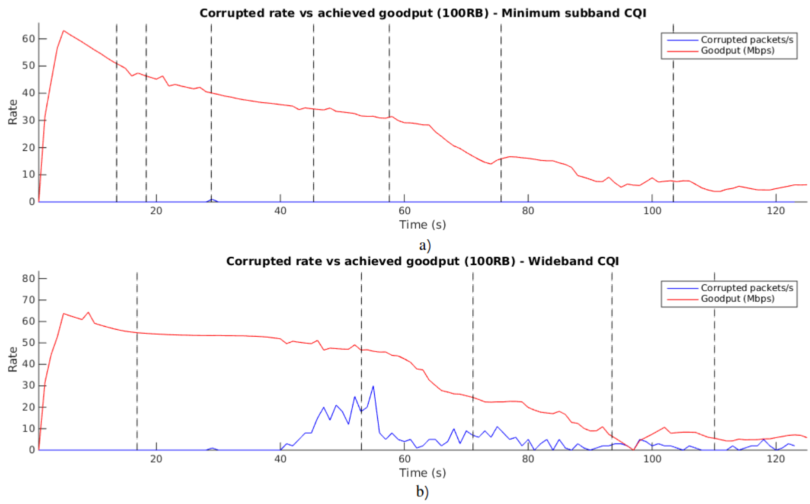


Figure 4.5: Achieved goodput vs. corrupted packets: a) Sub-band CQI; b) Wideband CQI

For instance, in a situation with 100 PRBs and an LTE bandwidth of 20 MHz (see Figure 4.4), the PRBs are grouped in 25 groups (4 PRBs each). The CQI of each PRB group (PRBG) is calculated by averaging the CQI values of 4 PRB so that a set of 25 CQI values is reported to the eNodeB. Under the use of proportional-fair scheduler in ns-3, among all the 25 values in the sub-band CQI reporting array, the weakest CQI among the PRBGs is translated as a representation of the CQI for the whole array when translating to MCS. So, for a situation in which the average CQI is 13 but has one PRBG with CQI 10 (see Figure 4.4), the eNodeB will either translate into MCS 1) a CQI index of 13 if reported by Wideband CQI and 2)10 if reported by Sub-band CQI. Therefore, even though the quality of the channel is the same in both situations, the reporting mechanism could lead to an underused channel and far from real-world behaviour (from the radio perspective). Figure 4.4 shows a simplified representation of different CQI assignments in relation to the base SINR.

Considering the reasoning of CQI assignment in Figure 4.4, Figure 4.5 depicts how different the results for the same scenarios are depending on the use of Wideband CQI or Sub-band CQI reporting. While the results show a greater goodput under the use of Wideband CQI, the outcome of Sub-band CQI reports higher percentage of corrupted frames in the radio layer. Both subplots summarize one of the main foundations of LTE. LTE allows up to 10% of BLER (Block Error Rate) because it has been widely demonstrated that is the best relation between corrupted frames and modulation level so as to achieve the highest possible performance. So, the presence of corrupted packets and greater goodput indicate that the performance of ns-3 under the use of Wideband CQI is closer to what LTE represents in reality according to the channel conditions.

Following the mechanism, if only 6 PRBs are assigned to the UE, "the weakest PRBG" is going to be closer to "Wideband CQI", if not the same. Therefore, in order to establish a fair CQI-MCS translation among different bandwidths and be closer to real world deployments, the proportional fair scheduler is changed to work with Wideband CQI. Leaving the mechanism by default, the consequences are manyfold:

- With big bandwidths and therefore many PRBs available, if some drastic fading is applied, ns-3 tends to be too robust, wasting achievable capacity.
- Under the same conditions, there is almost no impact of a greater CRR (the clock utilized to synchronize the CQI reports) because the robustness hides CQI variability. This is, waiting more time between two consecutive CQI reports and therefore managing for a longer time the variability of the network with the same CQI value, the performance is not affected.
- For single UE scenarios, we have demonstrated that there are two ways to obtain realistic LTE performance in which the BLER could have some impact on the final goodput performance. The two ways are by: 1) reducing the bandwidth of the UE (following the example of 6 available PRBs) or 2) changing the reporting mechanism to work under Wideband CQI (see Figure 4.5).

Trying to dodge such consequences, for this thesis the Wideband CQI reporting mechanism is applied in proportional fair scheduler.

4.3.2 Emulated testbed

The next approach in the experimentation stair comprises emulated environments. Precisely in LTE, the so-called digital radio tests, LTE-in-a-box or LTE emulators. These kinds of emulators play the role of the eNodeB, creating the LTE radio signal and all the necessary LTE protocols events to support the attachment and registration

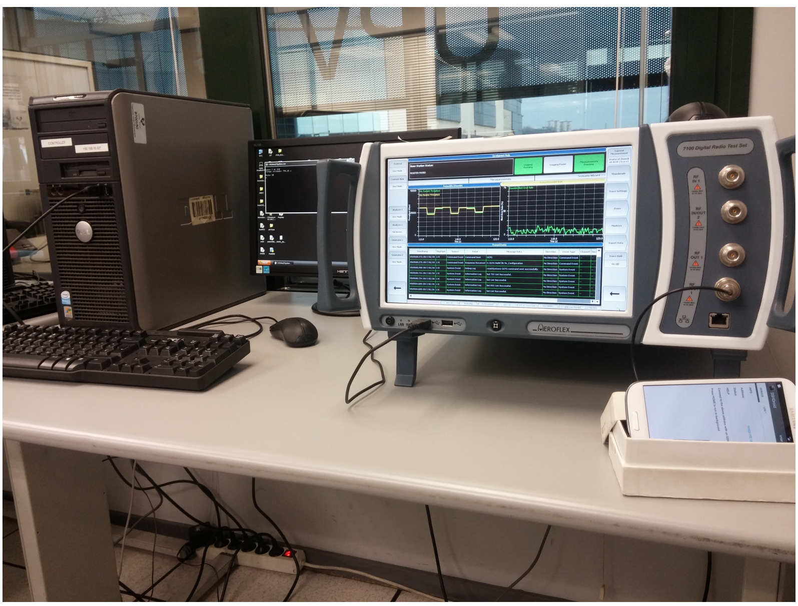


Figure 4.6: Emulated testbed: Aeroflex 7100, smartphone and controller

of any LTE device through an RF cable or over the air.

Since simulated environments where not realistic enough due to the lack of real UEs and LTE signalling among other limitations, we decided to use emulated environments in order to incorporate the use of real UEs that lead to realistic quality reporting to the eNodeB. As seen in the Background Chapter, the CQI reporting mechanism is the first step in a mapping chain until an available capacity is assigned to a precise channel. The realism of this process, with real UEs and a LTE-in-a-box that emulates the eNodeB side, allows measuring more accurately the impact of variability in the signal and how TCP is able to respond to such fluctuation. The LTE emulator also gives the opportunity to configure LTE parameters straightforward comparing with simulated environment. Finally, the LTE-in-a-box has the ability to collect meaningful traces such as CQI, BLER and so on for a more detailed and comprehensive results gathering. On the other hand, the utilized fadings have a random feature to make them closer to realism, but require a high number of experiments to have statistically representative samples to model the behaviour of both radio part and upper layers (TCP in this case). Apart from that, due to the limited coverage or use of RF cable, by-default emulated environments do not allow UEs' real movement.

For this thesis, we decided to use an emulated testbed in our lab with the following machines (part of the deployment is in Figure 4.6):

- UEs: Samsung Galaxy S III mobile phone and Samsung GT-B3730 LTE dongle. Both devices connected to the emulator through an RF cable in order to avoid undesired signal behaviour and to avoid legal issues due to radiating on licensed band.
- Server: Linux server with Linux kernel v4.10 to support the UEs during the experiments.
- Digital radio test set: Aeroflex 7100 is a full EUTRAN/EPC testbed with plenty of tests, configurable parameters and logging capabilities.
- Controller: Automation machine for the correct synchronization and commanding of the involved parts during the testing process.

4.3.3 Controlled deployment

Controlled deployments are a very good approach to experiment with real equipment and ad-hoc experiments out of the wild. Nowadays, there are many such facilities available due to the efforts made by federations such as Fed4Fire [117]. These infrastructures serve to model networks, study such networks under real conditions and validate new proposals among other purposes. Therefore, the usability of these infrastructures covers most of the research fields for their availability to test real interaction over the network.

Controlled testbeds constitute a good opportunity to perform experiments under different mobility patterns and speeds in a deployment with the whole equipment being real and having a real-world in small-scale.

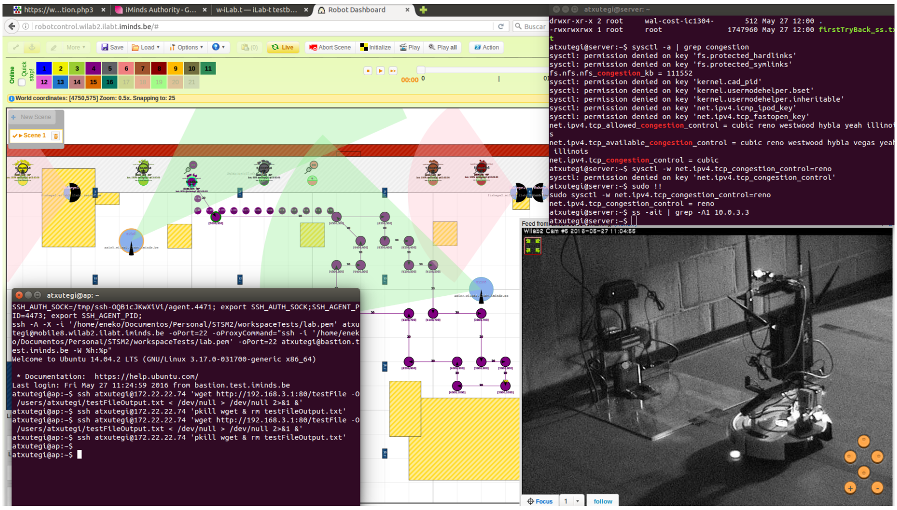


Figure 4.7: Example of experimentation over the LTE deployment of w-iLab.t

Looking for an alternative for emulated testbeds that could provide with mobility as well as the usage of fully real equipment, we decided to take advantage of iMinds' LTE facility (LTE w-iLab.t [118]) in Zwijnaarde, Ghent. In this particular case the deployment enables to research the performance at different TCP/IP layers. Besides, the deployment enables the design of ad-hoc mobility patterns helped by Roomba robots during the experiment (see a screenshot while experimenting over w-iLab.t in Figure 4.7).

Apart from all the positive features that have been highlighted regarding this kind of facilities, it has to be said that LTE transmission are done by air in a well prepared environment allowing a proper study of TCP and LTE events. Even though the movement is real, the space constraints of these small-scale deployments only allow very limited speeds.

4.3.4 Real-world deployment

There is nothing more representative than performing experiments in real-world deployments. However, the number of available ones (and the possibility to have access to one of them) is very low. Due to this lack of deployments and trying to cope with mobile networks problems, there is a significant effort dedicated to real-world measurements in FIRE [119]. All these federated testbeds [117] and pan-European projects [119] suggest an incremental requirement of studies over real-world and small-scale deployments so as to test, measure and improve different aspects of mobile broadband networks.

For instance, MONROE [120] (see a node used in this thesis in Figure 4.8) represents a good example of a pure-realistic deployment approach. It allows performing experiments in pan-European testing nodes with up-to-date equipment. Apart from

Figure 4.8: Experimentation node of MONROE MBB testing platform

the aforementioned strengths, real-world deployments give trustworthy insights of movement impact, cross-traffic loads impact and they open the gate to study traffic patterns and detect deficiencies, either due to coverage, equipment or resource lack.

As for the weaknesses, the possibility to gather information with high granularity is limited and so is the ability to infer inter-layer performance. Moreover, since they use commercial operators' coverage there usually is a limitation of data quota. This thesis has had the chance to experiment over such deployment during a limited period of time.

4.3.5 Comparability among steps

Considering all the aforementioned comments and described features of each deployment, Table 4.1 gathers the strengths and weaknesses of each step in the experimentation stair.

Once the experimentation stair has been explained, five important aspects need to be underlined:

- The ns-3 simulated/emulated environment is a useful deployment to start discovering interaction deficiencies between LTE and TCP due to its ability to collect information from different layer of the protocol stack. However, it lacks realism and other steps of the experimentation stair should confirm the findings of ns-3. Besides and in order to simulate scenarios closer to real-world, the use of Wideband CQI reporting mechanism has been proposed.
- The deployment of Aeroflex has one important limitation in order to be capable of serving as a realistic testbed for scenarios under mobility circumstances. To

Experimentation level	Strengths	Weaknesses
Simulated environment	-Repeatability -Cheapest option -Parameters gathering	-Synthetic fadings -Faked UEs -Hard modeling
Emulated testbed	-Real UEs -Easy LTE configuration -Radio info. collection	-Multi-experimentation required -No real movement
Controlled deployment	-Real movement -Ad-hoc patterns -Air transmission	-Speed/space limitation
Real-world deployment	-Real speeds -Real patterns -Ability to study realism	-Limited parameters study -Data quota

Table 4.1: Strengths and weaknesses of experimentation stair' steps.

cope with this drawback, the system must be able to change its SINR along the experiment, either by internal mechanism or by external commanding.

- Controlled deployments are needed to verify the findings before carrying out over the wild.
- The closest to real-world deployments, the harder impact on variable events in the performance due to the inclusion of more processing, more cross traffic, more queuing and so forth.
- With the proper configuration of scenarios, the experimentation stair is capable of confirming macroscopic findings and behaviors amongst different steps.

The current Subsection has been responsible for gathering the strengths and weaknesses of each step in the experimentation stair as well as describing the usability of each deployments considering its capabilities. This information is of utmost importance in order to select the most adequate testbed for studying each open-issue.

4.3.6 Importance of realistic UEs

Considering the emulated testbed as the first step in the experimentation stair that allows performing with real UEs and taking into account the substantial importance

No.	Propagation Condition	Base SNR
1	EPA5	20dB
2	EVA5	10dB
3	EVA70	20dB
4	EVA200	20dB
5	ETU70	10dB *
6	ETU300	10dB *

*Original testing point were placed in 0dB (edge of the cell).
Adaptation to the middle of the cell with 10dB.

Table 4.2: Testing points proposed by 3GPP.

of real UEs, the reported quality has been measured in the emulated testbed. The channel conditions of two UEs (Dongle and Mobile phone) have been modeled from their reported CQI in six of the LTE testing points that the 3GPP suggests (Table 5.5.4.5-2 in [16]). Table 4.2 gathers such testing points and their network conditions. Two modifications have been made in relation to the testing point with a base SNR of 0 dB. Due to the impossibility to hold a LTE attachment at 0 dB plus the impact of the fading, it has been decided to move the base SNR to the middle of the cell at 10 dB. All testing-points are static to establish the fading as the only SINR modifier and be able to study the behaviour of each UE in the same network conditions.

Considering 3GPP's testing points, Figure 4.9 shows as empirical cumulative distribution function (ECDF) the differences that the reporting of selected UEs have between each other in exactly same situations with sufficient repetitions (5) and logging time (120 seconds) [16]. It is clear that for exactly the same network conditions, different UEs may easily report a distinct quality and therefore the eNodeB will assign a different MCS for them, enabling in this case higher throughputs with the smartphone. Considering the maximum CQI value being 15, the existent difference between devices of 3-4 levels in the reported CQI values is very significant.

We conclude that, even using real UEs huge differences could appear. For that reason, it is important to perform experiments with up-to-date UEs and carry out comparative studies performing with similar or even same equipment, but also considering that it is difficult to infer from a certain outcome a general-purpose behavior.

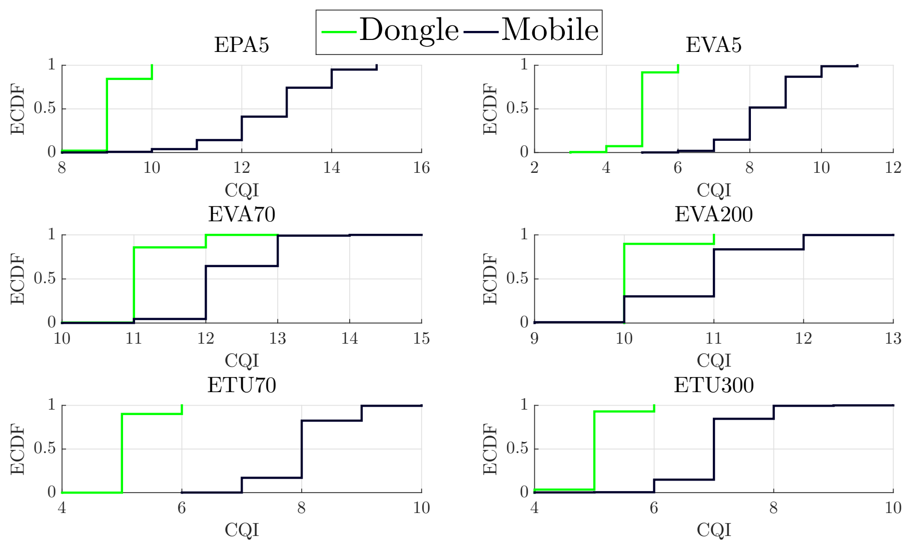


Figure 4.9: Channel quality by Dongle and Mobile phone in emulated testbed

4.4 Methodology

This section covers the methodology of this thesis, selecting the LTE deployments for the fulfillment of the analysis regarding the selected performance aspects (PA) to be considered in the intra-protocol selection framework (Defined as open-issues) as well as explaining the process to analyze them. Throughout the explanation of each part of the methodology, the section also explains the reasoning behind the experimentation over certain LTE deployment. The decision is based on the match between the technical requirements that each open-issue has so as to be appropriately analyzed together with the capabilities that each deployment offers. Table 4.3 shows the deployment selection wrap-up, allowing a better understanding in each part of the methodology.

Multi-testbed/ Open-issues	**Specific CCA features** - Real limitations - Microscopic eval. - Comparability	**Start-up performance** - Diverse variab. circumstan.	**Adaptability of CCAs** - Mobility - Dif. variabilities - Confirmation	**4G vs. low latency deployments** - Real UEs - Real delay - No cross-traffic
Simulated	X	X	X	
Emulated		X	X	X
Controlled		X	X	
Real-world	X	X		

Table 4.3: Open-issues and their relation with LTE deployments.

4.4.1 PA1: Impact of specific CCA features

In order to appropriately study the impact of specific CCA features in the same network conditions and analyze to which extent the selection of a certain CCA entails a precise performance or the achieved performance is obtained regardless the CCA, it is important to divide the analysis in two parts: 1) **PA1.1** is devoted to analyze the possible constraints that would prevent a certain CCA from achieving maximum potential in terms of throughput, radio resource utilization and so on. 2) Once the limiting factors are defined, the limitation are transposed into the simulated environment. Afterwards, the simulated environment is able to microscopically and individually assess every CCA and be capable of comparing them and detecting the features of each CCA and its performance-related implications (**PA1.2**).

4.4.1.1 PA1.1: Technical constraints

The analysis of technical constraints is divided in two tasks. On the one hand, the work with static constraints in real-world due to the necessity of the interaction between several clients with different servers. On the other hand, the study in simulated environment of TCP dynamics and the impact that may well have in final performance and the following confirmation of findings in real deployments.

In order to detect a static constraint that in the end could limit any transmission, the Window Scaling (WS) has to be taken into account. In order to better represent and manage the increasing buffer sizes in end-points, the WS field enables the expansion of TCP windows from a limiting management of the end-point buffers with 16 bits to 30 bits [23]. Therefore, the option modifies the ability of TCP and allows working with buffer sizes greater than 64 KB up to 1 GB. The WS option is negotiated and sent between end-points in the TCP handshake and once a negotiation is completely made, there is no room for reassigning (as stated in RFC7323 [23]). The space limitation that the WS establishes would be, throughout the whole transmission, the representation of the maximum manageable capacity of end-points. The main way of assessing such parameters is by gathering real traffic in a networking lab environment and capturing every single TCP handshake (where the WS is negotiated). **The dump aimed at having a perspective of current common WS values and study whether they are a limiting factor for the achievable rates or not in order to select the most appropriate TCP candidate with the intra-protocol selection framework**.

Regarding the dynamic constraint that could prevent a CCA from achieving its full potential, it is important to analyze the importance of buffer size in the bottleneck and its possible impact in the performance of TCP. In this regard, it is important to **extract to which extent network conditions like intermediate buffer length would equalize/normalize the behaviour of different CCAs and therefore affect the impact of our intra-protocol de-ossification proposal**. To this end and looking for an LTE deployment able to fully compare the dynamics of a certain CCA under the same network conditions with different buffer sizes, several tests need to be launched in the simulated environment under a dumbbell topology deployment.

Another important measurement set in the analysis of dynamic constraints is related to the testing of them in a real-world environment. Such analysis aims at assessing the CWND evolution in different real networks and **evaluating whether the limiting factors are located in the network itself (similar performance regardless the selected CCA) or in the end-nodes due to the evolution of TCP (improvements by selecting a more suitable TCP implementation with the intra-protocol selection framework)**.

4.4.1.2 PA1.2: Detection of CCA features

Once the limitations have been studied and their implication has been clarified, the analysis aims at analyzing the implications of the CCAs themselves by detecting their main features in distinct network conditions. To this end and willing to avoid variability sources, static positions are proposed for the evaluation. The static scenarios aim at providing insights of the evolution and responsiveness of TCP under different background traffic features. There are two main goals with this scenarios: the comparative study of TCP behavior with and without a loaded cell, and the responsiveness comparison of TCP variants with sudden capacity increase and decrease. **These experiments aim at identifying the effect of cross-traffic and available bandwidth changes into different CCAs so as to provide more input regarding the responsiveness of TCP flavours to the CCA selection logic in the intra-protocol selection framework**. In order to study the features and impact of different CCAs and be able to control the behaviour of the network, the simulated environment is picked. Among the selected TCP CCAs, TCP BBR is not tested due to the present impossibility to perform with it in native ns-3 or DCE.

The UE is located in an average static position that allows half of the maximum throughput of the cell (35 Mbps). Different experimental trials need to be carried out with and without background traffic to study the responsiveness of TCP and deduct if the background traffic has the same impact among the CCAs. The total target load of the background traffic is set to the 50% of the link capacity so as to allow bottleneck fairness by-default. Regarding the sudden changes in the traffic load, two type of simulations are designed: with the background traffic being stopped at 20 seconds of the test and with the background traffic being started at 20 seconds of the test. Regarding the impact of each CCA, four main parameters are studied: the CWND evolution, the throughput over time, the delay and the packet drop count.

4.4.2 PA2: Start-up performance

The Start-up phase of TCP is driven by different Slow Start mechanisms depending on the selected TCP candidate. In order to carry out a comprehensive review of the Start-up phase and its performance, the analysis is divided in two: 1) Since the Start-up phases cover a short period of time, the variability of the network context could lead to different results. Therefore, **PA2.1** covers the analysis of distinct Slow Start mechanisms over all presented LTE deployments (see Table 4.3). 2) **PA2.2** was added to the thesis during the last year due to the inclusion of TCP BBR in the Linux kernel and its proposal of a novel Start-up phase. This new method required analysis and comparison alongside less recent Slow Start mechanisms in order to asses the selection suitability of each Start-up method in the proposed framework.

4.4.2.1 PA2.1: Multi-deployment Start-up comparison

The analysis over all the LTE deployments validates the evidences under different circumstances and based on setups that rely on very distinct equipment. **The main point of the analysis is to study the behaviour of different Start-up methods in variable MBB conditions and extract from there the suitability of each mechanism in order to add such a logic to the intra-protocol selection framework**. In the initial stage of this thesis, two are the main Slow Start methods under test: The Standard Slow Start utilized by NewReno and Hybrid Slow Start by CUBIC.

- Standard Slow Start [111]: During this period the CWND typically increases by one packet for each ACK reception until it reaches *ssthresh* value. After that period, the CWND enters a congestion avoidance phase. The most common configuration avoids this behaviour in the beginning of the transmission and establishes the *ssthresh* with an infinite value, so it can ramp-up to greater CWND levels. The whole ramp up phase ends when a congestion event is detected by 3 DUPACK reception or by a time-out. After the triple DUPACK, the back-off policy is applied (in the case of basic NewReno, halving the CWND and the new *ssthresh* to previous CWND). However, if a time-out occurs the CWND will be decreased to one packet and the *ssthresh* to half of previous CWND so that it can enter congestion avoidance once it gets that value.

- Hybrid Slow Start [59]: This method aims at finding the proper exit point for Standard Slow Start in order to advance to congestion avoidance phase without causing heavy packet loss. The detection of such point is based on the measurements of ACK trains and RTT delay samples to detect whether the path's bottleneck is congested or the current value of the CWND has already achieved the available capacity.

4.4.2.2 PA2.2: Thorough comparison of Start-up methods in real-world

Previous Performance aspect analyzes the suitability of Start-up methods in variable MBB conditions. After the knowledge gathered with such an analysis, this Performance aspect **studies more complex and realistic scenarios by adding the transmission of a mixture of long and short flows in the bottleneck and the configuration conditions of distinct operators**. The aim of this analysis is to extract whether the mixture of traffic and different conditions in the operators could lead to a different behaviour that should be taken into account in the intra-protocol selection framework or not. Together with the inclusion of TCP BBR in the thesis during early 2017, a new Slow Start method was added in the Slow Start options to be tested.

- TCP BBR uses Startup phase to ramp-up as the classic Slow Start method but exit the increase when the gain in throughput during three consecutive transmission rounds is below 25% [36,44]. Within three rounds, BBR ensures that there is no limitation of the rate due to a underperformance of the reception-window.

In order to properly analyze classic Slow Start, Hybrid Slow Start and Startup and after the experience with the analysis of the previous performance aspect (Performance aspect 2.1), the selected deployment is the real-world one, MONROE testbed in our case. Considering that the usual Internet loads are a mixture of short and long flows, it is important to build the experiments based on a mixed traffic in order to better analyze the impact of Slow Start mechanisms. In long flows, both Slow Start and CA mechanisms of TCP are involved, whereas in the short flows the Slow Start mechanism has the highest impact. So, the CA phase is more significant in long flows during the steady-state period and the Slow Start method has an impact regardless the flow length but it is more prominent in the performance of short-lived transmissions. Therefore, the main focus of these experimentation is on short-lived transmissions.

The analysis is based on the following measurement proposal: Each set of experiments go through the steps depicted in Figure 4.10. The left part of the figure illustrates the steps of a single experiment in a flow chart, and the right part provides a flow chart to explain the automatic execution of a whole set of experiments. Each set comprises the execution of single experiments under different circumstances: different delays between the beginning of the long flow and the first short flow, different operators, different types of experiments (short and long), different CCAs and different iterations of the same configuration. In order to increase the number of experiments or samples with similar network configuration, the whole set of experiments was launched in four different days:

A) A download of a long file is launched in the background (10 MB in short experiments -X1- and 30 MB in long experiments -X2-).
B) A predefined delay is set (three possible options: 0.5 seconds -Y1-, 0.1 seconds -Y2- or nothing -Y3-). This delay lets the long flow grow in terms of CWND and achieve greater bandwidth than from the beginning of the transmission. The more bandwidth it takes, the more should reduce its speed in order to give a fair share to an incoming flow. Different delays are utilized in order to introduce the first short flow during different stages of the long flow.
C) A download of a short file is launched (always 1MB). A file of 1MB is considered short enough to be downloaded quick (in about 6 RTTs) but long enough to have some impact in the performance of the long flow.

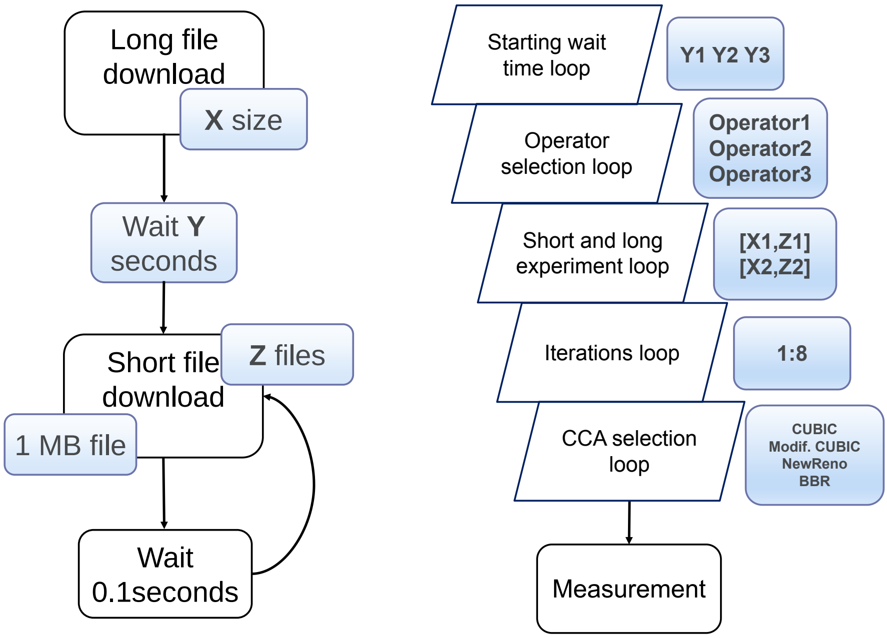


Figure 4.10: Measurement procedure: a) Measurement framework; b) Automatic measurement loop.

D) Once the download of the short file is finished, the next short flow is delayed (always 0.1s). This delay supposes approximately a couple of RTTs in the tested mobile networks and lets the long flow in the background get more bandwidth. This way, if another incoming short flow is launched, the long flow would need to reduce and adapt its pace.
E) Another short download is performed and the next one is delayed as well (10 times in total for short experiments -Z1- and 20 times for long experiments -Z2-).
F) The experiment ends either when the last short flow is downloaded or the long flow finishes (in case the download of the last short flow is completed before the download of the long flow).

In order to carry out measurements with all the experimental conditions one after the other and also launch different repetitions for the same experiment, a five dimension loop could be run (See the right side of Figure 4.10).

4.4.3 PA3: Adaptability of CCAs

The analysis regarding the adaptability of CCAs to challenging mobility circumstances is divided in four parts due to the implication of several factors while experimenting on the move and the necessity of individually studying them so as to achieve a more appropriate CCA selection. 1) **PA3.1** is devoted to analyze the impact of different movement patterns on the performance of goodput under considerably stable fading conditions. The application of a non-aggressive fading is done to properly measure the impact of movement pattern in a pseudo-isolated way. Besides, with the fading pattern we also avoid experimenting with an unrealistic "stepped" channel quality. 2) **PA3.2** analyzes how only a greater speed in similar network conditions could impact on the performance while moving. 3) **PA3.3** introduces different variabilities in the channel while moving and evaluates its implication in the performance of different CCAs at different speeds. 4) Finally, similar to the evaluation in static position in **PA2.2**, **PA3.4** analyzes and gathers the CCA features on the move.

4.4.3.1 PA3.1: Impact of movement pattern

The analysis aims at studying the impact of different movement patterns in the performance of TCP under considerably smooth fading conditions. This is, the important aspect of the analysis is not the fading itself, but the performance differences between distinct mobility scenarios. **This study seeks for the impact that moving to better or worse positions would have on transport protocol performance depending on previous network state**. The emulated testbed is selected due to its inclusion of real UEs and the possibility of performing long rounds of experiments with controlled mobility. This analysis will contribute to the study of UEs' movement while performing TCP. The movement and the mobility pattern itself has an impact in the actual resource assignment to a certain UE due to the possibility of a change in the UE's SINR. This is, the movement itself could provoke changes in the channel quality and therefore, assigned resources to the moving UE.

The simplest movement patterns are straight and with a constant speed, but even those, are capable of revealing important features. The mobility of a UE could be represented as a variation of SINR. In addition to this baseline SINR variation (modified by the controlled in Figure 4.11), a fading pattern is applied. Willing to better understand the pure relation between TCP's outcome and different mobility patterns, a relatively stable fading pattern is selected - Extended Vehicular A model 120 (EVA120). This fading makes the experiments closer to the effects that could happen in considerably stable channel conditions in real-world with air transmission and Doppler shift. The only lacking point in the emulator is the cross-traffic present in the wild. However, in order to study different movement patterns and be sure about the reason behind a certain performance or effect, it is better to avoid loaded networks.

Being more precise regarding the final movement patterns, our analysis considers two specific cases, the forward movement of a UE or decreasing quality movement and the backward movement or increasing quality movement.

- Forward movement or decreasing quality movement: This scenario evaluates the behaviour of TCP with a UE holding a constantly worsening channel quality. Considering the simplified single user scenario, the MCS assignment will be based on the "based" SNR (modified by the controlled in Figure 4.11) and the selected fading. The idea behind these experiments is to evaluate the CCA's adaptability in a continuous capacity reduction environment.
- Backward movement or increasing quality movement: The behaviour of TCP on a constantly improving channel quality is now presented. These tests aim at evaluating the CCA's adaptability in a continuous capacity increase.

The current analysis will be focused on the simplified impact that moving to better or worse positions would have on transport protocol performance depending on previous network state.

4.4.3.2 PA3.2: Impact of speed

This phase of the analysis is devoted to study the responsiveness of TCP under different speeds with similar network conditions. So, the main controversial effect in this analysis is the speed itself. The responsiveness of TCP is measured by counting the transmission opportunities that are wasted in the eNodeB because of the impossibility of TCP to sufficiently feed the network under such circumstances. The simulated environment allows gathering such parameters in the eNodeB and therefore it is selected to carry out the analysis regarding this performance aspect.

In previous explanation EVA120 has been utilized as a fading example, whereas this time EVA60 is going to be applied due to its existence in the available fading pattern in ns-3. The same fading is applied in different speeds in order to have the speed as the unique distinct parameter. The performance of 60 km/h or common limitation in rural roads and 200 km/h or high-speed trains is studied in this phase.

Throughout this analytical phase, it will be analyzed under different speeds but similar network conditions the ability or inability of TCP to take full advantage of the available transmission slots that the eNodeB offers (in terms of transmission opportunities or TxOps). Since different MCS values result in different available capacities and therefore different achievable throughputs and injection levels to be achieved by TCP, the analysis is divided by MCS ranges. Since the control of eNodeB stats are required, simulated environment is requested.

4.4.3.3 PA3.3: Impact of different variability conditions on the move

This stage in the analysis covers the study of different speeds and variable conditions and the impact in the performance of TCP. One of the main targets is to add different speeds and fadings to the already explained movement patterns so as to understand the impact of every factor in the final outcome. **This study serves to identify whether only speed and trajectory are the main mobility-related heuristics in the intra-protocol selection framework or the variability of the mobility scenario itself should be also added to the driving heuristics that will decide among the most suitable CCA**. Since the fadings may alter the reporting of CQI to the eNodeB and therefore, the final achievable rate could suffer modifications, real UEs are needed in this occasion in order to represent results and findings as close to real-world as possible. To that end and also taking advantage of the complex and realist generation of fadings, the emulated testbed is utilized.

Since different mobility schemes lead to distinct challenges for TCP, this analysis aims to cover the performance of TCP under three different speed and fading combinations. The patterns do not only represent different average variabilities of the available bandwidth over time, but also differences in terms of variability between consecutive available bandwidth samples. In our precise analysis, the selected speed and fadings are the following:

- A scenario at 60 km/h with the application of Extended Vehicular A model 60 (EVA60) fading model (common limitation in rural roads).

- A UE at 120 km/h modelling the maximum highway speed in any European country with EVA120.
- A scenario at 300 km/h with High Speed Train (HST) fading model (HST300).

4.4.3.4 PA3.4: Detection of CCA features on the move

Taking into account the previous phases in the analysis, **it is important to analyze the different TCP flavours and detect their main features under mobility circumstances**. It is an evolution and combination of previously covered mobility scenarios. In order to confirm the behaviour of the CCAs under mobility conditions, different LTE deployments are used.

4.4.4 PA4: 4G vs low latency deployments

The end-to-end delay differentiation that is treated in this thesis is related to the latency gap between realistic 4G deployments and low latency scenarios that are close to the future delays in 5G networks that would allow proximity service provisioning. While comparing the 4G and 5G deployments, we come across multiple differences in regards to the capabilities in terms of modulation, achievable rates, end-to-end latencies and so forth. Due to the lack of actual 5G deployments, in 4G scenarios that mimic 5G by low latency conditions, the two schemes only differ in the location of the server and therefore, in the end-to-end delay between the server itself and the UE. Two are the main targeting schemes: a) 4G latencies; b) Lower latencies that move towards the overall potential delays in 5G networks. In scenarios close to the target capabilities of 5G, it is common to find Radio Access Network (RAN) caching [18] or the ability of instantiating the server management as close as possible to the core mobile network [19], reducing the latency and shortening the server's period of times between consecutive decisions. **It is important to analyze the performance of different implementations of TCP with different end-to-end delays or latencies so as to infer which candidate fits better under which network circumstances**.

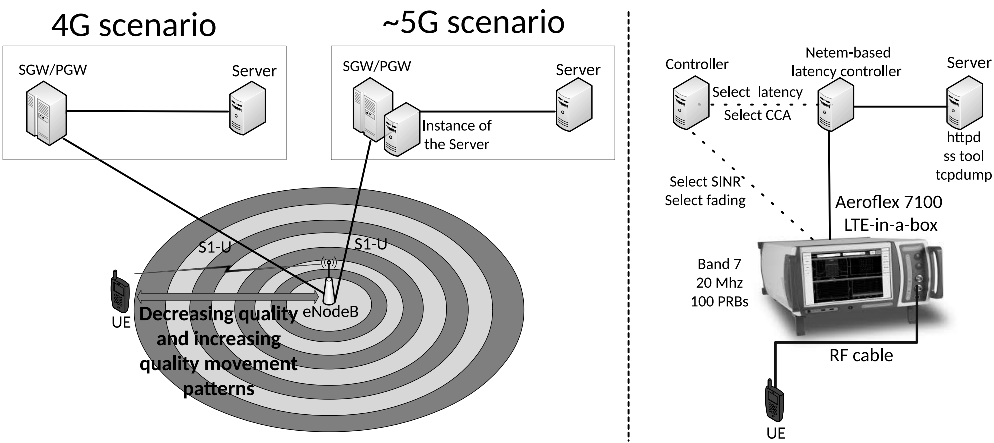


Figure 4.11: Testbed with Aeroflex 7100.

Figure 4.11 shows on the left the two use-cases based on final end-to-end latency. On the right part of Figure 4.11, the emulated version of such use-cases is represented. Among the deployment options, we select to use an emulated environment with an LTE-in-a-box (Aeroflex 7100), one server, one computer that plays the role of the bottleneck by controlling the latency with *netem*, a smartphone as a UE and a controller to synchronize the experiments. The controller not only synchronizes the execution of the experiments, but also configures and commands the rest of the equipment involved in the testing process.

When referring to realistic 4G latencies, the delays do not necessarily have to be huge constraints for the good performance of the experiments, but a representation of baseline RTTs present in real-world. Depending on the country and area, the experienced latencies could vary drastically. For this reason and due to the lack of operators' information disclosure, it is difficult to emulate meaningful LTE scenarios in terms of end-to-end delays. According to the following report [20], in USA during the third quarter of 2015, the average latencies were between 68ms to 85ms in 4G networks. In order to experiment with realistic 4G delays but being aware that the selected values are not applicable for all 4G scenarios, 68ms are assigned as the minimum latency value in the bottleneck. Regarding low delay scenarios as the 4G adaptation towards 5G, the system is configured to work under the minimum delay of the LTE-in-a-box deployment, being on average about 18ms.

4.5 Conclusions of the framework and methodology

This Chapter has been responsible for proposing an intra-protocol de-ossification framework. Considering the different pieces needed to appropriately build the framework, the explanation has been divided into Sections:

- Section 4.1 has described the intra-protocol de-ossification framework as well as the technical requirements in order to cope with the defined research open-issues in the previous Chapter. The framework aims at appropriately deciding the most suitable CCA for each situation based on the outcome of performance-related metrics (results of open-issues) and the application requirements. The term intra-protocol corresponds to the sole selection of TCP CCAs in the whole selection process in the framework. Part of the de-ossification is enabled due to the use of the selection criteria as a transport service together with a a context of transport protocol from where all CCAs are fed with initialization network information.

- Section 4.2 has described the selected CCAs for the analysis, introducing their most important phases and features and reasoning the selection and inclusion of them in the candidates under test. All in all, the selected group is formed by loss-based CCAs (NewReno and CUBIC), delay-based variants (TCP CDG), hybrid solutions without bandwidth estimation (Illinois) and hybrid solutions with bandwidth assessment (Westwood+ and BBR).

- Section 4.3 has covered the description of four different LTE deployments in order to allow the measurements needed by the open-issues: a simulated environment, an emulated testbed, a controlled deployment and a real-world deployment are described. Besides, the Section has compared the deployments highlighting the potential and limitations of each of them. Finally, the Section has underlined the importance of using real UEs in measurements over MBB.

- Section 4.4 has comprised the explanation of the methodology in order to cope with each part of the analytical open-issues. The Section has selected the assigned LTE deployments for the fulfillment of the analysis regarding the performance aspects (PA) to be studied (Defined as open-issues) as well as explaining the process to analyze them. The decision of one LTE deployment or other has been based on the match between the technical requirements that each open-issue has together with the capabilities that each deployment offers. Besides, the Section has described the methodology for the analysis of each performance aspect. PA1 (Impact of specific CCA features) has been split in PA1.1 (Technical constraints) and PA1.2 (Detection of CCA features), while PA2 (Start-up performance) has been divided in PA2.1 (Multi-deployment Start-up comparison) and PA2.2 (Thorough comparison of Start-up methods in real-world). Finally, PA3 has been cut in PA3.1 (Impact of movement pattern), PA3.2 (Impact of speed), PA3.3 (Impact of different variability conditions on the move) and PA3.4 (Detection of CCA features on the move), whereas the PA4 (4G vs. low latency deployments) has remained as a single study entity.

5 Analysis

Keeping in mind the long list of TCP implementations family and the idea of using and appropriately selecting them based on the network conditions and application requirements, we have detected four major performance aspects. These aspects can impact final outcome and require further analysis so as to understand the interaction effects between TCP and different network circumstances in MBB. Willing to cover the analysis of each Performance Aspects one by one, the current chapter is divided in sections as follows:

- Section 5.1 covers the technical constraints that could limit a CCA and prevent from achieving maximum capacities (**PA1.1**). Such limitation could prevent transport protocols from achieving full utilization of the radio resources that the cellular networks provide. The targets of this section are network-agnostic, thus, the analyzed limitations comprise end-points or general aspects such as intermediate queue length, outstanding data or the CWND evolution.

- Section 5.2 comprises the detection of CCA features by analyzing the implication of cross-traffic and responsiveness of TCP (**PA1.2**). As explained before, the selected scenarios aim at providing insights of the evolution and responsiveness of TCP under different background traffic features. There are two main goals in order to appropriately detect the characteristics of each CCA: the comparative study of TCP behavior with and without a loaded cell and the responsiveness comparison of TCP variants with sudden capacity increase and decrease.

- Section 5.3 analyzes in all selected LTE deployments the performance of Slow Start methods in order to detect the best and worst candidates that could have a significant effect in short-lived flows and the beginning of the transmissions (**PA2.1**). Besides, it provides with a more detailed analysis of different Slow-Start mechanisms over a real-world testing deployment in order to better understand the suitability of each Start-up method for its selection in real deployments (**PA2.2**).

- Section 5.4 is devoted to analyze the impact of different movement patterns in goodput performance under considerably stable fading conditions. As a simplification and in order to better understand the behaviour of TCP under mobility circumstances, forward movement and backward movement have been selected as mobility scenarios (**PA3.1**). The movement patterns are also analyzed together with the differences present in TCP performance in 4G scenarios and in low latency scenarios that moves towards the overall potential delays in 5G networks (**PA4**).

- Section 5.5 comprises the analysis of different performance aspects related to the speed itself and how TCP is capable of adapting to the fluctuation created in the channel conditions while moving. The Section is divided in three different parts: 1) It is analyzed the impact of faster UEs when the channel conditions are similar and the parameter of speed is the only source of disparity (**PA3.2**). 2) The Section covers the impact of different fading patterns attached to distinct speeds in order to evaluate the effect of mobility from a more realistic but complicated point of view (**PA3.3**). 3) Finally, the Section applies the knowledge regarding the detected features for each CCA and the understanding of mobility conditions and detects the behavioural patterns of each CCA on the move (**PA3.4**). Throughout this Section and in order to study the effects of mobility step-by-step for 4G and low latency deployments, the end-to-end latency is considered as well (**PA4**).

- Finally, Section 5.6 summarizes all the detected findings. Out of the selected Performance Aspects, these major findings would be considered as important decision points in the intra-protocol de-ossification framework and its decision routine. In order to help identify the treated performance aspects in each section, each title adds [PAx] so as to indicate that the Performance aspect x is covered.

5.1 Current technical constraints - PA1.1

This section covers the detection of possible network and system constraints that could be found in fixed and mobile networks. Such limitations could prevent transport layer from performing at its full potential and utilizing the resources that cellular networks provide. The organization is as follows:

- Subsection 5.1.1 describes the WS limitation and the possible impact that could cause in end-to-end systems. In case of a hard limitation, since the limitation is end-to-end, the whole transmission and capacity of the CCAs to operate would be constrained. Thus, the Subsection analyzes the current limitation from the transmission taken in a research lab environment.

- Once analyzed the possible static constraint, Subsection 5.1.2 studies and explains the possible dynamic limitations regarding the transport layer. Within the effects that could dynamically impact the final transport protocol performance, two different features have been covered: the impact of intermediate queue sizes and the impact of CWND evolution.

5.1.1 Static constraint: WS negotiation

As explained before, the WS [23] field represents the maximum manageable capacity in the end-nodes. The limitation is negotiated during the transmission establishment and there is no possibility for the end-nodes to change their negotiated values if they realize that their capacity is bigger than expected. Therefore, by a set of experiments we wanted to clarify whether this would impose some constraints to the maximum available capacity. As a result of a live WS dump, Figure 5.1 shows the gathered values .The live WS dump aimed at having a perspective of current common WS values and study whether they are a limiting factor or not for the performance of TCP. The results are shown in Figure 5.1. The top figure shows the WS related to the clients, whereas the bottom figure depicts the WS negotiated by the servers when researchers are accessing Internet resources.

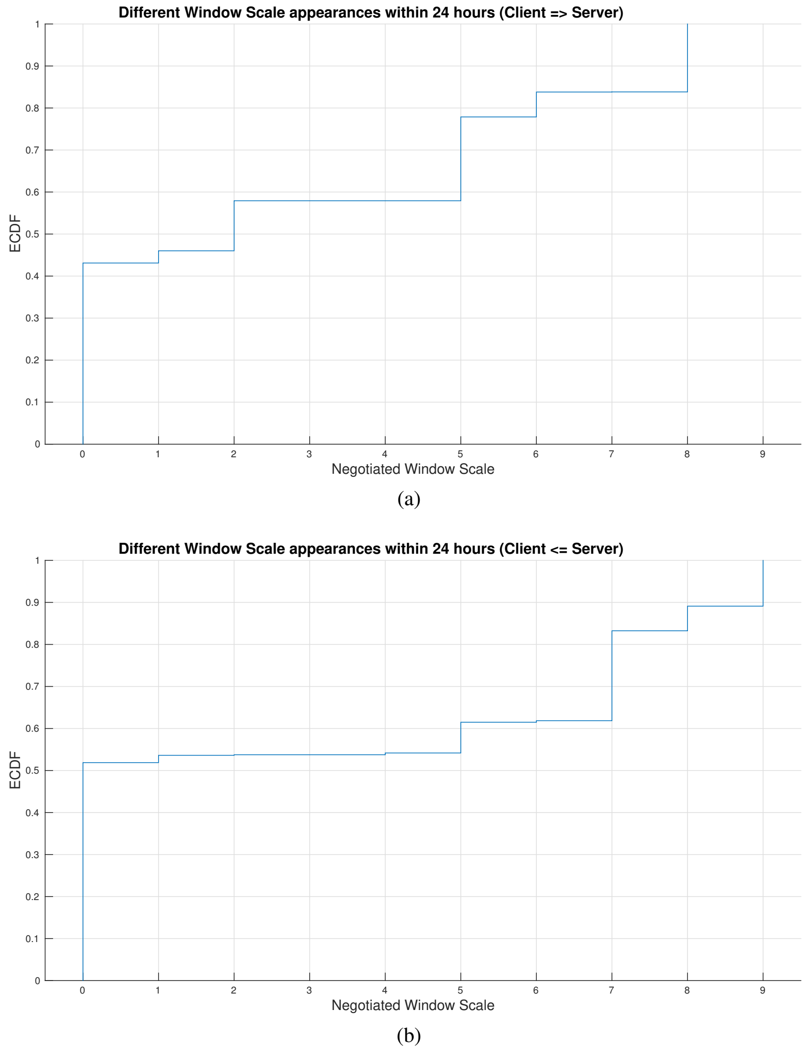


Figure 5.1: Window scaling negotiation: a) Clients negotiation; b) Servers negotiation.

The clients' related Subfigure 5.1a displays a wide variety of results, with multiple high and low WS values. Servers (see Subfigure 5.1b), in contrast, usually announce either very high values or no WS at all. The spread of the clients graph would be related to the heterogeneity of our lab equipment and the buffer diverse capacity, which consists of laptops, desktop computers, probes, smartphones, and so forth (depicted in Subfigure 5.1a). On the other hand, publicly available servers (Subfigure 5.1b) would have either very long buffers, and therefore high WS capable of serving a huge amount of traffic per connection, or they would not negotiate WS at all (non-WS) or have short buffers (WS 0) to support multiple concurrent connections without running out of RAM. Therefore, the final conclusion is that, in a **significant number of cases, either non-WS or WS 0 options were negotiated, resulting in the maximum capacity of the receiving buffer being 64 KB**. Figure 5.2 shows an example of the concurrent tranmissions needed to fill the pipe in diverse combinations of available capacity and negotiated WS taking into account that the baseline delay is 150ms.

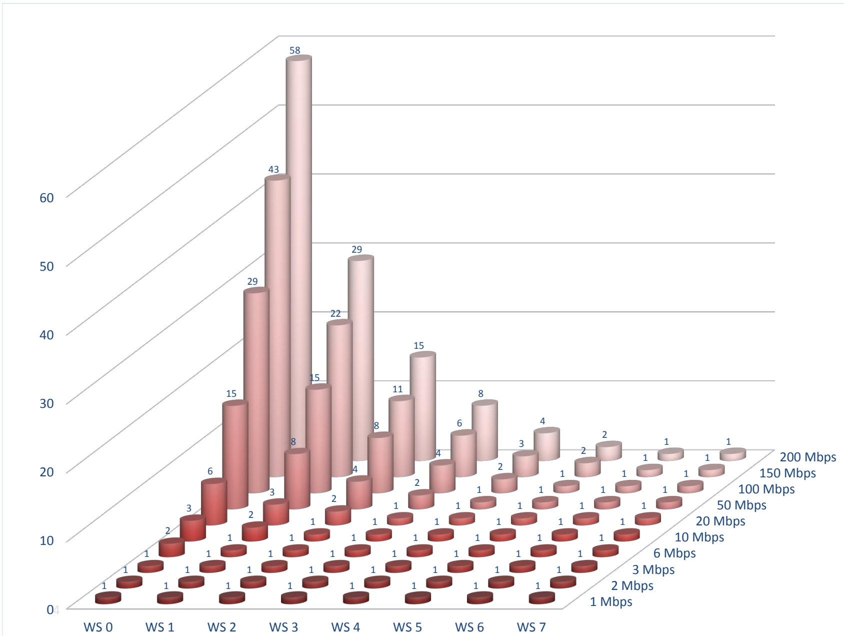


Figure 5.2: Relation between WS assignment, end-to-end delay and available bandwidth with the number of concurrent connections.

A trivial calculus points out that such a low value becomes a huge constraint in high BDP networks, because it does not allow the maximum capacity to be reached or demands an unrealistic number of parallel connections to be able to utilize all the available resources. For instance, if we calculate the required number of parallel connections for a network with high BDP (100 Mbps capacity and RTT=150ms in this example), the result will show the need for 29 parallel connections to fill the resulting BDP (1875000 bytes) with non-WS or a WS equal to 0 (limit of 64KB in the receiving buffer). The requirement of such a high number of parallel connections to be capable of filling the pipe so as to be able to utilize all resources is totally unrealistic. Therefore, it is **important to consider the WS** negotiation for two reasons:

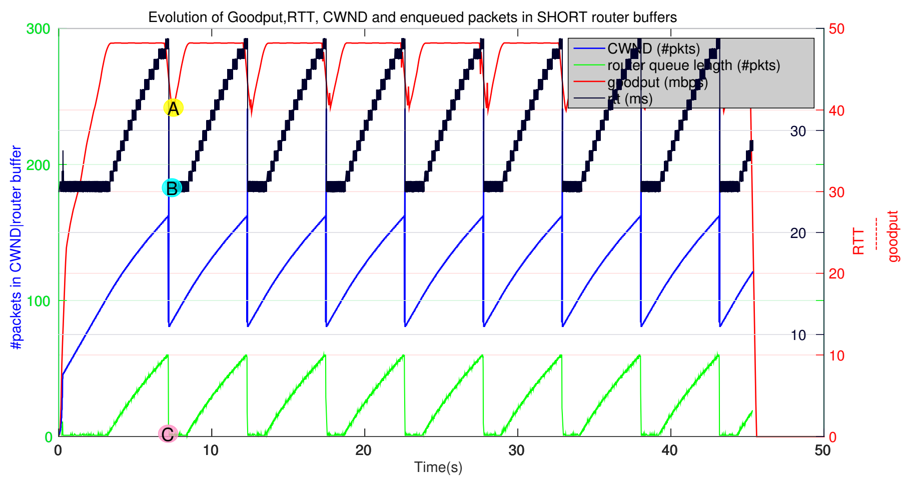
1. to **select a more appropriate TCP candidate** taking into account the achievable rates: more aggressive in case of big receiving buffers and successful WS ne-

gotiation and less aggressive implementations when the limitation only allows low capacities due to either short receiving buffer of inaccurate WS parameter.

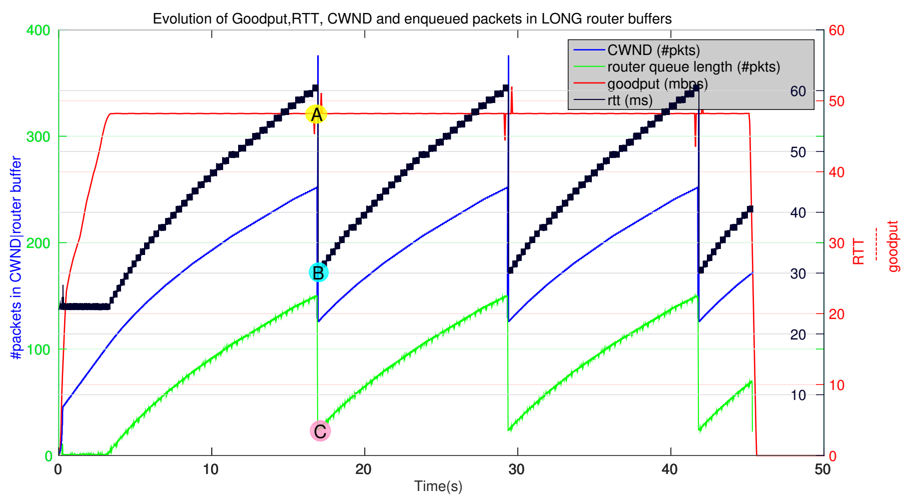
2. to **check** whether **this constraint** has prevented users from achieving the maximum available bandwidth during a precise situation throughout the experimentation phase of the thesis.

5.1.2 Dynamic constraints

Once the possible static constraint of WS negotiation and the relation with the length of the receiving buffer have been detected and in order to evaluate the actual impact



(a)



(b)

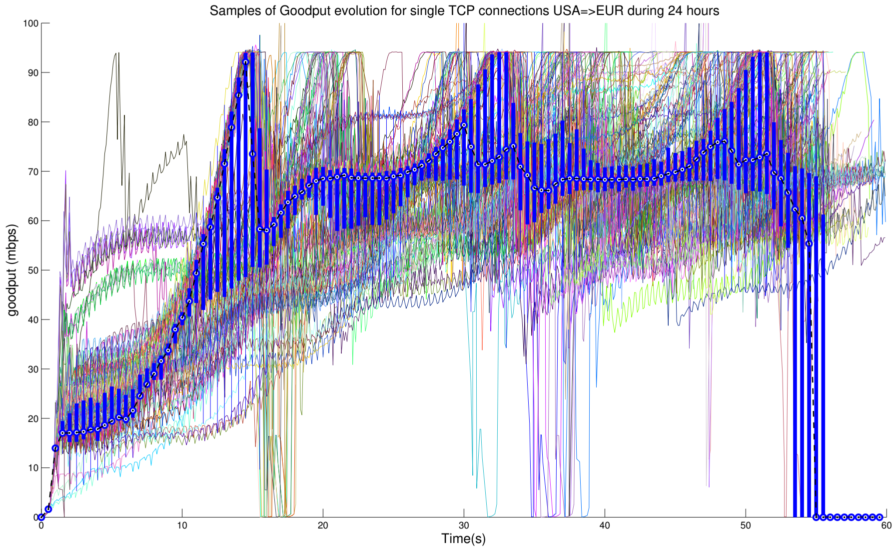
Figure 5.3: Impact of queue length: a) Short buffer; b) Long buffers (bufferbloat).

and the relation of the CCA mechanism on the goodput performance of TCP, several tests are launched regarding **TCP dynamics**. In this regard and after the analysis of buffer limitations in the end-nodes, we designed a set of experiments to isolate **to which extent network conditions like intermediate buffer length would equalize/normalize the behaviour of different CCAs and therefore affect the impact of our intra-protocol de-ossification proposal**. Thus, we analyzed importance of the intermediate queue sizes and the possible effect in TCP's performance. Figure 5.3 shows only two significant graphs for short and long buffer lengths for illustration purposes (NewReno is used as the CCA). In these graphs several performance-related parameters are gathered so as to give a broader view of the TCP behaviour with different lengths of intermediate buffers. In order to illustrate all parameters outcome in a single shot, different colours are used. In this regard, the evolution of the CWND (in blue), the number of enqueued packets (in green), the RTT (in black), and the TCP goodput (above in red) are represented.

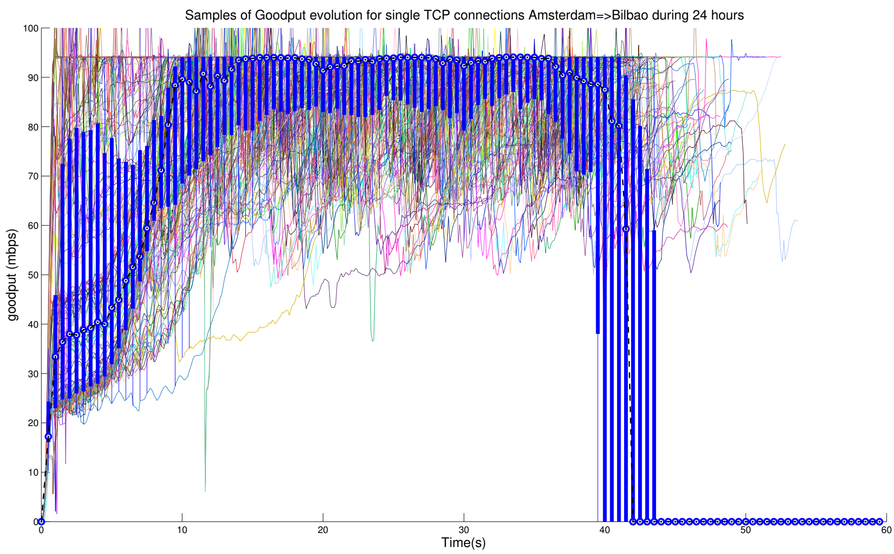
Subfigure 5.3a depicts the performance results while the intermediate buffer is short. As shown in the figure, the resulting goodput is CWND dependent (point A) for the short buffer configuration case. This is caused by the intermediate node's buffer starvation (point C) leading to available capacity underutilization. As a side effect, the component of the RTT related to queuing disappears, resulting in RTT close to pure transmission delay (point B). On the other hand, Subfigure 5.3b shows the overall performance under the same conditions but with longer intermediate queue size. In this case, we can find a couple of different effects due to no buffer starvation (e.g., see point C, the number of enqueued packets never drops to zero). First, since the buffer always has enqueued packets to transmit (*bufferbloat* effect), goodput remains stable (point A) immediately after a Slow Start phase and regardless of the CWND evolution.

Therefore, from the goodput's point of view, the effect of the CCA algorithm (and dependence on TCP flavour) is minimal under certain network circumstances that include long intermediate buffers. In terms of RTT, since the queuing delay follows CWND behavior, so does RTT, never falling back to transmission delay (point B) and being potentially harmful in a multi-client environment with shared bottleneck that will lead to impact as cross-traffic for the rest. That is, **only by evaluating the goodput outcome several CCAs may well report similar results that would reject the usability of the intra-protocol selection framework. However, depending on the performance metric or the driving heuristic, if other parameters such as RTT are involved, the framework would still be useful due to its capacity to decide the most appropriate candidate based on more balanced (goodput plus other aspects such as RTT to assess circular dependencies of TCP as well) performance-related metrics**.

Following the analysis of TCP dynamics and its possible constraints not only for the TCP performance, but for the usefulness of the intra-protocol selection framework, we wanted to check in a real-world set of measurements whether the limiting factors were located in the network itself (no consideration of the selected TCP candidate) or in the end-nodes due to the evolution of TCP (possibility of improvements by selecting a more suitable TCP implementation). After three days of measurements, a pair of graphs are selected to explain the two main cases regarding the gathered test behaviour.



(a)



(b)

Figure 5.4: Goodput evolution in real world measurements: a) From USA to Amsterdam; b) From Amsterdam to Bilbao.

Subfigure 5.4a shows a set of tests between Amsterdam and Virginia throughout an entire day. Every sample's goodput (every execution in a different colour) and the average goodput (black line and blue bars deviation) are depicted. Even visually, the obtained curves show very clearly that obtained goodput follows the typical evolution of CWND. This may well be caused by insufficiently long intermediate or client/server buffers, or the use of insufficiently aggressive (slow growth policy) TCP for this scenario, resulting in a non-filled intermediate buffer. Therefore, the graph demonstrates that the selected TCP candidate (CUBIC) was far away from the available capacity, opening the possibilities of the intra-protocol selection framework so as to decide in accordance to the network conditions and pick a more suitable CCA to make the most of the available network resources.

Finally, Subfigure 5.4b shows the result of a single day between Amsterdam and Bilbao. In this case, the average value looks very stable once the pipe is filled, due to no buffer starvation and having a TCP flavour aggressive enough to continue "feeding" the network. In a situation like this in which the baseline end-to-end RTT is by far shorter, even a Slow Start phase would be almost enough to achieve maximum capacity and be able to sustain it. In contrast, the greediness of this behaviour would affect the performance of flows that utilize the same bottleneck due to the induced increment in injected packets and therefore, in delays.

Therefore, **depending on the specific network configuration we check that both in simulation and real-world cases the actual impact of the CCA could be biased only by looking at the goodput but could bring important improvements while the evaluation and following intra-protocol selection considers other performance-related aspects**. All these findings foster the use of our CCA selection idea and therefore encourages the analysis and use-case validation of our proposed intra-protocol de-ossification framework.

5.2 Cross-traffic impact and responsiveness of TCP - PA1.2

Following the analysis, this Section aims at identifying the effect of cross-traffic and available bandwidth changes into different CCAs so as to provide more input regarding the responsiveness of TCP flavours to the CCA selection logic. The section is divided in two main experiments:

1. The comparison between a single-UE without cross-traffic and a loaded network in order to study the behaviour of CCAs with and without competing traffic in the bottleneck.

2. Sudden increase and decrease of the available capacity so as to measure the responsiveness of CCAs to quick and drastic changes in the capacity.

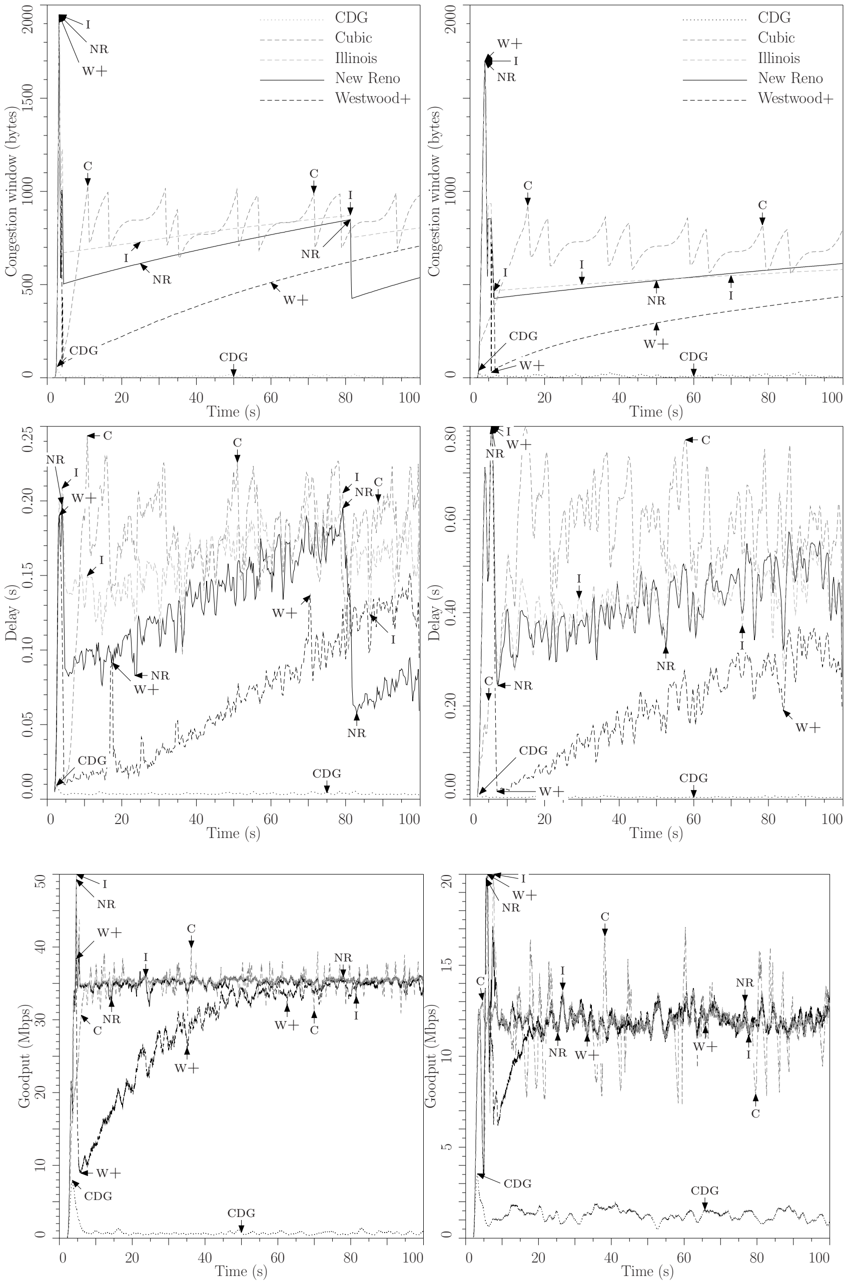
5.2.1 Base behavior and behavior in a loaded network

CCA	Conditions	Behavior
CUBIC	Base behavior	Slightly suffers for the deficient behavior of Hybrid Slow Start.
	Loaded network	No impact of Hybrid Slow Start.
NewReno	Base behavior	Easily achieves maximum capacity.
	Loaded network	Similar behavior but with higher delay and more unstable goodput.
Illinois	Base behavior	Easily achieves maximum capacity. However, it creates a a huge standing queue.
	Loaded network	Very similar to NewReno but with slightly higher delay.
CDG	Base behavior	Keeps the delay controlled but fails while trying to reach full resource utilization.
	Loaded network	The differences with loss-based CCAs are reduced.
Westwood+	Base behavior	Very aggressive back-off that impacts the time needed to ramp-up.
	Loaded network	The impact of the back-off application is minimized.

Table 5.1: Findings wrap-up in base behavior and behavior in a loaded network

Placing the UE in medium signal quality position in the cell, the UE has a maximum throughput around the half of the total maximum (35 Mbps). Different experimental trials are carried out with and without background traffic to study the responsiveness of TCP and deduct if the background traffic has the same impact among the CCAs. In order to make the reading easier, Table 5.1 gathers the most important points of the following explanation.

The three subfigures on the left of Figure 5.5 depict the results regarding the scenario with no background traffic. As shown in Figure 5.5 and gathered in Table 5.1, the differences between the loss-based TCP variants and delay-based ones is remarkable even in such a simplified scenario. Loss-based implementations (CUBIC,



(a) (b)

Figure 5.5: Performance comparison of the selected CCAs: a) Base single flow behavior; b) Single flow behavior over loaded network.

NewReno and Illinois) manage to achieve the maximum capacity and create a long standing queue delay (up to 250 ms), whereas delay-based variants, such as CDG, keep the delay controlled but fail while trying to reach full resource utilization. As collected in Table 5.1, every CCA has its particular characteristic regarding the performance outcome. In the case of Westwood+, it is clear that the applied back-off is very drastic and due to this, it takes longer to ramp-up again. Illinois minimally reduces the CWND, causing huge standing queue delay comparing with more conservative implementations like NewReno. In the case of CUBIC, it suffers for the deficient behavior of Hybrid Slow Start. The mechanism exits to the congestion avoidance phase in an early stage and therefore reduces its growth pace far from the maximum achievable capacity, severely impacting in the time it takes to converge. These results once again demonstrate that from the goodput's perspective there is little room for improvements among the TCP family. However, if the performance is evaluated by combining the achieved goodput and injected delay, the selection of most appropriate CCA is clearer. In this precise situation, among the loss-based CCAs that are capable of achieving maximum capacity, NewReno has shown lower induced delay, resulting in an overall better performance.

The three subfigures on the right of Figure 5.5 show the outcome for the same scenario but with background traffic. The total target load of the background traffic is set to the 50% of the link capacity. The capacity reduction minimizes the performance gap between loss-based and delay-based variants and still, the more capacity a CCA gets, the harder impact it inflicts in terms of queuing delay (Illinois as an example). As gathered in Table 5.1, big differences appear comparing with the base example without background traffic, mostly related to a significant increment in the queuing delay and the reduction of the gap in terms of capacity to reflect the differences amongst the CCAs. RTT-clocked CCAs suffer due to a lengthen of time between implementation decisions. In contrast, CUBIC behaves better because it does not suffer for RTT increase in this scenario and the available capacity is reduced, cushioning the underperformance of Hybrid Slow Start. It is clear that by knowing the approximate available capacity, the intra-protocol selection framework would decide better the candidate. For instance, in a non-loaded network situation, Westwood+ has demonstrated to suffer due to a drastic back-off policy. Whereas the same protocol achieves good results in a loaded network, getting almost the maximum rate and keeping the delay lower than other loss-based implementations. Therefore, the framework would consider that the suitability of Westwood+ is attached to low available bandwidths. The analysis gives insights in relation to the macroscopic behaviour of CCAs and helps study the CCAs in more complex scenarios. In this regard, the analysis has shown that the behaviour of CCAs in loaded network circumstances is the same as the non-loaded network conditions but in smaller scale, hiding many differences between the TCP candidates. Therefore, in order to notice and properly analyze performance gaps between CCAs, we consider non-loaded network conditions in more complex scenarios such as the mobility ones.

5.2.2 Sudden increase and decrease of the available capacity

Once the main features of the CCAs have been detected in loaded scenarios in comparison with the base behavior, it is important to study the responsiveness of CCAs in big and sudden capacity changes. To this end, two type of simulations were carried out: with the background traffic being stopped at 20 seconds of the test and with the background traffic being started at 20 seconds of the test. In order to make easier the reading, Table 5.2 gathers the most important points of the following explanation.

CCA	Conditions	Behavior
CUBIC	Inc. available cap.	Good performance without the impact of Hybrid Slow Start due to the low available capacity at the beginning of the transmission.
	Dec. available cap.	Impact of Hybrid Slow Start in the beginning. Aggressive behavior in congestion avoidance phase that leads to an instant huge increment of queue size while reducing the available capacity.
NewReno	Inc. available cap.	Good responsiveness and average delay impact.
	Dec. available cap.	Average loss-based solution that suffers and instant standing queue increase while reducing the available capacity.
Illinois	Inc. available cap.	Good responsiveness and greater induced delay than NewReno.
	Dec. available cap.	Its aggressiveness is harmful in this scenario and takes some time to stabilize the goodput.
CDG	Inc. available cap.	Fails to increase its pace and thus, the new available capacity is wasted.
	Dec. available cap.	Bad performance in terms of goodput but full control of the delay that is always close to the baseline delay.
Westwood+	Inc. available cap.	Its AIMD mechanism is very conservative and the enqueued packets tend to be very few, being not capable of responding to a sudden greater capacity assignment.
	Dec. available cap.	The combination of its dynamics (with a slow ramp-up ability) and the available capacity reduction happen to get the best performance due to the achievement of the maximum goodput and the lowest impact in terms of delay.

Table 5.2: Findings wrap-up in sudden increase and decrease of the available capacity

On the one hand, the left part of Figure 5.6 shows the results regarding the scenario with a sudden capacity increase. In general, as soon as the capacity increases, the queue size is lowered due to a release of previously enqueued packets. As explained by Table 5.2, it is clear that loss-based CCAs quickly respond to an additional bandwidth assignment. However, Westwood+ still suffers from the excessive reduction of the CWND after the Slow Start phase. During the congestion avoidance phase, its AIMD mechanism is very conservative and the enqueued packets tend to be almost 0, therefore with a new and greater achievable capacity, the adaptation ability of the CCA is very weak. In the case of delay-based variant, since CDG mainly focus on reducing the delay over the path, it usually fails to increase their pace and thus, the new available capacity is wasted.

On the other hand, the right part of Figure 5.6 depicts the case in which the background traffic is activated at 20 seconds. Due to the sudden reduction of available capacity, the queue size suffers an instant increment because of the relation between the same number of incoming packets to the eNodeB and the drastic reduction of outgoing ones. As shown by Figure 5.6 and gathered by Table 5.2, it is clear that all CCAs but CDG are able to successfully react to the capacity reduction. However, in some cases such as CUBIC, the CCA takes more time to stabilize to the new pace. These simulations reflect that most CCAs, even delay-based implementations, are capable of reducing their throughput with a sudden available capacity decrease

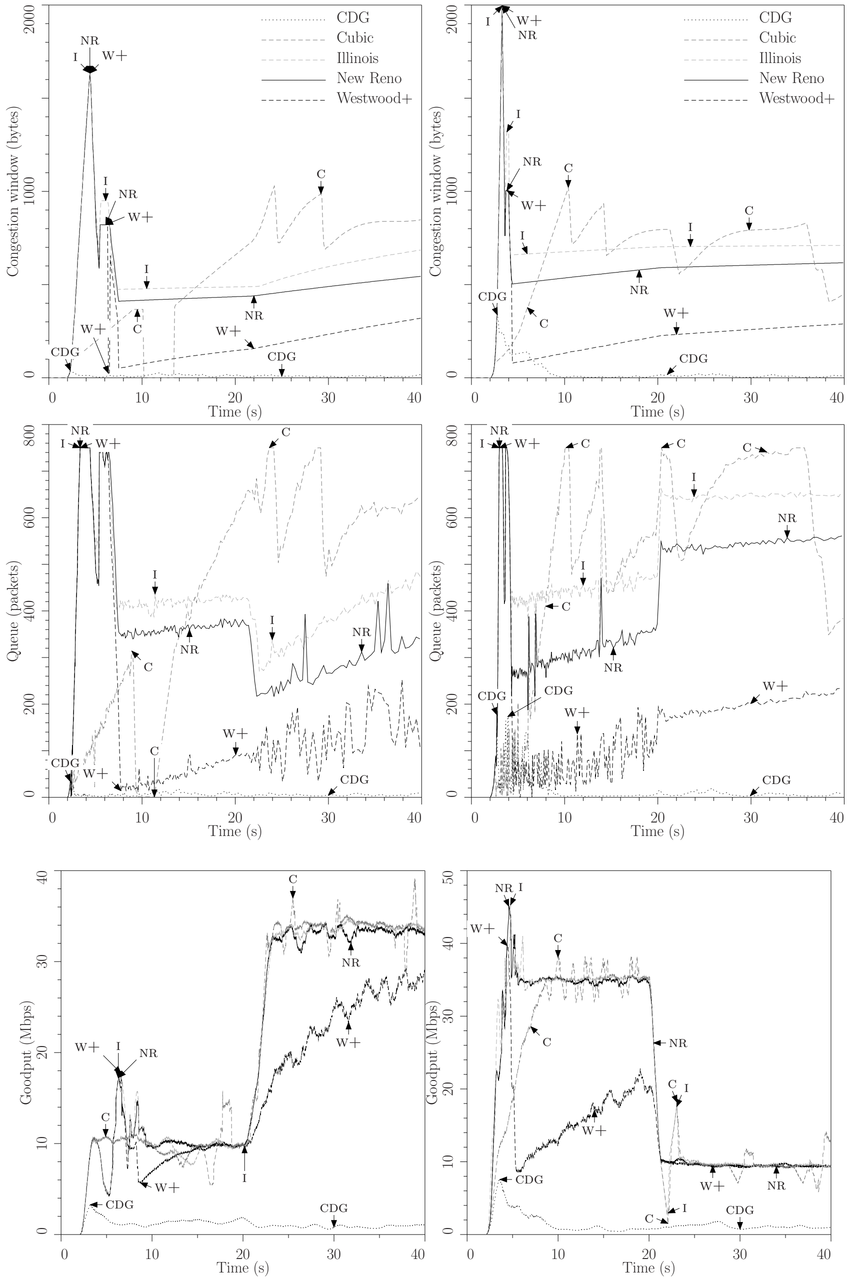


Figure 5.6: Performance comparison of the selected CCAs: a) Sudden capacity increase; b) Sudden capacity decrease.

but delay-based variants struggle to adapt their pace to a bandwidth increase. All these behavioural patterns could fruitfully feed the intra-protocol selection framework. For instance, in case of a forthcoming or ongoing capacity reduction or increment the framework could decide the CCA that best responds to the change not only by means of rate achievement but also considering the side effects (i.e. reaction time, stability after the reaction or induced delay). All these findings would be also very helpful in the understanding of mobility scenarios in next Sections with forward (decreasing quality) and backward (increasing quality) movement patterns in which the available capacity continuously fluctuates (fluctuations formed by increment and decrease changes).

5.3 Start-up performance over mobile networks - PA2

Since most of the Internet flows are short flows, the Start-up phase of TCP is a relevant part of the final performance outcome. Short flows are most of their duration in Start-up phase and thus, the performance of such phase is of utmost importance. This phase is different depending on the utilized mechanism, usually called Slow Start methods. Distinct CCAs have different Slow Start methods with completely different characteristics. It is crucial to determine whether the delay variability induced in mobile networks by resource sharing and the movement of the UE itself could have an impact on the performance of TCP or not and therefore, later on, be able to select the most suitable TCP candidate within the TCP family. Therefore, analyzing their performance, the study will provide insight of the actual impact of selecting one CCA or other in order to appropriately tackle short transmissions. We decided to study the three most important Slow Start methods: 1) The Standard Slow Start utilized by NewReno, Westwood+ and Illinois; 2) The Hybrid Slow Start used by CUBIC and CDG; 3) The Startup utilized by BBR.

The section covers the analysis in 2 stages. Firstly a comparison overview of Standard Slow Start and Hybrid Slow Start over the beforementioned 4 different cellular network deployments (simulated, emulated, controlled and real-world deployments) in order to assess the suitability of each method for its selection in variable MBB networks. The Startup method of BBR was not included in the first phase of the analysis due to the impossibility of testing it in all four testbeds. Secondly, considering all the gathered knowledge with previous analytical stage, the Startup phase of BBR was added to the options to be tested. In order to experiment with more complex and realistic variability conditions due to the significant impact that could cause in the performance of certain Slow Start methods and be able to properly test the different Slow Start mechanisms, we conducted several measurements with a mixture of traffic with long and short flows over different operators with distinct capabilities. The section finally concludes summarizing the findings.

5.3.1 Multi-deployment Start-up comparison - PA2.1

This Subsection aims to provide with a better understanding of the performance of the Start-up phase of TCP in order to help our selection framework select the most appropriate Slow Start method among the options that different CCAs include in their logic. To that end, we design a set of experiments in order to measure and compare the two most prominent and distinctive Slow Start methods: Standard Slow Start and Hybrid Slow Start. It is important to clarify whether the internal mechanisms of Standard Slow Start and Hybrid Slow Start could provoke an underperformance or not and therefore, act accordingly when selecting a TCP candidate in the intra-protocol selection framework. In principle and due to their main characteristics, the analysis wishes to resolve if the performance is deteriorated in the case of Standard Slow Start due to its harmful loss-based mechanism and in the case of Hybrid Slow Start due to a probable early exit from fast ramp-up.

Regarding the exit conditions of Hybrid Slow Start, two things need to be underlined. 1) While analyzing pacing and the possible benefit of it over mobile networks, some researches and companies have found out that the packet-train lengthen mechanism is not performing well, so they have decided to let the delay mechanism as the only exit condition. We can argue that any improvement should be made based on all network conditions and therefore, disabling the packet-train lengthen mechanism could lead to biased results. However, we found out that in mobile networks under distinct variabilities and use-cases, the packet-train mechanism never has an effect in an early exit. In fact, disabling it, the results are exactly the same if we compare with the Hybrid Slow Start working with both mechanisms (see Figure 5.7). Therefore, at least in cellular networks, if the delay mechanism is let as the only exit condition, there would not be any change in the final outcome.

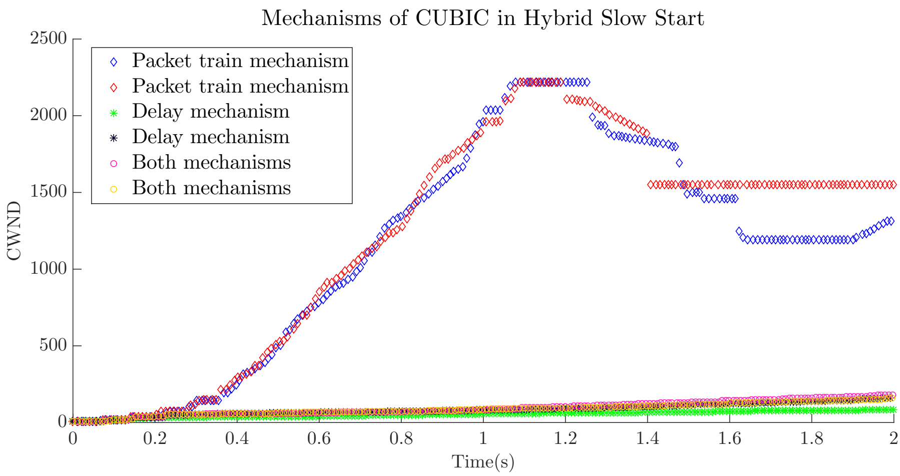
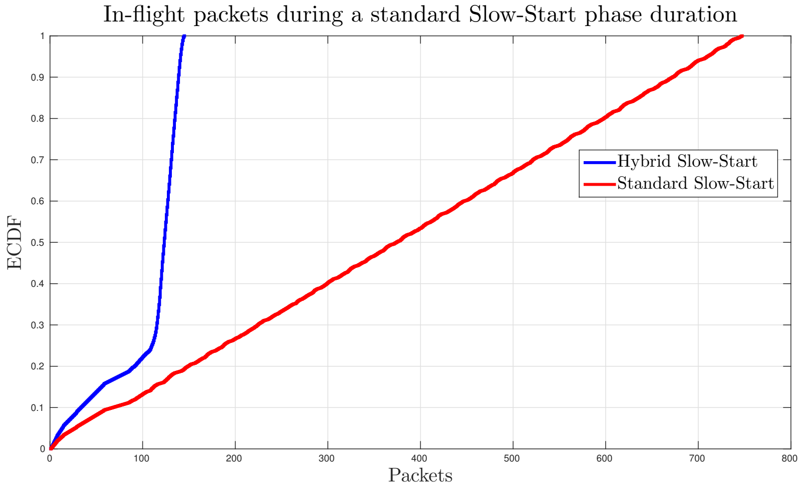


Figure 5.7: Comparison of skip-mechanisms of Hybrid Slow Start

The following 4 examples of both Slow Start methods illustrate their performance in the short-term with special impact on short-lived flows. The terms Hybrid Slow Start and CUBIC are indistinctinly used due to the importance of the latter and the utilization of the former in the latter. The same applies for Standard Slow Start and NewReno. Figure 5.8 shows, in a simulated multi-UE scenario, the ability to inject packets that both mechanisms have during the time Standard Slow Start takes to converge.

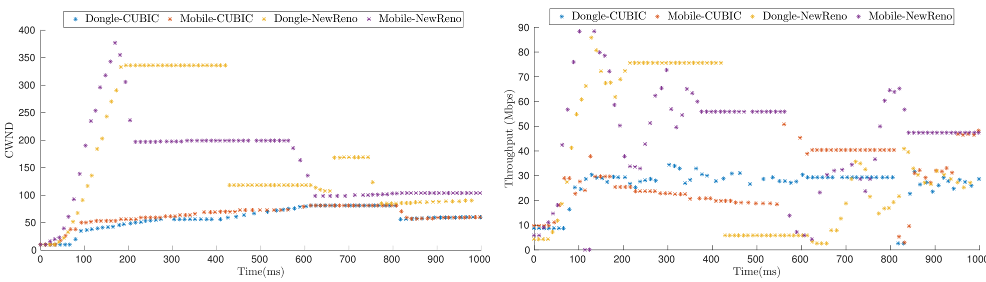
Figure 5.8 shows the ability to put packets in-flight during the time Standard Slow



(a)

Figure 5.8: Slow Start mechanisms in simulated static multi-UE scenario: Injected packets during a Standard Slow Start period.

Start takes to converge as an ECDF. Standard Slow Start has a gradual and continuous line shape, whereas Hybrid Slow Start has two stages formed by the period of time in which the method has ramped-up as Standard Slow Start and the period after detecting a delay variation and exiting Slow Start. The change between the periods (elbow in the graph) in this particular example occurs around 120 packets of outstanding data and it establishes the exit of the CWND from Slow Start in that precise value. It is clear that in the selected simulated scenario Hybrid Slow-Start suffers due to the detection of delay increment and the early trigger of exit condition from fast ramp-up. So, **under some delay variability circumstances Hybrid Slow-Start slows-down the ramp-up of TCP**. In some situations, this effect could lead to the underutilization of available radio resources and lengthens the time needed to converge, directly impacting in the quality experienced by users (QoE). Therefore, the utilization of Hybrid Slow Start under certain fluctuation conditions would be less interesting for the intra-protocol selection framework in order to tackle short flows.



(a) (b)

Figure 5.9: Slow Start algorithms comparison (emulated testbed): a) CWND evolution; b) Impact on throughput.

Considering abovementioned behavior and willing to involve real UEs in the measurement process and check out whether the outcome is similar to the one obtained with the simulated environment, we performed measurements in the emulated testbed with a smartphone and a dongle acting as UEs. Static network conditions are selected and the variability is provoked by the application of a fading model (EVA60 is selected). The testing point is picked from the ETSI/3GPP measurement suggestions [16]. Figure 5.9 shows that the experiments report a quick skip of the fast ramp-up for CUBIC due to the delay detection being triggered in Hybrid Slow Start. Subfigure 5.9b depicts the impact on throughput and how Hybrid Slow Start performs poorer in the beginning of the transmission. Thus, once again Hybrid Slow Start demonstrates to be not suitable enough to be utilized in short transmissions under precise variability circumstances in MBB.

The next comparison and verification example of the impact of Hybrid Slow Start is carried out in a LTE controlled deployment (Figure 5.10). To the correct dissertation of the results, different tests are launched over a UE. In this case the representation of final throughput is illustrative enough to understand the comparison.

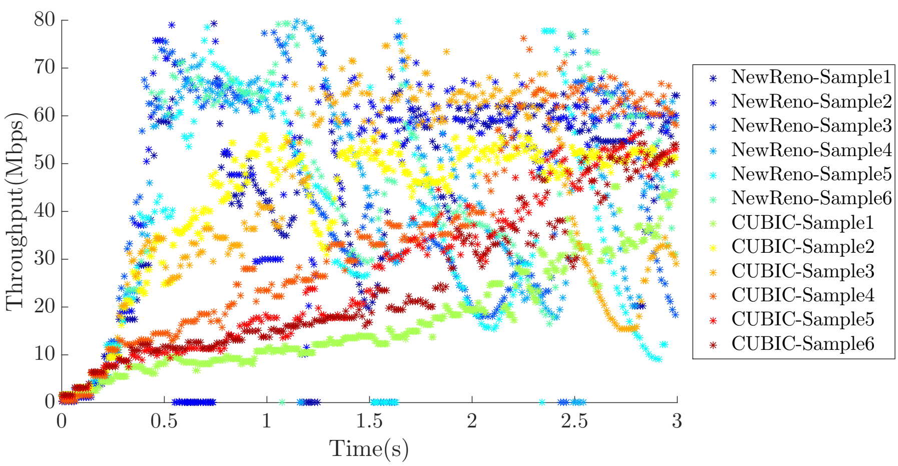


Figure 5.10: Throughput comparison of Slow Start mechanisms in w.ilab-t

The outcome of Figure 5.10 represents a greater impact of Hybrid Slow Start in final achieved throughput regardless the selected mobility pattern. The whole controlled deployment is composed of real equipment and the queuing and processing delay and interaction constraints are the closest possible to the real-world deployments. Even though there are experiments, such as the yellow line, in which Hybrid Slow Start ramp-ups to a point where exiting due to delay could be beneficial for the avoidance of bursty losses, on average Hybrid Slow Start underperforms over mobile networks, taking a long period of time to achieve the maximum capacity and wasting radio resources in the meantime. The last comparison example of Slow Start methods is carried out in live real-world MONROE testbed. Figure 5.11 shows the reported comparison over three different networks/operators.

To conclude, comparing with the results obtained with the emulated testbed or simulation environment, **the impact is even harder in real-world or controlled deploy-**

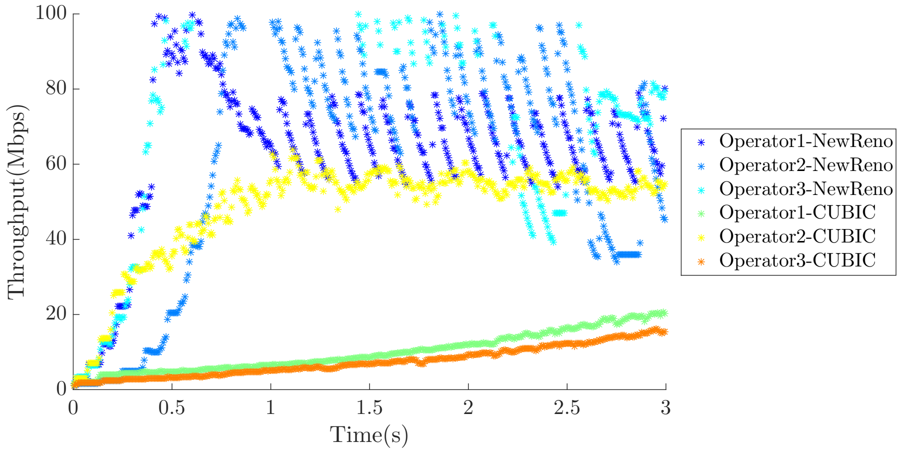


Figure 5.11: Throughput comparison of Slow Start mechanisms in MONROE

ment for the strong relationship between poor performance of Hybrid Slow Start and delay variability. The more realistic the delay over the system, the more challenging conditions of mobile network and therefore, more complex to adapt to. Linked with this finding, Annex A1 (On the use of RTT measurements and its implications with Hybrid Slow Start) describes in detail the relation between the present delays in different LTE testbeds and the incapability of modifying the delay sensitivity of Hybrid Slow Start to face the fluctuations in MBB due to the intrinsic unpredictability and massive variability of delay samples in TCP. Different testbeds confirm (but denote a very distinct impact) that the detected performance of Hybrid Slow Start is not appropriate for its utilization over variable MBB circumstances while the transmission is short. Therefore, the intra-protocol selection framework would avoid the selection of CCAs that use such Slow Start method (i.e. CUBIC or CDG) when the combination of network and traffic conditions match the abovementioned circumstances.

5.3.2 Thorough comparison of Start-up methods in real-world - PA2.2

The previous analysis regarding the analysis of Standard Slow Start and Hybrid Slow Start has demonstrated that the latter could misperform due to an early delay detection created by the variability of the network. Looking into the results, we have addressed that the deployments with real equipment prompt more variable conditions and therefore, lead Hybrid Slow Start to a even poorer performance. Considering these findings, **we tried to analyze more complex and realistic scenarios and know whether the combination of multiple transmission in the bottleneck and the configuration conditions of real operators could result in a different outcome that should be taken into account while selecting the CCA with the intra-protocol selection framework or not** (**PA2.2**). To that end, we designed a set of experiments with long and short flows (explained in PA2.2 in Section 4.4.2.2) over the real-world experimentation testbed called MONROE. The experiments were devoted to carry out traffic of different profiles as well as testing operators with distinct capabilities. Regarding the Slow Start methods we added the newly released Startup mechanism of BBR to the already tested Standard Slow Start present in NewReno, Illinois and

Westwood+ and the Hybrid Slow Start present in CUBIC and CDG.

Out of the two exit conditions of Hybrid Slow Start, the delay detection has been demonstrated to cause a deteriorated performance under certain mobile network circumstances. The main reason to this was the combination of delay spikes due to the accumulation of packets in schedulers (ACK compressing) and a very high sensitivity to delay changes that triggers the early exit from classic Slow Start ramp-up. Provided that the delay detection could be an occasional event and in order to avoid misperformance of Hybrid Slow Start in such cases, this experimentation phase also added a modified version of the Hybrid Slow Start in which the delay detection only triggers the exit conditions if it happens three times. In total, 4 different methods were tested with their selected CCA flavours among the options: Standard Slow Start (NewReno), Hybrid Slow Start (CUBIC), Hybrid Slow Start with triple delay detection (modified CUBIC) and Startup (BBR). Both the Start-up mechanism under test and its selected CCA to represent are equally utilized during the explanation in order to not only refer to the specific Start-up method, but to the CCA as well from the point of view of the intra-protocol selection framework.

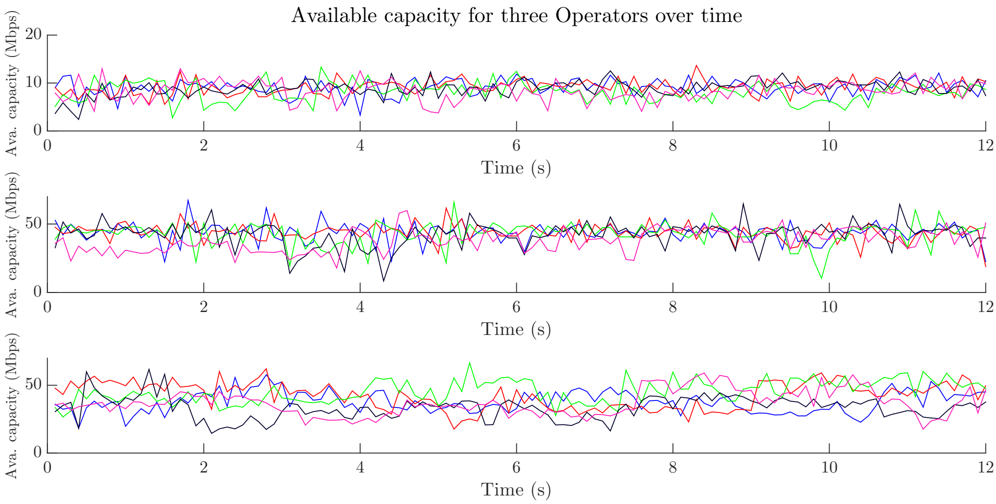


Figure 5.12: Temporal variability of available bandwidth.

When analyzing the results in nodes placed in different countries and with distinct operators, we realized that three behavioural patterns were repeated all over Europe. For that reason, the final measurements were carried out over three selected operators. These three operators represent three different scenarios in terms of capabilities and configurations that in the end could affect the performance of different CCA. Therefore, it is important to analyze them and study the impact factor of the operator configuration so as to add this logic to the intra-protocol selection framework and decide the most appropriate CCA in accordance to the detected conditions. The following explanation is divided operator-wise in order to separate the behaviour or performance of different Slow Start methods in distinct network conditions and be able to individually study them to later on contrast and wrap-up the findings.

The selected three different operators provide distinct conditions to which the CCAs would seek to adapt in order to make the most of the available radio resources. Figure 5.12 shows the conditions of the network for the selected UE in terms of available bandwidth with the representations of several measurements over time.

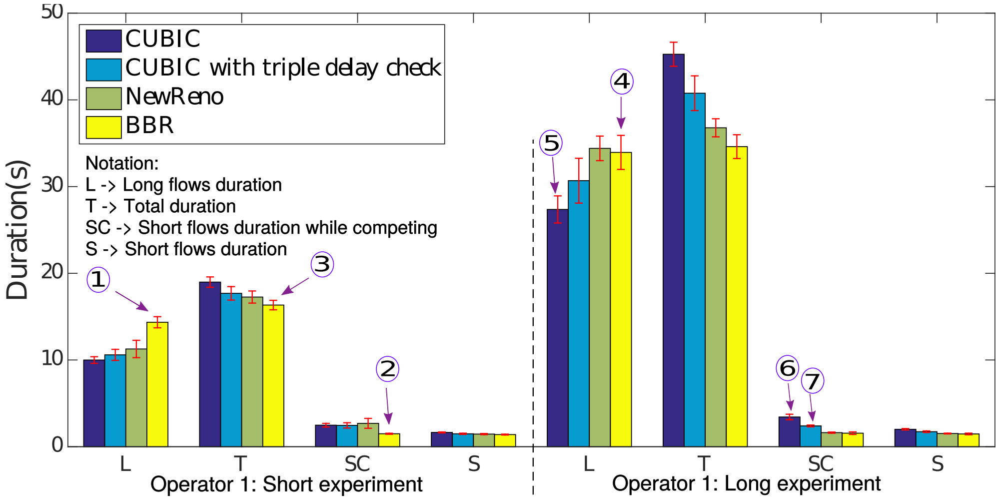
Figure 5.12 shows a low available bandwidth (7 Mbps) for **Operator 1** and higher available bandwidth (40 Mbps) for **Operator 2** and **Operator 3**. These two, according to additional traces, they differ in the length of the bottleneck queue, having the **Operator 3** shorther queue (based on the persistent lower achieved CWND and induced delays). Figure 5.13a gathers in a single plot all the outcome in relation to the completion times for the experiment types, CCAs and kind of flows in the first operator (Operator 1). The results show completion times with median and 95% confidence intervals in a bar plot style. From the left to the right, the bars are divided in two big boxes of results: Operator 1: Short experiment and Operator 1: Long experiment. Each result box is divided in 4 different fields:

1. Completion times for long flows (labelled with "L").
2. Completion times in total to finish the experiment (labelled with "T") as the ability of the selected CCA to download all the files of the experiment.
3. Completion times of short flows that compete with the long flow that runs in the background (labelled with "SC") in order to show results related to the fairness ability of the CCA.
4. Completion times of all short flows (labelled with "S") as the general behaviour of such flows.

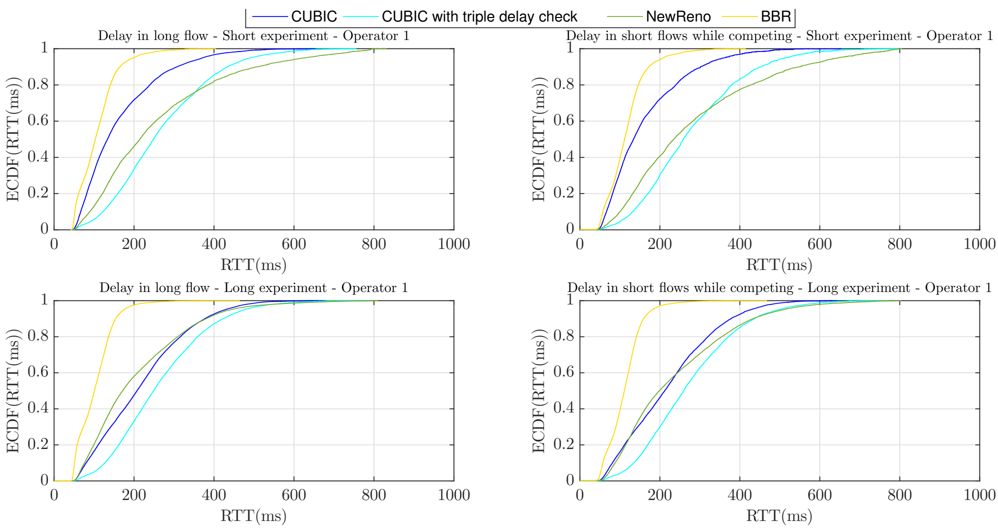
For each field, 4 different bars are depicted: dark blue for CUBIC, light blue for the modified version of CUBIC with triple delay check in Hybrid Slow Start, green for NewReno and yellow for BBR. In order to have a wider perspective of the collected results, Figure 5.13b depicts the measured TCP delays (RTT) in Operator 1 for long and short flows that compete against the former. Figure 5.13b depicts the ECDF of the delay in four subplots: Top subplots for short experiment and the two on the bottom for the long experiment. The left side represents the delay of the long flows, whereas, the right side shows the delay of short flows that compete against the long one.

Operator 1 (see Figure 5.13) provides a low available bandwidth and therefore the long flow takes a long time to download (at least 10 seconds in short experiments and 25 seconds in long ones), interacting this way with many short flows. The differences among the CCAs are the following:

- In the case of BBR, the long flow keeps a low delay (see Figure 5.13b), easily allowing a fair share of the bandwidth for incoming short flows (see point 2 in Figure 5.13a). The delay reduction prolongs the duration of the long flow (see point 1 in Figure 5.13a) but enables a rapid completion of short flows (see point 2 in Figure 5.13a). In fact, the delay reduction provided by BBR makes the whole experiment run faster than with the other CCA candidates (see point 3 in Figure 5.13a). The results in the long flows show that better results in terms of duration do not necessarily mean that the CCA performs more efficiently and



(a)



(b)

Figure 5.13: Performance evaluation in Operator 1: a) Completion times of total experiment, long flows and short flows; b) ECDF of the delay in (short/long) experiments for short and long flows.

could result in biased conclusions. The long flows of BBR take more time to finish but the CCA is more efficient in the whole experiment due to the avoidance of bufferbloat effect. Thus under low available bandwidth BBR tends to have a negative impact in the duration of long flows but a positive effect that allows a faster accomplishment of short ones. The results also show that, the bigger the size of the long download (Operator 1: Long experiment), the less noticeable and significant is the effect of the flow duration being lengthen in comparison with other alternatives (see point 4 in Figure 5.13a). In long experiment the duration of the long flow in BBR is comparable to other CCAs and therefore, the advantage of BBR in the total duration of the experiment becomes greater.

- CUBIC is the fastest candidate downloading the long flow (see point 5 in Figure 5.13a) due to the misperformance of Hybrid Slow Start in the short flows (see point 6 in Figure 5.13a). The poor ramp-up ability of short flows allows the long flow keeping more bandwidth for itself.

- CUBIC with triple delay check suffers the misperformance of Hybrid Slow Start less due to the necessity to detect three delay increases instead of a single one to match the exit conditions in the ramp-up phase (see point 7 in Figure 5.13a). The triple delay check allows avoiding occasional delay bursts and provides with an alternative between the Slow Start of NewReno and the Hybrid Slow Start of CUBIC.

- Even though in terms of total completion times, NewReno shows a great performance close to the one obtained by BBR, the induced delay is greater than with BBR. The extra induced delay could be harmful as cross-traffic, lowering the speed of competing flows.

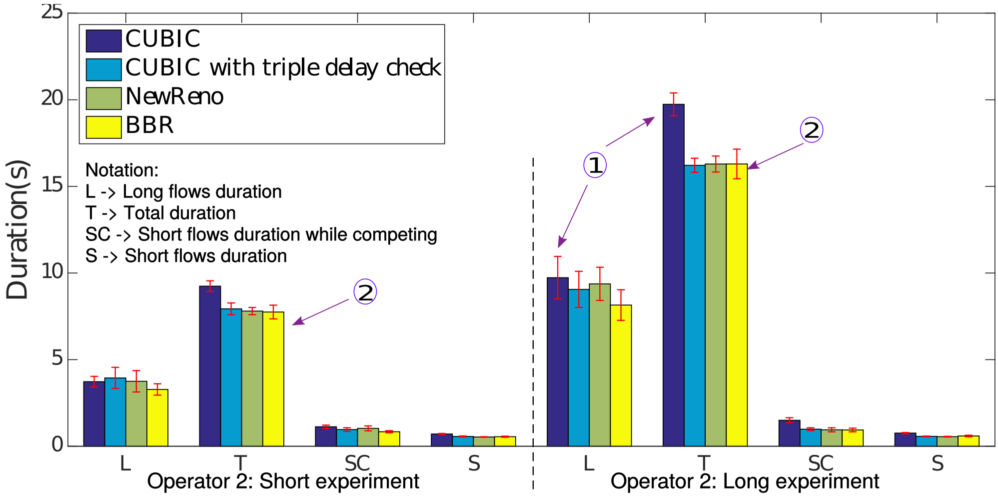
Three important aspects need to be highlighted regarding the delay: 1) BBR induces on average at least half of the delay comparing with other CCAs; 2) In short experiments in which the Slow Start period covers a larger part of the whole experiment run, CUBIC has lower delay values than NewReno due to the misperformance of Hybrid Slow Start; 3) The modified version of CUBIC bridges the gap between the misperformance of classic Hybrid Slow Start and NewReno with Standard Slow Start. Therefore the average delay is greater than using a classic Hybrid Slow Start method.

Operator 2 (see Figure 5.14) offers a high available bandwidth in which the download of the long flow happens quicker than with the Operator 1, interacting with less short flows in the experiments. Even though the CCA results look very similar in relation to the completion times, some differences are present.

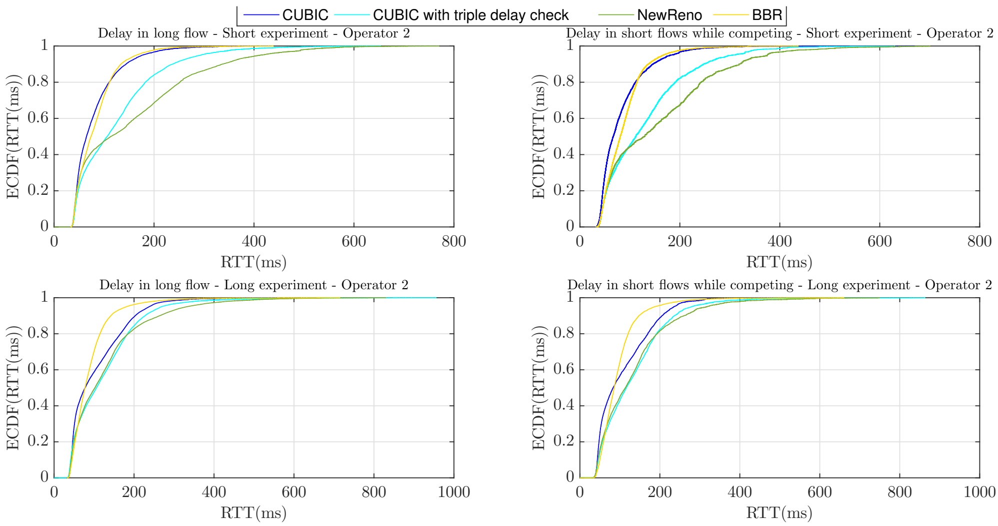
- CUBIC is equally affected by the misperformance of Hybrid Slow Start leading to greater completion times of both short and long experiments (see point 1 in Figure 5.14a). Besides, in the case of long experiments where the long flow interacts with more short flows than in short experiments, the short flows show longer completion times in comparison with other CCAs (as in Operator 1).

- CUBIC with triple delay check, NewReno and BBR report very similar results in terms of completion times, not being able to pick a clear and statistically significant winner (see point 2 in Figure 5.14a). Unlike in the Operator 1, these results show that when there is much available capacity, the ability of BBR to fairly share the bandwidth is comparable with the rest of the studied TCP implementations.

Figure 5.14b shows that although NewReno and the modified version of CUBIC get completion times close to BBR, there are large differences in the TCP delays provided by the different candidates. Out of the aforementioned options, BBR gets the lowest delay no matter the length of the flow. CUBIC offers low delay due to the underperformance of Hybrid Slow Start but it is clearly outperformed by BBR in completion times. Therefore, looking into every aspect of the experiment, BBR performs better



(a)

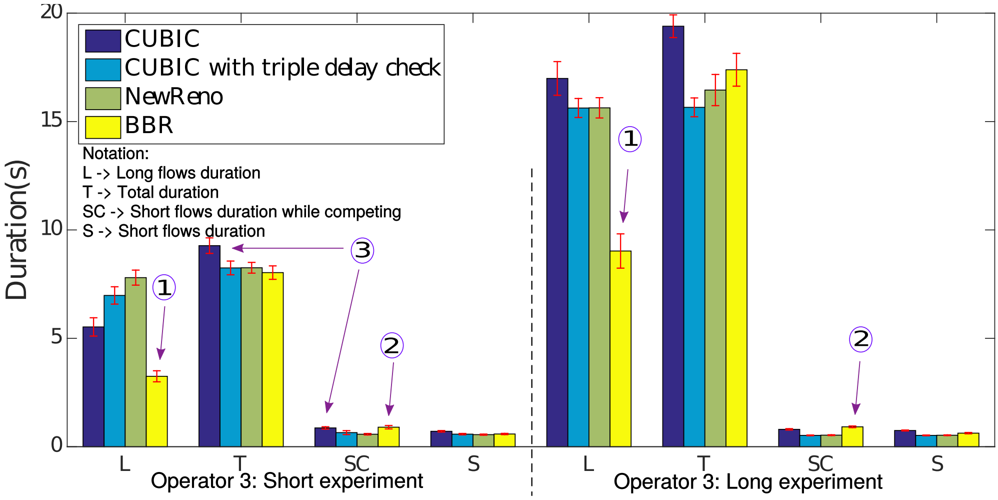


(b)

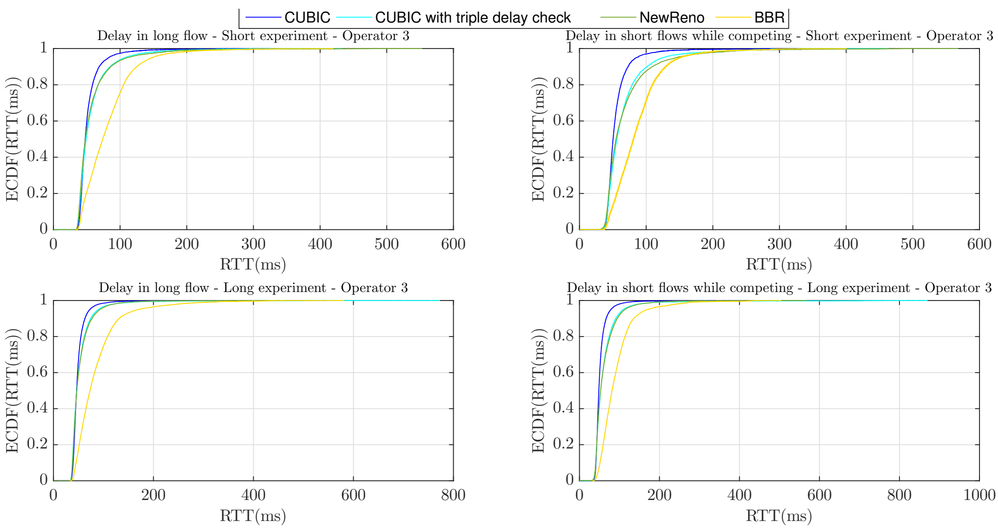
Figure 5.14: Performance evaluation in Operator 2: a) Completion times of total experiment, long flows and short flows; b) ECDF of the delay in (short/long) experiments for short and long flows.

than the rest, achieving similar completion times but being capable of preserving low delays and enabling the avoidance of bufferbloat effect. The results in Operator 2 are similar to the outcome of previous works [36,114,115] but only represent one fraction of the detected scenarios in mobile networks.

According to the gathered traces and metadata, **Operator 3** (see Figure 5.15) is similar to Operator 2 in terms of achievable rates. However, it has a shorter queue in the bottleneck (based on the persistent lower achieved CWND and induced delays) that completely changes the performance of the CCAs.



(a)



(b)

Figure 5.15: Performance evaluation in Operator 3: a) Completion times of total experiment, long flows and short flows; b) ECDF of the delay in (short/long) experiments for short and long flows.

- Both in short and long experiments, BBR demonstrates faster long flows than the rest of the CCAs (see point 1 in Figure 5.15a), downloading the file in half of the time needed by other TCP solutions. However, as a drawback, the high throughput in the long flow forces the incoming short flows of BBR to take longer to finish (see point 2 in Figure 5.15a).

- CUBIC still suffers the misperformance of Hybrid Slow Start (see point 3 in Figure 5.15a). The completion times are still longer than the rest leading to slower speed of flows and more wasted slots of radio resources.

Figure 5.15b shows the slightly longer induced delay by BBR that provokes the longer completion times of incoming short flows that compete against the long one. Regarding the delay of CUBIC, the modified version of CUBIC and NewReno, they maintain a low value due to the short length of the bottleneck queue. This feature makes the performance in terms of maximum achieved throughput worse than BBR but allows a rapid accomplishment of short file downloads while mixing the traffic.

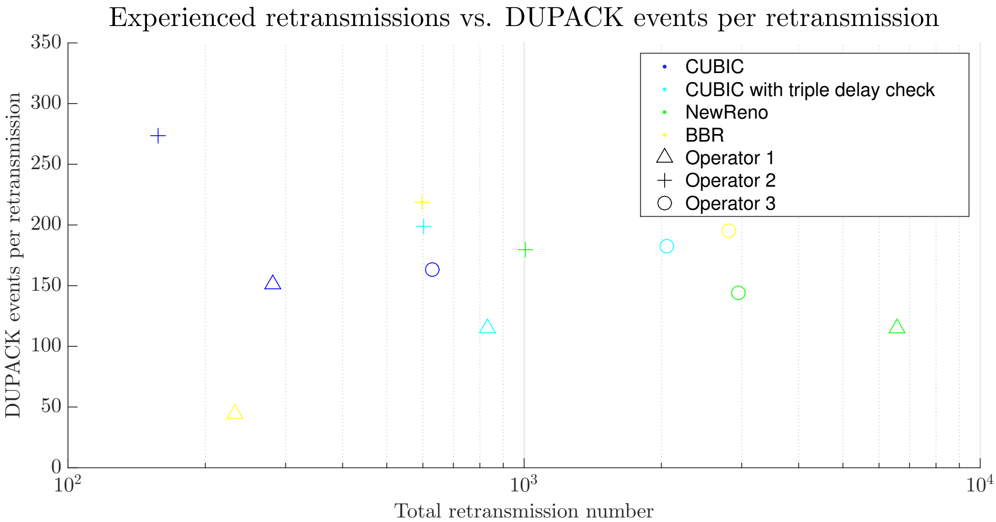


Figure 5.16: Total retransmission number Vs. DUPACK events per retransmission

In order to better understand the dynamics of BBR in comparison with different CCAs under distinct mobile network circumstances, Figure 5.16 depicts the overall packet retransmission events while performing short and long experiments over the operators. Since BBR does not modify its CWND depending on losses, whereas the rest of the candidates reduce the CWND when a 3DUPACK is detected, Figure 5.16 is capable of providing insights in the total amount of retransmission in each situation and the relation of the finally required retransmissions with DUPACK events. The x axis shows the total amount of retransmission events in a logarithmic scale, whereas the y axis graphs the mean DUPACK events for each retransmission. Triangles represent the results for Operator 1, crosses for Operator 2 and circles for Operator 3.

The results for Operator 1 clearly depict a lower amount of retransmission events for BBR, suggesting a controlled injection rate and enqueued number of packets with BBR under low available bandwidth with sufficiently long bottleneck queue to handle the overshooting events in the probing bandwidth state. With bigger available bandwidths (Operator 2 and 3), the retransmission events of BBR are very similar to the rest of the selected TCP alternatives. It is higher in the case of Operator 3 due to the abovementioned bottleneck queue length. However, BBR experiences more DUPACK events than the others for each actual retransmission, showing that BBR handles better than other CCAs the retransmission events, maintaining a pacing rate as close to the achievable rate as possible and not reducing the CWND more than necessary.

5.3.3 Conclusions regarding Slow Start performance over mobile networks

The current section has been devoted to study the performance of Start-up methods over mobile networks in order to study their suitability and define the best selection practises for the intra-protocol selection framework. So far, the findings are the following:

- It has been proven that under certain variability circumstances Hybrid Slow Start detects a sudden delay spike and exits the ramp-up in a too early stage, leading to a poor performance over cellular networks. Besides, the more realistic the delay over the system, the more challenging conditions of mobile network and therefore, more complex to adapt to. Therefore, **under variable network conditions, Hybrid Slow Start is not suitable for short transmissions**.

- Even though, we have found performance differences depending on the operator and flow type, when it comes to **operators with a queue in the bottleneck long enough to allow overshooting events, on average BBR and its Startup phase perform faster and more efficiently** than other CCA candidates and could bring great advances to the mobile developments.

- **NewReno and its Standard Slow Start** has shown an overall **good performance in terms of completion times** but has shown **the greatest induced delays** among the TCP candidates, being very harmful as a cross-traffic.

- **Hybrid Slow Start with triple delay check** offers results that **bridge the performance gap between NewReno and CUBIC in mobile networks** but it is not able to entirely resolve the overshooting events of NewReno with Standard Slow Start.

- Under **low available bandwidth** (6-7 Mbps), the **completion times of long flows are greater than other CCAs for BBR**, but the delay is preserved close to the baseline delay. This effect allows **faster incoming short flows**. Under these circumstances, the **Startup phase of BBR performs better** than the rest of the candidates but it is due to the strong implication of low induced delay in final fairness and therefore, completion time of the flow.

- In **higher available bandwidths** (40 Mbps) with a **queue in the bottleneck long enough, BBR performs as reported in previous works with greater or comparable throughput and lower delay** than other CCAs, improving this way the overall performance of the transport layer.

- In **higher available bandwidths** (40 Mbps), a **short buffer size** (about 80 packets) in the bottleneck leads **BBR to achieve greater throughput with the long flow** but induces slightly longer delays, provoking a **lengthening of completion times in incoming short flows**.

BBR **has demonstrated to handle better than other CCAs the retransmission events**, being more efficient in the CWND reduction and being able to maintain a pacing rate as close to the achievable rate as possible. This efficiency together with the ability to reduce the delay close to the baseline delay, **allows achieving high capacities while**

at the same time enables the rapid evolution of incoming flows. For all those reasons and even though we have detected flow-dependant and operator-dependant performance differences, **the coupled work of CA phases of BBR and the Startup phase are more beneficial than other CCA candidates** to work under MBB circumstances. Therefore, in general it is more beneficial to select BBR with the intra-protocol selection framework. However, **depending on the network and operator conditions and the type of traffic that the service provider wants to prioritize, other CCA candidates and Start-up methods could be selected due to their demonstrated appropriateness**.

5.4 Analysis of the impact of movement pattern - PA3.1,PA4

The current analysis will be focused on the impact that moving to better or worse positions would have on transport protocol performance depending on previous network state (**PA3.1**). In order to identify the effect into different kinds of CCAs, we have selected NewReno and CUBIC due to their differences regarding the Slow Start method, the behaviour related to being synchronized to RTTs and their contrasting congestion avoidance algorithms. It has been demonstrated that Standard Slow Start and Hybrid Slow Start behave very distinct over cellular network, however, this section will omit the existence of such a difference and will mainly focus on the congestion avoidance phase in order to assess the suitability of each CCA for each mobility scenario and help the intra-protocol selection framework better decide.

Apart from the aforementioned study and considering the non-RTT-synchronization of CUBIC and the synchronization of NewReno, different end-to-end latencies are tested in the bottleneck. The objective is to extract the impact that the end-to-end delay of 4G or the irruption of low latency deployments would have in cellular networks in general and during mobility circumstances in particular (**PA4**). In this sense, the section is split into three subsections: the analysis regarding the impact of 4G latencies with distinct movement patterns (in subsection 5.4.1), the implications in the same scenarios of low latencies that move towards the potential delays in 5G (in subsection 5.4.2) and a final comparison and wrap-up (in subsection 5.4.3).

5.4.1 Analysis of the impact of 4G latencies with distinct movement patterns

The analysis will try to point out important features related to the performance of NewReno and CUBIC under backward and forward movement. One of the main targets is to compare the performance difference of movement patterns while the UE holds the same position and be able to deduct the most adequate CCA for a particular mobility scenario. Besides, it is crucial to extract conclusions that could be used as an input to decide if at certain point it is better to continue using the same CCA or instead, it is more appropriate to switch to another CCA that provides better features for such network circumstances. In order to study all abovementioned questions, Figure 5.17 represents in a single graph the results of all testing combinations under 4G latencies of a mobility case at 120 km/h with EVA120 fading model. The figure depicts the comparison of the median goodput for forward movement of CUBIC (in green) and NewReno (in blue) in contrast with the median goodput result for reverse backward evolution of CUBIC (in black) and NewReno (in red). That is, for an emulation of length T, we plot the value obtained at time T-x at position x on the x-axis for the backward movement evolution. This allows us to directly compare the performance of the two CCAs under the same network conditions (same distance to the base station) for different movement patterns in terms of direction. The Subsection only wrap-ups the most important findings and features to be considered in the intra-protocol selection framework but more detailed and step-by-step explanation of results is provided in Annex A2.1 (Detailed study of movement patterns with realistic 4G latencies).

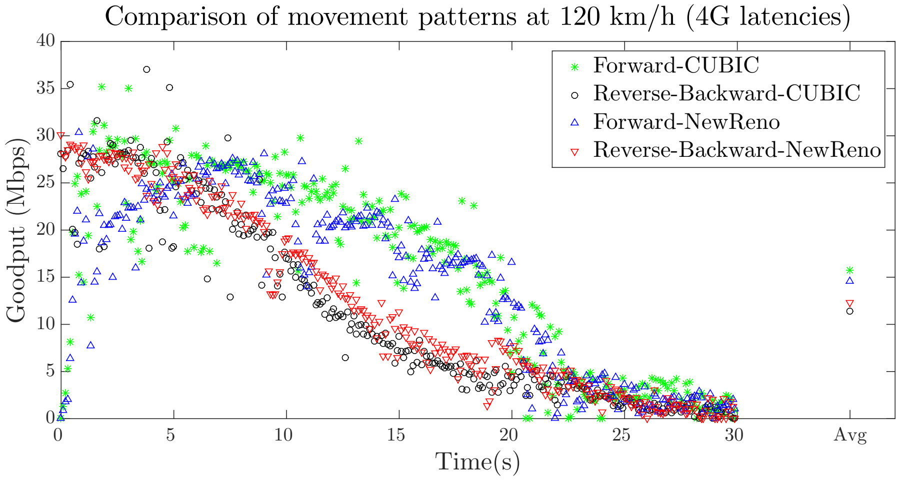


Figure 5.17: Movement patterns comparison (forward movement and reverse backward movement) with NewReno and CUBIC at 120 km/h (4G latencies).

Figure 5.17 shows that the time spent in Slow Start method and first phase of congestion avoidance phase is crucial in order to establish the goodput as close as possible to the maximum achievable rate. In forward movement for instance NewReno is not fast enough to achieve that goal. In the case of CUBIC, it gets a good performance but demonstrates more unstable and variable average outcome. However, only by considering the goodput as the metric to select a precise CCA, CUBIC performs better than NewReno. For NewReno, it is clear that the abrupt back-off policy after the Slow Start makes the CCA require more time to ramp-up and achieve the available capacity, giving to CUBIC a significant performance difference in the beginning of the transmission.

Even though CUBIC is not ACK-clocked or RTT-synchronized, it suffers almost in the same way the impact of end-to-end RTT, achieving similar performance. The higher the latency, the greater impact of packets that are already in-flight, being the forward movement the one overtaking the performance of backward movement due to the impossibility of backward experiments to accelerate and ramp-up from poorer radio conditions. In the case of backward pattern, the CCAs have underperformed ending in a very similar outcome. NewReno wins over CUBIC in backward scenarios under 4G latencies (see Annex A2.1 for more detailed explanation of CWND evolution).

In relation to the results, the intra-protocol selection framework should consider three important aspects:

1. Goodput-based CCA selection: According to the results, for realistic **4G latencies** and focusing merely on goodput, **CUBIC should be selected over NewReno in forward movement pattern and NewReno over CUBIC in backward movement**. However, even though an appropriate selection could bring performance

benefits, it is important to underline that the improvement possibilities that the intra-protocol selection framework has regarding the final performance is more significant in backward movement due to the greater gap between the available capacity and the achieved rate.

2. The obtained results demonstrate that **under realistic 4G latencies** both CCAs are unable to obtain the maximum available capacity in **backward movement**, reaching in some parts of the transmission about the 50% of such a maximum rate. The **intra-protocol selection framework** should consider these findings in order to **select a CCA candidate that performs more efficiently** in such network circumstances.

3. Limited effect of CCA in poor radio conditions: Under **poor radio conditions** due to the reduction in the available capacity, the goodput study shows **similar results regardless the selected CCA** and therefore 2 conclusions arise: 1) **Goodput as the sole driving metric rejects the usability of the intra-protocol selection framework** in some network conditions; 2) **More complex metrics are needed in order to appropriately select the most adequate CCA** that both fulfills the performance over a certain scenario and the requirements of the application layer.

5.4.2 Analysis of the impact of low latencies with distinct movement patterns

After the study of distinct mobility patterns for CUBIC and NewReno under 4G latencies, this subsection is responsible for covering the same study but under low latency conditions. There is no doubt that low latencies are beneficial for final performance because they shorten the cyclic periods of TCP decisions, but the question regarding the impact that have in different movement patterns still remains. The comparison and study will follow similar structure, if not equal, to the one used for realistic 4G latencies and will extract whether under the selected conditions and merely focused on goodput, the proposed intra-protocol selection framework is able to achieve some performance improvement or not. The Subsection explains the most significant findings in relation to its consideration in the intra-protocol selection framework. More detailed description of results is given in Annex A2.2 (Detailed study of movement patterns with low latency).

Even with the similarity between NewReno and CUBIC, it is worth-mentioning the tiny differences present. Figure 5.18 shows the complete median comparison of CCAs in both movement paths. It is clear by the results and average values on the right that CUBIC performs slightly better than NewReno in both movement patterns. Summing up, regarding the performance of TCP under low latency circumstances, the most important conclusion is that:

- With low latencies and merely focusing on goodput, the CCAs do not have any period of time in which their features comparing with the others could make a big difference. The low latency has the ability to equalize CCA's performance (i.e. resulting in CUBIC performing only slightly better than NewReno).

The obtained results demonstrate that very low latencies are capable of equalizing the goodput performance of different CCAs regardless the congestion avoidance phase.

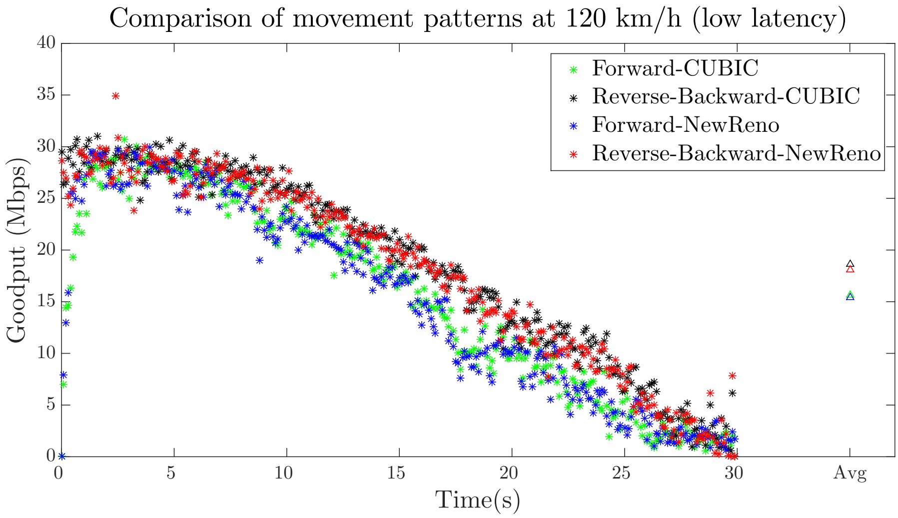


Figure 5.18: Comparison of the movement tendencies of NewReno and CUBIC at 120 km/h.

In upcoming 5G scenarios is which one of the main targets is the reduction of end-to-end latency to 1ms [17], the represented behaviours are applicable for considerably stable fading patterns. Yet, the forthcoming greater capacities and therefore available radio resource fluctuations may well open performance gaps between CCAs even under low latency conditions. Therefore, our intra-protocol selection framework is still applicable for forthcoming MBB deployments. In fact, under low latency network circumstances there are two conditions that increase the usability of the intra-protocol selection framework:

1. **Evolved performance metrics** that not only consider goodput, but more **complex representations of the circular dependencies of TCP**.
2. The **greater the achievable rates**, the **greater differences between CCAs**. Thus, being presumably able to experience performance differences as the ones obtained with 4G latencies but under low latency deployments.

5.4.3 Summary of movement patterns impact

This section summarizes the most important findings, serving as a final image of the outcome regarding different mobility patterns and end-to-end latencies for CUBIC and NewReno. The easiest way to explain and recapture such findings is by comparing the obtained goodput under the commented circumstances (see Figure 5.19). It is clear that there are situations in which the performance has been below the expected available capacity or the inflicted effects have been harmful. A sensible way to give solutions to the current deficiencies would be by the information available in the cellular network and being able to switch the utilized CCA for another one that better matches the network requirements among the options in the intra-protocol selection framework. This mechanism is potentially capable of enhancing the constrained behaviours such as:

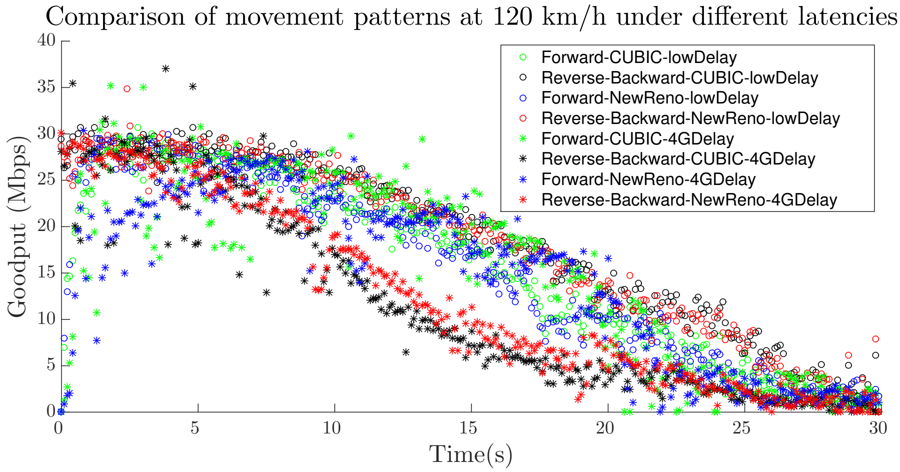


Figure 5.19: Comparison of movement tendencies at 120 km/h under different latencies.

- It has been explained that there is a time period in the experiments (which could be translated to some precise position far from the eNodeB) where the radio conditions are poor enough to serve as a very low ceiling for TCP. In this regard, it is important to notice that the possibility to achieve greater goodputs is constrained and the over-injection of packets is useless and could lead at so point to incur negatively. **In such situations a CCA that alleviates the network load in order to reduce the self-inflicted effects should be chosen by the selection framework**.

- As explained, under 4G latencies, the forward movement and backward movement have different requirements. The **forward movement suffers for the over-feeding of CCAs**. To solve this issue, the server would need to know that the UE is under the detected deficient circumstances so as to reduce the packet injection, being **able to switch to a more conservative CCA**. While, in **backward movement** at some point (belly-shape in Figure 5.19) from the goodput's point of view a **CCA with more ability to scale** would fill the BDP requirements and achieve the maximum throughput. To that end, the server should be capable of deducing the near-future with the movement pattern and speed or by receiving the instruction from a network assistant and proceed to switch to a CCA capable of injecting more packets as a response to a continuous capacity increase (better scalability).

- Regarding **low latency mobility scenarios**, a great **goodput-based performance** could be achieved **regardless the selected CCA**. Therefore, under the mentioned capacities, if in the application requirements only goodput is specified as the leading feature, the intra-protocol selection framework could select different candidates without altering the final outcome. However, if **from the application layer some other performance-oriented mandate is ordered (i.e. delay sensitive applications), the proposed framework will still achieve more adequate performance than randomly selecting a precise CCA (or by default option)**.

5.5 Impact of speed and study of TCP' responsiveness - PA3.2,PA3.3,P3.4,PA4

Once we have looked into the different performance effects created by distinct movement patterns and having detected distinct behavioral findings, we analyze in this Section the speed itself and how TCP responds to it. The section is split into three different parts; an explanation of the implications of a faster UE while the network conditions are similar (in subsection 5.5.1) and a presentation of different variable conditions and the importance in terms of achievable rate for different CCAs (in subsection 5.5.2).

5.5.1 The effect of a faster UE - PA3.2,P3.4

*** Disclaimer: NOTE that BBR is not included in this analysis because it is/was not available at DCE environment during the simulated/emulation tests, but next Chapter will consider it a validation of our CCA selection proposal in emulated mobility use-cases.

For the time being, several effects have been detected for those simplified movement scenarios where an UE is following lineal trajectories. Although forward and backward movement patterns are unlikely to happen in real world, they are capable of providing with knowledge about the performance of different implementations of TCP under certain radio and packet load conditions. In this subsection we will clarify how such effects vary under different speeds. This is, considering the same network conditions throughout the experiment for two different realizations and taking into account the abovementioned movement patterns, how does TCP interact and perform when the same area is covered faster (**PA3.1**). It is important to know whether TCP is affected by the speed itself or not in order to better select the TCP candidate (**PA3.4**) with the intra-protocol selection framework in accordance to the network conditions and mobility use-case. To that end, we designed a set of experiments with both forward movement and backward movement pattern with two different speeds but same network conditions (see demonstration of the same network conditions for different speeds in Annex 3 - Demonstration of similar network conditions to individually analyze the impact of speed). The speeds were the following: 60 km/h or common limitation in rural roads and 200 km/h or high-speed trains.

Throughout this part of the subsection, it will be analyzed the ability or inability of TCP to take full advantage of the available transmission slots that the eNodeB offers (in terms of transmission opportunities or TxOps). The analysis is divided in MCS values because each MCS could be mapped to a different final achievable capacity. In Figure 5.20, the wasted TxOps per second for each speed are depicted, for all selected TCP implementations that are available in ns-3, in forward movement, classified by MCS ranges. Since the representation is for forward movement (decreasing quality movement), the figure should be interpreted from right to left. As mentioned in previous Section, depending on the network conditions or position in the cell, TCP has different dependencies in regards to the maximum available capacity, getting in constrained network conditions no goodput difference between different CCAs. In this simulated mobility environment, two main CCA groups (Group 1 -CUBIC, NewReno and Illinois- and Group 2 -CDG and Westwood+-) and three main phases can be detected in the figure:

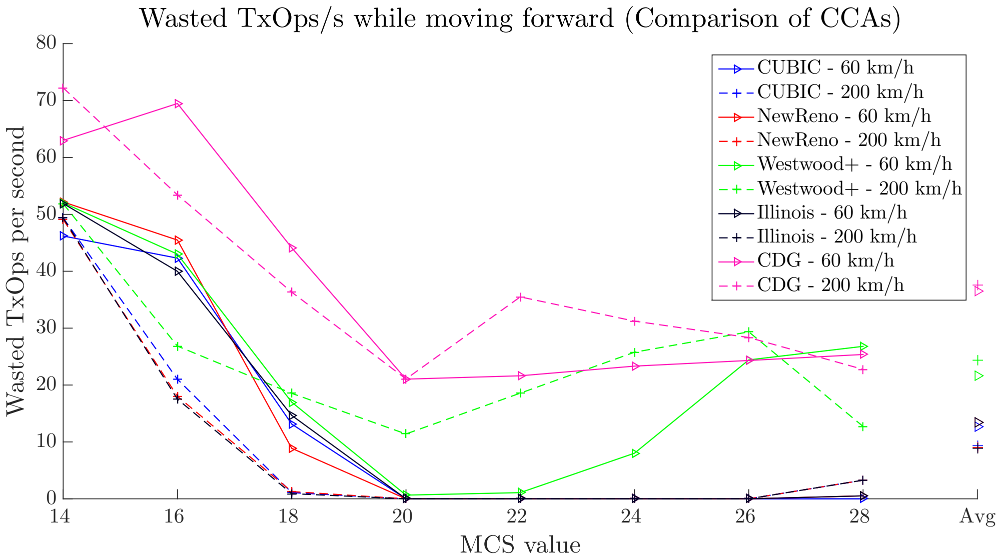


Figure 5.20: Impact of speed on forward movement with all CCAs.

- **Slow Start phase**: During the MCS 28 phase the connection establishment takes place. For the CCAs in Group 1, since the distance associated with a MCS is covered a lot faster at 200 km/h, the CWND has no time to grow quickly enough, and therefore the value of wasted TxOps/s is higher for this speed. In short, the total number of wasted TxOps may well be the same for both speeds, but it results in a more significant effect for 200 km/h for the shorter time spent in the MCS range. In the case of Group 2, both CCAs have an inefficient start-up aggravating their performance in MCS 28 at 200 km/h due to the higher delay variability and their sensitivity to it.

- ***Bufferbloat* area**: Throughout MCS 26 to 20, helped by the eNodeB's big queue length (*bufferbloat*) and the availability to serve previously enqueued packets when the CWND's slope decreases during congested intervals, there are is not a single wasted transmission opportunity (Group 1 of CCAs). The effect of *bufferbloat* itself is not a desirable feature because even though it denotes full utilization, packet overshoots could provoke high delays. In case of Group 2 of CCAs, they are not capable of achieving the full bandwidth utilization due to their mentioned deficiencies, demonstrating once again that as a general rule Westwood+ and CDG are not adequate for MBB and therefore, the intra-protocol selection framework will avoid their utilization and will only consider them under very specific conditions.

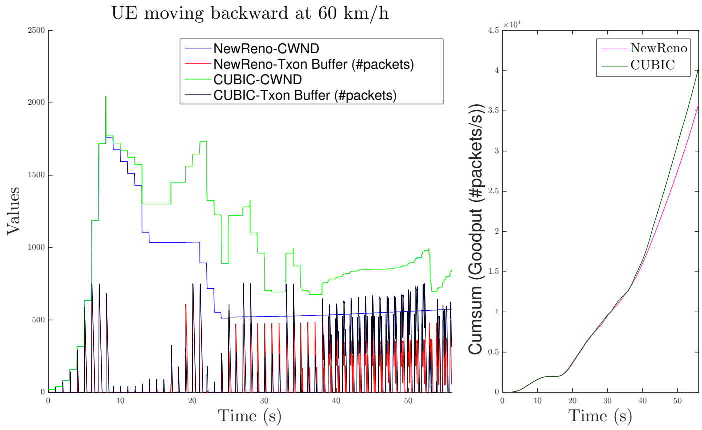
- **Queue draining zone**: From MCS 18 to 14, taking into account the available capacity and CWND evolution, the system tends to drain the queue. When it comes to faster UE scenarios, the slope for the available capacity is more tilted and closer to CWND's one, therefore, the draining effect is slower. **This zone is comparable to the one found in the study of movement patterns and gives another important hint in the resolution of the appropriate server decision with moving UEs under different speeds**.

Considering the average values reported over three phases, in slower scenarios more TxOps are wasted compared with faster ones (see average values on the right). This means that **depending on the network zone or the detected phase and the utilized speed, the created requirements for TCP are different**: during the Slow Start phase at 60 km/h, the Group 1 of CCA are aggressive enough to achieve a good utilization of radio resources, whereas at 200 km/h a more aggressive CCA is required in order to reach similar utilization results. In contrast, in the queue draining zone more aggressive CCA is needed at 60 km/h as compared to the outcome at 200 km/h. Therefore, the speed and zone marks the CCA requirements and the need for their intrinsic features.

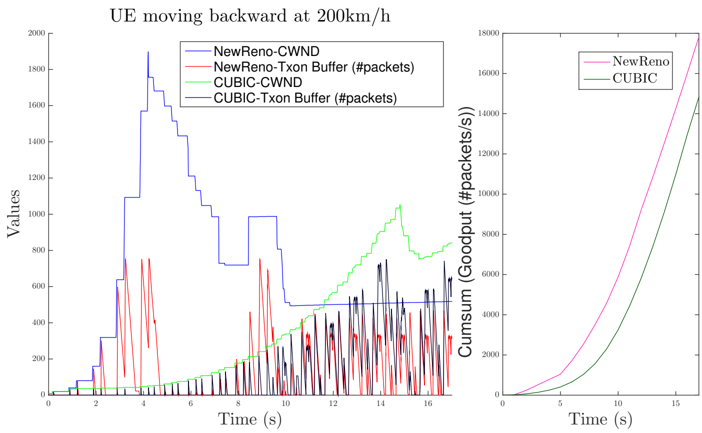
It is clear that delay-based variants are unable to cope with delay variability and they tend to underutilize the available bandwidth. "Off-the-shelf" CDG struggles under LTE environments, due to its high delay sensitivity resulting in rapidly exiting Hybrid Slow Start and the impossibility to grow in terms of goodput while keeping RTT under control. In regards to Westwood+, even though it is more capable of growing, it needs a long time to achieve full utilization, which happens to be very difficult at 60 km/h and completely impossible at 200 km/h. The main difference in Westwood+ stands in its congestion back-off after Slow Start, when a drastic AIMD policy is applied due to a poor available bandwidth estimation. Therefore, the intra-protocol selection framework will avoid the utilization of CDG and Westwood+ as a general rule.

As for backward movement, considering the difficulty to understand the relation between CWND evolution, eNodeB's queue, achieved goodput and the effect of speed, Figure 5.21 depicts the necessary information to understand such effects before the explanation of selected CCAs. CUBIC has been added to help explaining the substantial differences of start-up performance under challenging mobility conditions and its impact on final performance. For clarity purposes, the graphs have been split into four blocks: the results in relation to CWND evolution and transmission buffer occupancy (TxOn buffer) are on the left and goodput's cumulative sum on the right, dividing the results by speeds, at 60 km/h in Figure 5.21a and at 200 km/h in Figure 5.21b.

Regarding the results at 60 km/h, Figure 5.21a shows two details to be underlined: 1) Both NewReno and CUBIC have the same Slow Start phase, which means that the delay variability at 60 km/h in single UE simulated scenarios is not enough for Hybrid Slow Start to skip normal ramp up. Therefore, under such network and variability conditions CUBIC would be eligible at the beginning of the transmission. Even though in general CUBIC shows misperformance with its Hybrid Slow Start mechanism in MBB, there are precise conditions in which it would be usable. 2) The importance of aggressiveness and proper AIMD mechanism when applying back-off in this deployment and how CUBIC is able to get better goodput performance and radio resource utilization due to its ability to scale. The results are consistent with the scaling ability of CUBIC under challenging conditions that has seen in Subsection



(a)



(b)

Figure 5.21: NewReno vs. CUBIC in backward simulated scenarios: a) At 60 km/h ; b) At 200 km/h.

5.5.2.

In relation to the results at 200 km/h, Figure 5.21b depicts a major feature to be mentioned: The impact of Hybrid Slow Start, skipping the drastic ramp up very soon and performing poorly. The delay variability induced by propagation and fading at simulated 200 km/h backward scenario are by far sufficient to be detected by CUBIC as abrupt delay increase. The performance differences could be seen in cumulative goodput and how both lines separate their paths from the very beginning. The explained poor performance in high-speed backward environments together with the previous explanations of this effect are sufficient to highlight once again the Hybrid Slow Start deficiency under certain delay-sensitive mobile network circumstances. Therefore, as a general rule, the intra-protocol selection framework will avoid the utilization of CUBIC at the beginning of the transmissions due to the misperformance of Hybrid Slow Start and will only allow its use under very precise conditions of little channel and delay variability (i.e. the mobility scenario at 60 km/h in the simulated

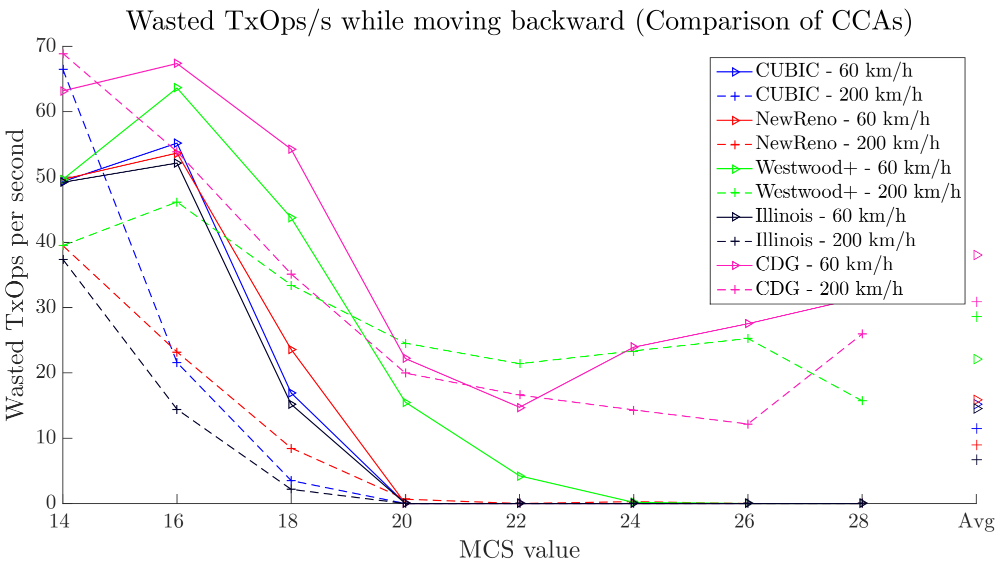


Figure 5.22: Impact of speed on backward movement with all CCAs.

environment).

In contrast to the previous figure, actual wasted TxOps per second are depicted for backward movement in Figure 5.22 for both speeds. At first glance, the figure looks very similar to Figure 5.20, but huge underlying differences are present. Since the representation is for backward movement (increasing quality movement), the figure should be interpreted from left to right. The behaviour of such scenario is divided in two phases.

- **Ramp-up phase**: Throughout the weakest radio positions (from MCS 14 to 18) and therefore constrained by the network, TCP tries to grow. This growth limitation is comparatively very similar for 60 km/h and 200 km/h during this phase and establishes an inevitable boundary for loss recovery. At higher speeds, the Slow Start is comparatively lengthened (since the available capacity grows and first loss is delayed) and the MCS periods are reduced. Due to the combination of both aspects, from Slow Start and beyond (until MCS 20), at 200 km/h on average less TxOps are wasted. Comparing with the results prompt in the analysis of movement patterns, the duration of ramp-up phase is slightly shorter but reflects the same network constriction problem and how when moving faster the impact is statistically less significant. The Group 1 CCAs are very similar and it is only under 200 km/h speed circumstances when CUBIC performs poorly due to Hybrid Slow Start (see number of wasted TxOps in MCS 14 for CUBIC at 200 km/h) and Illinois get advantage of its delay-awareness to make the most of using available TxOps. Therefore, considering the point of view of utilized transmission opportunities, the results do not vary significantly, being able to presumably achieve similar result in goodput as well.
- ***Bufferbloat* area**: Throughout MCS 20 to 28, TCP is able to take full advantage of available capacity (at 200 km/h in MCS 24 zone some wasted TxOps appear, but nothing resounding). However, it has to be mentioned that, due

to that transition speed and applied back-off, NewReno at 200 km/h during MCS 20 period is not able to rise sufficiently the CWND, causing wasted Tx-Ops. Whereas, at 60 km/h all radio resources are utilized due to CWND having enough time to grow. Anyway, the similarities among the CCAs in Group 1 are great and show a similar utilization of transmission opportunities. Therefore, considering TxOp utilization, any candidate within the group could be selected. In order to clearly select a candidate overt the rest, more complex performance metrics may well be applied. The CCAs in the Group 2 demonstrate again its inadequacy to face mobility scenarios and thus, the intra-protocol selection framework will avoid them while focusing on goodput or radio resource utilization.

The speed comparisons show some details regarding the dependencies of CCAs in faster movement under similar network circumstances (see Annex 3 - Demonstration of similar network conditions to individually analyze the impact of speed). As stated in forward movement pattern, **depending on the position of the UE while moving, the network zone could bring important differences in regards to the limitations and dependencies for TCP**. Therefore, the intra-protocol selection framework will require to switch the CCA to better utilized the available radio resources (if we only consider a goodput-oriented approach from the application layer) based on the UE's position, network conditions, speed and movement pattern. All these findings, together with the behavioural patterns of distinct CCAs, will serve as an important input for the next chapter in which the proposal of most appropriate CCA will be done according to the network conditions, application requirements and flexibility of the transport layer.

5.5.2 The effect of variability conditions on the move - PA3.2,PA3.3,P3.4,PA4

Once the effects of different speeds over similar network conditions has been described, the current subsection will cover the performance analysis of TCP when the achievable rates are distinct due to the direct impact of channel variability [121] (**PA3.3**). The target is to add different speeds and fadings to the already studied movement patterns (forward movement or decreasing quality movement and backward movement or increasing quality movement) in order to understand the impact of every factor in the final outcome and be able to extract performance-related and CCA-related conclusions for all the combinations. Thus, being able to identify whether only speed and trajectory are the main heuristics to switch the CCA in mobility scenarios or not. This subsection **checks the necessity of adding the variability of the mobility scenario itself to the driving heuristics that will decide among the CCA family**.

As explained in the methodology, three different mobility scenarios were selected for the analysis: at 60 km/h, at 120 km/h and at 300 km/h with EVA60, EVA120 and HST300 as the fading models respectively. Figure 5.23 depicts the ECDF comparison of reported CQIs when applying the aforementioned fadings into a base SINR of 20 dB so as to illustrate the impact of different fading models into a stationary base SINR. Even though the mobility scenarios forces the base SINR to gradually change, Figure 5.23 gives an idea of the constraints in terms of achievable rate and variability

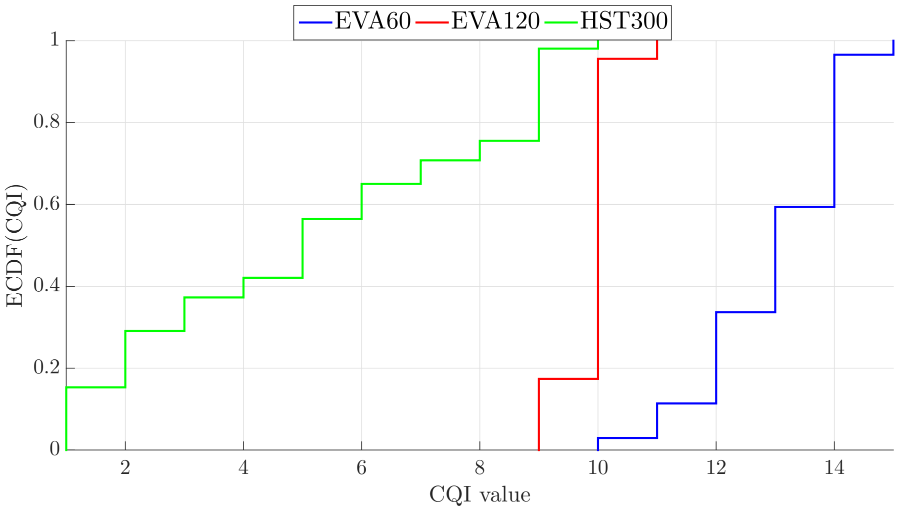


Figure 5.23: CQI reports for different mobility schemes and fadings.

along the experiment for a baseline SINR.

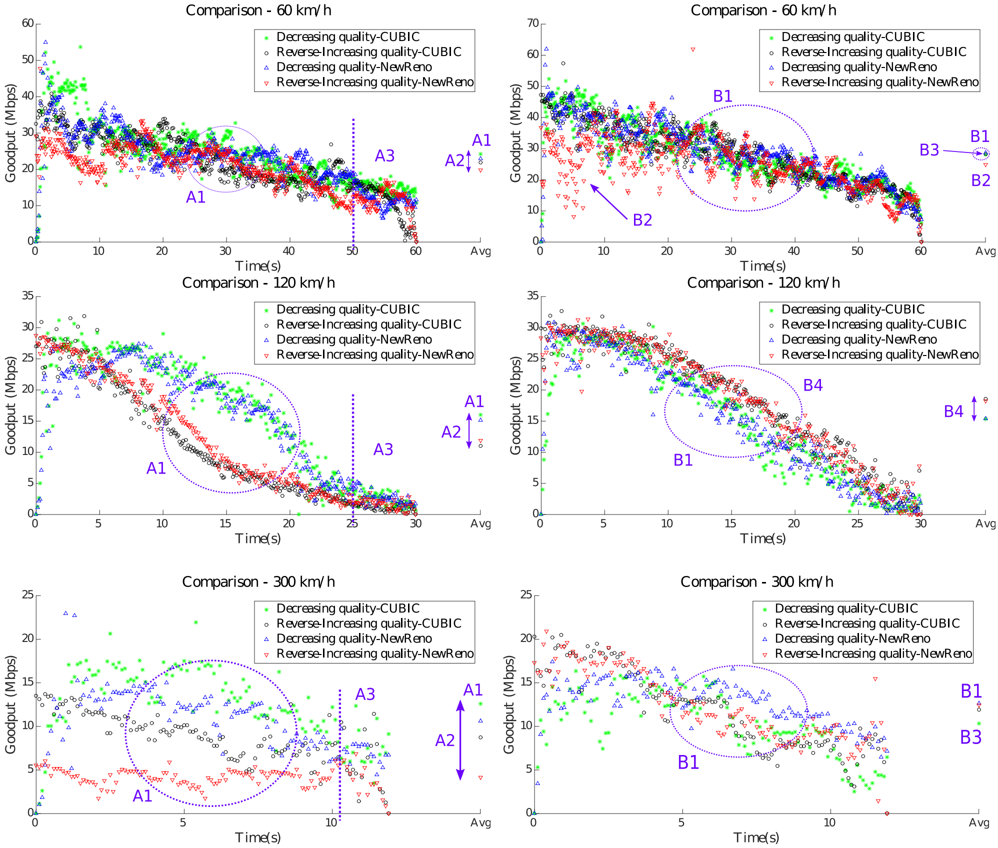
Figure 5.23 shows a very high and variable CQI reporting for EVA60 at 20 dB. CQI changes at highest levels are directly translated to high bandwidth changes. As for EVA120, the reported CQIs are lower but more stable. Finally, HST300 reports a very low and changeable behaviour. The scenarios represent different environments for the execution of TCP and could clarify the performance differences between stable network conditions and variable network conditions and the impact of high or low achievable capacities in the fulfillment of TCP.

The basic study will cover the analysis of NewReno as the basic CCA and CUBIC as the default CCA. This Subsection will comprise the summary of main results for the abovementioned mobility scenarios. More detailed and step-by-step results are provided by the Annex A4 - Detailed study of different variability conditions on the move.

Figure 5.24 shows the complete comparison of results regarding NewReno and CUBIC. The left column represents the performance comparison under 4G latencies organized by different speed scenarios (see Subfigure 5.24a), whereas on the right column, the results for low latencies are depicted (see Subfigure 5.24b). Many conclusions are extracted from Figure 5.24 in relation to variable fadings:

I) With 4G latencies and comparing decreasing quality and increasing quality movement patterns, the former achieves greater capacities than the latter (see A1 labels in average performance values and rounded A1 areas in the three subplots). This scenario shows that **regardless of the variability, in decreasing quality movement, NewReno and CUBIC perform very similarly**, being able to achieve big capacities due to the ability to use previously enqueued packets (under better radio conditions).

II) Under low latency, the depicted results show that very **low latencies are able**



(a) (b)

Figure 5.24: Comparison of the movement patterns with NewReno and CUBIC at different speeds and variability conditions: a) 4G latencies; b) Low latency.

to equalize the goodput performance of different CCAs regardless of the selected CCA (for instance, see the similar results for every experiment in rounded B1 areas). Due to the simplification of our movement patterns, we see a drastic reduction in the outcome differences between them. Even though the target rates are lower than the maximum ones envisioned for future 5G scenarios, the findings demonstrate that the **reduction of the end-to-end latency is effective in terms of final performance and reduced impact of the utilized CCA even under mobility circumstances in medium-to-high speed scenarios**.

III) Under 4G latencies, the combination between speed and fading makes the breach within decreasing quality and increasing quality proportionally bigger (see the outcome gap labelled with A2 and how it increases with faster scenarios). **The difference applies for both NewReno and CUBIC, but the breach is greater in NewReno** (under very variable conditions such as at 60 km/h and at 300 km/h, the red triangle, that represents NewReno in increasing quality movement pattern, gets the poorest performance). Thus, **CUBIC under a variable fading, such as the one in 60 km/h or 300 km/h scenarios, suffers less than NewReno from the impact of end-**

to-end latency, thus, achieving better performance. In increasing quality movement the third scenario with low CQI and high variability is a difficult scenario to successfully handle with a conservative AIMD CCA, leading to have better performance with CUBIC in comparison with NewReno. The changes between consecutive available bandwidth samples are variable enough to cause misperformance with NewReno.

IV) In **low latency** scenarios, there only exists a noticeable performance difference between CCAs in a scenario that **combines variability at high CQI values, big bandwidth changes and increasing quality movement** (e.g. experiments at 60 km/h). **A stable incremental behaviour is obtained with CUBIC, but NewReno is not able to grow between big bandwidth changes** with an incremental tendency that constantly requires the injection of more packets on average (see B2 point).

V) Under **poor radio conditions** with 4G latencies (see A3 areas), it does not matter whether the CCA has been able to put packets in-flight beforehand (case of decreasing quality movement) or not. The achievable capacity target is low and easily reached, **being not necessary to employ any kind of aggressiveness with utilized CCA**.

VI) TCP in general and **CUBIC in particular suffer for their aggressiveness** in those scenarios where such a feature is not required. Due to this overfeeding, we can find **worse results of CUBIC in decreasing quality movement** in comparison with NewReno (slight differences labelled with B3 in average values). **With low latency, NewReno demonstrates to perform more efficiently from the goodput's point of view in forward movement, while with 4G latencies the situation was the contrary**. Once again, we highlight the importance of selecting the most appropriate CCA for each network circumstances and scenario.

Context	60 km/h with EVA60	
	Low latency	Realistic 4G latencies
Forward	*	CUBIC
Backward	CUBIC	CUBIC
Context	120 km/h with EVA120	
	Low latency	Realistic 4G latencies
Forward	*	CUBIC
Backward	*	NewReno
Context	300 km/h with HST300	
	Low latency	Realistic 4G latencies
Forward	NewReno	CUBIC
Backward	*	CUBIC

Table 5.3: Selected CCA according to the goodput performance.

Considering all the abovementioned findings Table 5.3 gathers the best practises regarding the selection between CUBIC and NewReno merely based on goodput. The table contains three different boxes and each of them represents one speed scenario. Second and third columns show the mean goodput performance for low latencies and realistic 4G latencies, while the rows are for classifying forward and backward movement patterns. Each cell contains the most suitable CCA taking into account a goodput-oriented application requirement. In case of no clear winner between the two options, an asterisk is shown. In addition to the already explained findings, it is clear by the selection in the Table that the more challenging the variable conditions, **the greater the difference between CUBIC and NewReno, in clear favour of the performance of CUBIC and its suitability in the intra-protocol selection framework to face variable mobility circumstances**. However, it is important to note that NewReno has demonstrated great usability under low latency conditions and 4G latencies only under considerably stable channel status.

5.6 Findings wrap-up

This section will serve as a summary of all detected findings or heuristics considered in the intra-protocol de-ossification framework focused on challenging conditions such as the Start-up performance and mobility. Table 5.4 recapitulates all remarkable findings.

PA	**Findings to consider**
1	WS constraints consideration
2	Intermediate queue size impact
3	General performance constraint due to the selected CCA
4	Impact of variable available capacity into delay-based CCAs vs. loss-based CCAs
5	Impact of delay variability on Hybrid Slow Start
6	Efficient performance of BBR in general (depends on operator's network capabilities and flow type)
7	Impact of movement trajectory, speed and cell-position performance dependency
8	Impact of considerably stable network conditions and cell-position performance dependency
9	Impact of different variabilities while moving and cell-position performance dependency

Table 5.4: Findings wrap-up.

Finding 1, 2, 3 and 4 constitute the results for the defined **PA1: Impact of specific CCA features**:

Finding 1 ("WS constraints consideration" in Table 5.4) notes the inclusion of WS assessment in order to better select the TCP candidate in accordance to the maximum available capacity.
Finding 2 ("Intermediate queue size impact" in Table 5.4) looks for the intermediate queue size. It is of utmost importance in any case; either being short and preventing the CCA from reaching maximum capacity or being long and allowing long overshots that could lead to a standing queue and bufferbloat effect. The knowledge regarding the length of the intermediate buffer provides with important information in the selection of CCAs and their related aggressiveness.
Finding 3 ("General performance constraint due to the selected CCA" in Table 5.4) demonstrates that as a general rule, **the performance limitations are due to the CCA itself** rather than due to lack of appropriate network capabilities, **highlighting this way the importance of our proposal and the impact that may well have the selection of the most suitable CCA for precise network conditions**.
Finding 4 ("Impact of variable available capacity into delay-based CCAs vs. loss-based CCAs" in Table 5.4) looks into the features of CCAs and their implication into the performance, static simulations have demonstrated that most CCAs, even delay-based implementations, are capable of reducing their throughput with a sudden available capacity decrease but delay-based variants struggle to adapt their pace to a bandwidth increase.

Finding 5 and 6 cover the outcome of the defined **PA2: Start-up performance**:

Finding 5 ("Impact of delay variability on Hybrid Slow Start" in Table 5.4) has demonstrated the underperformance of Hybrid Slow Start under the majority of variable conditions in MBB. This finding avoids the possibility of using CUBIC (the by-

default CCA in Linux) for short transmission or selected to cope with the Start-up phase of some longer flow. Besides, it underlines that the RTT assessments could not be used as the unique value to take decisions in the server in a cellular end-to-end scheme due to the detected RTT spikes.
Finding 6 ("Efficient performance of BBR in general (depends on operator's network capabilities and flow type)" in Table 5.4) analyzes in the real-world the performance of different Start-up methods. Even though, we have found performance differences depending on the operator and flow size in which BBR has shown worse performance and adaptability, when it comes to operators with a queue in the bottleneck long enough to allow overshooting events, on average BBR performs faster and more efficiently than other CCA candidates and could bring great advances to the mobile developments. This effect is even more prominent in the Slow Start phase of BBR called Startup phase.

Findings 7, 8 and 9 comprise the results for the combination of **PA3: Adaptability of CCAs** and **PA4: 4G vs. low latency deployments**:

Finding 7 ("Impact of movement trajectory, speed and cell-position performance dependency" in Table 5.4) has demonstrated that depending on the position of the UE while moving and speed, the network zone or cell zone where the UE is located could bring important differences in regards to the limitations and dependencies for TCP, being able to separate the logic of the intra-protocol selection framework based on the UE's area.
Finding 8 ("Impact of considerably stable network conditions and cell-position performance dependency" in Table 5.4) considers considerably stable channel conditions under mobility. 1) Low latencies: Since the performance is very similar regardless the CCA under use, the least aggressive CCA candidate could be selected in order to avoid unnecessary overshots. NewReno has demonstrated to be solid enough to appropriately perform under such circumstances. 2) Under 4G latencies: NewReno is able to achieve greater mean capacities in backward movement, whereas in forward movement CUBIC makes the most in terms of sustained goodput.
Finding 9 ("Impact of different variabilities while moving and cell-position performance dependency" in Table 5.4) considers very variable channel conditions. 1) Low latencies: in both forward and backward movement the final goodput performance has little impact regardless the selected CCA. 2) Under 4G latencies: it is important to underline that under 4G latencies the tougher the speed scenario, the combination between speed and fading makes the breach within forward and backward proportionally bigger for both NewReno and CUBIC. However, the breach is greater in NewReno. Thus, CUBIC under a variable fading, such as the one in 60 km/h or 300 km/h scenarios, it is able to reduce the impact of lengthening the end-to-end latency, performing better than NewReno due to its aggressiveness.

In relation to the mentioned findings and always keeping in mind the reutilization of the features of distinct CCAs to tackle those situations that are appropriate to, we consider that depending on the flexibility of the transport layer, the intra-protocol selection framework will either select the best CCA candidate at the beginning of the transmission (limited flexibility) or the best candidate for each detected network area or cell position (more flexible transport layer). Regarding a more flexible transport layer that allows selecting the most suitable CCA candidate for each selected position in the cell, three different phases are underlined: the Start-up phase, the constrained area and, depending on the movement trajectory, the phase for sustaining the goodput in forward movement or the phase for scaling in backward movement.

In **forward movement**, after the **Start-up phase** little differences are present in the goodput performance among the different CCAs if we avoid the utilization of Hybrid Slow Start. Then, the selected CCA is required to **sustain the goodput**, taking advantage of previously enqueued packets and avoiding massive overshots that could have an impact in the performance. Either under stable circumstances or under low latency conditions, NewReno has shown a great performance. In hardest scenarios with very variable conditions and having a large end-to-end RTT, the phase needs a more aggressive CCA candidate to overcome the variability. When the mobility scenario reaches the **constrained radio conditions**, we have underlined the importance of CCAs with less ability to scale, which would benefit the final performance due to the loss events avoidance. Following the movement pattern, the areas will be in order: 1) **Start-up phase**; 2) **Goodput sustainer**; 3) **Constrained radio conditions**.

In backward movement, after the **Start-up phase** it is of utmost importance to **scale according to the network conditions** and area where the UE is located (either in or out of the **constrained radio conditions**). To that end, it would be important to know whether the variability is variable enough to use the scalability of CUBIC or otherwise NewReno could be used. It has been found that under 4G latencies the ramp-up or scaling phase needs more ability to scale that the one provided by CUBIC or NewReno. Following the movement pattern, the areas will be in order: 1) **Start-up phase**; 2) **Constrained radio conditions**; 3) **Scaling of ramp-up phase**.

6 Validation of the intra-protocol de-ossification framework

Network technologies move towards almost fully softwarized environments together with the capability to make intelligent and rapid decisions. Following this trend, future MBB scenarios would allow not only rapidly adapting the behaviour of each transport protocols' algorithm to the variability of cellular access, but also selecting between different CCAs depending on the network conditions and application requirements. Following the philosophy of cost-effectiveness and re-use of software components and protocols, the current work addresses the strengths and weaknesses of a selection of CCAs to solve the foreseen challenges in CCA's performance.

Previous chapter has shown the aspects to be considered to build such a de-ossification framework out of a comprehensive study of the performance of TCP in both 4G and low latency (towards 5G) scenarios under distinct mobility circumstances, with an special focus on the flow start-up performance. The different questions analyzed will determine the best CCA choice in every network situation including: a) The technical constraints that could prevent a CCA from achieving its maximum potential in certain circumstances ; b) The analysis of the effect of cross-traffic and responsiveness of TCP in static scenarios. c) The performance analysis of Slow Start methods in different testbeds together with a thorough study of them in real-world MBB networks; d) The analysis of the impact of certain mobility patterns under stable fading pattern; e) The effect of speed and responsiveness of TCP in different mobility scenarios with considerably variable fading patterns. Finally, the analysis has wrapped-up the findings and defined the different phases that are present in mobility scenarios according to the position of the UE in the cell and the dependencies that this location creates in the performance of TCP. All gathered findings will serve in this chapter to both understand the performance of certain CCA and justify the selection of a specific CCA based on the network conditions and application requirements.

Thus, this chapter will cover the feasibility analysis and validation of the intra-protocol de-ossification framework with mobility use-cases in the emulated testbed. Note that the term validation is used here to show how designed methodology and the results of the analysis carried out could be used to build an application-aware intra-protocol mechanism with higher performance than any-other-single-one in several scenarios. Furthermore, not only a single metric will be considered (typically goodput or/and delay) but the application requirements will be tried to get introduced in the CCA selection logic, through three different gradual approaches:

1. Criterion 1 - Pure goodput performance in order to assess how fast a CCAs is and which are the maximum achievable rates.

2. Criterion 2 - A performance metric that includes not only goodput but a broader metric tha also considers delay as well as the number of retransmission events (that would result on a waste of radio resources and smaller overall cell capacity). This is, this metric brings together three classic and most predominant parameters that are usually separatelly employed both to evaluate the performance of CCAs or as a driving mechanisms for the CCAs' behaviour. This way, a single value would be able to capture several important aspects of the performance of a certain CCA, according to the expression in (Eq 6.1):

$$\begin{aligned} A &= \frac{K}{Kt} \\ B &= \frac{Dmin}{D} \\ C &= \frac{BDP - 1500 * R}{BDP} \\ \alpha &= 1 \\ \beta &= 1 \\ \gamma &= 1 \\ PM &= \frac{(\alpha * A) + (\beta * B) + (\gamma * C)}{\alpha + \beta + \gamma} \end{aligned} \tag{6.1}$$

   - The first parameter, A, evaluates how much out of the available capacity is reached in a precise sample.
   - B indicates the growth of current delay considering the baseline delay of the transmission.
   - The third one, C, takes into account out of the current BDP, how many bytes are wasted in retransmissions.
   - α, β and γ are parameters to weight the importance of the three principal (A, B and C).

3. Criterion 3 - Taking into account the increasing demmand of transport solutions that consider applications that hardly depend on reduced delays (e.g. online gaming, augmented reality or video-conferencing), we focus on the application requirements that are delay sensitive and evaluate the presented performance metric weighting the importance of delay increment (β) in an order of magnitude in comparison with α and γ. To that end, we have modified the performance metric assigning ten times more value to β comparing with α and γ. The following equation shows the commented modification to the performance metric function.

$$\begin{aligned} \alpha &= 1; \beta = 10; \gamma = 1; \\ PM &= \frac{(\alpha * A) + (\beta * B) + (\gamma * C)}{\alpha + \beta + \gamma} \end{aligned} \tag{6.2}$$

With these three different criteria, this chapter shows the best candidate for each situation not only based on network and application conditions, but also based on the possible flexibility options provided by the transport layer: initially (Option 1) a dynamic selection of CCA during the session establishment and later (Option 2) with a more flexible real-time switching technique, covering therefore. The chapter is divided in three sections:

- Section 6.1 covers the preliminary selection of CCAs in mobility scenarios at different speeds, fading patterns, mobility pattern and end-to-end baseline latency. The section helps macroscopically a better understanding of the performance of the different CCAs in all network circumstances and provides inputs for the following sections in which the final selection of CCA is made.
- Section 6.2 analyzes CCAs' performance based on the CCA selection that a limited flexibility of the transport layer could perform while applying the intra-protocol selection during session establishment (Option 1). The application requirements are evaluated based on pure goodput performance (Criterion 1), the performance metric introduced in Methodology chapter (Criterion 2) and the same performance metric with delay-based measurement approach (Criterion 3).
- Section 6.3 analyzes the CCA performance based on the CCA selection that a more flexible transport layer could provide, applying a real-time intra-protocol selection (Option 2). The application requirements are evaluated based on pure goodput performance (Criterion 1), the performance metric introduced in Methodology chapter (Criterion 2) and the same performance metric with delay-based measurement approach (Criterion 3). These criteria are applied for each phase in the performance requirements building blocks. After deciding the best CCA switches and alternatives, an actual verification of the selection is carried out.

The last two sections have their own conclusion subsection and gather the most important gains and drawbacks with the proposed CCA selections.

6.1 Preliminary selection of the best CCA under mobility

This section covers the preliminary selection of CCAs under mobility in 4G latencies and under low latency scenarios. Until this point, the performance of CCAs has been mainly based on their goodput. Since during the chapter we will use a performance metric that also considers the injected delay and number of retransmissions, the overall performance representation of this section will take into account three parameters: mean goodput, mean delay and average number of retransmissions per time slots of 100ms. In order to avoid complicated graphs that may well be misinterpreted or messy to grasp, the overall performance is depicted in spider plots of three axis, one per performance parameter. Each figure comprises 6 subplots: the first 3 subplots in a row depict the results for forward movement pattern, whereas, the bottom line shows another 3 subplots for backward movement pattern. The three subplots per line are representative of the different speed and fading conditions: from the left to the right, the scenarios at 60 km/h, at 120 km/h and 300 km/h. The section is divided based on the end-to-end baseline latency. Subsection 6.1.1 covers the performance of 4G latencies and Subsection 6.1.2 does the same for low latency scenarios.

6.1.1 Preliminary selection of the best CCA under mobility under 4G latencies

All in all, previous analysis in mobility scenarios under 4G latencies has shown that: 1) In forward movement, regardless the channel variability, CUBIC shows better performance than NewReno and increasing their performance difference with more variable and drastic mobility scenarios (e.g. at 300 km/h). 2) In backward movement, both CCAs demonstrate to have difficulties to cope with the features of the mobility scenario. In considerable stable fading conditions, NewReno is able to beat CUBIC. However, the greater the channel variability, the more distance between the outcome of CUBIC and NewReno, being CUBIC the clear winner.

Considering the behaviour of CUBIC and NewReno as the baseline TCP performance of the by-default CCA and basic CCA in Linux, it is important to determine whether different TCP implementations are able to solve the encountered performance deficiencies. Figure 6.1 shows spider plot results for the abovementioned mobility scenarios for CUBIC, NewReno, Westwood+, Illinois and BBR.

Out of all results in Figure 6.1, the most important ones are the following:

- In forward movement pattern: 1) Under considerable stable circumstances (120 km/h), we find in Illinois and BBR better candidates in terms of goodput, but they also induce more delay and retransmissions. 2) Under more variable fading circumstances, the outcome is similar, with slightly better goodput for Illinois and BBR and impact in delay and number of retransmitted packets. Nevertheless, the self-inflicted effects of delay and retransmissions severely impact the goodput performance in faster and more aggressive fading conditions (300 km/h), dropping the goodput performance of BBR and Illinois and being surpassed by CUBIC.

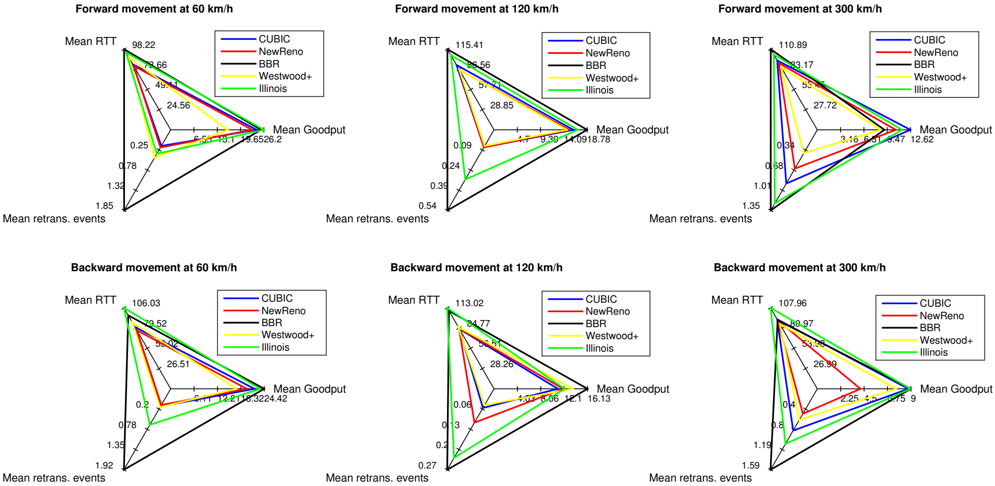


Figure 6.1: Performance spider plot of 5 CCAs while moving at different speeds under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

- In backward movement pattern: 1) Both Illinois and BBR have shows to scale better than CUBIC in all scenarios achieving greater goodput rates. 2) Under considerable stable fading conditions, even achieving similar delay injection and number of retransmitted packets, BBR performs significantly better than Illinois or any other TCP candidate, showing a more efficient evolution. 3) Under variable fading patterns, both Illinois and BBR outperform CUBIC but as a drawback, induce more delay in the network and suffer more retransmissions.

We have detected cases in which a similar goodput is achieved but significantly longer delay and more retransmissions are suffered. These examples, that suppose a difficult performance trade-off to analyze, are the foundation for the evaluation of the protocols based on different points of view in order to appropriately select the best candidate for each network circumstances but also considering the application requirements.

6.1.2 Preliminary selection of the best CCA under mobility under low latency

Previous analysis in mobility scenarios under low latency has shown that: 1) Since the performance is very similar regardless the CCA under use, the least aggressive CCA candidate could be beneficial in order to avoid unnecessary overshots. In this regard, NewReno has demonstrated to have a great behaviour in most of the network conditions and mobility patterns; 2) There only exist a noticeable performance gap for NewReno in a scenario that combines variability at high CQI values, big bandwidth changes and increasing quality movement (e.g. experiments at 60 km/h). A stable incremental behaviour is obtained with CUBIC, but NewReno is not able to grow between big bandwidth changes with an incremental tendency that constantly requires the injection of more packets on average.

Considering the obtained results, it is important to determine whether Westwood+, Illinois and BBR are able to enhance the overall behaviour of mobility scenarios under low latency or on the contrary, achieve a deficient behaviour. Figure 6.2 shows spider plot results for the abovementioned mobility scenarios for CUBIC, NewReno, Westwood+, Illinois and BBR.

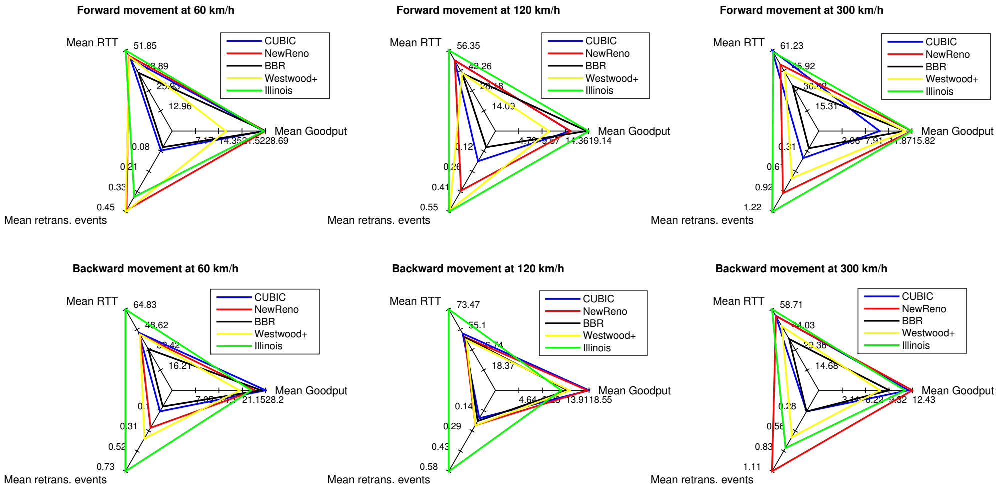


Figure 6.2: Performance spider plot of 5 CCAs while moving at different speeds under low latencies: Forward movement on top; Backward movement on bottom line.

The most important details in Figure 6.2 are the following:

- As a general performance and comparing with 4G latencies, Illinois still suffers due to a excessive delay injection and number of retransmissions, whereas BBR shows a more controlled behaviour with good performance in some scenarios in terms of goodput but also avoiding massive delay and retransmitted packets.

- In forward movement pattern: 1) Under considerably stable fading conditions, both Illinois and BBR perform better than the rest, showing that their CWND evolution is more efficient in network conditions that are not very changeable. 2) Under more aggressive fading patterns, there still exist huge similarities among CCA from goodput's point of view.

- In backward movement pattern: 1) Westwood+ and Illinois are not able to perform better than CUBIC or NewReno in any case. The former primarily due to its deficiency in the aggressive back-off after Slow Start phase and poor ability to scale. The latter, due to its self-inflicted effects, suffering by far the greatest delay. 2) In the case of BBR, as mentioned earlier, the behaviour under low latency looks more conservative. In this regard, in backward movement pattern that requires scalability, BBR performs worse than CUBIC in the mobility scenario at 60 km/h and underperforms in the scenarios at 120 km/h and 300 km/h earning very poor performance results.

Once again and even more noticeable than with 4G latencies, we have detected cases in which a similar goodput is achieved, but significantly more delay and retransmissions are suffered by different CCAs. Those examples require a evaluation of the protocols based on different application-layer requirements so as to appropriately select the best candidate whose performance matches the application requirements and network conditions.

6.2 [Option 1]: Intra-protocol selection during transmission establishment

We have previously introduced that there currently exist solutions [12, 13, 83, 84] that use protocol-independent mechanisms to set parameters on the transport layer based on the application requirements. The application only describes the required service or the constraints for an appropriate execution of the program running in the application layer (e.g. critical delay, maximum throughput, loss sensitiveness) and the API abstracts that info. The information is then utilized to select the most appropriate transport protocols among the options. A more recent advance [85] has added the fact that the API not only considers application requirements but also network conditions. Even though this mechanism requires further signalling and interaction, the method could provide with a more complex and complete API. Based on these advances and taking into account that currently the flexibility to select the most appropriate transport solution is limited to the beginning of the transmission, this section covers the selection and evaluation of the most suitable CCA for each mobility scenario based on macroscopic requirements from the application layer.

As in earlier section and analytical chapter, the explanation is divided depending on the end-to-end baseline latency. Subsection 6.2.1 covers the explanation and CCA selection of most reasonable TCP candidate from the beginning of the communication considering goodput requirements, performance metric requirements and delay-sensitive requirements under 4G latencies. Subsection 6.2.2 describes in the same format the CCA selection for low latency scenarios. Finally Subsection 6.2.3 concludes the section and gathers the most important CCA selections and best practises.

6.2.1 Intra-protocol selection during transmission establishment under 4G latencies

As stated before, the selection of most appropriate CCA would be carried out considering three different metrics. Firstly, the evaluation would be made merely based on the mean goodput. Secondly, the best CCA would be decided using the above-mentioned performance metric. Thirdly, we evaluate the CCA from the point of view of a delay-sensitive application. To that end, the performance metric is weighted to covert the delay parameter in the most significant.

6.2.1.1 Goodput based CCA selection with limited flexibility of the transport layer under 4G latencies

Previous section has depicted the goodput performance as one of the parameters in the spider plot. Now instead, Figure 6.3 covers the representation of the goodput showing the actual average evolution of it in a time axis. The figure not only provides with the most adequate CCA based on the average goodput, but also reveals details about the variability and dependency of the performance towards certain network

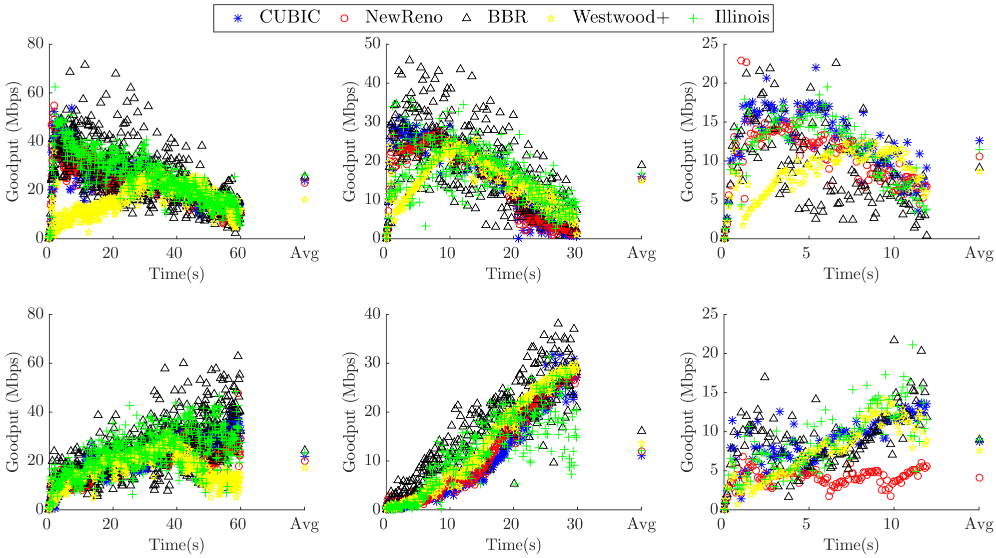


Figure 6.3: Goodput comparison of 5 CCAs under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

areas. The figure comprises 6 subplots: the first 3 subplots in a row depict the results for forward movement pattern, whereas, the bottom line shows another 3 subplots for backward movement pattern. The three subplots per line are representative of the different speed and fading conditions: from the left to the right, the scenarios at 60 km/h, at 120 km/h and 300 km/h.

Figure 6.3 shows that depending on the scenario a proper CCA selection could allow the achievement of greater capacities. All in all, the best practises regarding the goodput based evaluation are the following under 4G latencies:

- In backward movement pattern (bottom line of Figure 6.3) BBR is able to maximize the achieved goodput more than any other TCP candidate and demonstrated this way, the best ability to scale. It has no be noted that, the more stable conditions (at 120 km/h), the greater differences are provided by BBR in comparison with the rest. Under variable fading conditions, BBR shows great variability in the behaviour and obtained performance. Even with that, it is able to scale better than any other candidate. This ability could be of a great value in other MBB scenarios in which scalability is required, either as a necessary feature or as a response to available bandwidth increments.
- In forward movement pattern (first row of Figure 6.3), the three different scenarios require a completely distinct treatment and therefore CCA candidate. The best practises in forward movement pattern under 4G latencies are the following:
  1. Illinois is selected in scenarios with big bandwidth jumps between consecutive RTTs due to its aggressiveness to handle such variability but also due to its delay awareness. The combination of features provides the best and most consistent (little variation in the average behaviour) goodput

performance.

2. CUBIC is picked in variable scenarios with smaller changes between consecutive available bandwidth samples. The aggressive features of CUBIC are suitable to handle great fluctuation in smaller capacities but do not fit well when it comes to bigger bandwidth jumps.

3. BBR is selected in scenarios with considerably stable fading conditions. Even little variability makes the behaviour of BBR very variable and inconsistent. However, it is capable of performing by far better than the rest of the candidates.

The Figure 6.3 reveals that in forward movement, an appropriate CCA selection could provide with a great gain in terms of goodput even using a movement pattern that makes easier the task of TCP due to the avoidance of starvation events in the bottleneck buffer.

6.2.1.2 CCA selection based on performance metric with limited flexibility of the transport layer under 4G latencies

Once selected the best candidates for mobility scenarios under 4G latencies merely based on goodput, it is important to carry out a similar task but evaluating the performance of distinct TCP flavour from a point of view that gathers more information about the outcome and possible side-effect. To that end, Figure 6.4 depicts the ECDF results of the performance metric in all scenarios. The distribution of subplots is the same as before with forward movement scenarios in the first row and backward movement ones in the bottom line. Each column represents a different speed and fading combination, being from the left to the right the scenarios at 60 km/h, at 120 km/h and 300 km/h respectively.

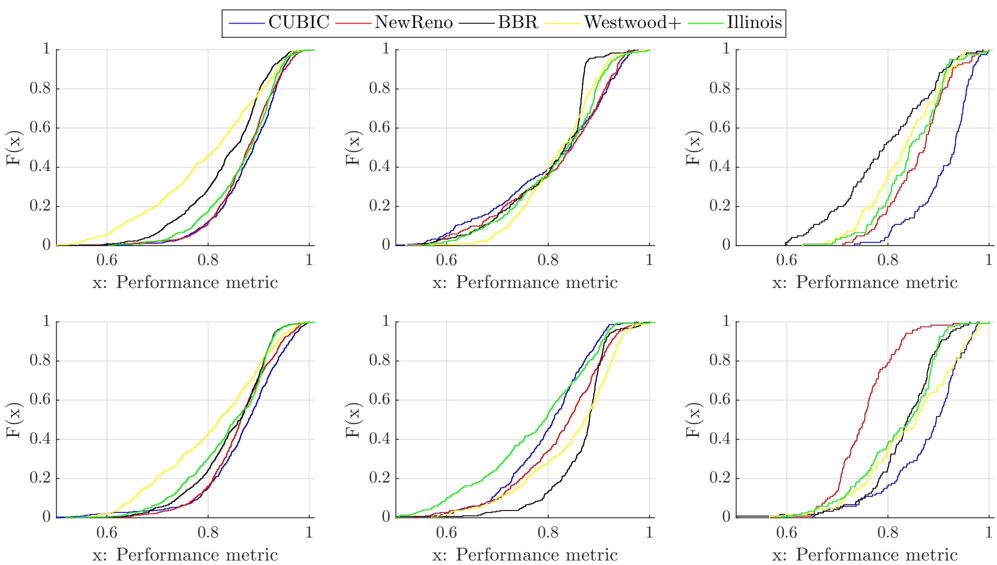


Figure 6.4: Comparison of 5 CCAs performance metric while movement at different speeds under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

Figure 6.4 contrasts with previous goodput-based evaluation in many ways. It shows that sometimes TCP performance is all about proportion in the parameters regarding the performance metric. In general in forward movement, Figure 6.4 depicts that the performance enhancements are not that clear in one CCA over the other. Maybe one CCA is able to perform better in terms of goodput but at the same time suffering from delay and retransmitted packets. Therefore, taking into account the three selected performance factors, it is not that easy to decide whether we should use one congestion control or the other. If the decision of the CCA is taken upon the evaluation of this performance metric, in the scenarios with similar outcome, the final CCA would be picked based on other circumstances. For instance, the decision is taken with: a) Developer preference; b) The by-default CCA prevails over the others. c) If at the point of selection there is one CCA already established and it is among the group with similar outcome, the CCA could remain selected. In contrast and still selecting the most appropriate CCA in forward movement pattern, the 300 km/h scenario is clearly dominated by CUBIC. This statement gives the credit to CUBIC not only achieving great goodput, but with a more general and extensive good performance.

Regarding the results in backward movement scenarios, 2 important aspects need to be highlighted:

1. Under considerably stable fading condition, BBR performs better than the rest, confirming its superior and more suitable behaviour based on both goodput and performance metric.

2. Under variable fading conditions, there is a clear pattern that reflects that CUBIC is more appropriate. The outcome contrasts the evaluation based on goodput that suggests the selection of BBR, proposing to pick CUBIC as a candidate that improves the overall performance based on the three selected performance aspects: goodput, delay and retransmissions impact.

6.2.1.3 Delay based CCA selection metric with limited flexibility of the transport layer under 4G latencies

Once determined the most adequate CCA selection based on goodput and the performance metric, it is important to decide which CCA is more appropriate for delay-sensitive applications. To evaluate and decide the best CCA among the candidates, we can not simply guide our selection based on the average induced delay in the transmission. A certain protocol could be capable of maintaining a very low delay but at the expenses of a very low achieved goodput as well. One of this examples may well be the performance of CDG in the analytical chapter. In order to provide with a more suitable evaluation based on delay, we have decided to use the aforementioned performance metric giving to the delay section more significance (10 times, as previously explained).

Figure 6.5 depicts ECDF plots of the delay-based evaluation for the selected TCP candidates in the mobility scenarios under 4G latencies. Considering the turn-around in the selection process, the results widely differ from the ones obtained based on goodput or the performance metric.

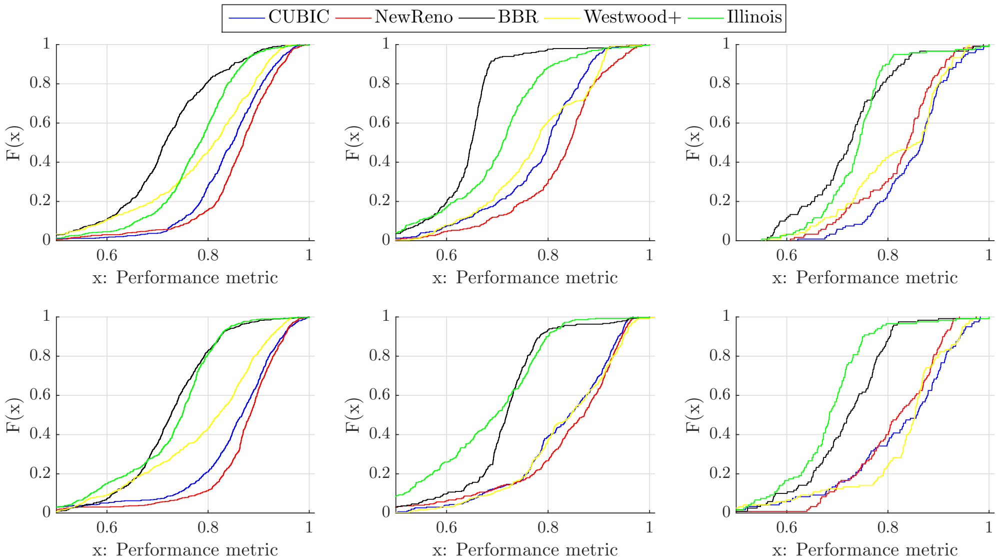


Figure 6.5: Comparison of 5 CCAs performance metric with weighted delay while movement at different speeds under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

All examples show that for our selected mobility scenarios BBR and Illinois are not appropriate from the delay's point of view. All selected TCP flavours try to maximize the goodput, while each of them is driven by different features. It is surprising to see that two delay-aware candidates such as BBR and Illinois, suffer due to the massive packet injection. It is therefore concluded that even though the TCP flavours are prepared to handle the delay across the transmission path, the variability present in the proposed mobility scenarios, even under considerably stable fading conditions, is enough to cause faulty interpretation or estimation of the network status in this regard.

It is the first time in the whole analysis and current evaluation of best TCP candidates that Westwood+ happens to be an eligible candidate for some mobility scenarios. Two are the scenarios that Westwood+ provides similar or comparable delay-based performance metric to the one given by NewReno and CUBIC: under considerably stable fading conditions and under variable conditions with low available capacities (mobility scenarios at 300 km/h with HST300 fading pattern). While in forward movement pattern, the results of Westwood+ are comparable but still below CUBIC and NewReno, in backward movement scenarios achieves greater results. The main deficiency of Westwood+ is the extreme back-off application after Slow Start that forces the CCA to take long time until it converges. The backward movement pattern minimizes this effect in some way due to its lower available capacities (comparing with forward movement) in the beginning of the transmission. Therefore, a combination of these details together with the delay-based evaluation, makes Westwood+ suitable to be selected.

Apart from the suitability of Westwood+, one CCA prevails over the others in most of the scenarios. NewReno achieves better delay-based performance than any other candidate under considerably stable fading conditions and under variable fading with big bandwidth jumps. Under variable fading conditions with lower achievable capacities, NewReno is one of the three eligible candidates together with Westwood+ and CUBIC with an unclear best CCA among the three. The presence of NewReno in all mobility scenarios with a delay-based evaluation, provides a valuable information. Even though NewReno represents the oldest and most basic version of TCP, there still exist circumstances where the selection of this archaic CCA could bring noticeable improvement to the overall performance of delay-sensitive applications.

6.2.2 Intra-protocol selection during transmission establishment under low latency

During the previous analysis of low latency mobility scenarios, we have detected many scenarios/cases in which a similar goodput is achieved by different CCAs but significantly more delay and retransmissions are suffered by some of the CCAs in the group. Those examples require a evaluation of the protocols based on different application-layer requirements so as to appropriately select the best candidate whose performance matches the application requirements and network conditions. This Subsection is responsible for evaluating and selecting the most suitable CCA candidates in low latency mobility scenarios. The structure and format of the explanation is the same as the one followed by 4G latencies and therefore is divided in three parts: the evaluation based on goodput, based on the performance metric and based on delay.

6.2.2.1 Goodput based CCA selection with limited flexibility of the transport layer under low latency

The previous section has shown the goodput performance as one of the evaluated parameters in the spider plot. This time, Figure 6.6 only covers the representation of the goodput in a time axis. The figure not only provides with the most adequate CCA or group of CCAs based on the average goodput, but also reveals details about the variability.

Figure 6.6 demonstrates that depending on the scenario, the selection of the most appropriate CCA is capable of allowing better performance, achieving greater capacities. All in all, the best practises regarding the goodput based evaluation are the following under low latency:

- In forward movement, CUBIC is damaging comparing with other candidates and its aggressiveness looks as if it does not properly fit for such scenarios.
- Due to the low end-to-end latency and taking into account that forward movement is an easier movement pattern to handle by TCP (because it allows having almost every time packets in-flight), we can barely decide one specific CCA in each scenario. The selection should be carried out among a group of CCA that have reported a very similar outcome in terms of goodput.
    1. BBR and Illinois have demonstrated to better handle considerably stable fading conditions.

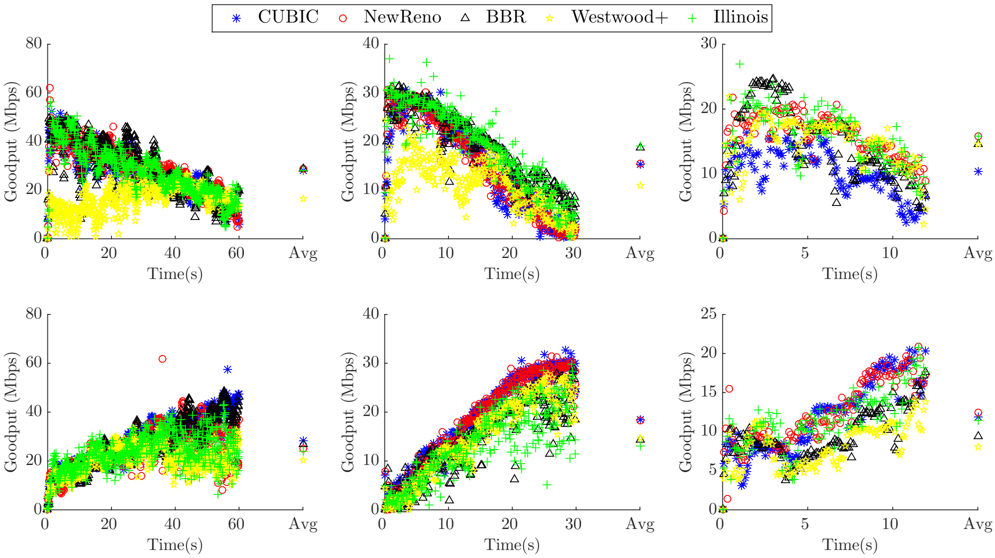


Figure 6.6: Goodput comparison of 5 CCAs under low latencies: Forward movement on top; Backward movement on bottom line.

2. NewReno and Illinois have shown a great performance in variable scenarios with low available capacities.
3. Under variable conditions with big capacities and big jumps between resource assignments, all CCAs but Westwood+ have proven to achieve very similar performance. Comparing with 4G latency scenarios, this similarities clearly come due to the reduction in the end-to-end delay.

- In backward movement, even with similarities among the CCAs, it is still clear that one or two candidates prevail over the others. Variable scenarios are better handled by CUBIC when the capacities are higher, whereas NewReno improves the performance of variable scenarios with low available bandwidths. No matter which CCA you select in stable fading conditions, both CCAs would prompt the best recorded performance. These results demonstrate that, yet, there is value in the performance of by-default CCA (CUBIC) and classic CCA (NewReno) in current and, more importantly, future networks with low latency scenarios.

6.2.2.2 CCA selection based on performance metric with limited flexibility of the transport layer under low latency

Once we have evaluated and selected based on goodput the best candidates for low latency mobility scenarios, it is crucial to evaluate the performance of the TCP flavours from the point of view of the performance metric. This metric is able to gather more information from the actual outcome of each CCA regarding the performance. Figure 6.7 depicts the ECDF results of the performance metric in all scenarios. The distribution of subplots is the same as before with forward movement scenarios in the first row and backward movement ones in the bottom line. Each column represent a different speed and fading combination, being from the left to the right the scenarios at 60 km/h, at 120 km/h and 300 km/h respectively.

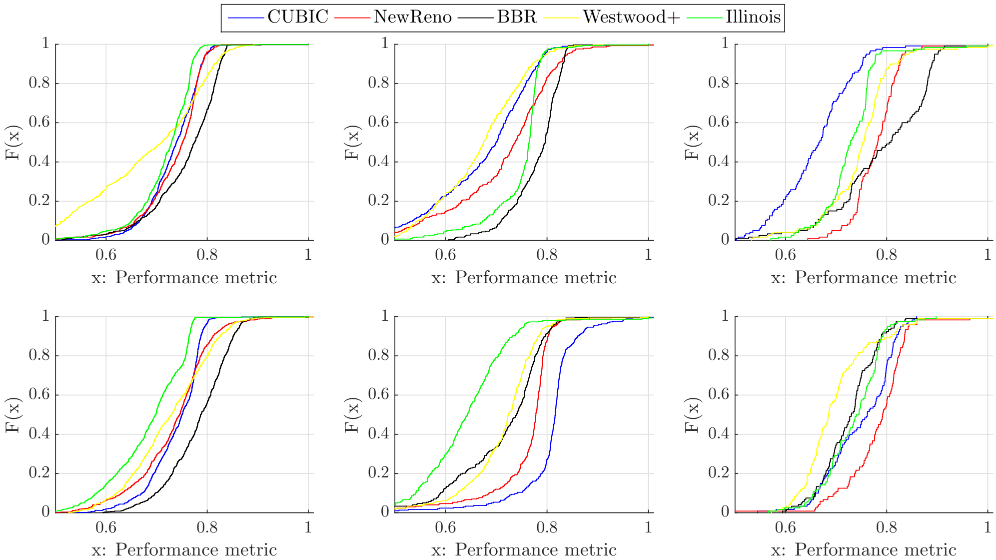


Figure 6.7: Comparison of 5 CCAs performance metric while movement at different speeds under low latencies: Forward movement on top; Backward movement on bottom line.

Taking into account that the current evaluation considers three different aspects of the overall performance, the results in Figure 6.7 widely differ with the outcome merely based on goodput. There is a clear pattern that confirms BBR as the best candidate in forward movement scenarios. The previous evaluation has shown that, based on goodput, in all scenarios BBR is within the group of best performers. Its capacity to reduce the delay close to the baseline delay in a movement pattern that strongly suffers due to the contrary, makes BBR the best candidate considering the performance metric.

In backward movement pattern, there are three clear candidates, one per each speed and fading combination:

1. BBR is able to better handle the required scalability under variable fading conditions with big available capacities and jumps between consecutive assignment samples. As under 4G latencies, BBR demonstrates great scalability. In addition, under low latency circumstances BBR is capable of better managing the induced delay, achieving a considerably better performance than the rest of the TCP implementations in the commented network use-case.

2. CUBIC shows that under considerably stable fading conditions, if the end-to-end latency is reduced, it is able to make the most of the available capacity. Under 4G latency conditions, CUBIC has demonstrated to suffer due to its incapability of properly ramping-up. With a reduced delay and under non-aggressive fading conditions, CUBIC proves to have the ability to scale. Even though goodput results have prompted similar outcome for NewReno and CUBIC, the latter clearly dominates based on the performance metric.

3. The combination of low latency together with low available bandwidths makes the performance playground very suitable for the selection of NewReno, showing great performance both in achieved goodput and based on the performance metric.

Overall the results in Figure 6.7 show two important facts in comparison with the mobility scenarios under 4G latencies:

- The performance of BBR shows a consistent good performance in terms of goodput. The main difference is that in low latency scenarios the behaviour of the CCA is more controlled and is able to take full advantage of its features, achieving great goodput while the delay is kept close to the baseline latency. This outcome shows that, the shorter response times (RTT) in each packet, the better for the accurate estimation of BBR in mobility scenarios.
- Based on the performance metric, it is clear that the reduction in the end-to-end latency makes Illinois perform worse in comparison with other candidates. While in 4G latencies, Illinois is eligible in many mobility circumstances that take advantage of it aggressiveness, the same scenarios under shorter latencies do not require its aggressiveness. Actually, this feature only increments the injected delay as well as the number of retransmitted packets.

6.2.2.3 Delay based CCA selection metric with limited flexibility of the transport layer under low latency

Once we have decided and explained the most appropriate CCA selection based on goodput and the performance metric, it is important to decide which CCA is more suitable for delay-sensitive applications. To evaluate and decide the best CCA among the candidates, we will again weight the values in the performance metric, giving to the field of the delay ten times more significance than the other two. Figure 6.8 shows the ECDF plots of the TCP evaluation based on delay in the mobility scenarios under low latency conditions. Since the point of view to evaluate the group of TCP flavours is very distinct comparing with the goodput-based and performance metric based selections, the results depict very different outcome.

Out of the results in Figure 6.8, there are some details that need to be addressed. As under 4G latencies, Westwood+ emerges as an eligible candidate that does not reach the maximum capacity, but it is able to perform minimizing delays and achieving a good performance proportion between goodput and delay for delay-sensitive application.

On average BBR performs better than the rest of the candidates, showing once again that a low latency allows a more controlled behaviour of the CCA. BBR is capable of fulfilling the task of one of its main features: the avoidance of excessive buffering and the drastic reduction of the delay. Therefore, BBR could be very important in future low latency scenarios due to its capability to scale, achieve great goodputs but at the same time reduce the self-inflicted effect of delay.

CUBIC and its good scalability is again confirmed in low latency scenarios from the perspective of induced delay, providing with the most suitable candidate to scale under considerably stable fading conditions in low latency mobility scenarios.

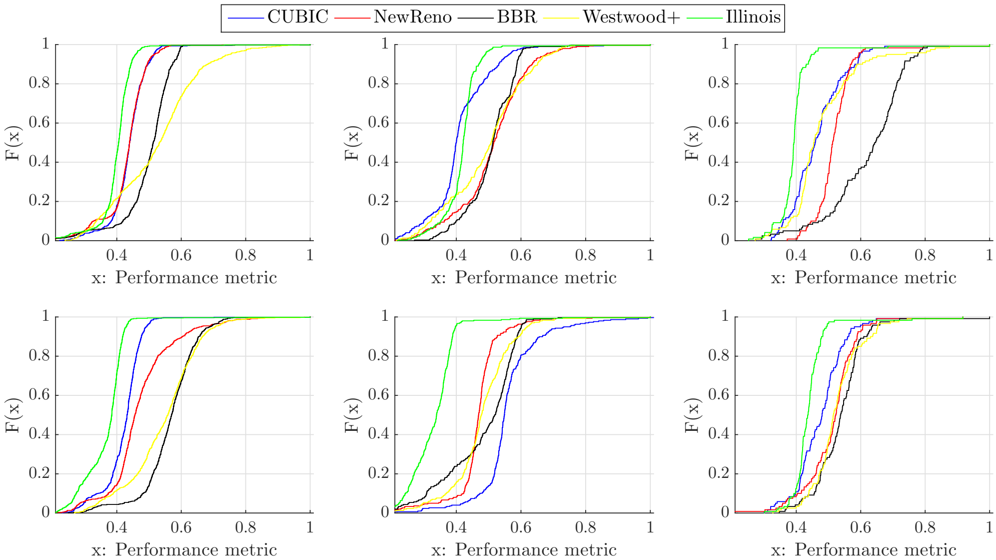


Figure 6.8: Comparison of 5 CCAs performance metric with weighted delay while movement at different speeds under low latencies: Forward movement on top; Backward movement on bottom line.

6.2.3 Conclusions in the intra-protocol selection of Option 1

After evaluating the selected mobility scenarios, based on goodput, based on delay and the performance metric, under both 4G latencies and low latency that moves toward the potential delays in 5G networks, we have been able to select the most appropriate CCA for each situation. The CCA selection in bounded by the limited flexibility of the transport layer that only allows a selection in the beginning of the transmission. So far, the selected CCAs are gathered in Table 6.1 with an asterisk for the cases in which more than one CCA could be picked. Speed 1 represents the results at 60 km/h with EVA60, Speed 2 the results at 120 km/h with EVA120 and Speed 3 the results at 300 km/h with HST300.

1. Illinois has shown an aggressiveness that is suitable for application with goodput requirements in both movement patterns in 4G latency scenarios, but only appropriate for backward movement under low latency. The reduction in the end-to-end delay increases the responsiveness of other CCAs, leaving no room for the aggressive behaviour of Illinois.

2. BBR has demonstrated good performance in terms of goodput under 4G latencies. However, it also induced a long delay and provokes a great number of transmitted packets. Related to that, Google released the information that primarily BBR was intended to drain more drastically. However, they were forces to change a little bit their strategy because in cellular networks, the eNodeBs were underassigning resources to them due to the few injected packets. So, they concluded that they should keep part of the queue built in order to get a fair share of resources. This increase in the aggressiveness could be part of the cause to experience such delays and retransmitted packets under 4G laten-

Context	Speed 1					
	Low latency			4G latencies		
	Goodput	Metric	Delay	Goodput	Metric	Delay
Forward	*	BBR	Westwood+	Illinois	*	NewReno
Backward	CUBIC	BBR	BBR	BBR	CUBIC	NewReno
Context	Speed 2					
	Low latency			4G latencies		
	Goodput	Metric	Delay	Goodput	Metric	Delay
Forward	*	BBR	*	BBR	*	NewReno
Backward	*	CUBIC	CUBIC	BBR	BBR	NewReno
Context	Speed 3					
	Low latency			4G latencies		
	Goodput	Metric	Delay	Goodput	Metric	Delay
Forward	*	BBR	BBR	CUBIC	CUBIC	*
Backward	NewReno	NewReno	BBR	BBR	CUBIC	*

Table 6.1: Wrap-up CCA selection with limited flexibility in the transport layer that only allow intra-protocol selection during transmission establishment (Option 1).

cies. In low latency scenarios instead, BBR has a more controlled and balanced behaviour, achieving a sufficiently good performance in goodput but also preserving a low delay.

3. Westwood+ has been labelled as an alternative to provide delay-sensitive service, not maximizing the goodput but allowing to ensure low induced delay.

4. NewReno provides with a valuable option for delay-sensitive applications in 4G mobility environments. Besides, under low latency, NewReno is able to achieve close to the maximum available capacity, being specially significant its performance and scalability in backward movement pattern under variable fading conditions with low achievable rates.

5. CUBIC has shown a balanced performance in variable scenarios with low available bandwidth under 4G latencies. Apart from that, under low latency circumstances, CUBIC is suitable to be selected for forward movements with big variability. Finally, under reduced end-to-end latencies CUBIC has proven to have the ability to appropriate scale.

6.3 [Option 2]: Real-time intra-protocol selection

After the evaluation of each mobility scenario as a whole and having selected the most appropriate CCA during the transmission establishment, this section covers a more complex CCA selection. The idea is to evaluate each scenario but taking into account the detected different phases in the previous analytical chapter. Each phase or network area has its own dependencies in order to better suit the performance of the TCP implementations. Therefore, it is important to evaluate in a similar way as the previous section but taking into account the performance of each phase. However, this section does not only aim to cover the evaluation and CCA selection process, but it is also devoted to test, evaluate and compare the CCA switches between distinct network areas.

This is, provided that in a close future, further evaluation of [12, 13, 83–85] as a more flexible transport layer would allow switching between similar transport protocols in-flight or real-time, the section covers the evaluation of such CCA switches and analyzes resulting scheme's strengths and weaknesses. To that end, four tasks need to be covered for each mobility scenario under certain criteria:

- 1) Evaluation of most appropriate CCA for each phase.
- 2) In case of a CCA switch is required, the CCA in the first phase will input the second by modelling its CWND in accordance to the induced delay.
- 3) Afterwards, starting from the switching point, several experiments will be carried out with the second CCA but the starting conditions extracted from the first one.
- 4) Final evaluation of the selection proposal (with or without CCA switches).

It is clear that the detected phases and performance dependencies are deployment-dependant and could not be extrapolated as a general-purpose management. However, the section provides with a valuable evaluation of bright and dark sides of the technique itself as well as performance-based examples of CCA selections that could be adapted to similar network conditions in other deployments.

Before the explanation and evaluation, Section 6.3.1 introduces the procedure to evaluate CCA switches when a performance metric prompts that such a change from one CCA to other CCA could bring important performance improvements. Afterwards, as in earlier section and analytical chapter, the explanation is divided depending on the end-to-end baseline delay. Section 6.3.2 covers the explanation, CCA selection, CCA switch experimentation and final evaluation of most reasonable TCP candidate for each scenario phase considering goodput requirements, performance metric requirements and delay-sensitive requirements under 4G latencies. Section 6.3.3 describes in the same format the CCA selection for low latency scenarios. Finally, Section 6.3.4 concludes the section and gathers the most important CCA selections and best practises.

6.3.1 Evaluation process of real-time CCA switches in-flight

In a transport layer with enough flexibility to allow an in-flight or real-time switch between CCAs, there is a need for transferring network status information from one CCA to the next one. In order to do that, we consider a simplified modelling of the CCAs in the switching points so as to feed the next CCA with transport protocol information.

Since the simplified modelling should be compliant with the parameters available in all selected CCAs and taking into account the limited information in precise TCP implementations, we model in accordance to the capabilities of the most limited one (NewReno). In this simplified modelling, we only considered two parameters: the evolution of both CWND and the RTT. In order to provide with a mechanism that is able to switch between CCAs, some information should be transmitted or transferred from the first CCA to the second one. In order to feed the second CCA with the information from the first on, we model the network situation of the first CCA by calculating the average CWND and RTT of the last 7 samples in the switching point so as to have a smooth value of CWND (sCWND) and RTT (sRTT or sD) and assess the final value based on an unloaded network (See Eq. 6.3). This is, if for instance the smooth CWND (sCWND) is 100 and the calculated smooth delay is 150 ms (sD) from a baseline delay (Dmin) of 60 ms, we take the percentage increment of delay from the baseline delay so as to calculate the final CWND (CWND). Therefore, the final sCWND (see Eq.) based on the baseline delay would be 40 segments. The calculated CWND value would serve to launch and evaluate the feasibility of the proposed CCA switch by checking the performance of the second CCA considering as an input the mentioned CWND value.

$$CWND = \frac{sCWND}{\frac{sD}{Dmin}} \tag{6.3}$$

Finally, the last step of the feasibility analysis validates and evaluates the obtained performance with the TCP intra-protocol switch of CCAs in comparison with previously obtained results using single CCAs. The evaluation process is also based on different criteria and need to be grouped in distinct mobility use-cases so as to finally conclude for which situations the proposal is able to improve the criteria-based performance.

6.3.2 Real-time intra-protocol selection under 4G latencies

Previous analysis has detected various phases in which the performance requirements are distinct. This Subsection is responsible for evaluating and selecting the most suitable CCA candidates in low latency mobility scenarios for each phase.

The structure and format of the explanation is similar to the one utilized before and therefore is divided in three parts: the evaluation based on goodput, based on the performance metric and based on delay. However, this time the explanation of each criteria is done in a different way. Firstly, a table will be introduced. Each table table is horizontally divided in 3 sections, one per speed and fading combination. Each section is vertically divided in 6 sections in accordance to the detected phases for each mobility pattern. The forward movement is comprised of a Start-up phase, a goodput Sustainer phase and a constrained area, whereas the backward movement is formed by the Start-up phase, the constrained area and the scaling area. The Start-up phase will be treated always as the first two seconds and the constrained area will be of ten seconds at 60 km/h, five seconds at 120 km/h and two seconds at 300 km/h. For each phase and mobility scenario, the table will show the evaluation of each CCA based on the criteria under study. In order to make the selection of the best CCA for each phase easier, the cell of the selected CCA is painted in green. The painted cells in yellow are possible alternatives to the green ones that do not require a CCA switch but could provide with a good enough result. However, since the main goal of the explanation is to clarify the strengths and weaknesses of the performance that a combination of CCAs could bring, the selected combination is the green one. After the table and once the CCA switching points have been selected, a figure is presented with the obtained results with the last experimentation regarding a first CCA in a switching point serving as an input to the second one. This step could either confirm that the selected CCA combination is more appropriate than a single CCA or reject showing that certain CCA could achieve good performance in a phase only due to its dynamics in that precise moment.

6.3.2.1 Goodput based CCA selection with flexible transport layer under 4G latencies

The evaluation, phase-wise CCA selection and CCA combination comparison will be explained based on goodput. Table 6.2 covers the representation of the goodput for each phase and mobility scenario. The table not only provides with the most adequate CCA or group of CCAs based on the average goodput, but also reveals details about whether a single CCA is appropriate for the entire transmission or instead, it is in principle better to switch.

Table 6.2 shows that in comparison with the results with limited flexibility of the transport layer, the CCA selection in backward movement pattern in much more consistent in every phase. In the mobility scenarios at 60 km/h and 120 km/h, BBR demonstrates that it achieves the greatest goodput throughout all the phases, showing great transmission initialization, ability to handle constrained network conditions and scalability. At 300 km/h instead, a combination of CCAs is detected to be presumably more convenient. With limited flexibility of the transport layer, both BBR and Illinois perform very similarly. Nevertheless, this is in principle based on the better ability of BBR to initialize the communication and the scaling capabilities of Illinois. The combination of CCAs needs to be verified in order to ensure that the good performance of certain CCA in an specific area in not due to its dynamics and the previous evolution. Therefore, we need to verify the CCA switch as stateless as possible with the basic input of an adapted CWND based on the induced delay.

Regarding the forward movement pattern, many different CCA combinations arise. With limited flexibility of the transport layer the selected CCAs have been Illinois, BBR and CUBIC for scenarios at 60 km/h, 120 km/h and 300 km/h respectively. However, in a more flexible transport layer environment the evaluation is prone to improve the performance of the constrained area with Illinois and Westwood+ in 120 km/h and 300 km/h scenarios. Apart from that, the variable scenarios of 60 km/h

CCA	Forward at 60 km/h			Backward at 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	24.009775	26.136320	14.830030	3.387359	12.930881	24.041202
NewReno	28.090239	25.083221	11.665474	7.652201	12.468487	21.548364
BBR	19.104916	28.319904	12.821943	9.847369	15.045444	26.498632
Westwood+	7.815343	17.079562	12.960179	4.384548	10.493897	18.734572
Illinois	22.793455	28.917083	13.839019	8.665801	12.762431	25.633693
CCA	Forward at 120 km/h			Backward at 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	18.765119	18.690868	2.464303	0.992367	1.274240	12.974852
NewReno	15.421204	17.864459	2.616826	1.224047	1.946112	13.985904
BBR	20.290345	20.925741	8.343183	1.835103	3.595545	18.777123
Westwood+	9.653337	16.781103	7.221466	0.583065	1.763342	16.113730
Illinois	14.300452	18.848679	8.883770	0.504873	0.904839	14.456446
CCA	Forward at 300 km/h			Backward at 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
CUBIC	11.621652	14.067320	7.813408	5.862473	9.344234	-
NewReno	10.721961	11.533803	6.900203	3.958836	4.190898	-
BBR	10.779881	9.727664	4.845008	7.371289	9.320486	-
Westwood+	4.674148	9.537976	8.809632	2.868975	8.517715	-
Illinois	9.738287	13.268989	6.051675	4.799641	9.773614	-

Table 6.2: Goodput in detected phases under realistic 4G latencies.

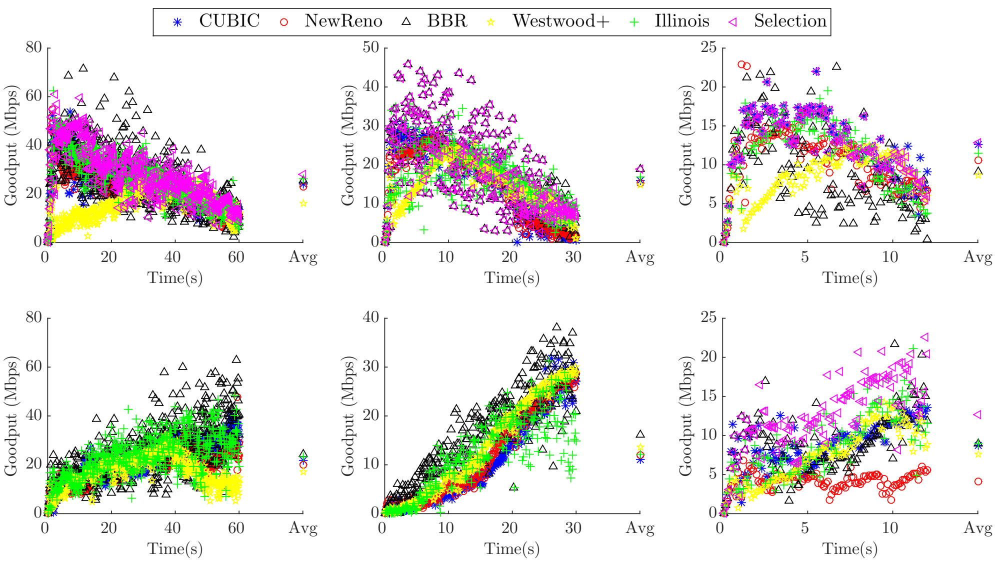


Figure 6.9: Goodput comparison of 5 CCAs + selected CCA combination under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

require the use of an aggressive CCA as NewReno in the transmission initialization, the ability to sustain the goodput of Illinois and the better management of the constrained area of CUBIC. As mentioned before, the performance of selected CCAs from the switching point in advance need be evaluated. After experimenting with switching CCAs, Figure 6.9 depicts the goodput of all TCP flavours in comparison with the decided combination of CCAs labelled as "Selection".

Figure 6.9 shows that the selected CCA combination achieves greater or similar goodput to the most appropriate CCA in limited flexibility of transport layer. On the one hand, there are two scenarios in which the differences are almost non-existent: forward movement pattern with considerably stable fading conditions (mobility scenario at 120 km/h) and variable fading conditions with low achievable capacities (mobility scenario at 300 km/h) report improvements of 0.23% and 1.32% with the CCA combination. The mentioned two scenarios are complicated to enhance due to the movement pattern itself (tendency to help CCAs have always in-flight packets and take full advantage of radio resources) and the stability of one scenario and low capacities of the other, leading to have little room for evolved mechanisms. On the other hand, our selected combination of CCAs improve the average goodput of forward movement with variable and high capacities and the backward movement at 300 km/h in 7.54% and 41.21% respectively. **These two scenarios demonstrate that there is room for our proposal of CCA switch and that in some circumstances it could bring a significant gain**.

6.3.2.2 CCA selection based on performance metric with flexible transport layer under 4G latencies

Having proved that in terms of achieved goodput, the combination of CCAs may well improve the performance, it is important to check whether the switching mechanism is applicable or not for other criteria with a wider view of the parameters involved in the final performance such as the performance metric. Table 6.3 gathers the evaluation based on performance metric of each phase or stage in each mobility scenario.

The results from the Table 6.3 show two different selection pattern divided by the movement patterns as well. In forward movement pattern, the scenarios at 60 km/h and 120 km/h suggest a combination of CCAs as the best solution based on the evaluation of each phase with the performance metric. In limited flexibility, both scenarios have demonstrated that several CCAs are able to achieve a similar outcome. During the main part of the transmission, NewReno is selected as the most balanced CCA to sustain the goodput. CUBIC and Illinois are selected to face the constrained network conditions. Even though it is not reasonable to think about the selection of more aggressive CCA solutions to deal with such limited network circumstances, the selected combinations of CCAs need to be evaluated to determine their suitability or inappropriateness.

Regarding backward mobility pattern, the selected CCA alternatives to drive the transmissions are in the main part of it the same as the ones selected for limited flexibility transport layer: CUBIC, BBR and CUBIC for scenarios at 60 km/h, 120 km/h and 300 km/h respectively. However, the evaluation of each phase prompts that in principle another CCA selection in the Start-up phase would help achieve better performance (Illinois for 60 km/h, Westwood+ for 120 km/h and BBR for 300 km/h). Taking into account all the suggested combinations of CCAs and experimenting with the proposed CCA switches, Figure 6.10 depicts the performance-metric-based comparison results in an ECDF graphs of the selection of CCA combination against each CCA by their own.

CCA	Forward at 60 km/h			Backward at 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.849943	0.876508	0.891923	0.683201	0.857607	0.878203
NewReno	0.912654	0.882414	0.819121	0.835962	0.842523	0.865342
BBR	0.850915	0.848214	0.777470	0.873315	0.852180	0.843479
Westwood+	0.777437	0.793647	0.819489	0.787850	0.821990	0.806111
Illinois	0.851940	0.879932	0.809620	0.915885	0.793931	0.843218
CCA	Forward at 120 km/h			Backward at 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.896179	0.834193	0.654994	0.760316	0.695770	0.814306
NewReno	0.838534	0.847879	0.670324	0.754736	0.761508	0.836605
BBR	0.895338	0.803627	0.756128	0.785263	0.825925	0.871402
Westwood+	0.809043	0.833051	0.759766	0.897523	0.745392	0.866068
Illinois	0.809144	0.818351	0.809439	0.695083	0.639612	0.796268
CCA	Forward at 300 km/h			Backward at 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
CUBIC	0.938419	0.918043	0.861951	0.843963	0.882451	-
NewReno	0.887811	0.855712	0.848949	0.806045	0.743098	-
BBR	0.869097	0.775462	0.781257	0.852164	0.766925	-
Westwood+	0.773913	0.837580	0.855491	0.743528	0.861353	-
Illinois	0.864424	0.857447	0.776710	0.813088	0.821311	-

Table 6.3: Performance metric in detected phases under realistic 4G latencies.

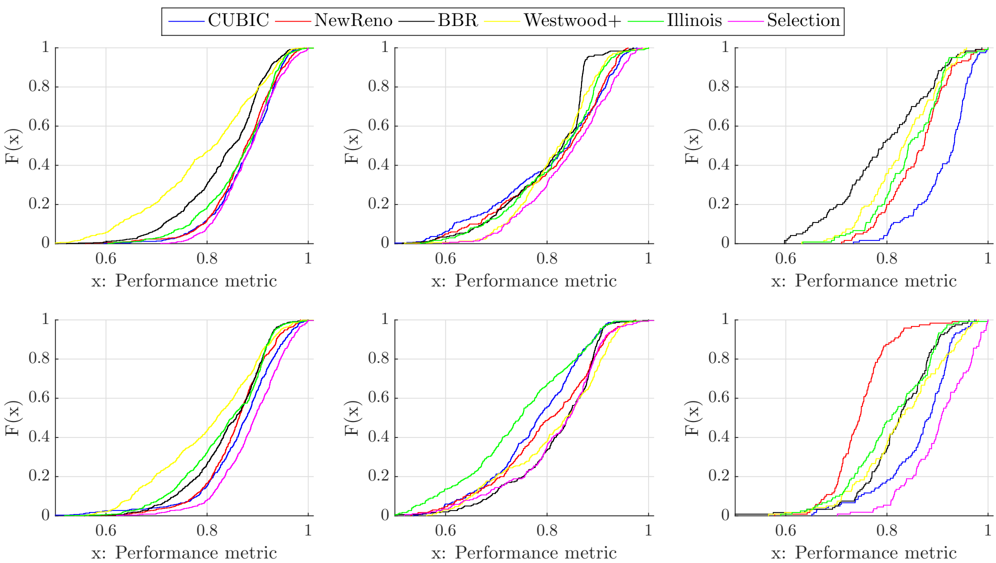


Figure 6.10: Comparison of 5 CCAs + selected CCA combination based on performance metric while moving at different speeds under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

The results in Figure 6.10 show that the proposed CCA combinations are capable of providing with more balanced TCP behaviour throughout the experimentation in the required scenarios. In forward movement at 60 km/h the inclusion of CUBIC in the constrained network area gives to NewReno a slightly more appropriate performance based on the performance metric. At 120 km/h, even in forward movement under considerably stable fading conditions, the combination of CUBIC, NewReno and Illinois prompts an enhanced performance, underlining the possible impact that the switching technique between distinct CCAs may well have in future networks. In the same way, in backward movement pattern, the use of Illinois and BBR in the Start-up phase help CUBIC achieve by far greater performance. Finally, backward movement at 120 km/h shows that the addition of Westwood+ in the Start-up phase of a transmission driven by BBR can only obtain similar results comparing with BBR in the whole experimentation. Summing up, **the selected combination of CCAs has gathered better results than using single CCAs in four cases out of five, highlighting again the potential of this simplistic technique**.

6.3.2.3 Delay based CCA selection metric with flexible transport layer under 4G latencies

Both evaluations based on goodput and performance metric have demonstrated under 4G latencies that the selected combinations of CCAs could improve the performance in the majority of the cases. Now, it is crucial to examine if the CCA switching mechanism is suitable from the point of view of delay-sensitive applications. Table 6.4 covers the evaluation based on a delay-driven weighted performance metric of each network phase in each mobility scenario.

Once evaluated the delay-based performance metric for every phase, Table 6.4 shows a combination of CCAs for each scenario as the preferable option. In the selection of most appropriate CCA for limited flexibility in the transport layer, NewReno has emerged as the best option for both movement patterns at 60 km/h and 120 km/h, while the scenarios at 300 km/h have depicted a similar performance of CUBIC, NewReno and Westwood+. Aligned with that outcome, in the selected CCA combinations at 60 km/h and 120 km/h, NewReno has been selected in the longest phases of the transmissions for both movement patterns: filling the goodput sustainer requirements in forward movement and the scaling ability in backward movement. Apart from that, two import details need to be addressed: 1) Westwood+ seems the best option in backward movement scenarios to handle the constrained network conditions and even with certain ability to scale in the scenario at 300 km/h where the maximum capacities are low. This is because in backward movement in which the first stage of the transmission happens under relatively low capacities, the underperformance of Westwood+ (massive back-off application after Slow-Start) is less significant and Westwood+ is able to achieve considerably good rates, while it keeps the delay low. 2) Almost all scenarios show that for Start-up phase delay-based performance, it is better to select other CCA, rather than the most prominent one (NewReno), showing that, a combination of a precise CCA in Start-up phase and NewReno in the majority of the transmission, has the potential to bring performance gains. After selecting the most appropriate CCA combinations, Figure 6.11 depicts the outcome of such mixture of CCAs in comparison with the delay-based performance of single CCAs.

CCA	Forward at 60 km/h			Backward at 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.799446	0.845469	0.813768	0.592770	0.803300	0.860929
NewReno	0.816017	0.874937	0.773247	0.805448	0.778506	0.883127
BBR	0.836482	0.734044	0.617415	0.802198	0.701656	0.724915
Westwood+	0.747761	0.799672	0.716216	0.773306	0.828443	0.783157
Illinois	0.815272	0.795255	0.655601	0.819894	0.652101	0.731117
CCA	Forward at 120 km/h			Backward at 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.836000	0.787821	0.708692	0.670766	0.686524	0.854401
NewReno	0.809080	0.841098	0.731668	0.602877	0.709772	0.867863
BBR	0.793805	0.644438	0.565446	0.613775	0.643117	0.726863
Westwood+	0.836728	0.794574	0.648897	0.711000	0.717744	0.856081
Illinois	0.733149	0.703924	0.661405	0.563740	0.611162	0.693713
CCA	Forward at 300 km/h			Backward at 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
CUBIC	0.920120	0.846501	0.791107	0.712600	0.842064	-
NewReno	0.834684	0.818346	0.819236	0.833584	0.807253	-
BBR	0.784755	0.702728	0.740885	0.689073	0.704949	-
Westwood+	0.801269	0.848013	0.700408	0.717692	0.848167	-
Illinois	0.794962	0.733246	0.700603	0.694632	0.677870	-

Table 6.4: Delay-based performance metric in detected phases under realistic 4G latencies.

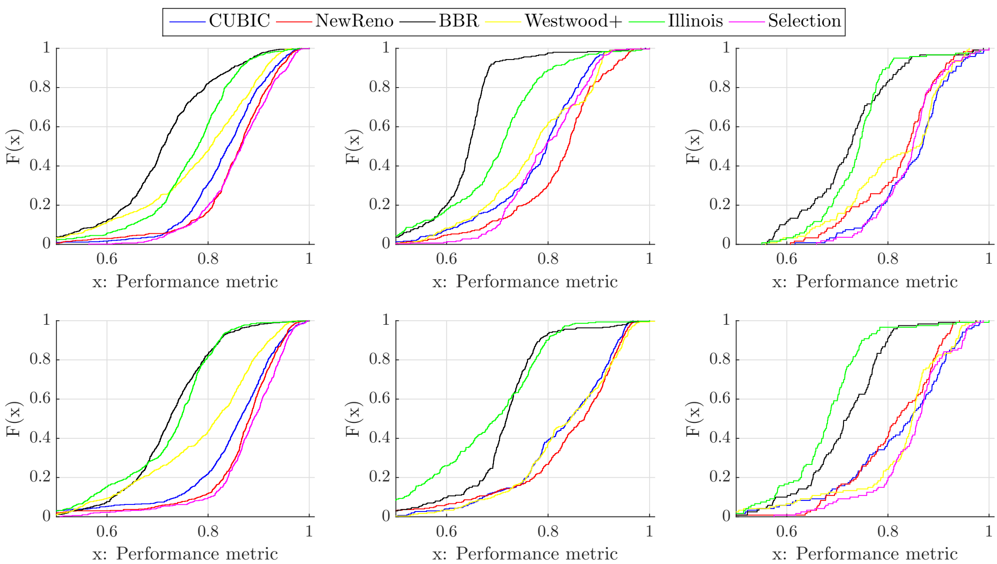


Figure 6.11: Comparison of 5 CCAs + selected CCA combination based on performance metric with weighted delay while moving at different speeds under realistic 4G latencies: Forward movement on top; Backward movement on bottom line.

The results in Figure 6.11 demonstrate that the proposed CCA combinations are not capable of providing with a better delay-based performance and they could only obtain similar results to the ones achieved utilizing a single CCA. Even though the evaluations based on goodput and performance metric have reported successful results, the delay-based interpretation have proved the contrary. Taking into account that the scenarios are under 4G latencies and therefore high BDP paths, it needs to clarify whether shorter RTT periods (shorter BDP and quicker response between consecutive decisions in the server) could be more convenient for the application of CCA switching and combining technique.

6.3.3 Real-time intra-protocol selection under low latency

This chapter has evaluated with a flexible of transport layer that allows real-time intra-protocol selection, the most suitable CCAs for every mobility scenario based on goodput, performance metric and delay under both 4G latencies and low latency circumstances. The previous analysis has detected various phases in which the performance requirements are distinct, presumably leading to a different CCA selection for each phase. In this lines, we have covered the evaluation of CCAs with a flexible transport layer that allows in-flight CCA switching under 4G latencies. Now it is important to clarify whether the conclusions and outcome under 4G latencies are equally applicable under low latency scenarios.

The evaluation and explanation will follow the same structure and format as the one utilized under 4G latencies. A big division is done in three so as to split the evaluation based on goodput, based on the performance metric and based on delay. Each part of the evaluation is formed by three steps: 1) A table is presented with the criteria-based evaluation for each phase and CCA in all mobility scenarios. The table not only provides with the most adequate CCA or group of CCAs based on the criteria, but also reveals details about whether a single CCA is appropriate for the entire transmission or instead, it is in principle better to switch; 2) The CCA switching points are modeled and new experiments carried out considering the input performance information of the first CCA in the switching point and the selection of the second CCA; 3) A final evaluation and suitability-check figure and explanation is done in order to finally gauge the appropriateness of the CCA switching technique from the perspective of a precise performance criteria.

6.3.3.1 Goodput based CCA selection with flexible transport layer under low latency

The first part of the evaluation will be explained based on goodput and will select the most suitable CCAs for each phase of the mobility scenarios. Table 6.5 covers the representation of the goodput for each phase in mobility scenario and highlights with green the selected CCA.

Table 6.5 shows that as with limited flexibility of the transport layer, there are huge similarities in the performance of CCAs. Although the differences are minimum, certain patterns are clear. On the one hand, in the forward movement the combination of NewReno in the majority of the transmission and CUBIC in the management of the constrained network area is in principle the best choice to obtain better goodput-based results in the mobility scenario at 60 km/h. The good performance of CUBIC in the constrained area of the combined circumstances of decreasing quality movement

CCA	Forward at 60 km/h			Backward at 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	37.142169	30.301331	16.224357	8.629119	17.098225	30.770695
NewReno	37.986836	30.949793	15.919119	11.330121	14.295380	27.125017
BBR	36.147876	31.124277	15.515030	9.159087	15.774271	29.320803
Westwood+	12.991460	16.790566	16.206402	8.159967	14.568328	21.698994
Illinois	35.300313	30.455422	15.915065	10.241225	16.503821	25.519282
CCA	Forward at 120 km/h			Backward at 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	19.866564	17.823915	2.176634	2.460639	5.040327	21.456039
NewReno	25.277257	17.420741	2.392096	2.054233	4.172171	21.201964
BBR	23.251023	20.607936	8.538180	2.124703	3.813710	16.543342
Westwood+	9.847369	12.715790	3.787582	1.438351	2.842585	17.176446
Illinois	26.406697	21.185457	6.837649	2.234751	3.627723	15.001126
CCA	Forward at 300 km/h			Backward at 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
CUBIC	11.344601	11.462609	4.764885	6.465807	12.993194	-
NewReno	15.636473	16.986488	11.162149	8.073087	13.302486	-
BBR	15.583380	16.039255	8.205333	8.042196	9.612789	-
Westwood+	12.076324	16.434800	10.142757	6.291081	8.467711	-
Illinois	16.987940	16.685304	11.163115	7.791209	12.189650	-

Table 6.5: Goodput in detected phases under low latency.

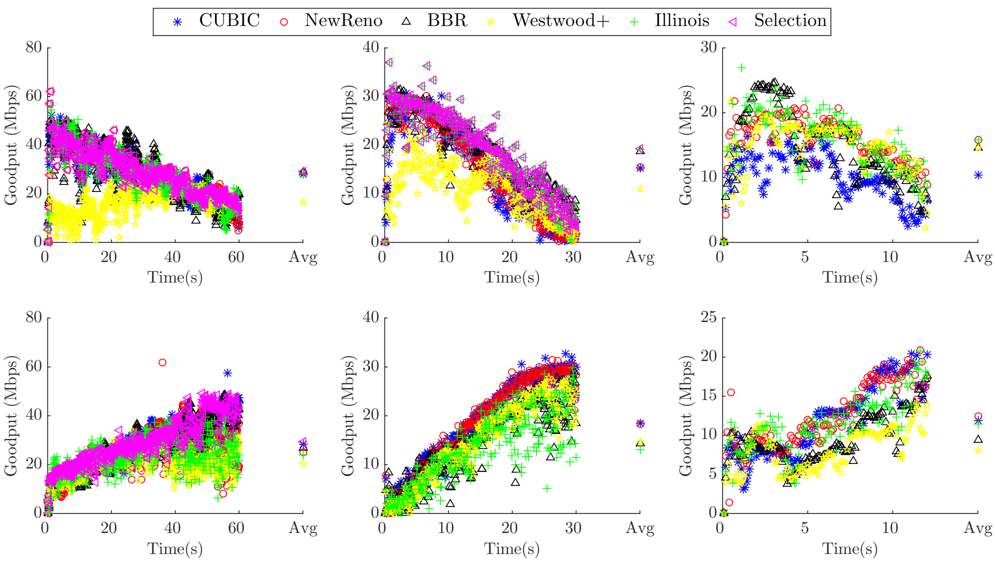


Figure 6.12: Goodput comparison of 5 CCAs + selected CCA combination under low latency: Forward movement on top; Backward movement on bottom line.

with variable fading conditions and overall high available capacities, confirms again (4G latencies scenario has prompted the same) such finding. At 120 km/h with considerably stable fading conditions Illinois makes the most in the whole transmission, but BBR is able to perform by far better than the rest in the constrained network area.

Regarding the scenario at 300 km/h, the similarities between Illinois and NewReno make almost any combination between these two CCAs a suitable option. We avoid the use of the CCA switching technique due to the little room for improvements and select Illinois as the best candidate for the whole transmission.

On the other hand, there is only one scenario in which presumably the combination of CCAs may well bring a noticeable performance enhance. Even though CUBIC is the best candidate to scale in backward movement with variable fading conditions and high available capacities under low latency circumstances (backward movement at 60 km/h), the support of NewReno in the Start-up phase could be beneficial. The rest of the mobility backward movement scenarios are manageable by either CUBIC (at 120km/h) or NewReno (at 300 km/h), highlighting again the importance of by-default CCA and classic CCA in the performance of mobility scenarios under low latency conditions. All the selected CCA combinations need to be checked in order to confirm or deny the suitability of CCA switching technique in low latency scenario withe the focus on goodput maximization.

Figure 6.12 depicts the mobility scenarios with the achieved goodput performance by single CCAs and the selected combination of CCAs. The results show that even under low latency where the responsiveness of every CCA is facilitated due to the reduction in the time gap between two consecutive CCA-based decisions, the selected combination of CCA could slightly improve the goodput-based performance. Two forward movement pattern schemes have been selected to work with a selection of CCAs and the final outcome has enhanced the obtained results with limited flexibility in the transport layer in 1.51% and 0.26% respectively. Even though the evolution in terms of achieved goodput is modest, the usability of the technique is confirmed once again. As for backward movement scenario, the obtained gain with the selected combination of CCA is established in almost the 5%. The increment proves that likewise under 4G latencies, the increasing quality movement scenarios provide with a more suitable playground to gather significant gains in terms of goodput in the required scenarios with the combination of CCAs. **Considering that precise 5G deployments would allow performing at rates much higher than the achievable ones by 4G, the potential performance enhancement while utilizing the CCA switching technique may well be even greater.**

6.3.3.2 CCA selection based on performance metric with flexible transport layer under low latency

It has been demonstrated that in terms of achieved goodput, the combination of CCAs is capable of slightly improving the performance. In this regard, it is important to check if the CCA switching mechanism is suitable for criterion with a more complete and balanced point of view of the performance. To this end, the performance metric is evaluated for each CCA in each phase of the mobility scenarios and the most appropriate candidates are highlighted in green. Table 6.6 gathers the evaluation based on performance metric of each stage in each mobility scenario.

Table 6.6 shows multiple combination of CCAs that in principle could benefit the performance based on the performance metric. While in forward movement with limited flexibility in the transport layer the selected CCA has been BBR for all scenarios, a more flexible evaluation that considers different stages suggests that other

CCA	Forward at 60 km/h			Backward at 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.758045	0.737830	0.703463	0.667170	0.732906	0.741479
NewReno	0.763244	0.742894	0.691693	0.787842	0.680377	0.739558
BBR	0.770395	0.770336	0.699342	0.772034	0.756842	0.786409
Westwood+	0.641814	0.671816	0.762163	0.769376	0.720482	0.725650
Illinois	0.754114	0.724560	0.676425	0.708591	0.715221	0.679328
CCA	Forward at 120 km/h			Backward at 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.751858	0.695220	0.317016	0.721686	0.795992	0.814020
NewReno	0.854650	0.735939	0.573044	0.407003	0.707917	0.775353
BBR	0.749019	0.788840	0.753223	0.429354	0.609983	0.728104
Westwood+	0.670600	0.684644	0.598438	0.603689	0.629479	0.726192
Illinois	0.812229	0.756434	0.670048	0.720031	0.563309	0.640193
CCA	Forward at 300 km/h			Backward at 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
CUBIC	0.690527	0.676068	0.575745	0.675523	0.762844	-
NewReno	0.787205	0.777678	0.774813	0.737796	0.785499	-
BBR	0.843191	0.800826	0.701898	0.776660	0.716589	-
Westwood+	0.768491	0.750220	0.717286	0.763179	0.686771	-
Illinois	0.780000	0.718355	0.719365	0.743782	0.731849	-

Table 6.6: Performance metric in detected phases under low latency.

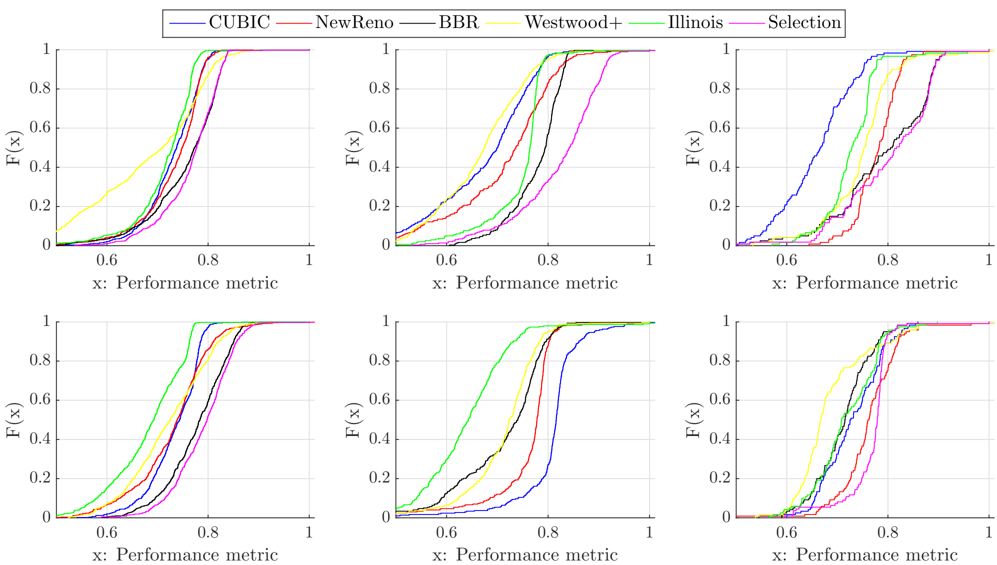


Figure 6.13: Performance metric of 5 CCAs + selected CCA combination while moving at different speeds under low latency: Forward movement on top; Backward movement on the bottom.

CCAs could help BBR achieve better results. In variable fading conditions (scenarios at 60 km/h and 300 k/h), the constrained network area is proposed to be covered by less aggressive solutions such as Westwood+ and NewReno. Whereas in considerably stable fading conditions, NewReno could bring a more balanced performance in the Start-up phase.

In regards to the backward movement pattern, the scenario at 120 km/h with stable fading conditions proposes CUBIC as the best candidate in all phases of the scenario. Nonetheless, under variable fading conditions, a change in the Start-up phase is recommended. With limited flexibility in the transport layer, BBR and NewReno have shown the best ability to scale in the mobility scenarios at 60 km/h and 300 km/h. Keeping in the scaling phase such CCAs, a different CCA could potentially bring overall performance gain while utilizing it in the Start-up phase and then switching to the selected candidates with the best scalability. Such proposals have been tested and Figure 6.13 depicts the comparison results in ECDF graphs.

Figure 6.13 proves that all scenarios that have been selected to work with the combination of CCAs report better results with the CCA switching method. Even though in some scenarios (e.g. forward movement at 60 km/h) the enhancements are not very significant, on average and overall the CCA switching technique demonstrates the ability to provide with a more balanced performance, adapted to the requirements from the application layer. In this regard, **once again it is shown that the implementation of such simplistic switching mechanism could benefit low latency scenarios, being considerably suitable for future 5G softwarized deployments**.

6.3.3.3 Delay based CCA selection metric with flexible transport layer under low latency

Both evaluations based on goodput and performance metric have demonstrated even under low latency conditions that the selected combinations of CCAs could improve the performance in a large number of the cases. Looking into the delay-based evaluation performance for 4G latencies and taking into account that in general the CCA switching mechanism has been unsuccessful, leading to achieve only similar or worse results that the ones obtained with single CCAs, it is crucial to determine whether the delay-based evaluation over low latency conditions prompts similar outcome or not. Table 6.7 depicts the evaluation of each network phase in each mobility scenario based on a weighted performance metric towards delay.

Regarding the CCA selection in Table 6.7 and only focused on the highlighted CCA switches, three important details need to be underlined:

- The combination of NewReno and BBR in forward movement schemes looks promising, not only due to the phase-wise results based on delay, but because in performance metric evaluation their presence has been important as well.
- Under low latency scenarios Westwood+ demonstrates to have a very significant impact while evaluating the suitability of CCAs for delay-sensitive applications. It reports good average results providing with a good candidate in both certain phases of forward movement patterns and in the Start-up phase of backward movement schemes.

After experimenting with the proposed delay-based CCA switching points, Figure 6.14 shows that under low latency conditions the selected combinations of CCAs are able

CCA	Forward at 60 km/h			Backward at 60 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.493066	0.446795	0.385195	0.383722	0.389045	0.428689
NewReno	0.491156	0.442214	0.373219	0.508341	0.377607	0.485950
BBR	0.532021	0.516794	0.405650	0.643308	0.505665	0.572276
Westwood+	0.565231	0.520792	0.523778	0.574969	0.462053	0.557967
Illinois	0.465040	0.409700	0.338620	0.413637	0.374087	0.363800
CCA	Forward at 120 km/h			Backward at 120 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
CUBIC	0.585520	0.407352	0.302067	0.450910	0.586674	0.565590
NewReno	0.632849	0.523055	0.414922	0.305750	0.471540	0.476703
BBR	0.490204	0.526370	0.426877	0.323457	0.313780	0.509598
Westwood+	0.549843	0.515622	0.380573	0.392795	0.434547	0.490980
Illinois	0.512439	0.425080	0.345400	0.368079	0.260921	0.335248
CCA	Forward at 300 km/h			Backward at 300 km/h		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
CUBIC	0.496265	0.480522	0.387878	0.412978	0.489973	-
NewReno	0.558254	0.510472	0.512038	0.464944	0.519911	-
BBR	0.732198	0.623331	0.508670	0.528186	0.537836	-
Westwood+	0.614074	0.466800	0.412867	0.604662	0.510420	-
Illinois	0.462019	0.383570	0.378451	0.460401	0.436021	-

Table 6.7: Delay-based performance metric in detected phases under low latency.

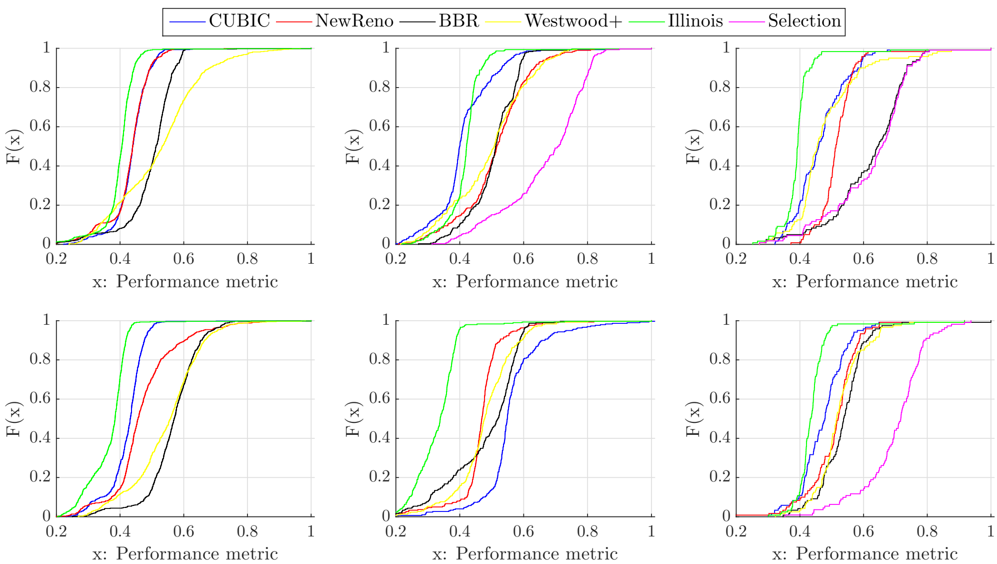


Figure 6.14: Delay-based comparison of 5 CCAs + selected CCA combination while moving at different speeds under low latency: Forward movement on top; Backward movement on the bottom.

to get similar or better results than the ones achieved by single CCAs. The best results have been in a combination of one CCA in the Start-up phase and the second one managing the rest of the transmission. This effect has been encountered in many occasions and highlight the necessity of the switching mechanism in order to provide with the suitability of certain CCAs for a specific network conditions and application requirements and input the next CCA for a more efficient performance than working by its own. This success contrasts with the previous evaluation with 4G latencies and proves to be important for future delay-sensitive deployments and applications QoS.

6.3.4 Conclusions in the intra-protocol selection of Option 2

We have evaluated the selected mobility scenarios under both 4G latencies and low latency that moves toward the potential delays in 5G networks, based on goodput, on the performance metric and on delay. Besides, we have carried out the study considering that the transport layer is flexible enough to allow in-flight or real-time CCA switches (Option 2). The analytical chapter has reported certain phases in the transmission that depending on the mobility scheme and network circumstances, the CCA requirements are different. Following those phases, we have selected the most appropriate CCAs for each situation. After picking the, in principle, most suitable CCA combination, we have performed actual experiments starting from the CCA switching points, utilizing the second CCA from such a point on and serving the information of the first CCA as the initialization information/configuration. Table 6.8 gathers the selected combinations. Four different colors are used in order to classify the obtained outcome: grey cells are mobility scenarios in which due to the phase-wise evaluation, the results have shown that the utilization of the most appropriate CCA is enough to make the most of the scenario. This is, grey cells represent mobility scenarios in which even with a limited flexibility in the transport layer, the achieved performance of the most suitable CCA is hardly improvable. Green cells show mobility scenarios where with a flexible transport layer and the application of a CCA switching technique, the criteria-based performance could be greatly enhanced (greater than the 5%). Yellow cells show the cases in which the CCA switching method has obtained similar or slightly better results that the ones gathered by the utilization of single CCA (equal results or improvements below the 5%). Finally, red cells represent the mobility schemes that have reported unsuccessful results with the CCA switching technique and therefore, the application of such procedure causes misperformance in comparison with simplistic CCA selection from the beginning of the transmission. In the left column "G" represents the goodput-based evaluation, "PM" the evaluation based on performance metric and "D" shows the delay-based assessments. Speed 1 represents the results at 60 km/h with EVA60, Speed 2 the results at 120 km/h with EVA120 and Speed 3 the results at 300 km/h with HST300.

The most important details in Table 6.8 are the following:

1. The great advances with CCA switching technique are achieved in both backward movement schemes with variable fading patterns or in forward movement with considerably stable fading conditions that target either balanced or delay-based performance. All greatly successful scenarios (with performance gains over the 5%) show that a CCA switch after the Start-up phase could help achieve a better criteria-based performance. Even though the most prominent CCA is used as a "Sustainer" in forward movement and as a "Scaling" CCA in backward movement, the utilization of a more appropriate CCA in Start-up phase is very beneficial and brings great gains.

/	Forward with Speed 1			Backward with Speed 1		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
G. - Low latency	NewReno	NewReno	CUBIC	NewReno	CUBIC	CUBIC
G. - 4G latencies	NewReno	Illinois	CUBIC	BBR		
PM - Low latency	BBR	BBR	Westwood+	NewReno	BBR	BBR
PM - 4G latencies	NewReno	NewReno	CUBIC	Illinois	CUBIC	CUBIC
D. - Low latency	Westwood+			BBR		
D. - 4G latencies	BBR	NewReno	CUBIC	Illinois	Westwood+	NewReno
/	Forward with Speed 2			Backward with Speed 2		
	Start-up	Sustainer	Constrain.	Start-up	Constrain.	Scaling
G. - Low latency	Illinois	Illinois	BBR	CUBIC		
G. - 4G latencies	BBR	BBR	Illinois	BBR		
PM - Low latency	NewReno	BBR	BBR	CUBIC		
PM - 4G latencies	CUBIC	NewReno	Illinois	Westwood+	BBR	BBR
D. - Low latency	NewReno	BBR	BBR	CUBIC		
D. - 4G latencies	Westwood+	NewReno	NewReno	Westwood+		
/	Forward with Speed 3			Backward with Speed 3		
	Start-up	Sustainer	Constrain.	Start-up	Scaling	-
G. - Low latency	Illinois			NewReno		-
G. - 4G latencies	CUBIC	CUBIC	Westwood+	BBR	Illinois	-
PM - Low latency	BBR	BBR	NewReno	BBR	NewReno	-
PM - 4G latencies	CUBIC			BBR	CUBIC	-
D. - Low latency	BBR	BBR	NewReno	Westwood+	BBR	-
D. - 4G latencies	CUBIC	CUBIC	NewReno	NewReno	Westwood+	-

Table 6.8: Wrap-up of selected CCAs with flexible transport layer that allows in-flight or real-time intra-protocol selection (Option 2).

2. In backward movement with considerably stable fading conditions, the selected CCA candidate is able to maintain a good performance throughout the whole transmission and the lack of flexibility in transport layer does not have any impact.

3. In forward movement schemes with variable fading conditions, the use of CCA switching technique is capable of providing with slight gains (performance gain but below the 5%). Actually, the improvements mostly come from the utilization of a second CCA to handle constrained network area. This modification and switch of CCAs gives slight enhancements that under higher capacities could be more noticeable and be able to greatly improve future communications.

4. CUBIC has demonstrated to successfully handle constrained network conditions in both decreasing quality movement with variable fading conditions and in increasing quality movement schemes.

5. The presence of BBR is of two types: 1) Goodput maximizer under 4G latencies; 2) Balanced and delay-sensitive behaviour under low latency conditions.

6. Westwood+ seems the best delay-sensitive option in backward movement scenarios to handle the constrained network conditions and even with certain ability to scale in the scenario at 300 km/h where the maximum capacities are low. Backward movement starts under relatively low capacities and the underperformance of Westwood+ (massive back-off application after Slow-Start) is less

significant. Therefore, Westwood+ is able to achieve considerably good rates, while it keeps the delay low.

7. A classic TCP solution as NewReno has shown great usability and suitability either as a support in the Start-up phase, to handle constrained area, in low latency scenarios or balanced and delay-sensitive 4G deployments. The wide presence of NewReno is one of the main foundations of the CCA switching technique that stands for the re-usability of CCAs in those network conditions and application requirements that fit better. NewReno proves that even classic TCP CCAs could be of a great value in current and future networks.

Out of 36 mobility use-cases, 11 have reported the best results only by appropriately selecting the best CCA candidate at the beginning of the transmission, 10 have shown performance improvements over the 5%, 13 have demonstrate improvements below the 5% and 2 have shown worse performance applying the intra-protocol selection mechanism. Overall, the intra-protocol CCA switching and selecting technique has shown a great response in the selected scenarios and it has demonstrated that current and more importantly, future communications, could safely take advantage of such a simplistic method to enhance the criteria-based performance.

7 Conclusions, contributions and future work

This thesis has analyzed the performance of TCP, in 4G scenarios and in low latency scenarios that moves towards the overall potential delays in 5G networks, under distinct MBB circumstances that mainly cover mobility scenarios as well as the flow start-up performance. The work has also evaluated, considering the flexibility of the transport layer, a solution based on the network status and the application requirements for an enhanced coexistence between distinct implementations of TCP (intra-protocol de-ossification) and MBB. To this end, several steps has been covered.

The **first** step has comprised the study and capabilities evaluation of different testbeds so as to determine and match each analytical parts with the most suitable deployment. After analyzing four different deployments, we have concluded that the controlled testbed could be suitable to confirm certain findings but it is not capable of providing the necessary test-field so as to perform realistic mobility measurements. However, after the addition of a controller that modifies the baseline SNR of the UE, the controlled testbed has been widely utilized to analyze and evaluate the mobility schemes.

The **second** step has been responsible for covering the detection of possible basic networking constraints in MBB that could prevent transport layer from performing at its full potential. In this regard, it has been suggested the inclusion of WS in the validation process of each measurements. Besides, we have highlighted the importance of queue size and its impact so as to understand the performance of different CCAs and be able to appropriately select the most suitable CCA according to the available capacities. Finally, we have underlined that the performance limitations are due to the CCA itself rather than due to lack of appropriate network capabilities, highlighting this way the importance of our proposal and the impact that may well have the selection of the most suitable CCA for precise network conditions.

In the **third** step, different CCAs and distinct traffic loads have been tested while the UE was static, serving as a first study of different CCAs responsiveness and detection of behavioural patterns. We have demonstrated that most CCAs, even delay-based implementations, are capable of reducing their throughput with a sudden available capacity decrease but delay-based variants struggle to adapt their pace to a bandwidth increase.

The **fourth** step has covered the performance study of Slow Start methods. We have suggested the avoidance of RTT as the unique value to take decisions in the server due to the detected RTT spikes in MBB. In order to experiment with realistic variability conditions due to the significant impact that could cause in the performance of certain Slow Start methods and be able to properly test the different Slow Start mechanisms in the wild, we have conducted a measurement campaign over the MONROE testing deployment. The results have shown that there exist performance differences depending on the operator and flow size not only among the CCAs but also considering a certain CCA and its suitability for precise conditions of network capabilities and flow type. In operators with a queue in the bottleneck long enough to allow overshooting events, on average BBR performs faster and more efficiently than other CCA candidates and could bring great advances to the mobile developments.

In the **fifth** step, we have covered the analysis of different simplified mobility patterns in order to understand the possible impact that moving to better or worse positions would have on transport protocol performance depending on previous network state, demonstrating that depending on the position of the UE while moving and speed, the network zone or cell zone where the UE is located could bring important differences in regards to the limitations and dependencies for TCP, being able to separate the logic of the intra-protocol selection framework based on the UE's area.

The **sixth** step has been devoted to analyze the speed and channel variability themselves and how TCP responds to it. The analysis of different speeds with similar network conditions has shown the dependencies regarding the TCP aggressiveness depending on the cell area. Besides, under very variable channel conditions, low latency mobility scenarios have reported little impact of the selected CCA or the mobility pattern in the final goodput performance. Whereas under 4G latencies, the tougher the speed scenario, the combination between speed and fading makes the breach within forward and backward proportionally bigger, being the breach greater in less aggressive TCP solutions such as NewReno. Other aggressive CCAs, such as CUBIC, are able to reduce the impact of lengthening the end-to-end latency. This part of the analysis has also detected and reported different phases in the mobility scenarios in which the requirements of the CCAs to appropriate perform are different. In this sense, we have presented the Start-up phase, the goodput sustainer phase and the constrained network area for forward movement schemes and Start-up phase, constrained network area and scaling phase for backward movement scenarios. These phases have been later utilized to divide the evaluation of CCAs for each scenario. The division is deployment-dependant and the length of each phase will highly depend on the resources and configuration. Therefore, our selected phases and their duration could provide with applicable results for similar deployments but it would require refurbishment for different testbeds and deployments.

Apart from that, this step in the analysis has also provided with the behavioural patterns of several CCAs and certain findings need to be remarked: 1) The ability of NewReno, CUBIC and Illinois to get full utilization of the available capacity in many mobility scenarios. 2) The underperformance of Hybrid Slow Start in precise delay variability circumstances, skipping the drastic ramp up very soon and leading CUBIC to achieve a poor performance. 3) Overall inadequacy of Westwood+ and CDG due to their weaknesses. Westwood+ due to its drastic CWND reduction after Slow-Start and CDG due to its main objective of reducing the delay, cause a poor radio resource utilization.

The **seventh** step has covered the evaluation of different CCAs in mobility scenarios under both 4G latencies and low latency and has decided the best candidate for each situation not only based on network conditions and application requirements, but also based on the flexibility of the transport layer. Even though the flexibility of the transport layer will increase with the time being and will allow switching between CCAs in-flight, this step in the thesis wanted to cover the limited flexibility present nowadays, providing with the ability to select the CCA at the beginning of the transmission. The evaluations have been based on goodput, delay and a performance metric, in order to give a wider view of the performance trade-off and avoid the simplistic performance reference to goodput rates.

Regarding this criteria-based evaluation, each CCA has shown its reusability and suitability for certain mobility schemes: 1) Illinois has shown an aggressive behaviour that is appropriate for application with goodput requirements in both movement patterns in 4G latency scenarios, but only appropriate for backward movement under low latency. In forward movement under low latency conditions, other CCAs increase their responsiveness, being able to improve the achieved outcome of the aggressiveness of Illinois; 2) BBR has proved to have a great goodput-based performance under 4G latencies. Nevertheless, it also induces a long delay and provokes a great number of transmitted packets. In contrast, in low latency scenarios BBR has a more controlled and balanced behaviour, achieving a sufficiently good performance in goodput but also preserving a low delay; 3) Westwood+ has demonstrated to be a good TCP alternative for delay-sensitive applications, not seeking for the maximum capacity, achieving reasonable rates but ensuring low delay; 4) NewReno has emerged as an eligible TCP implementation for delay-sensitive applications under 4G mobility environments. Apart from that, under low latency conditions, NewReno has been capable of achieving close to the maximum available capacity. Its performance in low latency scenarios has been very important, being specially significant in backward movement pattern under variable fading conditions with low achievable rates. The good results of NewReno are very important to sustain the reasoning of our CCA selection at the beginning of the transmission. NewReno proves that even classic TCP CCAs could be of a great value in current and future networks; 5) CUBIC has demonstrated a good performance in several circumstances: a) balanced performance in variable scenarios with low available bandwidth under 4G latencies; b) suitable for forward movements with big variability under low latency circumstances; c) ability to scale under low latency conditions.

The **eighth** and last step has comprised the evaluation of different CCAs in mobility scenario under 4G latencies and low latency conditions with a more flexible transport layer that allows switching the CCA in-flight. The evaluation has been performed again based on goodput, on the performance metric and on delay. The **sixth** step in the thesis has prompted different phases in the mobility schemes that require different CCA features for a proper execution. Such phases are the foundation for the segmentation in the evaluation. Following those phases, we have selected the most appropriate CCAs for each situation. After selecting the, in principle, most suitable CCA combinations, we have performed experiments to check the validity of our selections. Each experiment has been carried out starting from the CCA switching point, using the second CCA from that point on and serving the information of the first CCA as the initialization configuration. The results have shown that: 1) In backward movement with considerably stable fading conditions, the selected CCA candidate is able to maintain a good performance throughout the whole transmission. Therefore, a selection at the beginning of the transmission is more appropriate and the lack of flexibility in transport layer does not have any impact; 2) The CCA switching technique is more successful in both backward movement schemes with variable fading patterns or in forward movement with considerably stable fading conditions that target either balanced or delay-based performance. All these scenarios have demonstrated that a CCA switch after the Start-up phase could help achieve a better criteria-based performance. Although, the most notorious CCA is used as a "Sustainer" in forward movement and as a "Scaling" CCA in backward movement (longest phases with the greatest impact), the utilization of a more appropriate CCA in Start-up phase is very beneficial and brings great gains; 3) In forward movement schemes with variable fading conditions, the use of CCA switching technique is capable of providing with slight gains. Actually, the improvements mostly come from the utilization of a second CCA to handle constrained network area. This modification and switch of CCAs gives slight enhancements that under higher capacities could be more noticeable and be able to greatly improve 5G communications.

All in all, this thesis has demonstrated that current and future communications could take advantage of the intra-protocol CCA switching technique in order to enhance the criteria-based performance. Even if the in-flight switching method is not enabled, available or the operator requires a more conservative approach, by properly selecting at the beginning of the transmission the most adequate CCA that considers network conditions and well as application requirements, the obtained results and overall performance gain have been proven to be significantly important.

7.1 Thesis contributions

This thesis has contributed with its findings and proposals publishing a variety of conference papers, talks, book chapters and journal articles. This section will link each publication with the precise finding, evaluation of Performace aspect or part of the proposal.

Two publications have covered the **Findings 1** (WS constraints consideration), **2** (Intermediate queue size impact) and **3** (General performance constraint due to the selected CCA) inside the evaluation process of the **PA1.1**:

1.- **Eneko Atxutegi**, Fidel Liberal, Eduardo Saiz and Eva Ibarrola, "Why we still need standardized internet speed measurement mechanisms for end users", ITU Kaleidoscope: Trust in the Information Society (K-2015), December 2015, Barcelona, Spain. DOI: 10.1109/Kaleidoscope.2015.7383645

2.- **Eneko Atxutegi**, Fidel Liberal, Eduardo Saiz and Eva Ibarrola, "Towards standardized internet speed measurements for end users: current technical constraints", IEEE Communications Magazine, Volume: 54, Issue: 9, September 2016, pp 50-57. DOI: 10.1109/MCOM.2016.7565272

A single publication was responsible for **Finding 4** (Impact of variable available capacity into delay-based CCAs vs. loss-based CCAs) regarding the analysis of **Performance aspect 1.2**:

3.- Rémi Robert, **Eneko Atxutegi**, Åke Arvidsson, Fidel Liberal, Anna Brunstrom and Karl-johan Grinnemo, "Behaviour of common TCP variants over LTE", 2016 IEEE Global Communications Conference (Globecom 2016), 4-8 December 2016, Washington, USA. DOI: 10.1109/GLOCOM.2016.7841626.

Different publications and a talk were devoted to study the **PA2** and most of the **PA3** with the **Finding 5** (Impact of delay variability on Hybrid Slow Start), **6** (Efficient performance of BBR in general (depends on operator's network capabilities and flow type)), **7** (Impact of movement trajectory, speed and cell-position performance dependency) and **8** (Impact of considerably stable network conditions and cell-position performance dependency):

4.- **Eneko Atxutegi**, Fidel Liberal, Habtegebreil Kassaye Haile, Karl-Johan Grinnemo, Anna Brunström and Åke Arvidsson, "On the use of TCP BBR in cellular networks", IEEE Communications Magazine. Accepted for publication. In press. Expected to appear in March 2018 issue.

5.- **Eneko Atxutegi**, Fidel Liberal, Karl-johan Grinnemo, Anna Brunstrom, Åke Arvidsson and Rémi Robert, "TCP behaviour in LTE: impact of flow start-up and mobility", IFIP Wireless and Mobile Networking Conference - WMNC, 11-13 July 2016, Colmar, France. DOI: 10.1109/WMNC.2016.7543932

Talk - **Eneko Atxutegi** and Åke Arvidsson, "Transport protocol performance over cellular access: adaptability issues and future steps", COST ACROSS WG meeting in Bilbao, 13 October 2016.

6.- **Eneko Atxutegi**, Åke Arvidsson, Fidel Liberal, Karl-Johan Grinnemo and Anna Brunstrom, "TCP Performance over Current Cellular Access: A Comprehensive Analysis", In book: Autonomous Control for a Reliable Internet of Services: Methods, Models, Approaches, Techniques, Algorithms and Tools, Chapter: 7, Publisher: Springer, 2018. (In press)

A publication was devoted to present the experimentation stair with 4 LTE testbeds and show the bright and dark side of each deployment so as to realistically test TCP over MBB:

7.- **Eneko Atxutegi**, Jose Oscar Fajardo, Eva Ibarrola and Fidel Liberal, "Assessing Internet performance over mobile networks: from theory to practice", ITU Kaleidoscope 2016 - ICTs for a Sustainable World, 14-16 November 2016, Bangkok, Thailand. DOI: 10.1109/ITU-WT.2016.7805705

A single publication reported the **Finding 9** (Impact of different variabilities while moving and cell-position performance dependency) related to **PA3** and **PA4**:

8.- **Eneko Atxutegi**, Karl-Johan Grinnemo, Andoni Izurza, Åke Arvidsson, Fidel Liberal and Anna Brunstrom, "On the move with TCP in current and future mobile networks", The 8th International Conference on the Network of the Future (NoF), Nov 22-24 2017, London, UK.

Finally, a publication gathered the feasibility analysis and evaluation of appropriately selecting the most suitable CCA considering network conditions as well as application requirements:

9.- **Eneko Atxutegi**, Jose Oscar Fajardo and Fidel Liberal, "Transport protocol performance and impact on QoS while on the move in current and future low latency deployments", In InTechOpen book "Broadband Communications Networks - Recent Advances and Lessons from Practice", 2018. (In press)

Besides, during the thesis other contributions have been made in related topic of the field:

- Eduardo Saiz, Eva Ibarrola, **Eneko Atxutegi** and Fidel Liberal, "A unified framework of internet access speed measurements", ITU Kaleidoscope: Trust in the Information Society (K-2015), December 2015, Barcelona, Spain. DOI: 10.1109/Kaleidoscope.2015.7383644

- Alberto Carreras, Isabel M. Delgado-Luque, Francisco J. Martin-Vega, Mari Carmen Aguayo-Torres, Gerardo Gomez, J. Tomas Entrambasaguas, **Eneko Atxutegi**, Ruben Solozabal, Bego Blanco, Jose Oscar Fajardo and Fidel Liberal, "Impact of Front-Haul Delays in Non-Ideal Cloud Radio Access Networks", 2nd Global Conference on Wireless and Optical Communications (GCWOC), Sep 18-20 2017, Malaga, Spain.
Link: http://gcwoc17.ic.uma.es/wp-content/uploads/TS-A.pdf

- **Eneko Atxutegi**, Jose Oscar Fajardo, Eva Ibarrola and Fidel Liberal, "Experimental suitability evaluation of standardized QoS measurements over mobile broadband networks", 6th International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2017), Nov 28-30, 2017, Paris, France.

-Ruben Solozabal, Jose Oscar Fajardo, **Eneko Atxutegi**, Begoña Blanco and Fidel Liberal, "Exploitation of Mobile Edge Computing in 5G Distributed Mission-Critical Push-to-Talk Service Deployment", In major revision for the IEEE Access in Special Section: Mission Critical Public-Safety Communications: Architectures, Enabling Technologies, and Future Applications.

7.2 Future lines

Related to this thesis, three **future research lines** are envisioned. Firstly, **it is important to move the analysis and evaluation of mobility scenarios toward the real-world**. To that end and in order to extract meaningful and realistic conclusions, it is important to build a radio-aware information control. This is, the execution of TCP tests over certain roads and railways, needs the assistance of positioning metadata (to extract speed, mobility pattern and so on) as well as radio information such as CQI, MCS or the distribution of the transport block size. This information will allow a better understanding of the performance and will enable the creation of models that consider both mobility and network circumstances. These models will serve in the validation and benchmarking process of transport layers in general and TCP flavours in particular. This thesis is of utmost importance in the creation of such models because it provides with the knowledge regarding simplified mobility patterns and could help divide and understand complicated models and mobility schemes.

Secondly, **it is crucial to extent the application requirements that have been used in the current thesis**. In this regard, the idea is to evaluate precise flow patterns and build upon the performance under mobility QoE models. The models themselves could be further used to a better or more appropriate selection of each CCA for the precise circumstance in terms of network conditions and application requirements. This method will be very helpful for servers that are prone to serve similar content, being able to adjust the transport layer to their content needs in terms of performance and only requiring to be informed about the specific network conditions.

Finally, **the presented analysis and evaluation needs to be evolved with the addition of greater achievable rates in low latency scenarios**. Even though the presented results are applicable for the first stage of 5G, the long future carries higher capacities that would require special treatment in the TCP intra-protocol CCA selection.

Annex A - Detailed description of the analysis

A1 - On the use of RTT measurements and its implications with Hybrid Slow Start

Once it has been demonstrated that the behaviour of Hybrid Slow Start could lead the CCAs using that Slow Start method (i.e. CUBIC and CDG) to have a poor performance under high variability MBB circumstances, a deeper analysis is needed to determine the reasons behind the early triggering of the exit condition due to detection of a sudden delay increment. Considering that the abrupt variation of delay is causing a different performance among the Slow Start methods, the main target of explanation is to analyze the behaviour of the delay, measured as the RTT.

Only by observing the results of continuous ping measurements, we could infer some network patterns regarding the delays. Figure 7.1 shows the ping results of several MONROE nodes (selected due to their common operator -Orange-) throughout 9 complete days with measurements every second (blank spaces are non-operative time-slots or days for the specific node). Since the nodes are not transmitting anything else than the ICMP packets (no auto-induced delays), the resultant delays are provoked by cross-layer delays, resource sharing and scheduling techniques. Thus, giving insights of the possible latencies and network conditions variability that a transmission would have to face.

The delay happens to be very volatile. Figure 7.1 depicts many situations in which RTT values of less than 100 ms are alongside values about 1 s. Such variability makes the network unpredictable for the server and has the potential to vary the outcome of certain transmissions by means of abrupt delay detection.

In order to better understand the RTT variability and the impact that may well have in the exit condition of Hybrid Slow Start, it is important to study the behaviour or latencies with similar metrics to those present in Hybrid Slow Start. Basically, the

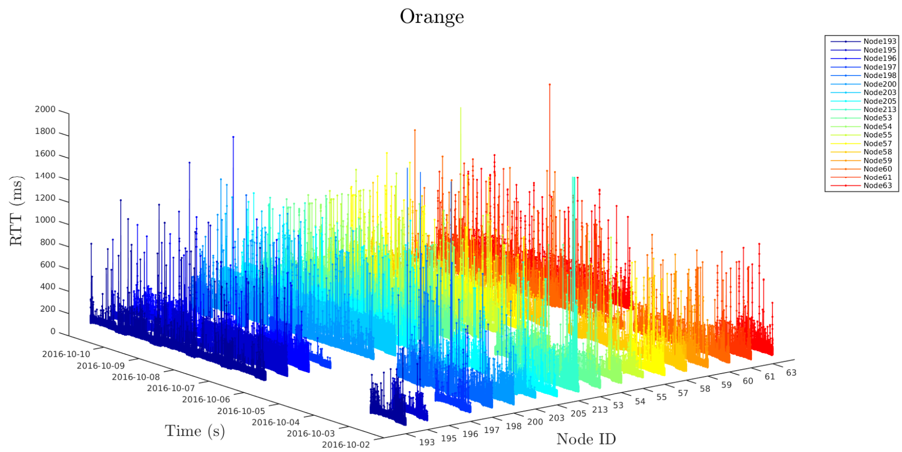


Figure 7.1: RTT samples of a selection of MONROE nodes throughout 9 days.

Hybrid Slow Start delay-based mechanism/algorithm compares the minimum RTT values over the first 8 samples (called *curr_rtt*) within a round trip with the minimum registered RTT from the beginning of the transmission or the latest RTO (named *delay_min*). Several experiments have been performed along different deployments in order to have delay-based results under different circumstances and testbeds in order to evaluate which are the conditions that have a greater impact into delay based CCAs or Hybrid Slow Start. The following figure (Figure 7.2) depicts representative traces of the aforementioned RTT comparison in commented deployments while using NewReno as the CCA (behaviour of Standard Slow Start) in order to gather RTT values in Slow-Start and post-process with the detection mechanism of Hybrid Slow Start.

In short, the following use-cases are represented:

- Simulated environment (see Subsection 4.3.1): Backward and forward movement at 60km/h and 200km/h.
- Emulated testbed (see Subsection 4.3.2): Backward and forward movement at 60km/h, 120km/h and 300km/h.
- Controlled deployment (see Subsection 4.3.3): Backward and forward movement following 3 different mobility patterns.
- Real-world deployment (see Subsection 4.3.4): Different static and widespread MONROE nodes.

Figure 7.2 shows the RTT comparisons (*curr_rtt* with *delay_min*) in percentage difference. Besides, the Y axis has been formatted as logarithmic so as to allow in the same graph large and small differences.

The graphs illustrate how easy the RTT of the transmissions suffers from initial burst in the Standard Slow Start with values over 100% comparing with the so-called *delay_min*. It is also clear how different the results are for different deployments and

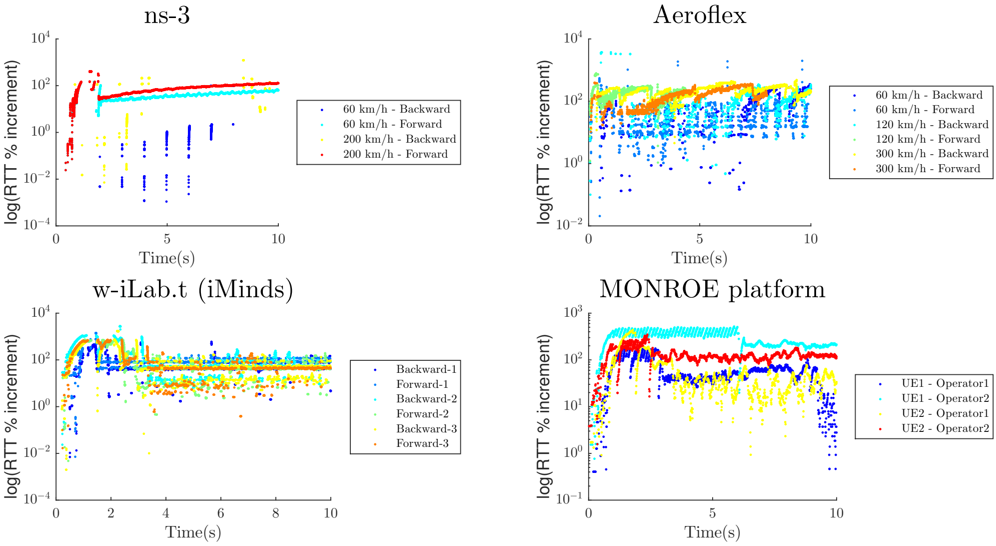


Figure 7.2: RTT% increment along 4 deployments with NewReno.

use-cases. The figure does not aim to thoroughly compare among the subplots, but to present the behaviour of the delay in different circumstances in order to cover possible outcomes depending on the testbed and give an overall picture.

The variability not only happens in a "long term". Sample by sample the RTT is very volatile and Figure 7.3 presents the RTT variability comparing each RTT sample with the previous one.

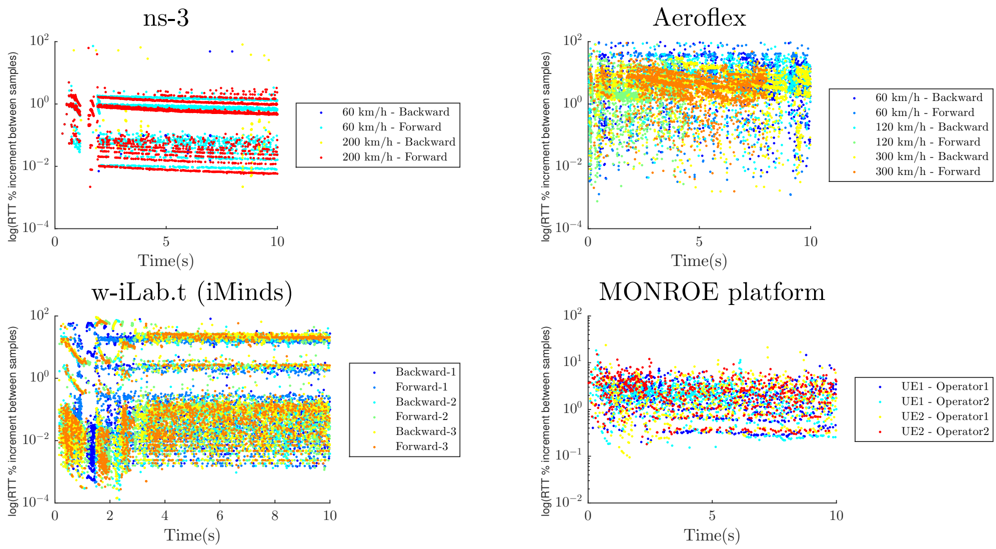


Figure 7.3: RTT% increment sample by sample along 4 deployments with NewReno.

It is noticeable that in most cases we come across values up to the 100% RTT variability between consecutive samples. Since it is represented in a logarithmic way, no negative values are depicted. Even though the figures are not intended to explain differences amongst deployments, some information should be given to help understand the reported results.

- ns-3: The lack of air communications and the modelling and simplification of them make simulated environment very robust to sudden changes. Even in backward movement and under high speeds, in which the Slow Start phase takes place in very hard radio conditions, the system is capable of continuing the transmission. All presented examples for ns-3 prompt a low sample by sample RTT variability.
- Aeroflex: Since the testbed's baseline RTT is quite low (about 12-25ms), the transmission is able to achieve maximum capacity within few RTTs. Due to the aforementioned low baseline RTT, there exist RTT changes of up to 100% sample by sample.
- W.iLab-t: Considering that the whole LTE system runs with real equipment, the scheduling, queueing and the rest of the features that could have an impact in delay are realistic and represent more clearly what would happen in live scenarios. It represents a very variable deployment in terms of RTT, in general terms and sample by sample.
- MONROE platform: The most network-status-dependant deployment among the four. The RTT variability between consecutive samples is not very high considering percentages. However it has to be considered that these experiments are under medium-high baseline RTTs of about 60-110ms. Therefore, the low percentage increment is understandable. In addition, after the end of the Standard Slow Start, the TCP's saw-tooth does not vary the RTT considerably.

Once the different RTT patterns have been observed, it is important to know how Hybrid Slow Start handles such network situation in current implementation and know if those situations can result in occasional RTT spikes that would cause a premature exit, resulting in low throughput or low radio resource utilization. The following definition of parameters establishes different boundaries for the method. a) 8 are the number of samples used to compare the minimum RTT (called *curr_rtt*) within a round trip with the minimum registered RTT from the beginning of the transmission or the latest RTO (named *delay_min*); b) A delay threshold is defined between 4ms and 16ms. The values that are going to clamp in this threshold are portions of *delay_min*.

```
#define HYSTART_MIN_SAMPLES 8
#define HYSTART_DELAY_MIN (4U«3)
#define HYSTART_DELAY_MAX (16U«3)
#define HYSTART_DELAY_THRESH(x)
        clamp(x, HYSTART_DELAY_MIN, HYSTART_DELAY_MAX)
```

How small or big the portion is will be only determined in the following step. As it is now, if a delay increment exceeds the 12.5% of the *delay_min*, the delay increment is enough to be detected as a significant delay increment and therefore, the exit condition is met.

```
if (ca->curr_rtt > ca->delay_min + HYSTART_DELAY_THRESH(ca->delay_min » 3))
                    ca->found  |= HYSTART_DELAY;
```

Considering the aforementioned volatility of current cellular networks in which RTT increments up to 100% have been shown, the fixed 12.5% seems not enough or at least very sensitive against occasional bursts. Taking into account the measured impact that Hybrid Slow Start has shown in the deployments of W-iLab-t and MONROE, a closer look is desirable. Figure 7.4 reflects the comparative performance of CUBIC using Hybrid Slow Start and NewReno using Standard Slow start over the aforementioned deployments during the first three seconds of the transmission, in which the distinct behaviour is more remarkable.

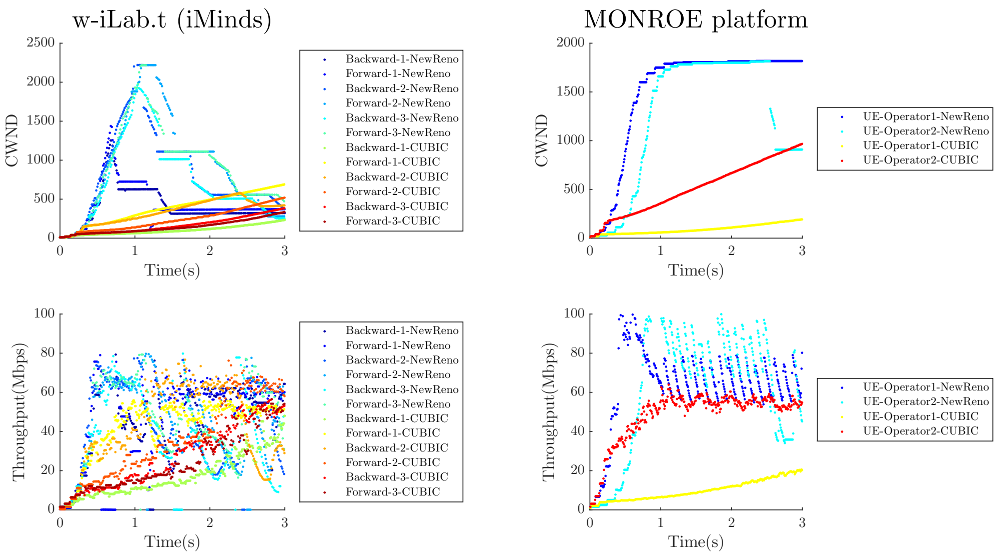


Figure 7.4: CWND and throughput over W-iLab-t and MONROE.

It is clear that NewReno (using Standard Slow Start) outperforms CUBIC (using Hybrid Slow Start) in this Slow Start comparison. It is also worth-mentioning those cases in which the Hybrid Slow Start mechanism of CUBIC is capable of exiting later than the average, causing a benefit of using Hybrid Slow Start. For instance, the yellow trace in w.iLab-t and the red one in MONROE are the biggest examples of such benefit. Yet, they are corner-examples and far away from the overall performance of Hybrid Slow Start.

Looking into average results, it is clear that almost every single burst is capable of forcing Hybrid Slow Start to skip the fast ramp-up very early, provoking an under-utilization of the network or the assigned resources to a certain UE. Even thought the idea of avoiding the last harmful rounds of a loss-based CCA is sensible, it does not seem to appropriately work under some delay variability circumstances in mobile networks. From the main idea of the Hybrid Slow Start proposal [59], we can extract part of the tentative reasoning: "*Since packets arrive at the beginning of each train do not suffer from queuing caused by packet bursts, these samples return more accurate estimations of persistent queuing delays.*". The problem in LTE is that ACKs are queued in the eNodeB and served in slots. This way, in the same slot is feasible to send packets that have been queued in different bursts and therefore, they would have different RTT delays. Avoiding this conceptual mismatch between wired networks and mobile networks, the variability of the cellular network itself is capable of provoking huge delay "jumps" that force Hybrid Slow Start to prematurely exit the fast ramp-up. Besides, it has been proven that, the more we move an experiment to the evaluation "in the wild", the more we notice the effect of Hybrid Slow Start. Therefore, the detected performance of Hybrid Slow Start is not appropriate for its utilization in the intra-protocol selection framework over variable MBB circumstances while the transmission is short.

Regarding the explained behavioural pattern of Hybrid Slow Start and its internal mechanism, a great amount of research efforts have been dedicated to solve the issues. Years before, regarding the requirements of fixed networks and being the research community aware of the need for delay adaptability, several patches have been proposed in the Linux kernel. The first one [122] aimed at making the delay threshold of HyStart less sensitive to sudden delay variations. The second one [123] was intended to refine the HyStart (Hybrid Slow Start) delay so as to establish the aforementioned 12.5% of the *delay_min*. Both were accepted and still running in the kernel but do not properly fit in mobile networks either. Other suggestions, such as [124] proposed modifying that 12.5% into 200%, but it was never considered to be pushed to the master branch of Linux. Even with such modification, there is no clear prospect about the benefits of modifying the delay sensitivity for Hybrid Slow Start in mobile networks due to the massive and unpredictable delay variations.

A2 - Detailed impact of movement pattern

A2.1 - Detailed study of movement patterns with 4G latencies

This subsection of the Annex aims at explaining in detail and step-by-step the results obtained for NewReno and CUBIC in the study of distinct movement patterns under 4G latencies in order to decide over their suitability for each use-case.

As a first step, it is more desirable to present the results for NewReno due to its simplicity comparing with CUBIC. Figure 7.5 depicts the behaviour of NewReno for each mobility pattern using realistic 4G baseline RTTs. On the left side of Figure 7.5 it is represented the achieved goodput in forward movement, while on the left, the backward movement is presented for the same metric. Both outcomes are shown in a statistical way, with medians and 95% confidence intervals. Below, there is a comparison of the median goodput in forward pattern in contrast with the median goodput result for reverse backward evolution. The Subfigure tries to contrast the performance of NewReno in the same network conditions, therefore it could be seen as the same distance of the UE from the eNodeB, for different movement patterns and study how could the previous network situation has an effect on the final outcome.

Both movement patterns report a completely distinct behaviour along the experiments. The figure shows that both patterns make a difference between each other and the results bring noticeable performance gap among them. In such scenario, two important things could be highlighted:

- Mostly **after the Slow Start phase, NewReno suffers to sustain the goodput** close to available capacity due to the constraints in terms of longer RTTs, the significant CWND reduction while recovering from losses, together with a poor scalability.
- **Mitigation of the queueing effect under poor radio conditions** under poor radio conditions. It does not matter whether the CCA has been able to put packets in-flight beforehand (case of forward movement) or not because the target capacity is low and easily achievable. Therefore, under low channel quality conditions, there is no requirement of CCA scalability.

It is of utmost importance to clarify whether the same network conditions affect differently the performance of distinct types of CCAs such as NewReno and CUBIC as a Proof of Concept of the intra-protocol selection framework on the move. Considering the features of CUBIC, it is important to analyze whether the non-synchronization of CUBIC with RTTs may well be able to prompt a different outcome in scenarios with traditional and very low latencies. Besides, in NewReno, we have detected that the backward movement pattern is not able to achieve the same capacities as the forward movement one. In these lines, it is important to check the performance of CUBIC with distinct mobility patterns and certify if the incapability of ramping-up in backward movement is also present. These findings would clarify whether each CCA behaves similarly in different network conditions or if certain CCA is more suitable for precise

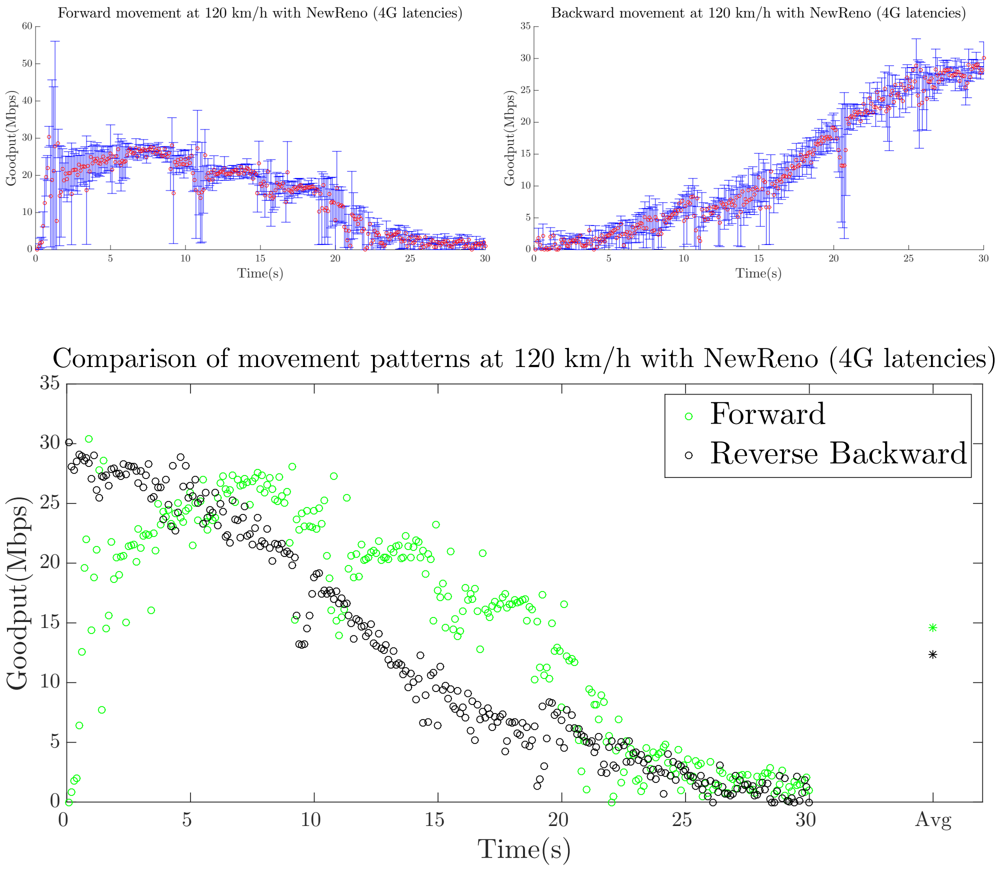


Figure 7.5: Comparison of NewReno's movement patterns at 120 km/h (4G latencies).

conditions over the rest of the candidates.

At first sight the result in Figure 7.6 looks similar to NewReno's (see the comparison as well in Figure 7.7). However, a couple of distinctions can be identified. After the loss event caused by Slow Start, the combined effect of a less strict back-off, the aggressiveness while recovering from losses and forward tendency with the radio conditions going worse, gives CUBIC the ability to sustain a goodput close to maximum capacity. NewReno it is not capable of accomplishing such a task due to its need for longer time to recover the dropped packets. CUBIC confirms that under realistic 4G RTT values the backward schemes underperform due to the constraints while ramping-up. So for both CCAs, under 4G delay circumstances for a certain UE position, it is more beneficial for the goodput performance to come from better radio conditions and have already some packets in-flight (forward movement or decreasing quality movement). Therefore, **the backward movement leaves greater improvement space for the intra-protocol selection framework** due to the rate difference between the available one and the achieved one. It demonstrates that it is better, from the goodput's point of view, to self-inflict delay with packets' overfeeding and

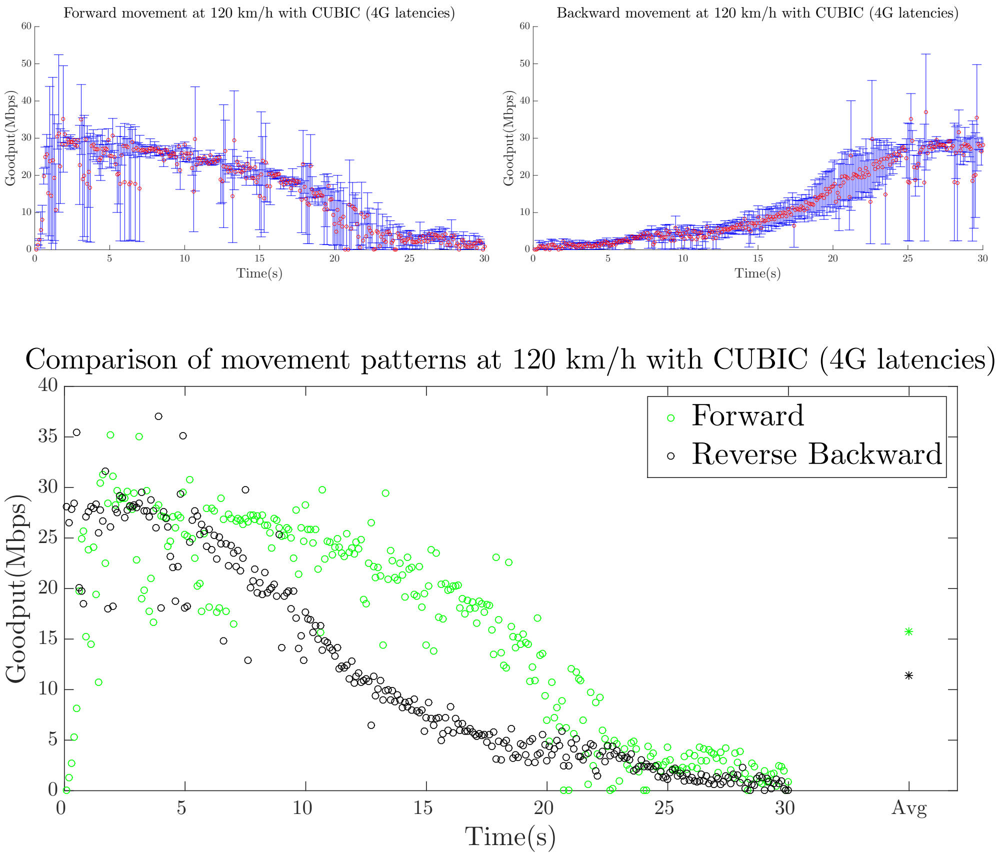


Figure 7.6: Comparison of CUBIC's movement patterns at 120 km/h (4G latencies).

recover from occasional losses due to TCP's capacity greediness forward movement or decreasing quality movement) rather than raise from worse positions and have an instant packet disposal backward movement or increasing quality movement). Thus, **it is harder to achieve a big performance enhancement by switching the utilized CCA in the forward movement**.

See Figure 7.8 to notice the difference of some experiment samples while ramping-up. In order to explain the reasons behind this effect, a recap of CUBIC is required. According to the cubical equation of CUBIC, three different stages are defined. Firstly, after a multiplicative decrease the CWND ramps-up very quickly following a concave shape. Secondly, in the transition of achieving *cwnd_max* (CWND recorded before the last window reduction), it slows down having an increment close to zero. Thirdly and once the *cwnd_max* has been overtaken, the CWND will accelerate its growth in a convex way until a new loss event happens. Quoting [113], "*CUBIC increases the window to (or its vicinity of) Wmax (CWND_max) very quickly and then holds the window there for a long time. This keeps the scalability of the protocol high, while keeping the epoch long and utilization high.*".

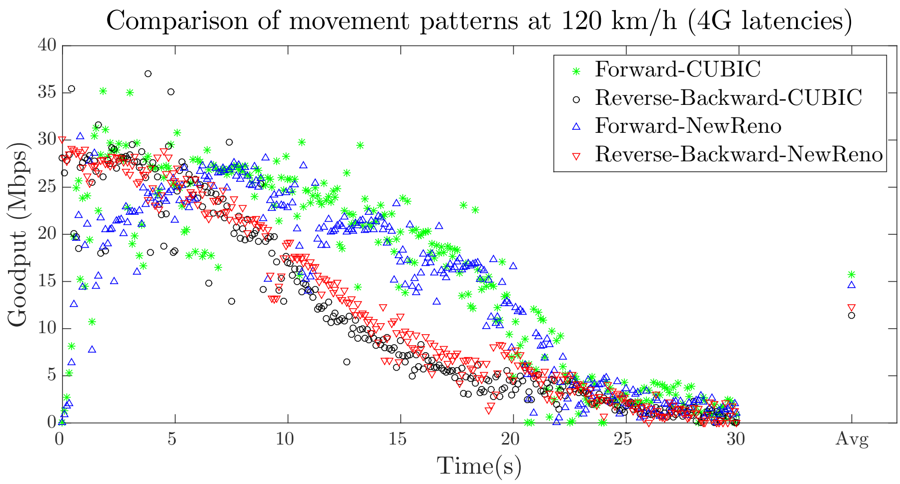


Figure 7.7: Comparison Movement patterns of NewReno and CUBIC at 120 km/h (4G latencies).

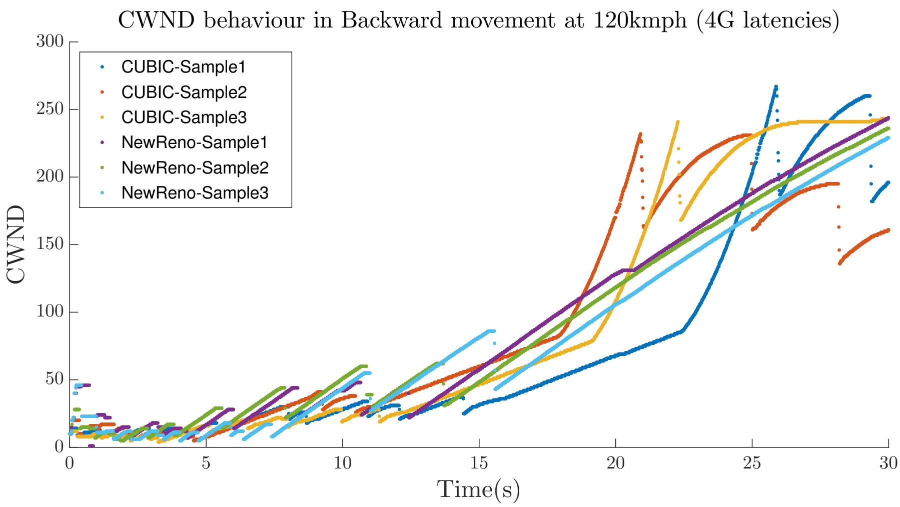


Figure 7.8: Comparison of CWND behaviour in backward movement at 120 km/h (4G latencies).

CUBIC is configured to scale with a concave shape, slowing down until a precise CWND value is achieved (the explained *cwnd_max*). In backward movement, the capacity constantly increments and therefore, the CWND will need to ramp-up always higher than expected by the *cwnd_max* value. That being said, in order to increment the CWND to positions greater than *cwnd_max*, CUBIC needs to pass through the second stage of congestion avoidance phase in which the increment is almost zero, wasting a valuable time in comparison with NewReno (see CWND evolution of CUBIC samples in Figure 7.8).

In order to understand the differences between NewReno and CUBIC, it is required to dig a little bit in the traces and look into those effects that are clear representations of harmful events for TCP. One of these examples is the reception of a duplicated ACK (so-called DUPACK).

When a sender sends a packet, it attaches the sequence number to it. The receiver responds with an ACK with the same sequence number so as to know the sender that the receiver has correctly received that packet. Since TCP is sequential, the receiver will be continuously expecting a certain sequence number in the arriving packet. If the TCP segment contains a sequence number higher than the expected one, then the receiver replies with an ACK referring to the packet it is waiting for. Therefore, the ACK is a duplicate of a previous ACK and tells the sender that at least one TCP segment is missing or it has been dropped. For that reason, the number of mean DUPACKs could give an insight regarding the packets that have been dropped due to excessive queueing and the metric will help understand the performance differences, even modest, between NewReno and CUBIC in forward and backward patterns.

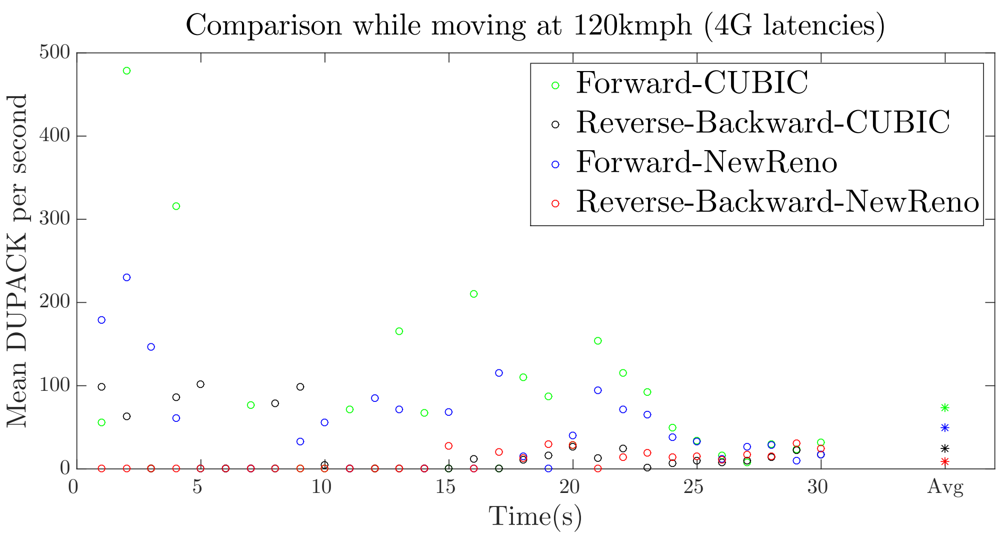


Figure 7.9: Comparison of DUPACK in movement tendencies of NewReno and CUBIC at 120 km/h (4G latencies).

Figure 7.9, represents a DUPACK study in order to help understand the analyzed results and be able to compare the scenarios from another point of view. In the forward movement, the obtained results show that based on the DUPACK events CUBIC either receives more DUPACK packets for actual retransmission or suffers more retransmission events. Even with that, in comparison with NewReno, CUBIC is able to more efficiently maintain a rate closer to the available capacity. In contrast, in backward movement the results are similar. Nevertheless, the evolution of NewReno's CWND keeps the injection rate closer to the available capacity. Besides, at the end of the transmission CUBIC self-inflicts losses (see CWND in Figure 7.8 and DUPACK events in Figure 7.9) and needs to pass through the consequent recovery process that makes the average goodput performance more unstable. Thus, the goodput results among CCAs are alike but self-caused effects make CUBIC perform slightly poorer.

A2.2 - Detailed study of movement patterns with low latency

This subsection of the Annex aims at explaining in detail the results obtained for NewReno and CUBIC in the study of distinct movement patterns under low latencies that move towards the potential delays in 5G.

Under low latency, the Figure 7.10 depicts the behaviour of NewReno for each mobility pattern following the same look-and-feel explained in Annex A2.1. As shown in the figure and mainly due the stability of fading pattern, both movement patterns report a very stable behaviour in terms of goodput. All red dots, representing median values, follow an almost straight shape and denote a great capability of achieving maximum radio capacity. Looking at the comparison between median results, we come across an important detail. Even though **both scenarios prompt a similar re-**

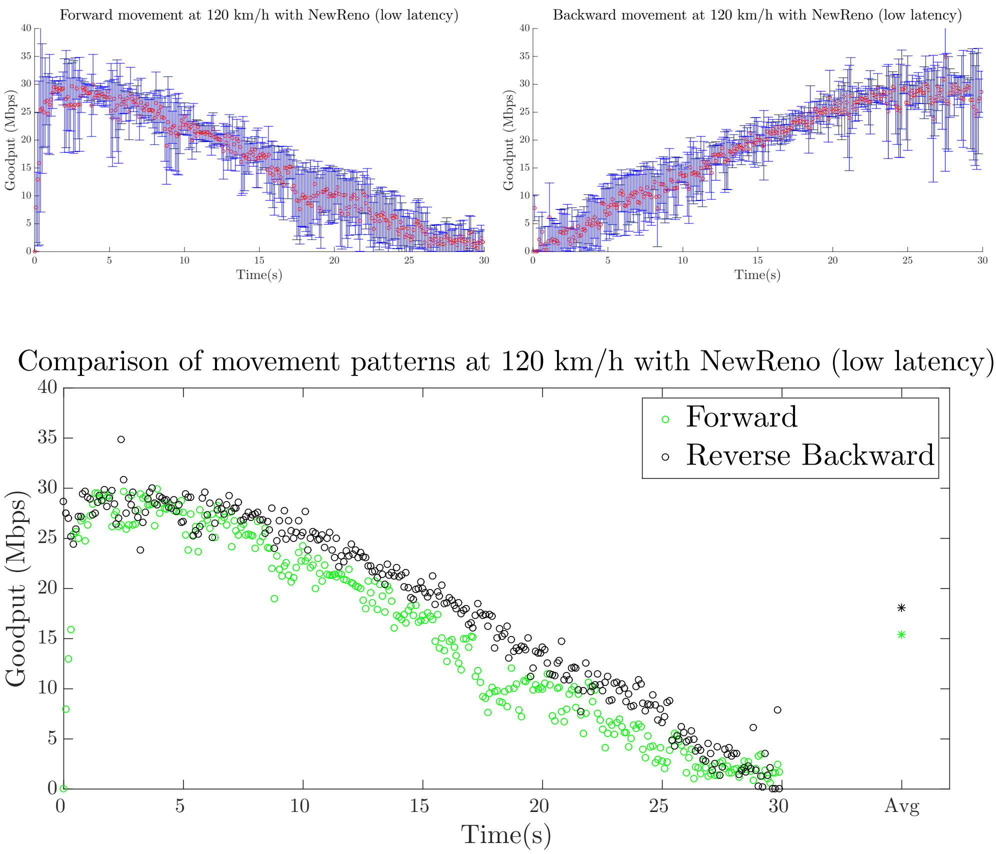


Figure 7.10: Comparison of NewReno's movement patterns at 120 km/h.

sult, backward movement outperforms forward movement. This means that **the impact of the mobility scenario itself is very weak in regards to the performance of a certain CCA**. Considering this information and the reported results in Annex A2.1, the question is to clarify whether CUBIC performs in the same way or differs from the explained results of NewReno. In principle, there are two features that could make a difference: the non-synchronization of CWND management with RTTs and the cubical congestion avoidance phase. The Figure 7.11 depicts the resultant goodput graphs for CUBIC.

At a glance the outcome looks very similar to the one obtained with NewReno and the backward movement still remains with better goodput performance than the forward movement. So for both CCAs (loss-based classic AIMD protocol -NewReno- and loss-based cubical implementation -CUBIC-), under low delay circumstances and applying a reasonably stable fading, for a certain UE position it is more beneficial for goodput performance to move from worse to better radio conditions (see comparison in Figure 7.12). It seems illogical, but the low latency helps reducing TCP's ramp-up periods and therefore allows TCP to perform close to the maximum capacity. Since

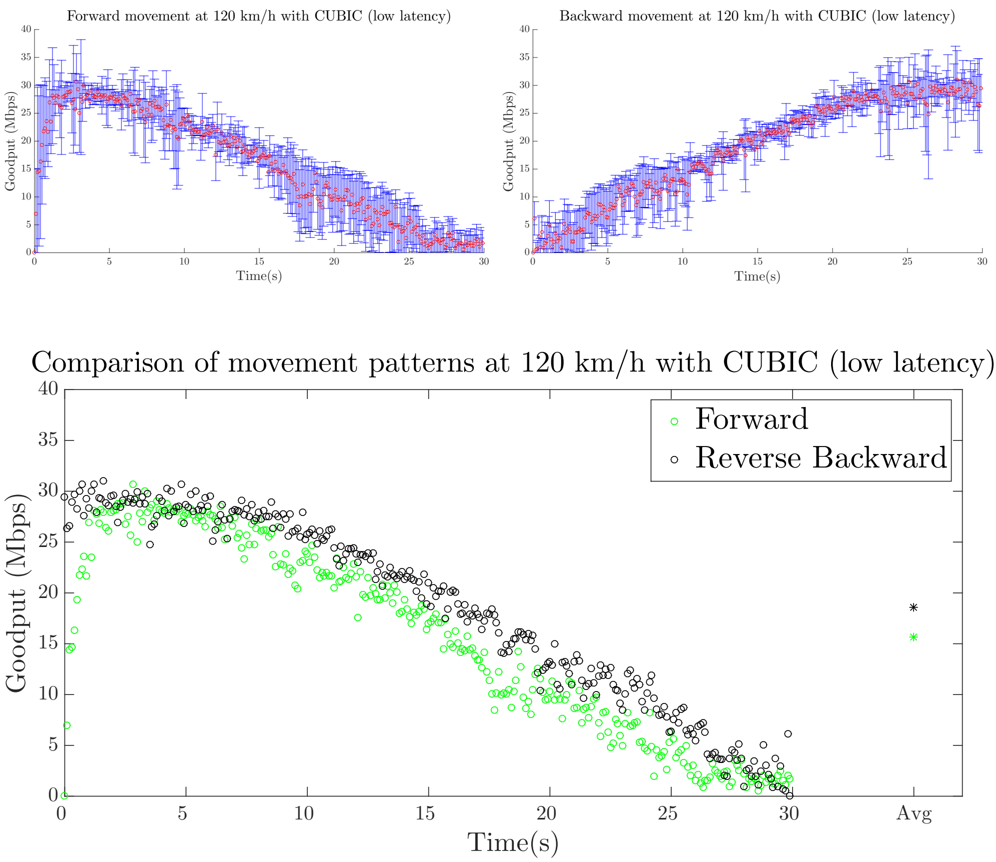


Figure 7.11: Comparison of CUBIC's movement patterns at 120 km/h.

TCP is always looking for more capacity, it tends to overfeed the bottleneck. For that reason, when backward movement constantly offers a greater maximum capacity, it cushions such overfeeding, utilizing extra packets to respond to the new and greedier capacity demand. In contrast, the forward movement suffers the contrary, the overfeeding packets are stored more and more with the evolution of the maximum achievable capacity to worse positions, until the bottleneck queue is full and drops some of the incoming packets, forcing the CCA to recover from those losses. However, the recovery from drop packets is quick due to the low baseline latency, helping again to achieve remarkable goodput performance.

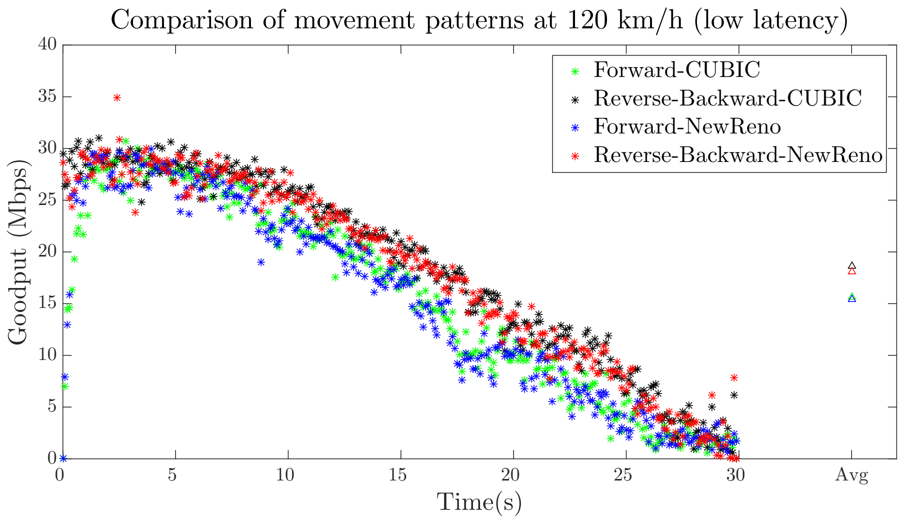


Figure 7.12: Comparison of the movement tendencies of NewReno and CUBIC at 120 km/h.

In order to understand the differences between NewReno and CUBIC, it is required to dig a little bit in the traces and look into specific TCP events. One of these examples is the reception of DUPACK due to the relation with actual TCP retransmissions.

The results in Figure 7.13 show two important aspects of the comparison. While in forward movement CUBIC has more DUPACK on average due to its aggressiveness seeking for maximum capacity, in backward movement such aspect of CUBIC does not seem to be equal or at least the impact is very distinct. CUBIC while moving backward gets the least DUPACK by far, getting time-slots in which tends to zero. This result reveals one of the reasons for CUBIC to win in the comparison of Figure 5.18. Small number of DUPACK does not necessarily mean a great goodput performance. However, in our low latency scenario in which every experiment is very close to the outcome reported by the rest, lowest DUPACKs on average is directly mapped to better performance due to the avoidance of loss events.

As explained with 4G latencies, CUBIC suffers to ramp-up to higher capacities once it presumes it has achieved the maximum capacity. It stores the CWND in which the loss events occurred and the ramp-up ability is drastically reduced around such a CWND value. In an increasing quality movement scenario, this effect provokes

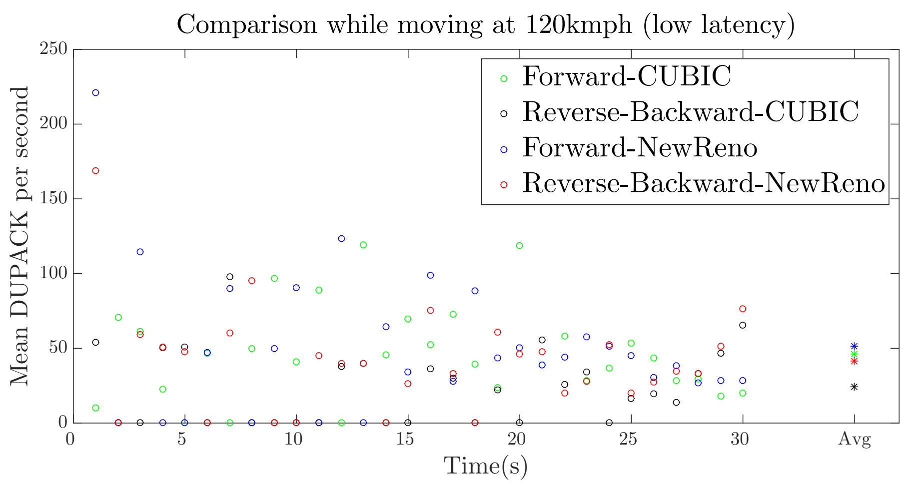


Figure 7.13: Comparison of DUPACK in movement tendencies of NewReno and CUBIC at 120 km/h.

longer epoch times of CUBIC in comparison with NewReno, wasting radio resources. For that reason NewReno ramps faster than CUBIC in backward movement under low latency circumstances as well, getting similar goodput performance but provoking a greater number of DUPACKs. Figure 7.14 shows an example of three different experiments of CUBIC and NewReno in backward pattern. It is clear that both CCAs report a slight difference in relation to epoch times.

So far, in **simple low latency conditions and focusing on the final goodput CUBIC has shown a good enough scalability and performance in order to beat NewReno.**

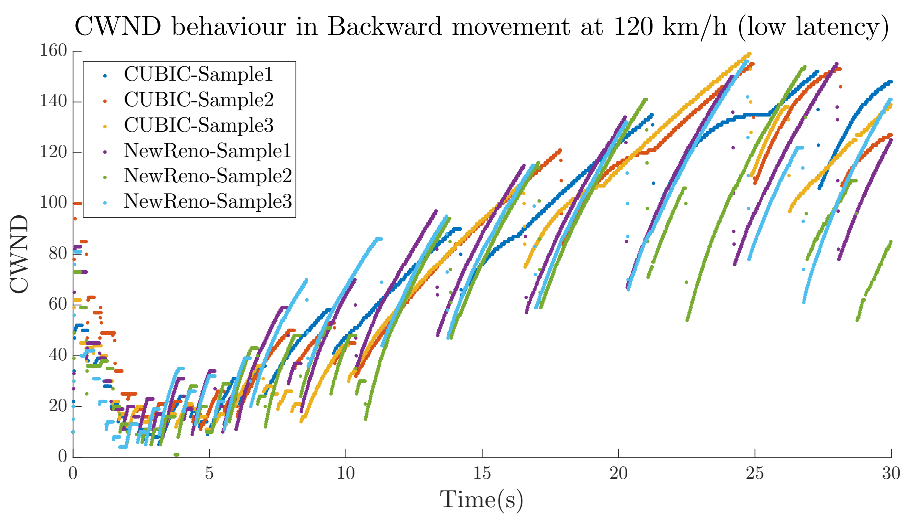


Figure 7.14: Comparison of CWND behaviour in backward movement at 120 km/h.

However, the performance gap is almost not noticeable, being possible to select a more conservative CCA (NewReno) and not noticing big difference in the final performance. Aligned with these results **under low latency conditions, only more complex performance metrics (considering circular dependencies of TCP) or greater achievable data rates would increase the usability of the intra-protocol selection framework**.

A3 - Demonstration of similar network conditions to individually analyze the impact of speed

Regarding the mobility scenarios in Section 5.5.1 that study the impact of faster UEs in similar network conditions, this section in the Annex demonstrates that such a network circumstances are indeed comparable and therefore the speed could be appropriately study in an isolated way. In the commented mobility cases, 3GPP-compliant (3GPP TS 36.104) EVA60 was applied for both speeds. Nevertheless, the use of the same fading pattern does not necessarily mean that the MCS assignment and variability are equal for both speed scenarios. In this regard, this section analyzes the MCS assignment for both scenarios. Figure 7.15 depicts the statistical assignment of MCS and MCS variability for both speeds. The graph contains four parts: 1) On the top left, a comparison of MCS assignments in percentage bar style; 2) On the bottom left, a comparison of MCS variability in the same way; 3&4) On the right, two pictures regarding the MCS assignment and variability as ECDF so as to be able to give more information about the total statistical distribution of values.

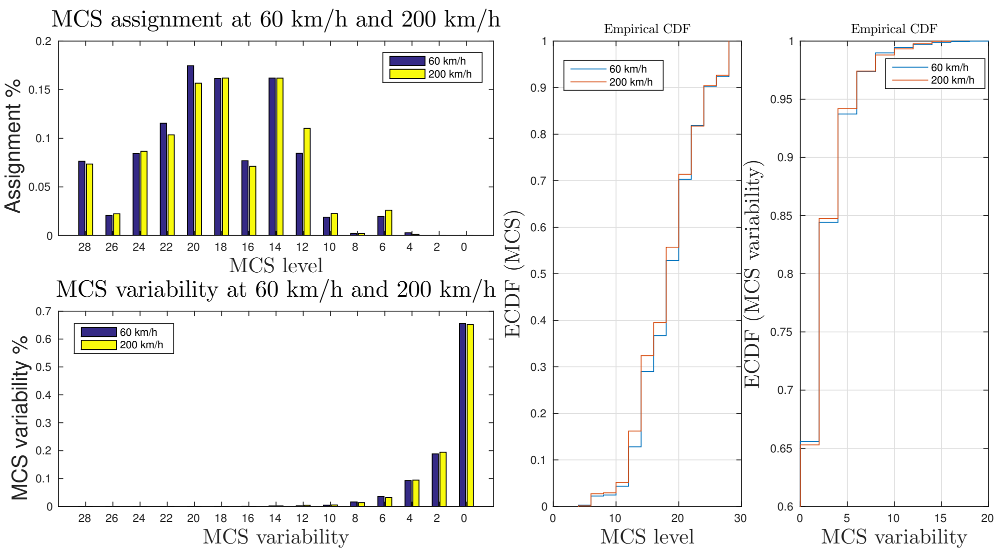


Figure 7.15: MCS assignments and variability at different speeds.

According to the results, in general terms, there is an analogous MCS assignment.

However, the distribution of values is different for both scenarios. For instance, the MCS 6 assignment is more likely to happen in 200 km/h experiments and on the other hand, MCS 4 is more frequent in 60 km/h. This finding reveals that even using the same fading trace, the scenario at 60 km/m has fades that achieve lower peaks. Besides, the ECDF of the MCS assignment shows that in general the scenario at 200 km/h has lower values. The gap between them is not remarkable but it clearly prompts harder fades in relation to the percentage meaning within the complete trace. Nonetheless, looking at the variability, we perceive differences in terms of "bigger jumps" at 60 km/h. Summing up, in general under the use of the same fading trace the experiments under 200 km/h circumstances have more fades, but under 60 km/h are slightly more drastic.

Apart from the analysis of MCS assignment and variability, it is interesting to look into the numeric variability sample-by-sample as a verification of changes in short time-gaps. This is, from a concrete sample number we need to analyze how probable is to change the MCS value in the following samples. Figure 7.16 shows the probabilistic MCS assignment of following three samples for both scenarios. The metric consists of checking the MCS value of a certain sample (n sample) and recording the next three MCS assignments (n+1, n+2 and n+3). The probabilistic MCS changes are depicted as a contour plot. The Figure 7.16 only represents those MCS changes that are statistically significant.

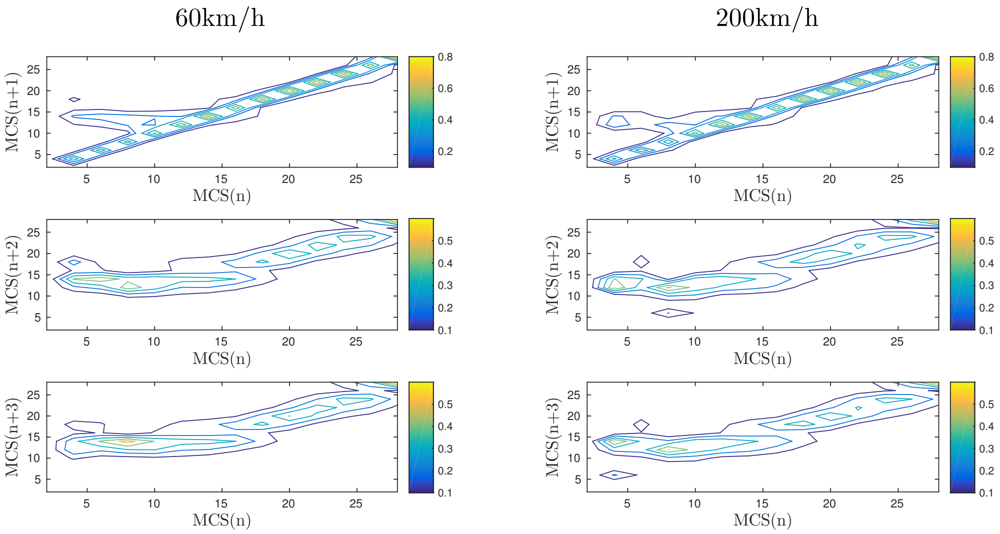


Figure 7.16: MCS level change between samples at different speeds.

First row of figures clearly show that the tendency between consecutive samples is to keep the value. Thus, being more times in which the MCS value is maintained rather than is changed. Another important fact is that starting from low values, some important number of following samples "jump" to greater values, showing how and under which values happen the recovery of average values from fadings. Besides, looking into the second row, we could detect that it is greatly probable to return in a couple of samples from very low values to values over 12. Regarding the specific behaviour of different speeds, in 60 km/h scenarios not only some fades achieve deeper values but also quicker. A simple comparison of the aforementioned effects is represented for MCS 8 while analyzing the third row, in which the returning point for MCS has been barely established in MCS 12 for 200 km/h together with some significant MCS changes between consecutive MCS values, while, for 60 km/h, the most significant value is over 14.

The presented findings prompt the huge similarities between both scenarios from the point of view of the assigned modulation and therefore the maximum achievable rate. This way, it demonstrates that the analysis regarding different speeds has been carried out under similar network conditions with little impact of the portion of the fading pattern that has been utilized. However, it still exists some doubts in relation to the performance dependency detected in the so-called "queue draining zone" and where TCP is not able to follow the cellular network's pace. It is important to clarify whether in that precise area both speed scenarios suffer differently leading to a performance gap or not. The study tries to find out if the explained impact of a faster UE is only due to a more rapid movement or also due to the fading trace impact in such an area (network conditions or UE position).

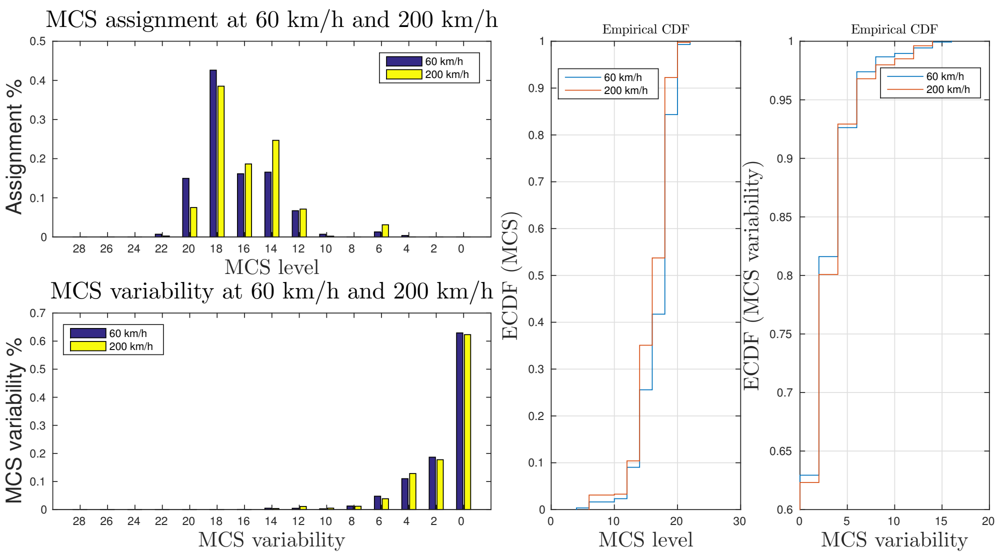


Figure 7.17: MCS assignments and variability at different speeds (Queue draining portion of time).

Figure 7.17 shows a very similar effect as before, giving the idea of consistent behaviour of the fading trace along the whole path. Before a deeper analysis, it is important to proceed to look into the MCS level changes between consecutive samples. Figure 7.18 depicts the results for the portion of time of the so-called "queue draining zone".

These last results depict a similar behaviour to the previously described one, but with a greater breach between the stats for 200km/h and 60km/h. The lowest values in general terms are for 200 km/h scenarios and few of the fades are greater for 60 km/h. However, in this case in lower values (see right part of Figure 7.17) the variability is

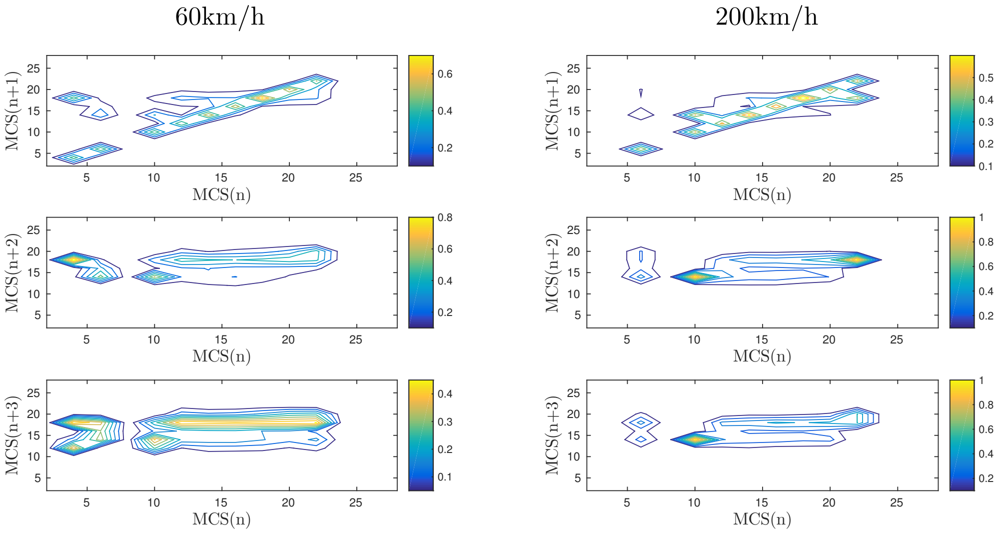


Figure 7.18: MCS level change between samples at different speeds (Queue draining portion of time).

almost equal and only slightly differs in the distribution among different MCS values, leading to a similar network conditions for both speeds.

As for Figure 7.18, we detect that in 60 km/h plots, mostly in the first row, it is more or less equally probable for MCS 4 to keep the value or return to MCS 18. In top left of Figure 7.17, the total percentage assignment of MCS 4 is depicted and prompts a very low value. Therefore, MCS 4 is very unlikely to appear and when it does, it equally goes back to high MCS values or keeps the assignment. So, the low MCS values are not maintained more than two samples, being the fades very drastic and quick in the "queue draining zone". Second row in Figure 7.18 back-ups this statement with low MCS assignments changing to higher values in consecutive samples. Similar effect is present in 200 km/h but for MCS 6. In the "queue draining zone", it is clear that the possibility of keeping the MCS level for a long time has been almost vanished, leading to a more changeable environment comparing with the whole trace.

All in all, **it has been proven that the obtained results for 60 km/h and 200 km/h have very similar MCS assignment (network conditions) and therefore achievable rates, enabling an isolated study of the impact of speed alone**.

A4 - Detailed study of different variability conditions on the move

This section provides with a detailed analysis of the performance of NewReno and CUBIC with the selected movement patterns at 60 km/h, 120 km/h and 300 km/h under realistic 4G latencies (Subsubsection 7.2) and low latencies (Subsubsection 7.2).

A4.1 - Different speed and fading patterns' study under 4G latencies

This Subsection covers the analysis under 4G latencies of different movement patterns with distinct speeds and fadings in order to better select a CCA for its selection over such mobility scenarios. Part of the encountered effects regarding stable fading patterns in Annex 2 may well be useful to explain the differences among distinct speed scenarios. Therefore, the already explained examples at 120 km/h are kept in the figures for comparison purposes. However, only the singularities over the more variable network circumstances and the outcome differences between NewReno and CUBIC are explained. Figure 7.19 depicts the goodput performance of NewReno with different movement patterns at 60 km/h, 120 km/h and 300 km/h respectively.

First row covers the outcome at 60 km/h with EVA60 fading. The greater the capacity, the greater jump between consecutive samples due to the impact of fading. So far, this scenario shows that the effect of end-to-end delay is very similar for both movement patterns, only detecting a slight advantage of forward movement in high capacities, being able to achieve around 40 Mbps while backward movement can hardly make around 30 Mbps. This effect demonstrates that the biggest impact is obtained in high capacities where the variability between consecutive bandwidth assignments is greater.

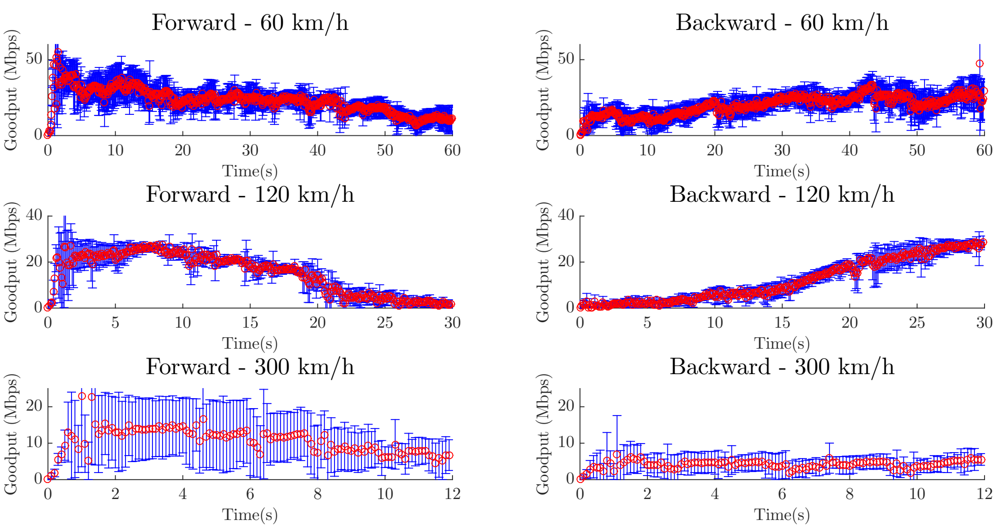


Figure 7.19: NewReno at different speeds and movement tendencies (4G latencies).

Second row corresponds to the scenario at 120 km/h (explained in Subsection 5.4.1). Under 4G delay circumstances and considerably stable fading conditions, it is more beneficial for goodput performance to come from better radio conditions and have already some packets in-flight. It demonstrates that it is better, from the goodput's point of view, to self-inflict delay through packets' overfeeding and recover from occasional losses due to TCP's capacity greediness rather than raise from worse positions and have an instant packet disposal. This means that backward movement requires in a greater way the use of a intra-protocol selection framework that picks a more aggressive CCA to cope with the increasing quality movement scenario.

At 300 km/h and having applied HST300 fading model, it is noticeable in the third row of the figure that in forward movement average goodputs over 10 Mbps are achieved, while in backward scenario NewReno is not able to scale close to 10 Mbps in any time. It is therefore important to underline that a conservative AIMD mechanism as the one present in NewReno, is not capable of ramping-up with a very variable and constant movement towards better radio conditions. As mentioned before, since the reported CQI is very volatile, the Slow Start phase and first period of congestion avoidance are crucial and pave the way for the rest of the transmission. If during such a critical period low CQIs are under use and TCP is not allowed to ramp sufficiently or encounters loss packets in an early stage, the transmission will need more time to achieve around 10 Mbps values. Apart from that, the NewReno case shows that sometimes even with long experiments, a conservative CCA is not able to reach full utilization at any point of the transmission, being important to address this constraint in the intra-protocol selection framework.

Figure 7.20 covers the main representation of the behaviour of CUBIC under the same speed, fading and latency conditions in which CUBIC prompts a very similar results to the ones obtained by NewReno. However, in general, the achieved rates are higher than NewReno (e.g. around 50 Mbps in forward movement at 60 km/h, 40 Mbps in

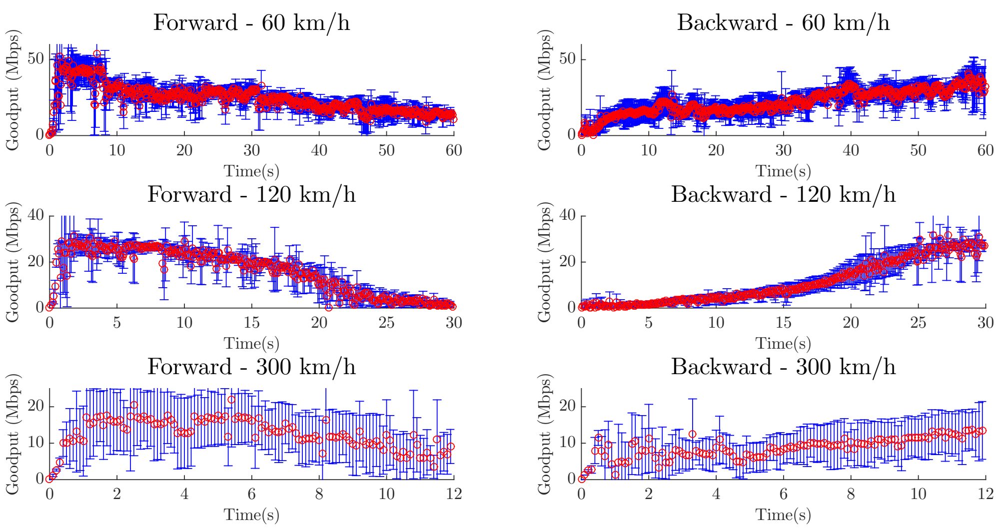
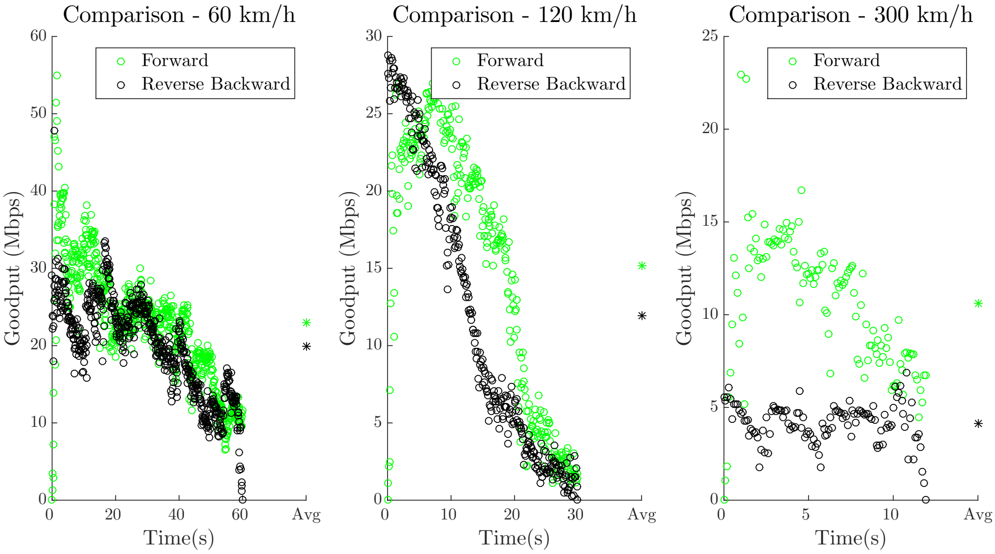
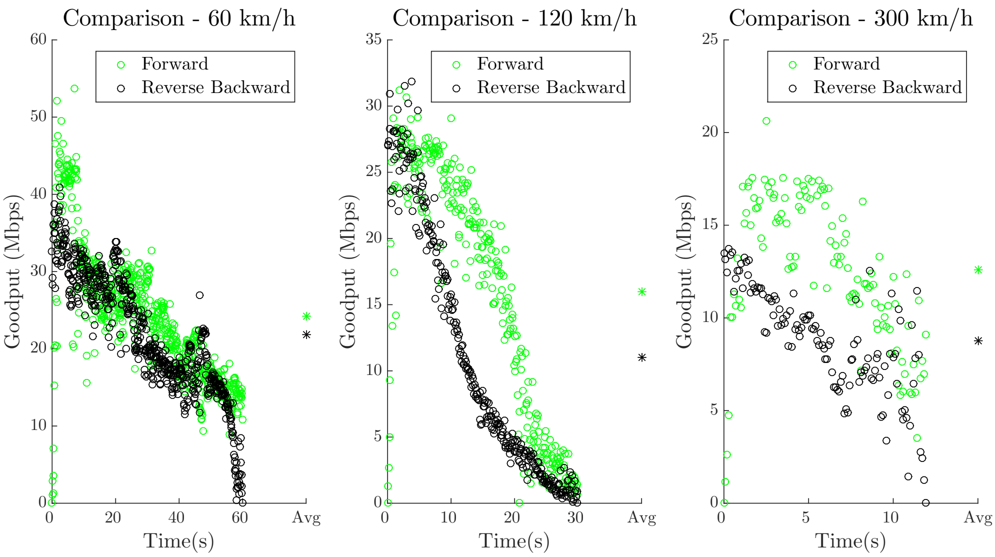


Figure 7.20: CUBIC at different speeds and movement tendencies (4G latencies).

backward movement at 60 km/h), being more noticeable in backward movement. Figure 7.21 graphs the average goodput performance for different speeds under longer RTT circumstances with the focus on the comparison between movement patterns for both NewReno and CUBIC. As in previous comparison graphs, the figure shows median goodput in forward pattern, contrasting with the median goodput result for reverse backward evolution. Thus comparing the performance of the CCA in the same network condition for different movement tendencies and study how could the



(a)



(b)

Figure 7.21: Comparison of movement tendencies at different speeds (4G latency): a) NewReno ; b) CUBIC.

previous network situation has a effect in final outcome.

Looking at Figure 7.21, there is a clear pattern that reflects that under realistic 4G latencies, forward movement outperforms backward inertia. Even though it is something already explained, it is important to underline that under 4G latencies the greater the speed and the tougher conditions of the scenario, the combination between speed and fading makes the breach within forward and backward proportionally bigger, being the effect less notorious in CUBIC. Thus, CUBIC under a variable fading, such as the one in 60 km/h or 300 km/h scenarios, it is able to reduce the impact of the end-to-end latency. Therefore, looking for optimization, CUBIC is closer to the optimal performance, being less critical to switch to another CCA that could resolve the network circumstances. In other words, the performance of NewReno in backward movement is not appropriate and its use should be avoided.

A4.2 - Different speeds and fading patterns' study under low latency

Following the description and findings under realistic 4G latencies, this Subsection covers the analysis and explanation of the same experiments and procedure over low latencies that characterize the potential delays in 5G networks. Figure 7.22 depicts the goodput performance of NewReno with different movement tendencies at 60 km/h, 120 km/h and 300 km/h respectively.

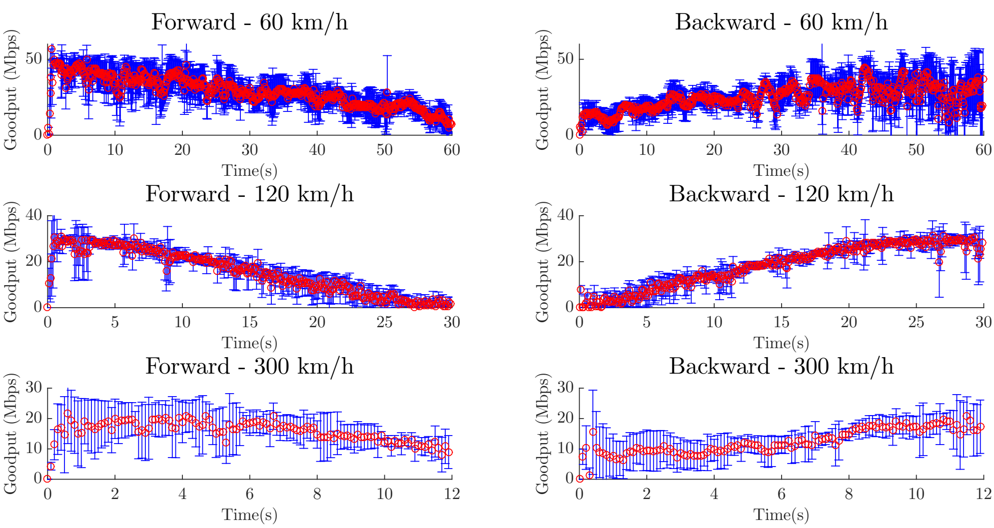


Figure 7.22: NewReno at different speeds and movement tendencies.

In Figure 7.22, the first row corresponds to the results for 60 km/h movement with an EVA60 fading. The performance on average is very changeable due to the variability of fading but it is more prominent in backward movement for the incapability of a moderated AIMD mechanism to grow between big bandwidth changes with an incremental movement pattern that constantly requires the injection of more packets on average. On the other hand, forward movement looks more stable regardless the variability because the eNodeB takes advantage of sufficiently loaded in-flight packets to respond adequately to different bandwidth assignments. That would suggest that in forward movement even the weak AIMD policy of NewReno is capable of achieving and sustaining close to maximum capacities, being a good TCP CCA to face such network conditions.

Second row corresponds to the scenario at 120 km/h (explained in Subsection 5.4.2). The lack of instability with the fading and low RTT are beneficial for the achievement of maximum capacities.

The third row depicts the performance of NewReno at 300 km/h with the usage of HST300 fading model. Regardless the movement quickness and the applied variable fading, NewReno is able in both movement patterns to converge very fast (less than a second). In contrast to the results depicted with 4G latencies, NewReno is able to cope with the variability and difficulties of the scenario in backward movement. This finding shows that the reduction of the end-to-end delay could lead to solve the underperformance problems of TCP under certain network circumstances. All in all, under low latencies, NewReno becomes an eligible candidate among the TCP family due to its good performance based on goodput.

Figure 7.23 shows the result for CUBIC. At a glance, in backward movement scenarios with very variable fading models, the depicted curves look more stable and adapted to the average incremental MCS (e.g. backward movement at 60 km/h). This is, even though the SINR, CQI and MCS vary for the same "base SINR" due to the fading, at every moment there is an average MCS to which optimally the CCA should adapt to. In the case of backward inertia the curve of CUBIC looks more stable due to the better adaptability comparing with NewReno, demonstrating more suitability to cope with such network conditions.

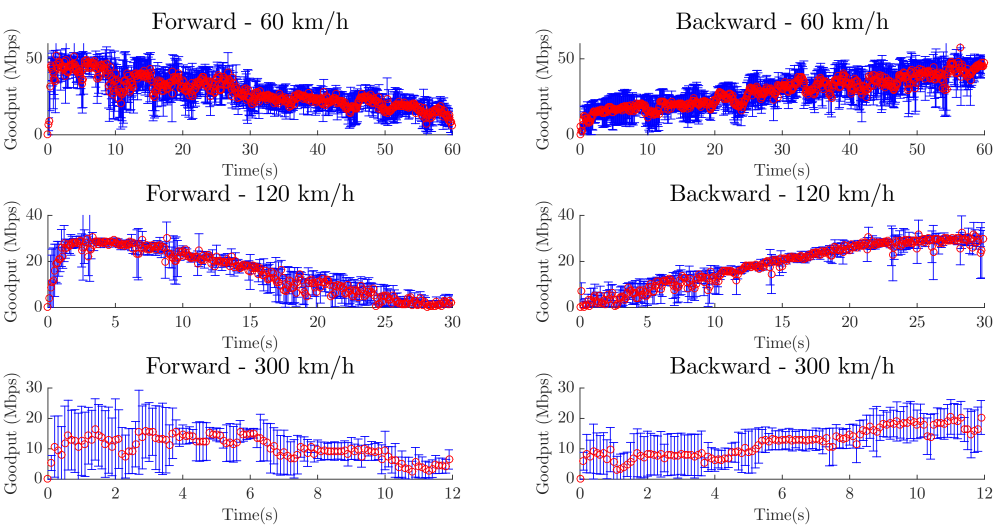


Figure 7.23: CUBIC at different speeds and movement tendencies.

Figure 7.24 graphs the average goodput performance for different speeds under low RTT circumstances with the focus on the comparison between movement patterns for both NewReno and CUBIC.

The conclusions are twofold:

- The variability at high CQI values, big bandwidth changes, in forward movement **has little impact regardless the selected CCA** because the eNodeB takes advantage of the previously enqueued packets.
- In contrast and even under low RTTs, backward movement's resultant perfor-

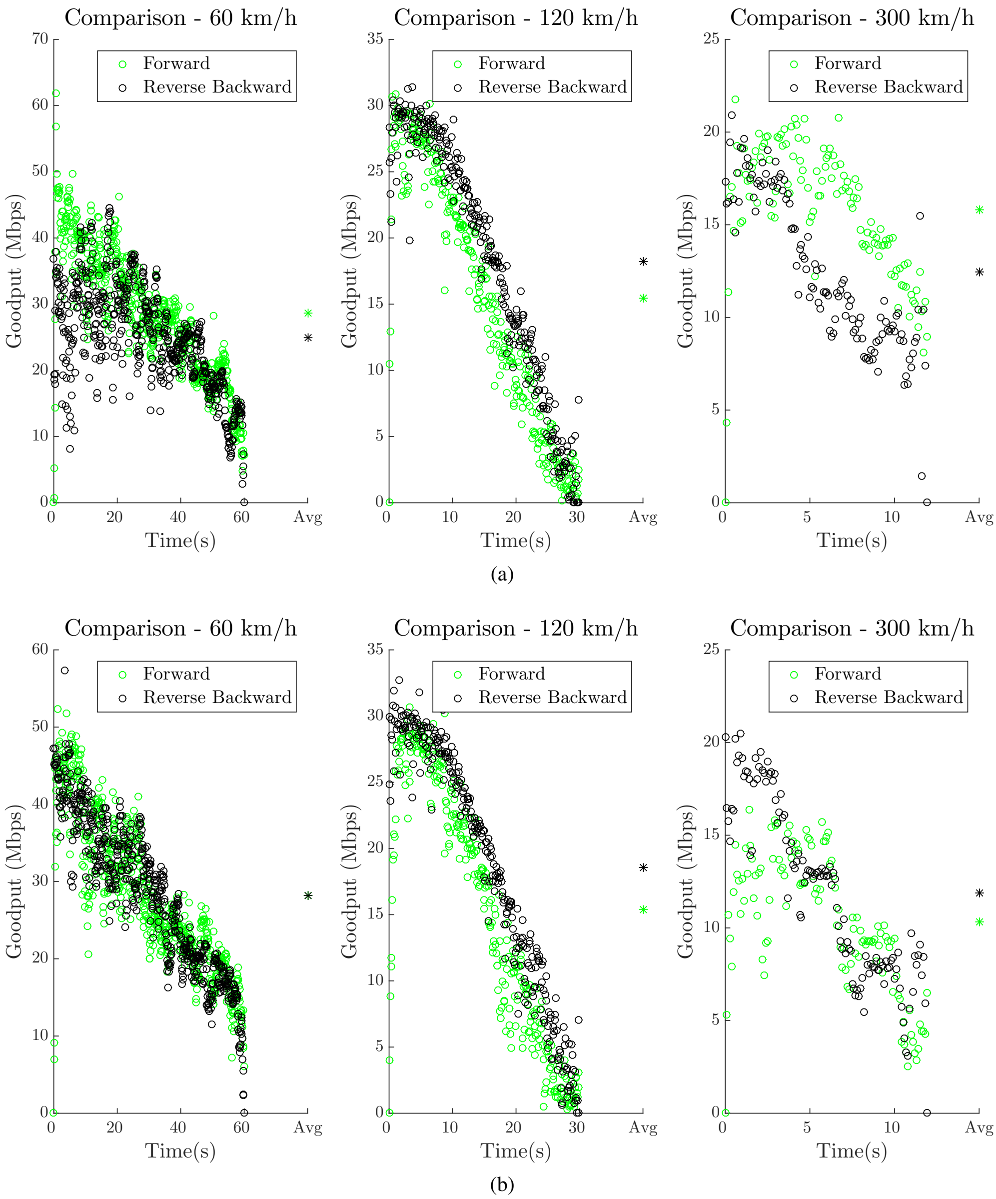


Figure 7.24: Comparison of movement tendencies at different speeds: a) NewReno ; b) CUBIC.

mance slightly suffers to meet the pace of the available capacity in NewReno. Therefore, it is clear that **even under low RTT, the impact of variability is translated to performance deficiencies** in those CCA cases in which the AIMD policy is very moderated (i.e. NewReno). However, in general **CUBIC is capable of better following the channel variability** in backward movement because it uses the packets that have been previously put in-flight.

As explained before, upcoming 5G scenarios will be partially devoted to drastically reduce latencies and therefore, the presented results would serve to deduct the possible behaviour of TCP in such future network conditions and the best practises in terms of most appropriate CCA selection on the move. Basic TCP, as NewReno is, has reported no impact of the variability on forward movement and slight dependency in backward movement due to the combined effect of buffer starvation due to the movement pattern to better radio conditions, the low scalability of NewReno and the variability itself. Even though NewReno struggles to achieve maximum capacities in backward movement, TCP flavours with greater aggressiveness and better responsiveness, such as CUBIC, have reported better behaviour. So far, it has been demonstrated that big bandwidth variabilities could be faced up with certain aggressiveness. This assumption brings us to the same dichotomy (or even bigger due to a bigger variability) of current cellular network or even fixed network regarding the aggressiveness of TCP flavours. It is complicated to resolve the trade-off between aggressiveness and high resource utilization against less aggressiveness and less buffering impact to the flow under test and the rest of the flows. In order to resolve the detected issues considering different perspectives of the performance, our work proposes a valid solution based on the network status (completed with different performance-related heuristics) and the application requirements for an **enhanced coexistence between distinct implementations of TCP (intra-protocol de-ossification)**.

Bibliography

[1] Rysavy Research/5G Americas, "Mobile broadband transformation LTE to 5G," Rysavy Research/5G Americas, White paper, Aug. 2016.

[2] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-Host Congestion Control for TCP," *IEEE Communications Surveys Tutorials*, vol. 12, no. 3, pp. 304–342, 2010.

[3] A. Wilk, J. Iyengar, I. Swett, and R. Hamilton, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2," Internet Engineering Task Force, Internet-Draft draft-hamilton-early-deployment-quic-00, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-hamilton-early-deployment-quic-00

[4] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper," 2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[5] A. Özer, M. T. Argan, and M. Argan, "The effect of mobile service quality dimensions on customer satisfaction," *Procedia-Social and Behavioral Sciences*, vol. 99, pp. 428–438, 2013.

[6] B. Nguyen, A. Banerjee, V. Gopalakrishnan, S. Kasera, S. Lee, A. Shaikh, and J. Van der Merwe, "Towards Understanding TCP Performance on LTE/EPC Mobile Networks," in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, ser. AllThingsCellular '14. New York, NY, USA: ACM, 2014, pp. 41–46. [Online]. Available: http://doi.acm.org/10.1145/2627585.2627594

[7] C. E. Koksal, K. Jamieson, E. Telatar, and P. Thiran, "Impacts of Channel Variability on Link-level Throughput in Wireless Networks," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '06/Performance '06. New York, NY, USA: ACM, 2006, pp. 51–62. [Online]. Available: http://doi.acm.org/10.1145/1140277.1140285

[8] D. Lee, B. E. Carpenter, and N. Brownlee, "Media streaming observations: Trends in UDP to TCP ratio," *International Journal on Advances in Systems and Measurements*, vol. 3, no. 3 & 4, pp. 147–162, 2010.

[9] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Behavior analysis of TCP Linux variants," *Computer Networks*, vol. 56, no. 1, pp. 462–476, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611003677

[10] N. Brownlee *et al.*, "Internet stream size distributions," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 282–283, 2002.

[11] "QUIC, a multiplexed stream transport over UDP - The Chromium Projects." [Online]. Available: https://www.chromium.org/quic

[12] K. J. Grinnemo, T. Jones, G. Fairhurst, D. Ros, A. Brunstrom, and P. Hurtig, "Towards a flexible internet transport layer architecture," in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, June 2016, pp. 1–7.

[13] T. Jones, G. Fairhurst, and C. Perkins, "Raising the Datagram API to Support Transport Protocol Evolution," in *Workshop on Future of Internet Transport (FIT 2017)*, 2017.

[14] A. Capone, L. Fratta, and F. Martignon, "Bandwidth estimation schemes for TCP over wireless networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 2, pp. 129–143, 2004.

[15] ETSI, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," European Telecommunications Standards Institute, Internet-Draft ETSI TS 136 213 v12.4.0, Feb. 2015.

[16] ——, “Universal Mobile Telecommunications System (UMTS); LTE; User Equipment (UE) application layer data throughput performance,” European Telecommunications Standards Institute, Internet-Draft ETSI TR 137 901 V11.13.0, Jan. 2015.

[17] “Understanding 5G: Perspectives on future technological advancements in mobile, Dec 2014.” [Online]. Available: https://www.gsmaintelligence.com/research/?file=141208-5g.pdf&download

[18] X. Wang *et al.*, “Cache in the air: exploiting content caching and delivery techniques for 5G systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.

[19] ETSI, “Mobile Edge Computing (MEC); Technical Requirements,” European Telecommunications Standards Institute, Internet-Draft ETSI GS MEC 002 v1.1.1, Mar. 2016.

[20] “3G/4G wireless network latency: How did Verizon, AT&T, Sprint and T-Mobile compare in Q3 2015?, Feb 2016.” [Online]. Available: http://www.fiercewireless.com/special-report/3g-4g-wireless-network-latency-how-did-verizon-at-t-sprint-and-t-mobile-compare-q3

[21] M. Allman, V. Paxson, and E. Blanton, “TCP Congestion Control,” RFC 5681 (Draft Standard), Internet Engineering Task Force, Sep. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5681.txt

[22] “Usage Statistics and Market Share of Operating Systems for websites in November of 2017.” [Online]. Available: https://w3techs.com/technologies/overview/operating_system/all

[23] D. Borman, B. Braden, V. Jacobson, and R. Scheffenegger, “TCP Extensions for High Performance,” RFC 7323 (Proposed Standard), Internet Engineering Task Force, Sep. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7323.txt

[24] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment Options,” RFC 2018 (Proposed Standard), Internet Engineering Task Force, Oct. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc2018.txt

[25] E. Blanton, M. Allman, L. Wang, I. Jarvinen, M. Kojo, and Y. Nishida, “A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP,” RFC 6675 (Proposed Standard), Internet Engineering Task Force, Aug. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6675.txt

[26] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, “The macroscopic behavior of the TCP congestion avoidance algorithm,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 3, pp. 67–82, 1997.

[27] R. Ludwig and R. H. Katz, “The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions,” *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 1, pp. 30–36, 2000. [Online]. Available: http://doi.acm.org/10.1145/505688.505692

[28] M. Handley, J. Padhye, and S. Floyd, “TCP Congestion Window Validation,” RFC 2861 (Experimental), Internet Engineering Task Force, Jun. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2861.txt

[29] I. Petrov and T. Janevski, “Evolution of TCP in High Speed Networks,” *International Journal of Future Generation Communication and Networking*, vol. 8, no. 2, pp. 137–186, Apr. 2015. [Online]. Available: http://www.sersc.org/journals/IJFGCN/vol8_no2/12.pdf

[30] K. J. Winstein, “Transport architectures for an evolving Internet,” Thesis, Massachusetts Institute of Technology, 2014. [Online]. Available: http://dspace.mit.edu/handle/1721.1/91037

[31] Y.-T. Li, D. Leith, and R. N. Shorten, “Experimental Evaluation of TCP Protocols for High-Speed Networks,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1109–1122, Oct. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4346548

[32] M. A. Alrshah, M. Othman, B. Ali, and Z. Mohd Hanapi, “Comparative study of high-speed Linux TCP variants over high-BDP networks,” *Journal of Network and Computer Applications*, vol. 43, pp. 66–75, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804514000903

[33] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links,” in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’01. New York, NY, USA: ACM, 2001, pp. 287–297. [Online]. Available: http://doi.acm.org/10.1145/381677.381704

[34] D. A. Hayes and G. Armitage, “Revisiting TCP Congestion Control Using Delay Gradients,” in *NETWORKING 2011*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6641, pp. 328–341. [Online]. Available: http://link.springer.com/10.1007/978-3-642-20798-3_25

[35] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "TIMELY: RTT-based Congestion Control for the Datacenter," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: ACM, 2015, pp. 537–550. [Online]. Available: http://doi.acm.org/10.1145/2785956.2787510

[36] N. Cardwell *et al.*, "BBR: congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, 2017.

[37] M. Dong, Q. Li, D. Zarchy, B. Godfrey, and M. Schapira, "PCC: Re-architecting Congestion Control for Consistent High Performance," *arXiv:1409.7092 [cs]*, Sep. 2014, arXiv: 1409.7092. [Online]. Available: http://arxiv.org/abs/1409.7092

[38] J. Iyengar and I. Swett, "QUIC Loss Detection and Congestion Control," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-quic-recovery-04, June 2017, http://www.ietf.org/internet-drafts/draft-ietf-quic-recovery-04.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-quic-recovery-04.txt

[39] "Different CCA alternatives for QUIC." [Online]. Available: https://chromium.googlesource.com/chromium/src/net/+/master/quic/core/congestion_control/send_algorithm_test.cc

[40] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)," RFC 6817 (Experimental), Internet Engineering Task Force, Dec. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6817.txt

[41] Mirja Kühlewind, "TCP SIAD: Congestion Control supporting Low Latency and High Speed," in *Proceedings of IETF-91*. Honolulu: IETF-91, Nov. 2014.

[42] S. D'Aronco, L. Toni, S. Mena, X. Zhu, and P. Frossard, "Improved Utility-based Congestion Control for Delay-Constrained Communication (Extended Version)," *arXiv:1506.02799 [cs]*, Jun. 2015, arXiv: 1506.02799. [Online]. Available: http://arxiv.org/abs/1506.02799

[43] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010, pp. 63–74. [Online]. Available: http://doi.acm.org/10.1145/1851182.1851192

[44] N. Cardwell, Y. Cheng, S. Yeganeh, and V. Jacobson, "BBR Congestion Control," Working Draft, IETF Secretariat, Internet-Draft draft-cardwell-iccrg-bbr-congestion-control-00, July 2017. [Online]. Available: http://www.ietf.org/internet-drafts/draft-cardwell-iccrg-bbr-congestion-control-00.txt

[45] J. Gettys, "Bufferbloat: Dark Buffers in the Internet," *IEEE Internet Computing*, vol. 15, no. 3, pp. 96–96, May 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5755608

[46] K. Miller and L. W. Hsiao, "TCPTuner: Congestion Control Your Way," *arXiv:1605.01987 [cs]*, May 2016, arXiv: 1605.01987. [Online]. Available: http://arxiv.org/abs/1605.01987

[47] M. Allman, C. Hayes, and S. Ostermann, "An Evaluation of TCP with Larger Initial Windows," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 41–52, Jul. 1998. [Online]. Available: http://doi.acm.org/10.1145/293927.295114

[48] N. Dukkipati, T. Refice, Y. Cheng, J. Chu, T. Herbert, A. Agarwal, A. Jain, and N. Sutin, "An Argument for Increasing TCP's Initial Congestion Window," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 3, pp. 26–33, Jun. 2010. [Online]. Available: http://doi.acm.org/10.1145/1823844.1823848

[49] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, "Increasing TCP's Initial Window," RFC 6928 (Experimental), Internet Engineering Task Force, Apr. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc6928.txt

[50] Matt Mathis, "Scaling IW with the Internet, an engineering argument," in *Proceedings of IETF-78*. IETF-78, Jul. 2010.

[51] J. T. <touch@isi.edu>, "Automating the Initial Window in TCP," Internet Engineering Task Force, Internet-Draft draft-touch-tcpm-automatic-iw-00, Dec. 2010. [Online]. Available: https://tools.ietf.org/html/draft-touch-tcpm-automatic-iw-00

[52] A.-L. Beylot, C. Baudoin, E. Chaput, R. Sallantin, F. Arnal, and E. Dubois, "Safe increase of the TCP's initial window using initial spreading," Internet Engineering Task Force, Internet-Draft draft-sallantin-iccrg-initial-spreading-01, Mar. 2014, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-sallantin-iccrg-initial-spreading-01

[53] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A.-L. Beylot, “An end-to-end alternative to TCP PEPs: Initial Spreading, a TCP fast start-up mechanism,” *International Journal of Satellite Communications and Networking*, vol. 34, no. 1, pp. 75–91, 2016. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/sat.1097/abstract

[54] “Initcwnd settings of major CDN providers.” [Online]. Available: http://www.cdnplanet.com/blog/initcwnd-settings-major-cdn-providers/

[55] Y. Cheng, J. Chu, S. Radhakrishnan, and A. Jain, “TCP Fast Open,” RFC 7413 (Experimental), Internet Engineering Task Force, Dec. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7413.txt

[56] S. Floyd, “Limited Slow-Start for TCP with Large Congestion Windows,” RFC 3742 (Experimental), Internet Engineering Task Force, Mar. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3742.txt

[57] S. Floyd, M. Allman, A. Jain, and P. Sarolahti, “Quick-Start for TCP and IP,” RFC 4782 (Experimental), Internet Engineering Task Force, Jan. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4782.txt

[58] D. Liu, M. Allman, S. Jin, and L. Wang, “Congestion control without a startup phase,” in *Proc. PFLDnet*, 2007.

[59] S. Ha and I. Rhee, “Taming the Elephants: New TCP Slow Start,” *Comput. Netw.*, vol. 55, no. 9, pp. 2092–2110, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2011.01.014

[60] M. Scharf, “Comparison of end-to-end and network-supported fast startup congestion control schemes,” *Computer Networks*, vol. 55, no. 8, pp. 1921–1940, Jun. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611000491

[61] M. Allman, K. Avrachenkov, U. Ayesta, J. Blanton, and P. Hurtig, “Early Retransmit for TCP and Stream Control Transmission Protocol (SCTP),” RFC 5827 (Experimental), Internet Engineering Task Force, May 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5827.txt

[62] M. Mathis, N. Dukkipati, and Y. Cheng, “Proportional Rate Reduction for TCP,” RFC 6937 (Experimental), Internet Engineering Task Force, May 2013. [Online]. Available: http://www.ietf.org/rfc/rfc6937.txt

[63] P. Sarolahti, M. Kojo, K. Yamamoto, and M. Hata, “Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP,” RFC 5682 (Proposed Standard), Internet Engineering Task Force, Sep. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5682.txt

[64] Y. C. N. Dukkipati, N. Cardwell and M. Mathis, “Tail Loss Probe (TLP): An Algorithm for Fast Recovery of Tail Losses,” Internet Engineering Task Force, Internet-Draft draft-dukkipati-tcpm-tcp-loss-probe-01, Feb. 2013, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-dukkipati-tcpm-tcp-loss-probe-01

[65] Y. Cheng, N. Cardwell, and N. Dukkipati, “Rack: a time-based fast loss detection algorithm for tcp,” Working Draft, IETF Secretariat, Internet-Draft draft-ietf-tcpm-rack-02, March 2017, http://www.ietf.org/internet-drafts/draft-ietf-tcpm-rack-02.txt. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-tcpm-rack-02.txt

[66] A. P. P. Hurtig, A. Brunstrom and M. Welzl, “TCP and Stream Control Transmission Protocol (SCTP) RTO Restart,” RFC 7765, Internet Engineering Task Force, Feb. 2016. [Online]. Available: https://tools.ietf.org/rfc/rfc7765.txt

[67] M. Rajiullah, P. Hurtig, A. Brunstrom, A. Petlund, and M. Welzl, “An Evaluation of Tail Loss Recovery Mechanisms for TCP,” *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 1, pp. 5–11, 2015. [Online]. Available: http://doi.acm.org/10.1145/2717646.2717648

[68] B. Briscoe, “Tunnelling of Explicit Congestion Notification,” RFC 6040 (Proposed Standard), Internet Engineering Task Force, Nov. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc6040.txt

[69] B. Trammell *et al.*, “Enabling Internet-Wide Deployment of Explicit Congestion Notification,” in *Proceedings of IETF-92*. Dallas, Texas, USA: IETF-92, Mar. 2015.

[70] R. Adams, “Active Queue Management: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1425–1476, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6329367

[71] Toke Høiland-Jørgensen, “The State of the Art in Bufferbloat Testing and Reduction on Linux,” in *Proceedings of IETF-86*. IETF-86, Mar. 2013.

[72] T. Hoeiland-Joergensen, P. McKenney, dave. tahtgmail.com, J. Gettys, and E. Dumazet, “The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm,” Internet Engineering Task Force, Internet-Draft draft-ietf-aqm-fq-codel-06, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-aqm-fq-codel-06

[73] “TCP Small Queues, Jul 2012.” [Online]. Available: https://lwn.net/Articles/506237/

[74] K. Winstein and H. Balakrishnan, “TCP Ex Machina: Computer-generated Congestion Control,” in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM ’13. New York, NY, USA: ACM, 2013, pp. 123–134. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486020

[75] K. Winstein, A. Sivaraman, and H. Balakrishnan, “Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks,” in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, ser. nsdi’13. Berkeley, CA, USA: USENIX Association, 2013, pp. 459–472. [Online]. Available: http://dl.acm.org/citation.cfm?id=2482626.2482670

[76] A. Sivaraman, K. Winstein, P. Thaker, and H. Balakrishnan, “An Experimental Study of the Learnability of Congestion Control,” in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM ’14. New York, NY, USA: ACM, 2014, pp. 479–490. [Online]. Available: http://doi.acm.org/10.1145/2619239.2626324

[77] X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall, “How Speedy is SPDY?” in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. Seattle, WA: USENIX Association, Apr. 2014, pp. 387–399. [Online]. Available: https://www.usenix.org/conference/nsdi14/technical-sessions/wang

[78] Y. Elkhatib, G. Tyson, and M. Welzl, “Can SPDY really make the web faster?” in *Networking Conference, 2014 IFIP*. IEEE, 2014, pp. 1–9.

[79] M. Belshe, R. Peon, and M. Thomson, “Hypertext Transfer Protocol Version 2 (HTTP/2),” RFC 7540 (Proposed Standard), Internet Engineering Task Force, May 2015. [Online]. Available: http://www.ietf.org/rfc/rfc7540.txt

[80] G. Carlucci, L. De Cicco, and S. Mascolo, “HTTP over UDP: An Experimental Investigation of QUIC,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, ser. SAC ’15. New York, NY, USA: ACM, 2015, pp. 609–614. [Online]. Available: http://doi.acm.org/10.1145/2695664.2695706

[81] J. Hildebrand and B. Trammell, “Substrate Protocol for User Datagrams (SPUD) Prototype,” Internet Engineering Task Force, Internet-Draft draft-hildebrand-spud-prototype-03, Mar. 2015, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-hildebrand-spud-prototype-03

[82] G. Papastergiou, G. Fairhurst, D. Ros, A. Brunstrom, K. J. Grinnemo, P. Hurtig, N. Khademi, M. Tüxen, M. Welzl, D. Damjanovic, and S. Mangiante, “De-ossifying the internet transport layer: A survey and future perspectives,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 619–639, Firstquarter 2017.

[83] A. A. Siddiqui and P. Mueller, “A requirement-based socket api for a transition to future internet architectures,” in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, July 2012, pp. 340–345.

[84] N. Khademi, D. Ros, M. Welzl, Z. Bozakov, A. Brunstrom, G. Fairhurst, K. J. Grinnemo, D. Hayes, P. Hurtig, T. Jones, S. Mangiante, M. Tuxen, and F. Weinrank, “Neat: A platform- and protocol-independent internet transport api,” *IEEE Communications Magazine*, vol. 55, no. 6, pp. 46–54, 2017.

[85] P. Hurtig, S. Alfredsson, Z. Bozakov, A. Brunstrom, K. Evensen, K.-J. Grinnemo, A. Fosselie Hansen, and T. Rozensztrauch, “A neat approach to mobile communication,” in *The ACM SIGCOMM 2017 Workshop on Mobility in the Evolving Internet Architecture (MobiArch 2017)*. ACM Press, 2017.

[86] G. Papastergiou, K.-J. Grinnemo, A. Brunstrom, D. Ros, M. Tüxen, N. Khademi, and P. Hurtig, “On the cost of using happy eyeballs for transport protocol selection.” in *ANRW*, 2016, pp. 45–51.

[87] H.-S. Park, J.-Y. Lee, and B.-C. Kim, “TCP performance degradation of in-sequence delivery in LTE link layer,” *International Journal of Advanced Science and Technology*, vol. 37, pp. 27–36, 2011.

[88] H. Jiang, Z. Liu, Y. Wang, K. Lee, and I. Rhee, “Understanding Bufferbloat in Cellular Networks,” in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, ser. CellNet ’12. New York, NY, USA: ACM, 2012, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2342468.2342470

[89] S. Alfredsson, G. Del Giudice, J. Garcia, A. Brunstrom, L. De Cicco, and S. Mascolo, “Impact of TCP congestion control on bufferbloat in cellular networks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*. IEEE, 2013, pp. 1–7.

[90] J. Garcia, S. Alfredsson, and A. Brunstrom, “A measurement based study of TCP protocol efficiency in cellular networks,” in *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2014, pp. 131–136.

[91] L. De Cicco and S. Mascolo, “TCP Congestion Control over HSDPA: an Experimental Evaluation,” *arXiv:1212.1621 [cs]*, Dec. 2012, arXiv: 1212.1621. [Online]. Available: http://arxiv.org/abs/1212.1621

[92] J. Garcia, S. Alfredsson, and A. Brunstrom, “Examining TCP Short Flow Performance in Cellular Networks Through Active and Passive Measurements,” in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ser. AllThingsCellular ’15. New York, NY, USA: ACM, 2015, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/2785971.2785974

[93] S. Alfredsson, A. Brunstrom, and M. Sternad, “Cross-layer analysis of TCP performance in a 4G system,” in *Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference on*. IEEE, 2007, pp. 1–6.

[94] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, “An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance,” in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM ’13. New York, NY, USA: ACM, 2013, pp. 363–374. [Online]. Available: http://doi.acm.org/10.1145/2486001.2486006

[95] L. Zhang, T. Okamawari, and T. Fujii, “Performance Evaluation of End-to-End Communication Quality of LTE,” in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, May 2012, pp. 1–5.

[96] Y. Xu, Z. Wang, W. K. Leong, and B. Leong, “An End-to-End Measurement Study of Modern Cellular Data Networks,” in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, M. Faloutsos and A. Kuzmanovic, Eds. Springer International Publishing, Mar. 2014, no. 8362, pp. 34–45, dOI: 10.1007/978-3-319-04918-2_4. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-04918-2_4

[97] R. Merz, D. Wenger, D. Scanferla, and S. Mauron, “Performance of LTE in a High-velocity Environment: A Measurement Study,” in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, ser. AllThingsCellular ’14. New York, NY, USA: ACM, 2014, pp. 47–52. [Online]. Available: http://doi.acm.org/10.1145/2627585.2627589

[98] L. Li, K. Xu, D. Wang, C. Peng, Q. Xiao, and R. Mijumbi, “A measurement study on TCP behaviors in HSPA+ networks on high-speed rails,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2015, pp. 2731–2739.

[99] K. I. Pedersen, M. Niparko, J. Steiner, J. Oszmianski, L. Mudolo, and S. R. Khosravirad, “System Level Analysis of Dynamic User-Centric Scheduling for a Flexible 5G Design,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[100] M. G. Sarret, G. Berardinelli, N. H. Mahmood, and P. Mogensen, “Impact of Transport Control Protocol on Full Duplex Performance in 5G Networks,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–5.

[101] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, “Adaptive Congestion Control for Unpredictable Cellular Networks,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM ’15. New York, NY, USA: ACM, 2015, pp. 509–522. [Online]. Available: http://doi.acm.org/10.1145/2785956.2787498

[102] H. Jiang, Y. Wang, K. Lee, and I. Rhee, “Tackling Bufferbloat in 3G/4G Networks,” in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC ’12. New York, NY, USA: ACM, 2012, pp. 329–342. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398810

[103] H. Im, C. Joo, T. Lee, and S. Bahk, “Receiver-Side TCP Countermeasure to Bufferbloat in Wireless Access Networks,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2080–2093, Aug. 2016. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7283642

[104] I. Johansson, “Congestion control for 4g and 5g access,” Internet Engineering Task Force, Internet-Draft draft-johansson-cc-for-4g-5g-02, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-johansson-cc-for-4g-5g-02

[105] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, “Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations,” RFC 3135 (Informational), Internet Engineering Task Force, Jun. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3135.txt

[106] G. Siracusano, R. Bifulco, S. Kuenzer, S. Salsano, N. B. Melazzi, and F. Huici, "On-the-Fly TCP Acceleration with Miniproxy," *arXiv:1605.06285 [cs]*, May 2016, arXiv: 1605.06285. [Online]. Available: http://arxiv.org/abs/1605.06285

[107] F. Ren and C. Lin, "Modeling and Improving TCP Performance over Cellular Link with Variable Bandwidth," *IEEE Transactions on Mobile Computing*, vol. 10, no. 8, pp. 1057–1070, 2011.

[108] K. Liu and J. Y. Lee, "Mobile accelerator: A new approach to improve TCP performance in mobile data networks," in *2011 7th International Wireless Communications and Mobile Computing Conference*. IEEE, 2011, pp. 2174–2180.

[109] F. Lu, H. Du, A. Jain, G. M. Voelker, A. C. Snoeren, and A. Terzis, "CQIC: Revisiting Cross-Layer Congestion Control for Cellular Networks," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '15. New York, NY, USA: ACM, 2015, pp. 45–50. [Online]. Available: http://doi.acm.org/10.1145/2699343.2699345

[110] Péter Szilágyi, "Mobile Content Delivery Optimization based on Throughput Guidance," in *Proceedings of IETF-93*. Prague: IETF-93, Jul. 2015.

[111] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582 (Proposed Standard), Internet Engineering Task Force, Apr. 2012.

[112] S. Liu, T. Başar, and R. Srikant, "TCP-Illinois: A Loss and Delay-based Congestion Control Algorithm for High-speed Networks," in *Proceedings of the 1st International Conference on Performance Evaluation Methodolgies and Tools*, ser. valuetools '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1190095.1190166

[113] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-speed TCP Variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008. [Online]. Available: http://doi.acm.org/10.1145/1400097.1400105

[114] Neal Cardwell and Yuchung Cheng, "BBR Congestion Control: An Update," in *Proceedings of IETF-98*, Mar. 2017.

[115] Jae Won Chung and others, "Driving Linux TCP Congestion Control algorithms around the LTE network Highway," in *Proceedings of NetDev 2.1*, Apr. 2017.

[116] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous, "Direct code execution: Revisiting library os architecture for reproducible network experiments," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 217–228.

[117] "Testbeds – Fed4fire facilities." [Online]. Available: http://www.fed4fire.eu/testbeds/

[118] S. Bouckaert, W. Vandenberghe, B. Jooris, I. Moerman, and P. Demeester, "The w-iLab.t Testbed," in *Testbeds and Research Infrastructures. Development of Networks and Communities*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, T. Magedanz, A. Gavras, N. H. Thanh, and J. S. Chase, Eds. Springer Berlin Heidelberg, May 2010, no. 46, pp. 145–154, dOI: 10.1007/978-3-642-17851-1_11. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-17851-1_11

[119] "FIRE family projects." [Online]. Available: https://www.ict-fire.eu/projects/

[120] A. Brunstrom, "Mobile Broadband Measurement Platform - MONROE," in *Proceedings of IETF-93*. Prague: IETF-93, Jul. 2015.

[121] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong, "TCP performance over mobile networks in high-speed mobility scenarios," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014, pp. 281–286.

[122] "tcp_cubic: make the delay threshold of HyStart less sensitive, Dec 2014." [Online]. Available: http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=2b4636a5f8ca547000f6aba24ec1c58f31f4a91d

[123] "tcp_cubic: refine Hystart delay threshold, Dec 2014." [Online]. Available: http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=42eef7a0bb0989cd50d74e673422ff98a0ce4d7b

[124] "Make CUBIC Hystart more robust to RTT variations, Mar 2011." [Online]. Available: http://patchwork.ozlabs.org/patch/85945/