5G-RACOM

Franco-German Innovation Project on 5G for Resilient and Green Rail Communications

Work Package 3: Hybrid FRMCS Networks

D3.1 Report on Assessment and Implementation of Multipath Technologies for Hybrid FRMCS Networks

Annex 1 - High-Level Descriptions of Candidate Technologies *(public)*

June 2025

6 Candidate Multipath Technologies

6.4 High-Level Descriptions of Candidate Technologies

For details refer to Annex 1.

Main candidate technologies assessed by the project are:

- *Extended Access Traffic Steering, Splitting & Switching (ATSSS)* – framework
- *Multi-Access Management Services (MAMS)* – framework
- Multipath TCP (MPTCP)
- Multipath QUIC (MP-QUIC)
- Stream Control Transmission Protocol (SCTP)
- Mutipath Data Congestion Control Protocol (MP-DCCP)
- Multipath UDP (MPUDP)
- Software Defined WAN (SD-WAN)
- Load Balancing Based IP Routing

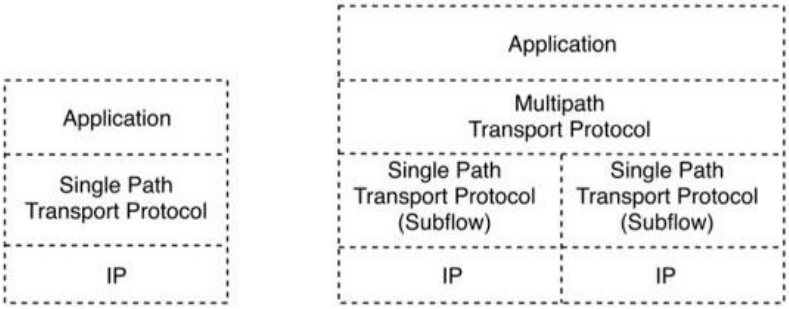

Figure 1: Single-path vs Multipath Transport Protocol Reference Architecture

The first two technologies of MAMS and ATSSS are generic recommendation drafts for implementing the multipath protocols, that cover the architecture design, features, messages and placement of modules. While the rest are multipath protocol implementations. The description in the

6.4.1 Extended ATSSS Framework

Access Traffic Steering, Switching & Splitting (ATSSS) is not a protocol, but rather a multiple access technology at the core of a 5G network. It is a technology introduced in 3GPP Release 16 to enable seamless and intelligent traffic management between 3GPP and non-3GPP access networks (e.g. WiFi) in 5G systems. It aims to improve performance and quality of service by aggregating Data Paths from different networks and dynamically switching traffic between them based on the required QoS.

ATSSS, as a multi-access technology residing within the transport stratum, does not align with the architecture defined by ETSI standards and adopted for the 5G-RACOM project. Its current design lacks support for multiple 3GPP access networks and does not accommodate the deployment of multipath proxies or gateways in middleboxes positioned between the transport and service strata. Alignment would require either extensions to the existing 3GPP specifications or adjustments to the FRMCS standards – both of which fall outside the scope of this project, even though the eventual deployment architecture of operational FRMCS may differ from that employed in 5G-RACOM. Note that such extensions has historically been analysed by ETSI - “Above-the-core using ATSSS-Emulated solution” as documented in [8].

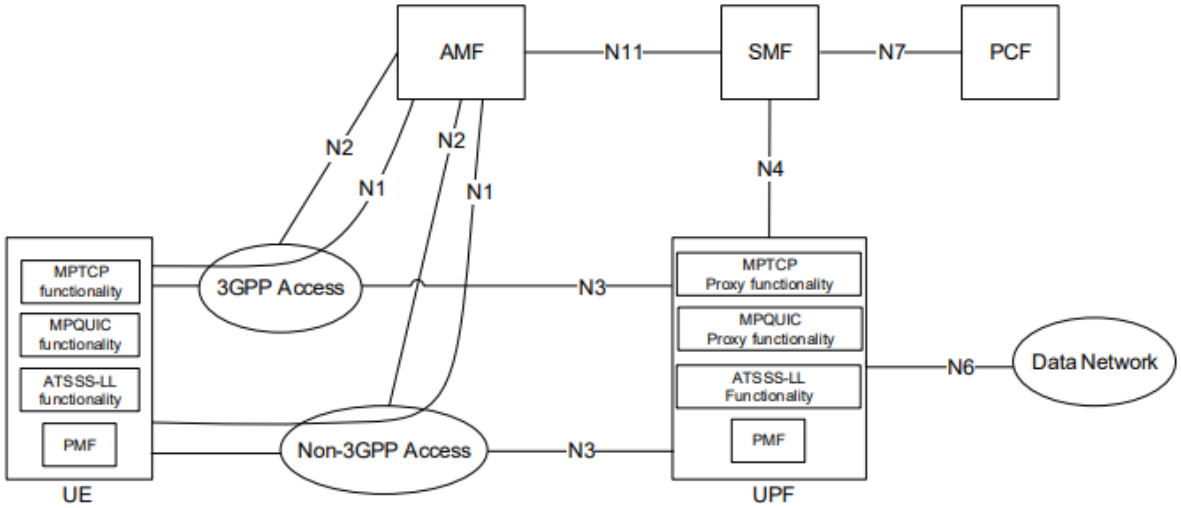

Figure 2: Architecture Reference Model for ATSSS Support

6.4.1.1 ATSSS Functionalities

Access Traffic Steering: The procedure that selects the appropriate access network for a new data flow and directs the traffic of this flow over the chosen access network.

Access Traffic Switching: The procedure that smoothly moves ongoing data flow traffic from one access network to another while maintaining continuity of the data flow.

Access Traffic Splitting: The procedure that divides the traffic of a data flow across multiple access networks, allowing for load balancing and increased reliability.

3GP defines [9] three options to implementing ATSSS: 1) using an ATSSS Lower Layer (ATSSS-LL) below the IP layer for non MPTCP nor MP-QUIC flows, 2) the MPTCP-based approach for TCP flows – requiring an MPTCP proxy aligned with the TCP-Convert RFCs—is well-known and readily available, 3) and an MP-QUIC based approach for UDP, Ethernet and IP flows (note that 3GPP Rel-19 investigates use of MP-QUIC for TCP flows).

ATSSS supports different steering modes for handling access traffic:

1. Active-Standby: Traffic is sent to one access network (active) until it becomes unavailable, then it is switched to the other (standby).
2. Smallest Delay: Traffic is directed to the access network with the shortest Round-Trip Time (RTT).
3. Load balancing: Traffic is distributed between access networks based on assigned weight factors.
4. Priority-based: Traffic is managed based on priority weights assigned to available access networks.
5. Redundant: The traffic is duplicated on both accesses if access networks are available.

Some of the ATSSS limitations are:

- ATSSS can simultaneously use one 3GPP access network and one non-3GPP access network and not multiple 3GPP access networks (note that 3GPP investigates simultaneous use of multiple 3GPP accesses - DualSteer).
- ATSSS requires client side to support multipath transport protocol and doesn’t support standalone multipath proxy or gateway function, only 5G core integrated one.

- ATSSS MPTCP requires two fully connected paths, but broadcast nature lacks uplink packet delivery, making MPTCP unviable.
- ATSSS-LL lacks an overlaying protocol for traffic switching or splitting, limiting its use to traffic steering.
- Untrusted convergence scenarios require modifications to hindering connectivity between different radio networks.
- Encryption of content leaving the 3GPP network poses challenges in point-to-multipoint systems, where decryption keys need to be known by all users.
- Lack of synchronization protocol between non-3GPP and 3GPP radios can cause buffer memory issues during traffic splitting due to different latency and configurations.
- Rerouting Control Plane signalling via 5G NR is not supported.

ATSSS is possible to use in railway applications, but with a number of limitations, this requires support for MPTCP, MP-QUIC or ATSSS-LL on each railway OB-GW unless abovementioned adjustments/extensions aren't available. It is worth concluding that ATSSS technology is not the best candidate for implementation within the 5G-RACOM project. The provision of multiple access takes place directly in the 5G core network, which causes several restrictions when using the implementation of this technology. At the moment, no papers have been found on the ATSSS extension for use outside the 5G core except [8].

[10] presents a survey on multipath transport protocols for 5G Access Traffic Steering, Switching, and Splitting (ATSSS) to achieve enhanced Mobile Broadband (eMBB) and Ultra Reliable Low Latency Communications (URLLC) services.

[11] provides an overview of the 3GPP's Access Traffic Steering, Switching, and Splitting (ATSSS) service, discusses ongoing discussions for enabling ATSSS for non-TCP with what they call "defines "ATSSS phase 2", it proposes the use of QUIC. 2 of the ATSSS modes, "Active-stand by" and "Smallest Delay" can be directly supported by QUIC, also the connection migration feature from QUIC allows the switching BUT it does not support the splitting functionality. Also, QUIC connection cannot be intercepted since they are secure, so a direct translation is not feasible in case the traffic is not QUIC. The document also provides some solutions for "Unreliable quic extension" to use QUIC with unreliable traffic and discusses a possible deployment with MP-QUIC which will provide to the ATSSS all 3 aspects Steering, Switching and Splitting.

Note: A (MP)QUIC connection initiated between the UE and a server without the ATSSS UPF assistance cannot benefit from any direct application of the ATSSS steering methods based on network input given that the steering policy as currently defined in ATSSS is local to the UE and the ATSSS UPF and there are no means to signal that policy to a remote server.

[12] proposes an efficient multi-access (MA) session management for ATSSS in 5G networks, achieving radio resource saving and signalling reduction through existing policy rules and signalling procedures.

[13] explores the use of ATSSS technology for IP layer convergence between 5G and ATSC 3.0, detailing its limitations and proposing a high-level converged architecture with ATSSS Release 17 characteristics to overcome these limitations.

Deutsche Telekom project has a practical implementation of Multipath for Fixed Mobile Convergence on Campus that uses ATSSS with 5G [14].

A commercial solution of ATSSS is available from the Tessares firm that provides the deployment of Multipath TCP proxies supporting Hybrid Access in different networks that use them to combine xDSL and LTE. 5G ATSSS solution is a software package that can be added in virtualised 5G cores [15].

6.4.2 MAMS Framework

Multi-Access Management Services (MAMS) is a programmable framework designed to handle multi-connectivity scenarios where clients can simultaneously connect to multiple networks based on different access technologies such as Wi-Fi, LTE and NR.

MAMS provides mechanisms for flexible selection of network paths in a multi-access communications environment, taking into account the specific needs of applications. It uses network intelligence and policies to adjust traffic distribution across selected paths and user plane treatments (such as tunnelling or encryption) to optimise network performance.

The functional elements of the MAMS architecture include:

- Network Connection Manager (NCM) and Client Connection Manager (CCM) at the control plane: These elements handle MAMS control plane procedures, configure user plane functions, negotiate with the client for the use of available access network paths, and determine link monitoring procedures.
- Network Multi-Access Data Plane (N-MADP) and Client Multi-Access Data Plane (C-MADP) in the user plane: These elements handle the forwarding of user traffic across multiple network paths, as well as encapsulation, fragmentation, reordering and other user plane functions.

The MAMS framework is not dependent on specific access network types or user plane protocols, allowing it to co-exist and complement existing protocols. It allows protocols to be negotiated and configured to suit their use in a given multi-access scenario based on client and network capabilities.

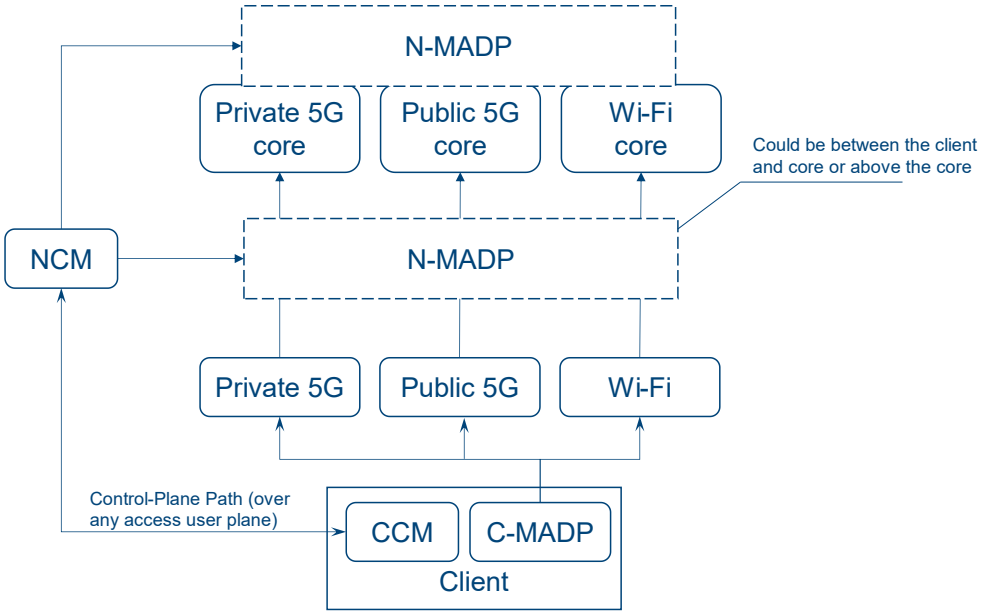

Figure 3: MAMS Reference Architecture

MAMS provides support for the following functionalities:

- Steering: MAMS enables traffic, both uplink and downlink, to be steered across different access network paths based on required QoS. For example, in uncongested scenarios or when a user's Wi-Fi coverage is good, MAMS can steer traffic towards Wi-Fi for optimal performance. Conversely, in congested situations or when Wi-Fi coverage is poor, MAMS can redirect traffic to LTE or other available access paths to ensure reliable and efficient data transmission.
- Switching: MAMS supports switching capabilities, enabling seamless handover of user traffic between different access network paths. If the quality of a particular network path degrades or

becomes unavailable, MAMS can switch traffic to an alternative path without disrupting the user experience.

- Splitting: MAMS can split user traffic across multiple access network paths, spreading the load and optimising resource utilisation. For example, in scenarios where a user's device is simultaneously connected to both Wi-Fi and LTE, MAMS can intelligently split traffic between the two paths based on application requirements, network conditions or user preferences. This load balancing approach helps prevent network congestion and improves overall efficiency.

These steering, switching and splitting capabilities are enabled by the coordination between the Network Connection Manager (NCM) and the Client Connection Manager (CCM) in the MAMS architecture. The NCM and CCM exchange control plane messages to negotiate the best combination of access and core network paths, as well as user plane treatments, to ensure optimal application performance. Similarly to ATSSS extensions, MAM has historically been analysed by ETSI - “Above-the-core using MAMS” as documented in [8].

The distinguishing feature of MAMS from ATSSS is that MAMS is completely independent of the type of communication protocols (could be any multipath transport protocol under) and technologies used, be it LTE, NR, WiFi or even wired communications. In addition, the ability to place N-MADP between the kernel and the client or above the kernel makes this framework more flexible.

The main problem is that MAMS is only described in RFC 8743 [16]. It is not an Internet Standards Track specification and may not be widely standardised and interoperable in railway systems. The lack of standardisation may hinder its widespread adoption and implementation. In addition, no scientific or practical implementations of the framework are currently publicly available. Another limitation of MAMS in use is the real-time requirements of railway applications. Some railway applications, especially those related to signalling and control systems, have stringent real-time requirements. MAMS may introduce additional latency due to its dynamic path selection and switching mechanisms, which could be critical in safety-critical railway systems.

6.4.2.1 High-Level Architecture

Control Plane

In MAMS control plane protocol stack, a WebSocket is used for transporting management and control messages between the NCM and the CCM.

The main functions of the MAMS control plane are:

- Configuration of Network and Client User-Plane Functions: Determines available access paths, protocols, handovers, and rules for user-plane traffic processing.
- Discovery of NCM by CCM: Facilitates NCM discovery through provisioning or DNS queries.
- Exchange of Capabilities and Negotiation of User-Plane Parameters: Enables CCM and NCM to exchange capabilities, negotiate user-plane parameters, and configure user-plane paths.
- Adaptation to Dynamic Network Conditions: Allows adaptive traffic steering and user-plane treatment based on link status information.
- Transport Protocol: WebSocket is used for management and control message communication between NCM and CCM.

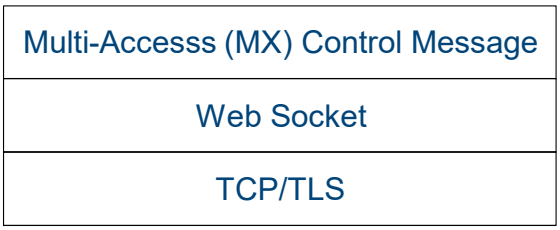

Figure 4: MAMS Control Plane Protocol Stack

User Plane

MAMS user-plane protocol stack is used for transporting the user payload, e.g., an IP Protocol Data Unit (PDU). It consists of the two layers of the MAMS user plane protocol: the Multi-Access (MX) Convergence Layer and the Multi-Access (MX) Adaptation Layer.

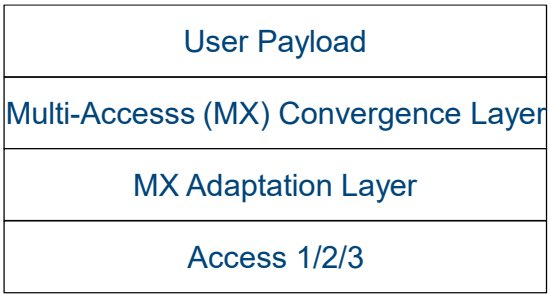

Figure 5: MAMS User Plane Protocol Stack

Multi-Access (MX) Convergence Layer

The MX convergence layer within the MAMS user plane protocol is responsible for handling multi-access specific tasks. These tasks include access (path) selection, multi-link (path) aggregation, splitting/reordering, lossless switching, fragmentation or concatenation of data packets. The MX convergence layer can be implemented using several existing user plane protocols, such as Multipath TCP (MPTCP) or Multipath QUIC (MP-QUIC). Alternatively, it can use encapsulating header/trailer schemes such as Generic Routing Encapsulation (GRE) or Generic Multi-Access (GMA).

Multi-Access (MX) Adaptation Layer

The MX Adaptation Layer in the MAMS user plane protocol focuses on addressing transport network-related aspects, ensuring accessibility and enhancing security at the user layer. It handles functions such as tunnelling, network layer security and Network Address Translation (NAT). The MX Adaptation Layer can be implemented using protocols such as IPsec, Datagram Transport Layer Security (DTLS) or Client NAT, which involves source NAT at the client with inverse mapping at the Network Multi-Access Data Plane (N-MADP). Importantly, the MX adaptation layer can be configured independently for each access link. For example, in a deployment with LTE/NR (considered secure) and Wi-Fi (considered less secure), the MX Adaptation Layer can be omitted for the LTE/NR link, while configured with IPsec to secure the Wi-Fi link.

6.4.3 MPTCP

The multipath TCP (MPTCP) protocol is a project that started around 10 years ago and that has been evolving until arriving at the proposed standard RFC 8684 [17]. This protocol is an extension of the widespread TCP, and it allows the usage of different interfaces under a single MPTCP connection to provide better capacity and more reliability leading to an overall better user experience. MPTCP is able to fall back and work alongside with the standard TCP which aims to an easier deployability. It also provides the reliable transport that TCP provides, as the main concept is to extend the TCP semantic to

a “bundling” of TCP subflows under the same TCP connection. The main differences that are added to MPTCP compared to TCP are:

1. Connection set up: An MPTCP connection is initiated with a normal 3-way handshake as a standard TCP connection, but the SYN packets and SYN/ACK packets carry a “MP CAPABLE” option. This option allows the connection partner to know that the initiating side is capable of MPTCP and is willing to do it with a specific version that comes inside of this option. Moreover, inside of this MP CAPABLE, keys are negotiated to authenticate the new flows that will be possibly added since each connection has a Token that is a cryptographic hash of this key. It is worth mentioning that the overall connection is closed with a connection-level FIN.
2. Subflow creation and addition: This is done in a similar way as the connection establishment, but instead of an “MP CAPABLE” an “MP JOIN” option is used. This sync is sent with the additional origin address and to the server’s destination address alongside the token generated with the keys and sets up the subflow. It is important to note that this implicitly tells the server that an additional address exists and that it may be possible to use it, this will depend on the path management. Note that the subflow is closed with a 4-way FIN handshake.
3. Path advertising: the last biggest difference is the “MP ADD” option. This option allows any of the ends to inform the other end that additional addresses are available in case they are needed. This may be helpful to surpass NAT as A can inform B about the address, while B can make the JOIN to establish the additional flow.
4. Data sequence mapping: Since there is a separation between subflows, there is a connection or Data sequence number space (random non-zero as in TCP) and a subflow sequence number space (allowed to be 0 or any other space), this means that the information from the application layer is numbered with one sequence space and it will be mapped to the different subflow spaces depending on the requirements. Having 2 number spaces allows the MPTCP to re-organize the traffic from different paths.

Between other highlights of the protocol there is the possibility to change the priorities of the paths, this will be done by the path management entity, also there are connection-level acknowledgements for the overall data, as well as path-level acknowledgements for the chunks that go over the specific flow. Moreover, if a packet fails, it is possible to send it over different flows or receive its ACK from any other flow since the numbering of the data sequence will be clear, leading to an avoidance of misinterpretations.

It is important to understand how the congestion control is managed in MPTCP. As mentioned before, different types of algorithms may be used, but they base the information on what each protocol has to offer. In this sense, the semantics of MPTCP define a specific connection congestion window (CWDN) instead of a per-flow congestion window. What this means is that the announced congestion window in any of the flows contains the information of the status of the overall connection receive buffer. This information is used by the congestion control algorithm as well as the scheduler to appropriately allocate the traffic according to the network congestion state and according to the policies respectively.

While MPTCP has been proposed to be used within ATSSS, there are drawbacks when being used to encapsulate unreliable traffic as it blindly retransmits each lost frame leading to excessive delay and potential head-of-line blocking. A decision for MPTCP leaves the increasing share of UDP in today's traffic mix unconsidered [18].

MPTCP supports load-balancing, traffic shifting among the multiple paths and capacity aggregation [19]. Further, it leverages the inherent congestion control from TCP which adapts the sending rate by observing congestion signals from the network. By design, MPTCP is limited to TCP services as it blindly re-transmits lost packets.

6.4.3.1 High-level Architecture

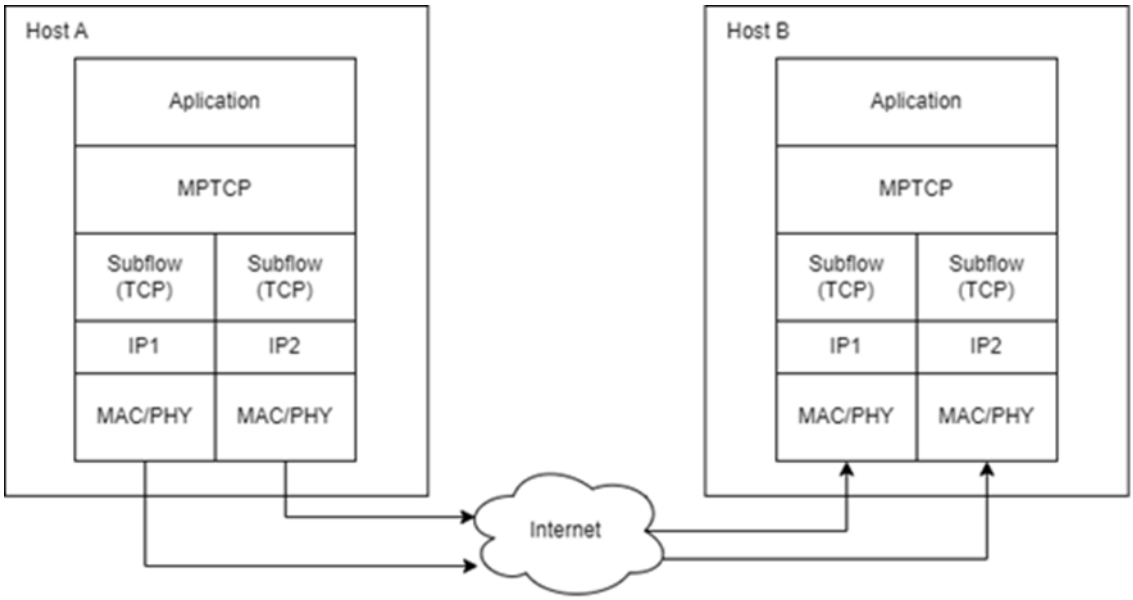

Figure 6: MPTCP high level architecture between two nodes

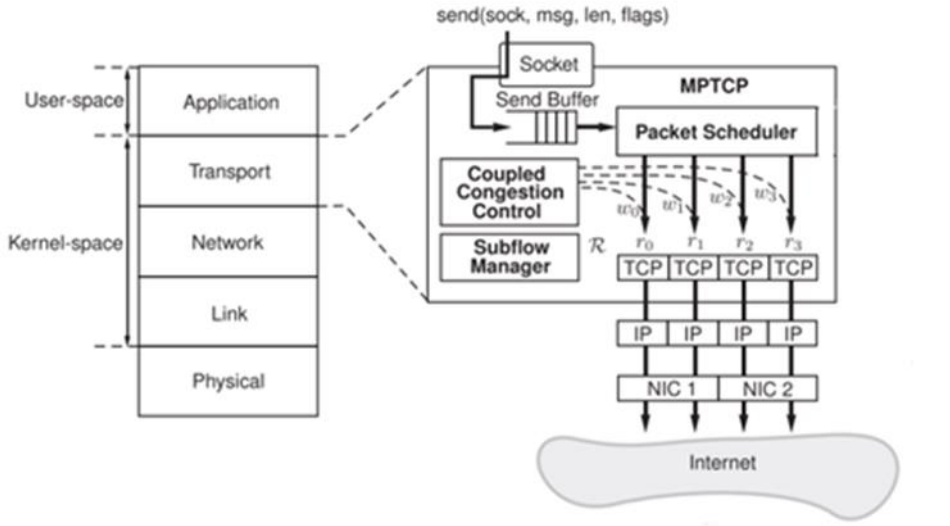

Figure 7 MPTCP Modules

The figures above are mentioned in [20].

6.4.3.2 Standards

- RFC 6824 → 8684– TCP Extensions for Multipath Operation with Multiple Addresses.
- RFC 6181 – Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses.
- RFC 6182 – Architectural Guidelines for Multipath TCP Development.
- RFC 6356 – Coupled Congestion Control for Multipath Transport Protocols.
- RFC 6897 – Multipath TCP (MPTCP) Application Interface Considerations.
- RFC 8041 – Use Cases and Operational Experience with Multipath TCP.
- RFC 8684 - TCP Extensions for Multipath Operation with Multiple Addresses.

6.4.3.3 Use and Deployments

The document below is the MPTCP used by Apple, the implementation is led for reliability and seamless experience rather than increased throughput. Apple's MPTCP implementation combines Wi-Fi with 4G/5G, using cellular networks as a backup whenever Wi-Fi performance drops below the required QoS. When this happens, MPTCP automatically transitions traffic to 4G or 5G. Users can enable this feature if they agree to the data limit and provide confirmation.

- MPTCP developed by Linux: Version 5.4 [https://www.multipath-tcp.org/]
- Versions >5.4 following MPTCP RFC 8684 [mptcp.dev]
- MPTCP for Apple iOS [https://developer.apple.com/documentation/foundation/improving-network-reliability-using-multipath-tcp]
- A proxy that uses both the interfaces to divide a request and is able to force MPTCP [megaleecher.net]
- FreeBSD Development of the MPTCP protocol [freebsd.org]
- Traffic load balancer that now supports MPTCP [techdocs.f5.com]
- Netscaler also has now available MPTCP for the application delivery controller [docs.netscaler.com]

6.4.3.4 Support of Steering/Switching/Splitting

There are several implementations of MPTCP that indeed have steering, switching, and splitting capabilities (some before and all after kernel 5.4). For this purpose, a conjunction work between the path manager, the congestion control, and the scheduler allows MPTCP to take decisions on the suitability of a path (Path management+scheduler=steering), adjust the traffic between paths according to the policies (Scheduler+path management+congestion=Splitting), and changing paths when necessary (Scheduler+path management+congestion control=Switching). All the capabilities are developed in Linux, but as of version of Linux kernel 6.8 there is only one scheduler by default and different possibilities for congestion control and path management. Some of the functions can be found under sysctl [21]. Nonetheless, since it is open-source, the code can be updated to provide different schedulers, adjust the path management and add other congestion control algorithms.

6.4.3.5 Support of Rail Applications

MPTCP will support all the applications that need reliable communication [22], [23]. The UDP-based applications are not inherently supported with MPTCP since there is not a mechanism to allow "un-acknowledged" packets.

6.4.3.6 Integration with Transport Network Infrastructure

MPTCP Protocol inherently only has path management and congestion control, both being a crucial part of knowing which paths are suitable and which are not. Nonetheless, the scheduler and how it should work is not regulated, thus there is flexibility on the Scheduler to provide QoS. MPTCP does have one capability which is MP_PRIO that allows the system to select the path with priority. The MPTCP options field has an additional value that is the "MP_EXPERIMENTAL". This field can be used as desired for experimental usage.

Taking this into account, the QoS can be provided on a per-packet or per-IP flow depending on the designed decision. Nonetheless, the developed schedulers mostly are on a per-flow basis since they are designed at the user plane.

6.4.3.7 Availability of Commercial Products and/or Open-Source

Yes, available in Linux kernels from 5.4, already included in Ubuntu image 22.04 (LTS). It is also possible to generate a kernel image that forces all communications to work under MPTCP V0, the one that is for Linux kernel 5.4.

6.4.3.8 Protocol Specific Information

MPTCP is promising since it removes the head of the line blocking and falls back to TCP when necessary. Also, provides appropriate handover and with an appropriate management of the paths, the scheduler and the congestion control, multiple results can be achieved according to the needs of the network. For rail activities, there should be an important focus on congestion control due to the high speeds.

Pros

- Increased reliability under fair MPTCP usage (meaning that the system does not overuse the link compared to non-MPTCP-capable systems), handover easiness.
- Possibility to distribute the traffic according to the link possibilities and traffic characteristics, the port number from the client or internal policies that can be developed.
- Path aggregation under “unfair” usage when capacity is the goal.

Cons

- It is not a straightforward task to use MPTCP in middle parts of the network; the best-case scenario would be a user and a server using MPTCP.
- if the client and server do not have MPTCP (TCP(Client)-MPTCP(client-gateway)-MPTCP(Server gateway)-TCP(server)) and a gateway/proxy is used, then the overall control loop is broken in N+1 control loops with N being the number of gateways/proxies. This can impact the TCP part since the overhead in the MPTCP may cause delays. In this sense, it is important to pay attention to the development of the gateways to affect as less as possible the TCP congestion control.

6.4.4 MP-QUIC

The Multipath QUIC (MP-QUIC) protocol is an extension of the QUIC protocol, aiming to provide multipath characteristics, similar to MPTCP, but using QUIC as its base. Despite not yet being an official standard, the protocol [24] has gained attention and promise. The primary goal is to address head-of-line blocking found in TCP. MP-QUIC achieves this through its numbering spaces, ensuring proper packet identification for reordering. The destination connection ID defines the utilized path, adding a distinctive feature to MP-QUIC. The main differences that are added to MP-QUIC compared to QUIC are:

- Connection Establishment: MP-QUIC follows the initial connection process of QUIC, with the option "enable multipath" indicating its multipath capabilities. MP-QUIC uses "connections" to refer to each path, bundled under the MP-QUIC connection. An interesting trait is the ability to explicitly constrain the number of paths through the "active connection id limit," and connection termination is signalled with a "CONNECTION CLOSE" message.
- Subflow Creation and Addition: Paths in MP-QUIC are created using a mechanism similar to path discovery in QUIC. The entity sends a "PATH_CHALLENGE" with "NEW_CONNECTION_ID" to establish the new path. Upon receiving a "PATH_RESPONSE," traffic transmission can commence. An additional "PATH_STATUS" option informs the status of a path and its availability for use.

- Data Sequence Mapping: MP-QUIC numbers packets per path, each with its sequence. The overall connection management involves using the pair sequence number and connection ID to reorganize packets before passing them to the next layer. The ACK MP frame is introduced for MP-QUIC to accommodate the connection ID in the ACK.
- Congestion Control: MP-QUIC inherits explicit congestion notifications and works similarly to TCP with a congestion window. Joint management is required, allowing the use of congestion control algorithms like LIA or OLIA. Research on congestion control algorithms specifically designed for MP-QUIC is limited.
- QUIC Extension: A QUIC extension (datagram mode) under development provides unreliable transmission based on unacknowledged configuration, potentially advantageous for different services and gateways [25]. The encrypted nature of QUIC, however, makes it challenging for middleboxes to handle translations.

6.4.4.1 High-level Architecture

The high-level architecture of MP-QUIC involves a path manager for defining path suitability, handling new paths, and managing path additions or removals. Congestion control is embedded, and the scheduler plays a crucial role, offering flexibility for providing Quality of Service (QoS). Available schedulers include minRTT and Round Robin but highly depend on the implementation. Although MP-QUIC theoretically supports path steering, switching, and splitting, the scheduler is crucial to implementing these capabilities in practice.

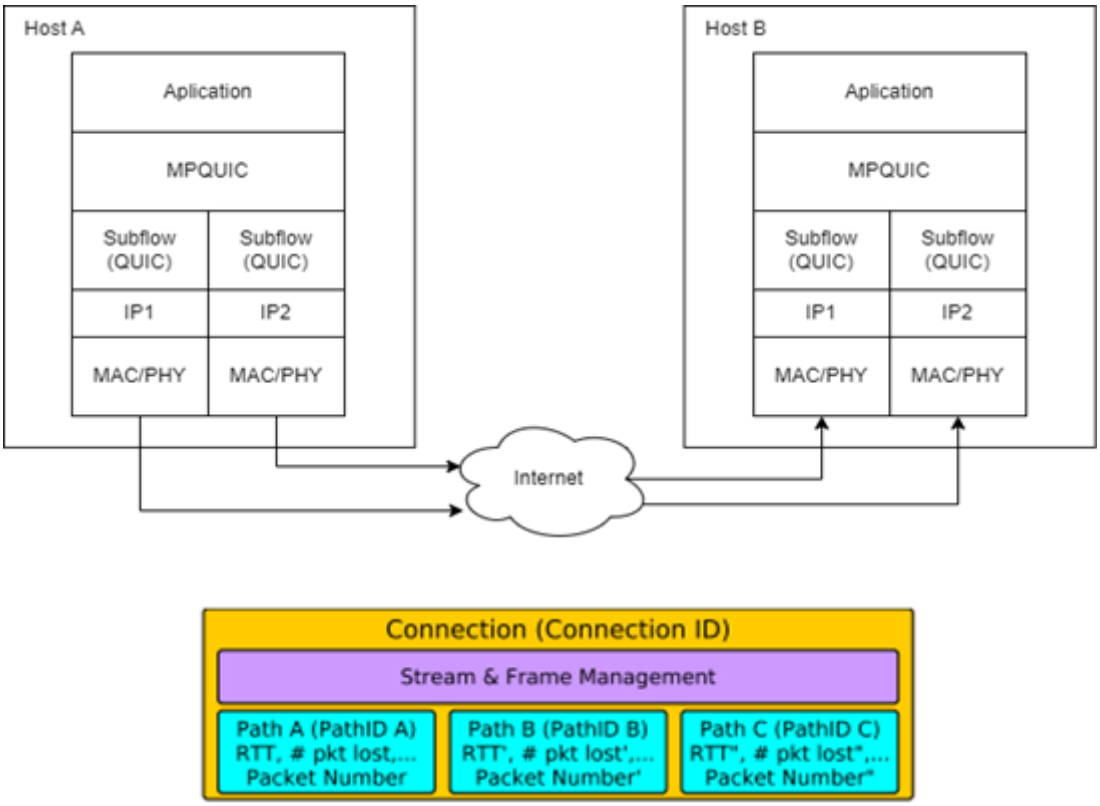

Figure 8: High-level architecture of MP-QUIC [3]

6.4.4.2 Standards

The MP-QUIC protocol is not yet an established standard, and its development is outlined in the internet draft [26].

There have been some ongoing studies on MP-QUIC, as of the research in [7], additional studies have been conducted to understand the feasibility of the protocol [27] does a comparison between MPTCP and MP-QUIC in mobile environments for an iOS application. Also, in [28] a discussion about challenges related to MP-QUIC is conducted, this is mostly due to its encrypted nature.

6.4.4.3 Use and Deployments

As of August 2023, real-world deployments of MP-QUIC are scarce, with no notable implementations by companies, including Google, the creator of QUIC.

6.4.4.4 Support of Steering/Switching/Splitting

MP-QUIC theoretically supports steering, switching, and splitting capabilities. The path manager defines path suitability, handles new paths, and manages additions or removals. The scheduler, not explicitly regulated, offers flexibility in providing QoS. While MP-QUIC supports steering and splitting through path management, the scheduler is essential for full steering and splitting capabilities.

6.4.4.5 Support of Rail Applications

MP-QUIC inherently supports all applications requiring reliable communication, similar to MPTCP. However, MP-QUIC is designed to work with UDP, and its usage in rail applications may need careful consideration of congestion control due to high speeds.

6.4.4.6 Integration with Transport Network Infrastructure

MP-QUIC protocol includes path management and congestion control, crucial for identifying suitable paths. The scheduler, not explicitly regulated, provides flexibility for QoS. Schedulers such as minRTT and Round Robin are available, and QoS can be provided on a per-packet or per-IP flow basis, depending on design decisions.

6.4.4.7 Availability of Commercial Products and/or Open-Source

As of August 2023, there are at least three open-source implementations of MP-QUIC.

MP-QUIC

- Source: https://multipath-quic.org/ that leads to the git hub implementation https://github.com/qdeconinck/mp-quic
- The MP-QUIC solution is implemented over the standard implementation of Go available on github (https://github.com/quic-go/quic-go). One feature of focus for MPF implementation as a gateway is the disabling of acknowledgements, which exists in the latest version of QUIC and is not compatible with MP-QUIC, and hence the need for looking into other two solutions of xQUIC and PicoQUIC.
- A VM (virtual machine) solution is available with the earlier version of QUIC is available which does not support the disabling of Acks. A considerable amount of implementation effort is needed to adapt the MP-QUIC solution over the latest QUIC, especially looking into the Acknowledgement relevant code in MP-QUIC.

XQUIC

- Source: https://github.com/alibaba/xquic
- The MP-QUIC solution is from Alibaba, with most of the documentation in Chinese, but the project is kept updated with the draft. It seems a good fit for the MPF gateway implementation, with some of its features like disabling acknowledgements, disabling security (which is not

available in others open-sources). Features of steering, switching and splitting solutions are supported.

PicoQUIC

- Source: https://github.com/private-octopus/picoquic
- The solution is not well maintained and has few runtime errors. Features of steering, switching are supported, but splitting (aggregation) is not supported yet.

6.4.4.8 Protocol Specific Information

Pros

- Faster protocol, as indicated by simulations and studies.
- Works with UDP, making it suitable for middleboxes to identify and handle it.
- Embedded TLS security makes it suitable for HTTP traffic, aiming to replace HTTP2+TLS+TCP.
- Faster connection setup and aggregation of new paths compared to MPTCP.

Cons

- Early stages of development with potential outdated implementations.
- Not implemented in kernels, primarily an application running on top of UDP.
- Challenges in middlebox/gateway usage due to the need for translation at the application layer.

6.4.5 SCTP, CMT+SCTP, MPSCTP

There are different approaches to provide the multipath capabilities to SCTP, there is not a specific consensus, and studies were found referring to “multipath SCTP” and others to concurrent multipath transfer CMT. They will be referred accordingly.

6.4.5.1 SCTP

the pure Stream Control transmission protocol (SCTP), defined in RFC9260, is a connection-oriented message switched protocol that ensures error-free delivery and uses UDP underneath. The idea behind it is to create an association where different paths can live. In other words, it is a bundling of different paths under the same association. It offers:

- Acknowledged error-free, non-duplicated transfer of user data
- Data fragmentation to conform to discovered Path Maximum Transmission Unit (PMTU) size
- Sequenced delivery of user messages within multiple streams, with an option for order-of-arrival delivery of individual user messages (if the applications wants, ordered delivery can be mandatory or not, this is defined by the U flag in the DATA chunk)
- Optional bundling of multiple user messages into a single SCTP packet (this means that the chunks can have different kinds of data in it, all together. There is the option to avoid bundling.
- Network-level fault tolerance through supporting of multi-homing at either or both ends of an association (when a data packet is lost, another path can be used for re transmission of the data, the server can also send the SACK over other connection, this helps also to know if a link is broken).

The architecture of SCTP is based on a multihoming approach using different IPs under the same association, where both the sender and receiver can have multiple IP addresses representing different network paths. The protocol establishes associations between these IP addresses and creates separate connections that can be used in case of failures. Note that in SCTP is NOT POSSIBLE to send over all streams simultaneously.

IMPORTANT: SCTP has an INIT message to initiate an association. This message MAY have, in its variable length field, additional IPV4 or IPV6 fields for multihomed services, thus, alongside with the source address, creates the various possible IPs that the sending and receiving host can use.

Pure SCTP can handle the switching characteristic when required. Whenever there is a duplicated packet or there is absence of reply to the heartbeats (messages used to check on the status of the possible links) the system can decide to switch to another path. It could be misunderstood from the SCTP characteristics that there is steering for the association initiation, nonetheless, there is not any kind of “characteristic analysis” to check which of the addresses shared in the association request is the “best link for the traffic”, it is generally defined by the application with the “SET_PRIMARY”. If there is no clear “SET_PRIMARY”, the selection of the primary address follows other reasons (for instance the type of address IPv6 or IPv4). Taking that into account, there is NO explicit steering functionality for SCTP. Finally, by means of the splitting, SCTP does not use all the paths, it only uses the other connections when the main one fails, thus There is no splitting.

Originally SCTP was designed to have failovers and not to send over all streams at the same time, the evolutions found are: using Concurrent multipath transfer (CMT) or the Multipath SCTP proposal. SCTP is not widely developed due to the lack of support in operating systems.

High-level architecture contained in [29] is depicted below.

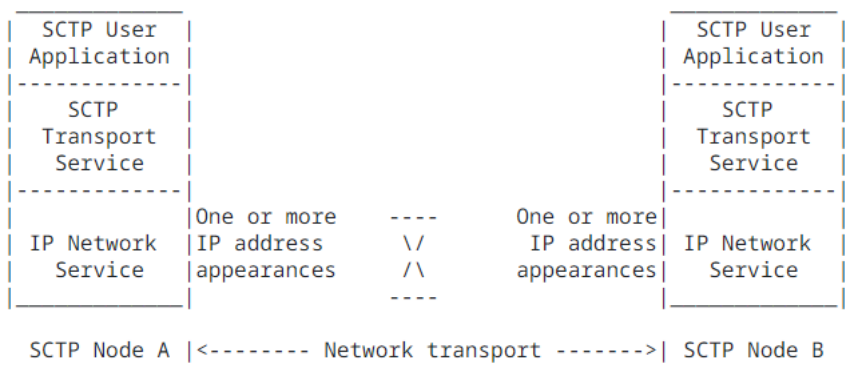

Figure 9: An SCTP association

6.4.5.2 CMT

The Concurrent multipath transfer (CMT) is a concept of concurrent paths that was proposed with traditional SCTP underneath that enables concurrent data transfer over multiple paths. Designed to overcome the limitations of single-path communication, it enhances data delivery reliability, throughput, and fault tolerance through the utilization of multiple available network paths. CMT is built on top of the SCTP protocol based on RFC 9260. CMT has an important characteristic - it has only one numbering space for all paths, making it sensitive to out of order packets. Note that the out of order packets do not create head of the line blocking. Nonetheless, it can give miscalculations for the congestion window since the “non differentiation” of the paths can lead to a wrong derivation of the RTT [30]. CMT supports various architecture options, including load balancing, failover, and bandwidth aggregation [31]. Load balancing ensures optimal distribution of data across available paths (Splitting) to maximize network resource utilization. Failover capability (Switching) guarantees uninterrupted data transmission by automatically switching to an alternative path in case of primary path failure. Bandwidth aggregation combines the available bandwidth of all paths, resulting in increased overall throughput. The Path Management module is responsible for discovering and monitoring available paths between the sender and receiver. It dynamically detects path changes, including path failures or recoveries, and updates the path status accordingly. This module plays a critical role in path selection and steering during data transmission.

The Association Establishment process in CMT is like the traditional SCTP. It involves the exchange of INIT and INIT-ACK chunks between the sender and receiver, negotiating various parameters, and forming associations for data exchange. CMT incorporates Stream Control mechanisms, like SCTP, to manage the simultaneous transmission of multiple streams of data. The streams are mapped to different paths, enabling the parallel transmission of data over each path.

6.4.5.3 MPSCTP

MPSCTP is also an extension of SCTP since it works on top of it. This approach gives also the multipath capability but has a different approach than CMT since it uses two numbering spaces one for each association (MPSCTP connection) and one for the packets per path, the main advantage of the separate numbering spaces is that RTT can be appropriately calculated per path and thus, the congestion control can use accurate data. This approach also takes care of the Fast retransmission algorithm from SCTP since this algorithm works under the supposition of the gap under one numbering space. the author provides a way around to solve that problem. There is not much elaboration on this approach [32].

NOTE: The main difference between MPSCTP and CMT is that MPSCTP uses two different number spaces to facilitate the reordering instead of one as CMT uses. That's why CMT is highly affected by out of order packets. Even though, the most adopted (yet not highly adopted) solution to extent SCTP has been CMT.

6.4.5.4 High-Level Architecture

High-level CMT architecture contained in [33] is depicted below.

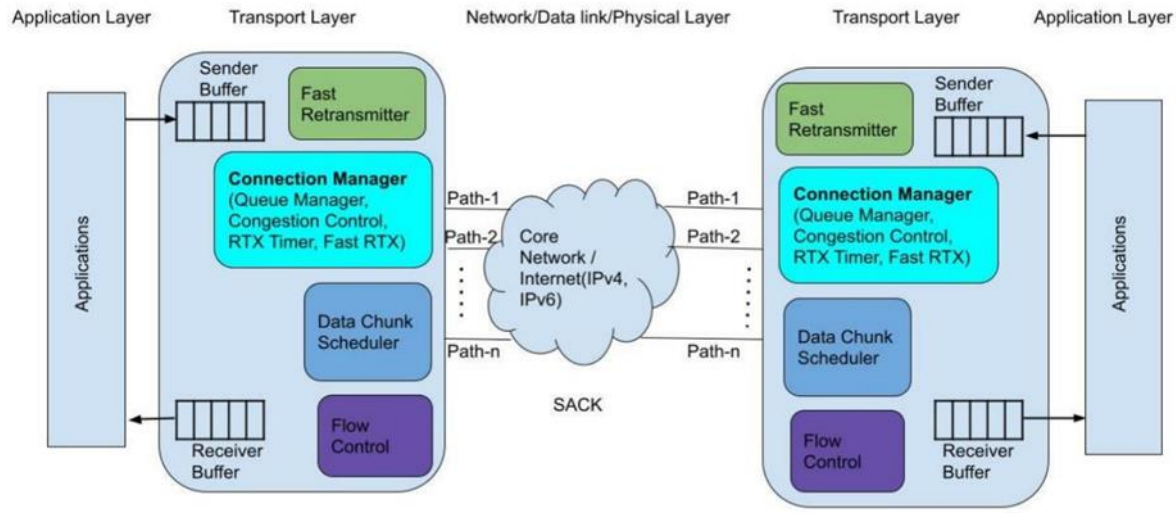

Figure 10: CMT Protocol Stack Architecture.

6.4.5.5 Standards

- RFC for SCTP: RFC9260 [https://datatracker.ietf.org/doc/html/rfc9260]
- Draft for "CMT-SCTP" : *Load Sharing for the Stream Control Transmission Protocol (SCTP)* . Internet Engineering Task Force. [https://datatracker.ietf.org/doc/draft-tuexen-tsvwg-sctp-multipath/]
- API extension: "SCTP Socket API Extensions for Concurrent Multipath Transfer" [https://datatracker.ietf.org/doc/draft-dreibholz-tsvwg-sctpsocket-multipath/24/]

Related papers and studies contain the following:

- [34], [35] are the beginning of the evolution for SCTP with CMT solution. They provide a description of the number space and the benefits and difficulties that this evolution brings. There is a clear explanation on how the single number space introduced by CMT can have problems due to the un-ordered delivery. Fiver different retransmission policies were also evaluated.
- [36] describes an evolution for the CMT protocol using the DB-CMT that is a congestion control using command window adaptation policy with 3 added policies. In this study also different CMT approaches are compared by means of the average transmission times.
- [37], [38] are the papers referred to MPSCTP approach that did not progress. The papers show the reasons of the researchers to use two number spaces instead of one as in CMT. Concluding on the benefit of reducing the out of order problem from single number space.

6.4.5.6 Use and Deployments

Pure SCTP has been deployed in several operative systems:

- AIX Version 5 and newer
- NetBSD since 8.0
- Cisco IOS 12 and above
- DragonFly BSD since version 1.4, however support is being deprecated in version 4.2
- FreeBSD, version 7 and above, contains the reference SCTP implementation
- HP-UX, 11i v2 and above
- Linux kernel 2.4 and above QNX Neutrino Realtime OS, 6.3.0 to 6.3.2, deprecated since 6.4.0
- Tru64 with the Compaq SCTP add-on package
- Sun Solaris 10 and above
- VxWorks versions 6.2.x to 6.4.x, and 6.7 and newer

CMT+SCTP and MPSCTP:

- It was not possible to find actual deployments of this, just studies and simulations.

6.4.5.7 Support of Steering/Switching/Splitting

- Pure SCTP: It does support switching but not the other capabilities.
- CMT+SCTP: It does have the capability of steering, depending on the scheduling policies the main path is chosen; switching is managed by the path manager that defines which paths are available and which aren't and splitting this is done based on policies for the data chunks that are processed by the scheduler.

6.4.5.8 Support of Rail Applications

- Steering switching splitting in each IP flow (Client+app / in the end is the full connection)
- The considerations here are the same as for MP-QUIC or MPTCP, this is dependent on the implementation of the scheduler.

6.4.5.9 Integration with Transport Network Infrastructure

6.4.5.10 Availability of Commercial Products and/or Open-Source

Pure SCTP is being commercially used here:

- http://spot-on.sf.net - P2P library
- http://goldbug.sf.net - Instant Messenger

6.4.5.11 Protocol Specific Information

SCTP by itself is a good reliable protocol that has the inherit capability to be failure resilient, nonetheless by itself it does not increase the throughput as it does not aggregate the capacity of each individual path.

CMT which allow SCTP to provide further capacity aggregation, has limitations when the networks are heterogeneous as the packets may arrive out of order and this causes some issues with the reordering, though it has the capability to do so. Also, CMT is not widely developed yet, turning it into a big limitation.

Seems that there is not much work going on with the MPSCTP or CMT-SCTP, which would make difficult the deployment.

6.4.6 MP-DCCP

MP-DCCP, or Multipath Datagram Congestion Control Protocol, is a transport layer protocol designed for real-time and multimedia communication, providing congestion control and unreliable data delivery. It extends the capabilities of Datagram Congestion Control Protocol (DCCP) by adding multipath support.

The Datagram Congestion Control Protocol (DCCP) is a transport layer protocol standardized in RFC 4340. It is designed to offer congestion control and unreliable data delivery for applications requiring real-time or multimedia communication. DCCP is based on UDP, is unicast, and connection-oriented, providing unreliable delivery for services using it. It is particularly suitable for applications like streaming media, where trade-offs between delay and reliable, in-order delivery are crucial.

6.4.6.1 High-level Architecture

MP-DCCP operates on a client-server architecture, allowing the sender to establish multiple paths to the receiver. It divides data into smaller chunks and sends them across different available paths. The receiver reassembles the datagrams, which may arrive unordered, into the original data. The protocol supports various architecture options, including load balancing, failover, and bandwidth aggregation.

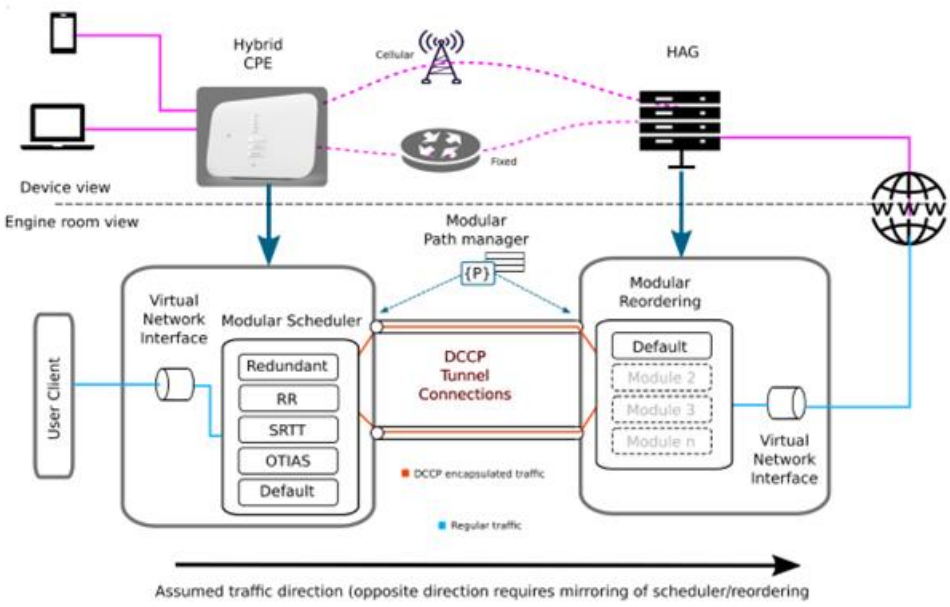

Figure 11: MP-DCCP Hybrid Scenario

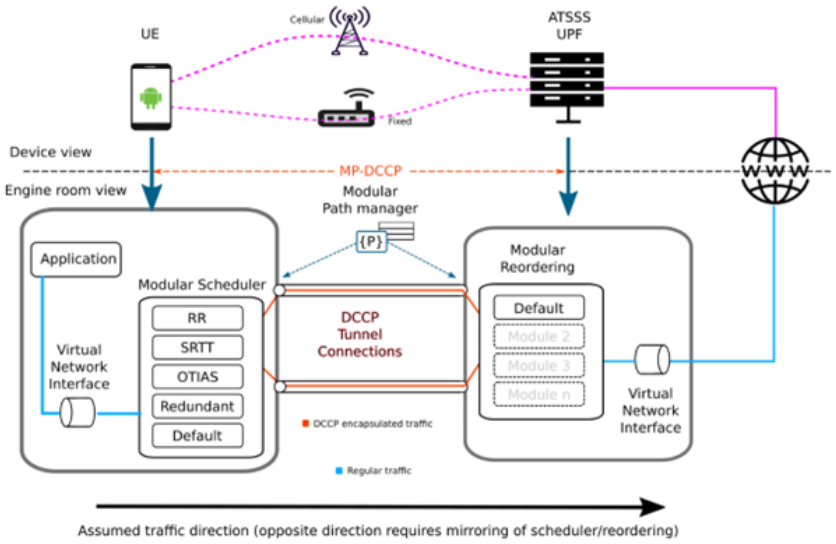

Figure 12: 3GPP ATSSS Scenario

The previous images were taken from [39].

```
Service          |<-            MP-DCCP          >|           Service
or Device                                                    or Device
+-------+          ___ +-----+ DCCP Flow 1 +------+          +--------+
|       |     _   |Seq||Path |-------------|Re-    |     _    |        |
| Sender|____| \__\ /_|      |      :      |order  |____/ |___|Receiver|
|       | IP|_/      |Sched|      :      |       |    \_|IP  |        |
|       |    VNIF_in  |uler |-------------|engine| VNIF_out  |        |
+-------+             +-----+ DCCP Flow n +------+           +--------+
```

Figure 13: IP compatible multipath framework based on MP-DCCP

```
 Device        Middlebox 1          Middlebox 2        Device
+------+    +--------------+    +-------------+    +--------+
|Sender| -> |MP-DCCP entry| -> |MP-DCCP exit| -> |Receiver|
+------+    +--------------+    +-------------+    +--------+
```

Figure 14: Sender and receiver independent MP-DCCP

```
       Device                      Middlebox         Device
+-----------------------+    +-------------+    +--------+
|Sender + MP-DCCP entry| -> |MP-DCCP exit| -> |Receiver|
+-----------------------+    +-------------+    +--------+
```

Figure 15: Sender integrated but receiver independent MP-DCCP

```
 Device        Middlebox                Device
+------+    +--------------+    +------------------------+
|Sender| -> |MP-DCCP entry| -> |MP-DCCP exit + Receiver|
+------+    +--------------+    +------------------------+
```

Figure 16: Sender independent and receiver integrated MP-DCCP

```
        Device                       Device
+-----------------------+    +------------------------+
|Sender + MP-DCCP entry| -> |MP-DCCP exit + Receiver|
+-----------------------+    +------------------------+
```

Figure 17: Sender and receiver integrated MP-DCCP

The previous images were taken from [40].

6.4.6.2 Standards

- DCCP: Datagram Congestion Control Protocol (DCCP) is standardized in RFC 4340.

- MP-DCCP: MP-DCCP is detailed in the white paper [41].

6.4.6.3 Use and Deployments

MP-DCCP is implemented in a proxy for 5G multi-access scenarios, enhancing traffic steering, switching, and splitting. An example implementation is available for download here. Additionally, interoperability tests with Xiaomi mobile devices have been conducted.

6.4.6.4 Support of Steering/Switching/Splitting

MP-DCCP supports traffic steering, switching, and splitting capabilities. The splitting functionality is highly dependent on the scheduler used.

6.4.6.5 Support of Rail Applications

6.4.6.6 Integration with Transport Network

MP-DCCP does not define Quality of Service (QoS) but can be used with schedulers that provide QoS. When integrated into the network layer framework, per-packet QoS may be achieved. In the user plane, schedulers can implement per-flow (per-application) QoS.

6.4.6.7 Availability of Commercial Products and/or Open-Source

- MP-DCCP is available as an open-source implementation, accessible for research and development. Commercial networking products and solutions may also incorporate MP-DCCP as part of their offerings.
- Wireshark also has the possibility to dissect MP-DCCP packets.
- Development of IPERF supporting for MP-DCCP: [https://github.com/NathalieRM/iperf/tree/mpdccp]
- Development of IPERF for Android mobile: [https://github.com/NathalieRM/platform_external_iperf3/tree/android-10.0.0_r40-mpdccp]
- Code developed in the Linux kernel: [https://github.com/telekom/mp-dccp/]

6.4.6.8 Protocol Specific Information

Pros

- Unreliable data transfer with reordering .
- Steering, switching, splitting capabilities.

Cons

- Requires further development.
- Not suitable for reliable or TCP-like traffic.
- Potential issues with unordered traffic arrival in heterogeneous networks; a suitable scheduler is required.

6.4.7 MPUDP

MPUDP had a draft that is expired, the proposal was to use MP-DCCP as the base of a framework to define “MPUDP” [42]. Nonetheless, there is not a specific deployment of MPUDP, it makes sense since UDP is a connectionless protocol and therefore, creating a connection to manage MPUDP sessions

would change the concept. Taking this into account, a possibility could be to rely UDP traffic to the network layer where there are protocols to aggregate different network interfaces and thus use multiple paths.

6.4.7.1 High-level Architecture

The UDP packets would be distributed, and the steering, splitting, and switching could be possible to some extent. For instance, to provide with the multipath functionality to UDP, the ATSSS-LL would be an option, or protocols like FatVAP [43] or that is an 802.11 driver design that aggregates the bandwidth available at nearby APs and load balances traffic across them. This is only for 802.11 connections. Nonetheless, unofficial attempts to deploy it were found.

6.4.7.2 Standards

N/A

6.4.7.3 Use and Deployments

This is a deployment of MPUDP to install in a client and server between the actual client and server, like a gateway that uses a tunnel. It is configured to forward to ONE specific server https://github.com/greensea/mptunnel. In https://github.com/zehome/MLVPN/ several networks are bonded. Since MPUDP is not a protocol per se, this is what is closer to the concept. Official documentation can be found here: [https://mlvpn.readthedocs.io/en/latest/].

6.4.7.4 Support of Steering/Switching/Splitting

N/A

6.4.7.5 Support of Rail Applications

N/A

6.4.7.6 Integration with Transport Network

N/A

6.4.7.7 Availability of Commercial Products and/or Open-Source

N/A

6.4.7.8 Protocol Specific Information

Unofficial attempts to deploy MPUDP were found. The following deployment of MPUDP to install in a client and server between the actual client and server, uses a tunnel. It is just configured to forward to ONE specific server https://github.com/greensea/mptunnel.

https://github.com/zehome/MLVPN/ is another deployment to bond several networks, since MPUDP is not a protocol per se, this is what is closer to the concept. Official documentation here: https://mlvpn.readthedocs.io/en/latest/.

6.4.8 SD-WAN

Software Defined WAN (SD-WAN) is a newer technology that can use load-balancing capabilities for WAN connections.

SD-WAN separates the data plane from the control plane and virtualizes much of the routing functionality. Leveraging the secured control plane, the context definition for the data plane is established. Additionally, the orchestration plane is introduced to provide specific business policies for Quality of Service (QoS). The control plane serves as a centralized entity, making decisions that the data plane follows, thereby reducing node complexity. Some key premises of SD-WAN include reduced network costs, increased speed, enhanced security and visibility through analytics, simplified failover for improved availability, and optimized network performance and bandwidth.

6.4.8.1 High-level Architecture

In the context of SD-WAN, the architecture involves the separation of the data, control, and orchestration planes. The control plane plays a central role in decision-making, while the data plane executes these decisions for efficient forwarding. The addition of the orchestration plane allows the implementation of specific business policies for QoS. SD-WAN aims to simplify network management, reduce hardware requirements, and enhance overall network capabilities. Relevant references include [44], [45] and [46].

6.4.8.2 Standards

The SD-WAN Service Attributes and Service Framework Standard defines the externally visible behaviour of a MEF SD-WAN Service - MEF 70 [47].

6.4.8.3 Use and Deployments

SD-WAN can be deployed to achieve various objectives, including reduced network costs, increased speed, improved security and visibility, enhanced availability through faster failover, and optimized network performance. System administrators can configure SD-WAN gateways based on client policies, with offerings from vendors like Cisco and Fortinet. SD-WAN can work with different technologies simultaneously, such as 5G/4G, Ethernet, or MPLS.

6.4.8.4 Support of Steering/Switching/Splitting

Theoretically speaking, yes. Nonetheless, the mechanisms differ too much and since it is not related to cellular networks and oriented to branch-office or data-center edge.

6.4.8.5 Support of Rail Applications

N/A

6.4.8.6 Integration with Transport Network

The integration of SD-WAN with transport networks involves the separation of the data and control planes, with the control plane making centralized decisions for efficient data plane forwarding. SD-WAN can be configured to work with various transport technologies, such as 5G/4G, Ethernet, or MPLS.

6.4.8.7 Availability of Commercial Products and/or Open-Source

Commercial products from vendors like Cisco and Fortinet offer SD-WAN solutions. The proprietary nature of these solutions may impact clarity on how certain functionalities, such as traffic balancing, are achieved.

6.4.8.8 Protocol Specific Information

N/A

6.4.9 Load Balancing Based IP Routing

Load balancing based IP routing is a multipath technique designed to enhance the transmission capabilities of multihomed devices through packet routing at the IP layer [48]. Unlike transport layer approaches, IP-layer-based load balancing is faster, but it faces challenges as IP-layered methods lack comprehensive knowledge of the traffic profile, making it difficult to provide Quality of Service (QoS) for all types of traffic. This technique involves concepts such as "Flow Splitting" for concurrent link usage to aggregate capacity and "traffic engineering" for optimizing Data Paths.

6.4.9.1 High-level Architecture

The load balancing approach involves creating paths and managing traffic. The number of paths and their opening requirements are crucial considerations, leading to a trade-off between performance and computational needs. Data Paths can be configured concurrently for simultaneous usage or as backups. Backup options, discussed in [49] [50], involve strategies to maintain alternative paths in case of the main path failure. Concurrent multipathing utilizes a scheduler approach, where an entity uses network information to schedule traffic based on network parameters. OSPF version 2 (RFC 2328) and other routing algorithms, such as the one proposed in [51], fall under the Quasi-Static Load balancing concept.

6.4.9.2 Standards

N/A

6.4.9.3 Use and Deployments

Load balancing based IP routing aims to distribute traffic across different networks connected to a specific router using information at the network layer. The technique requires an algorithm, known as a scheduler, to decide how to route packets. Various routing algorithms and approaches, such as RTT-based routing presented in [52], [53], have been proposed to maximize throughput and utilize bandwidth aggregation with different policies.

6.4.9.4 Support of Steering/Switching/Splitting

N/A

6.4.9.5 Support of Rail Applications

N/A

6.4.9.6 Integration with Transport Network

The integration involves utilizing network layer information to distribute traffic across connected networks. Various approaches, including virtual tunnelling with bandwidth aggregation, have been explored, as presented in [52], [53]. These approaches compare RTT-based routing with Throughput-based and Equal Cost Multipath (ECMP) routing.

6.4.9.7 Availability of Commercial Products and/or Open-Source

N/A

6.4.9.8 Protocol Specific Information

N/A