Workgroup: ICCRG Working Group
Internet-Draft:
draft-amend-iccrg-multipath-reordering-01
Published: 2 November 2020
Intended Status: Experimental
Expires: 6 May 2021
Authors: M. Amend
DT
D. von Hugo
DT

Multipath sequence maintenance

Abstract

This document discusses the issue of packet re-ordering which occurs as a specific problem in multi-path connections without reliable transport protocols such as TCP. The topic is relevant for devices connected via multiple accesses technologies towards the network as is foreseen e.g. within Access Traffic Selection, Switching, and Splitting (ATSSS) service of 3rd Generation Partnership Project (3GPP) enabling fixed mobile converged (FMC) scenario.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 May 2021.

Copyright Notice

Table of Contents

1. Introduction
2. State of the Art
3. Problem Statement
4. Scheduling mechanisms
5. Resequencing mechanisms
   5.1. Passive
   5.2. Exact
   5.3. Static Expiration
   5.4. Adaptive Expiration
   5.5. Delay Equalization
   5.6. Fast Packet Loss Detection
      5.6.1. Connection sequencing
      5.6.2. Per-path sequencing
      5.6.3. Combination connection and per-path sequencing
6. Recovery mechanisms
   6.1. FEC (Forward Error Correction)
   6.2. Network Coding
7. Retransmission mechanisms
   7.1. Signalling
   7.2. Anticipated
   7.3. Flow-selection
8. Informative References
Authors' Addresses

1. Introduction

Mobile end user devices nowadays are mostly equipped with multiple network interfaces allowing to connect to more than one network at a time and thus increase data throughput, reliability, coverage and so on. Ideally the user data stream originating from the application at the device is split between the available (here: N) paths at the sender side and re-assembled at an intermediate aggregation node before transmitted to the corresponding host in the network as depicted in Figure 1.

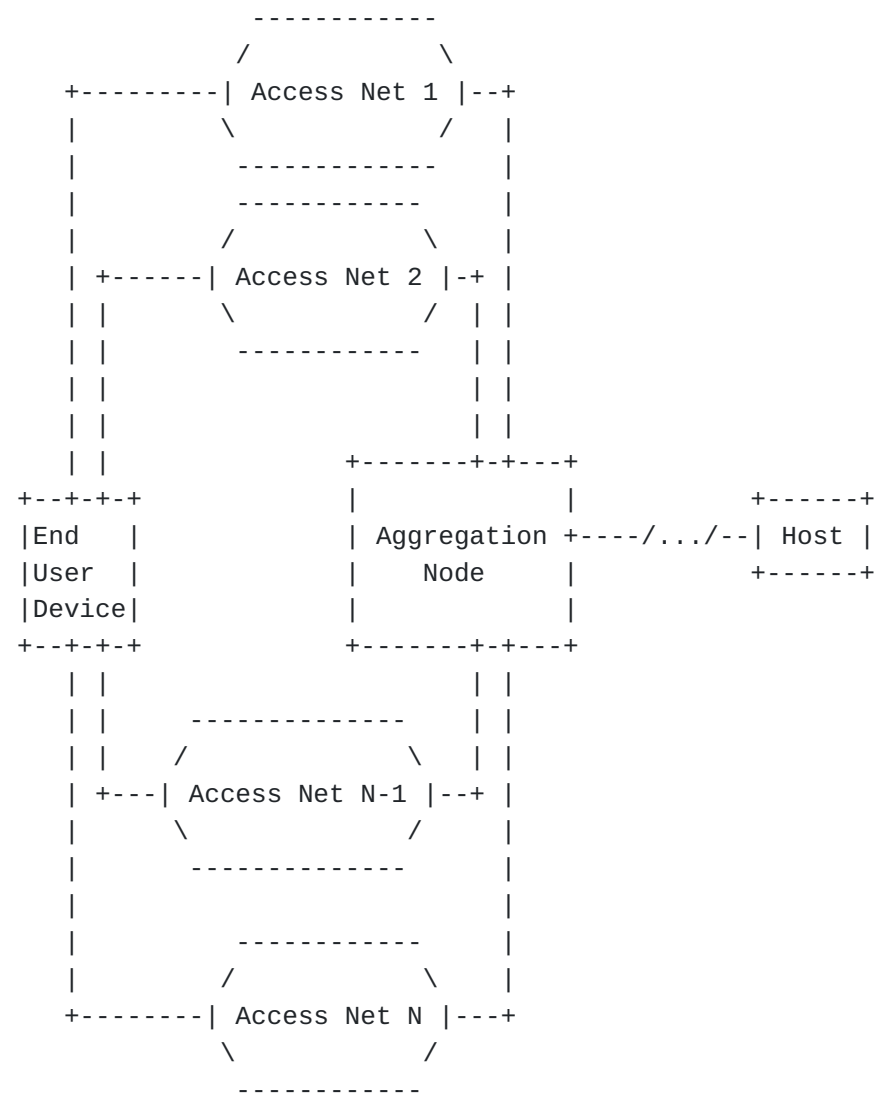

Figure 1: Reference Architecture for multi-path re-ordering

However, when several paths are utilized concurrently to transmit user data between the sender and the receiver, different characteristics of the paths in terms of bandwidth, delay, or error proneness can impact the overall performance due to delayed packet arrival and need for re-transmit in case of lost packets. Without further arrangements the original order of packets at the sending UE side is no longer maintained at the receiving host and a re-ordering or re-arrangement has to occur before delivery to the application at the far end site. This can be performed at earliest at the aggregation node with a minimum additional delay due to re-transmission requests or at latest either by the application on the host itself or the transmission protocol.

It is a goal of the present document to collect and describe mechanisms to maintain the sequence of split traffic over multiple paths. These mechanisms are generic and not dedicated to a specific

multipath network protocol, but give clear guidance on requirements and benefits to maintainers of multipath network protocols.

2. State of the Art

Regular TCP protocol [RFC0793] offers such mechanism with queues for in-order and out-of order (including damaged, lost, duplicated) arrival of packets.

This is also provided by MPTCP [RFC6824] as the first and successful Multipath protocol which however also requires new methods as sequence numbers both on (whole) data (stream) and subflow level to ensure in-order delivery to the application layer on the receiver side [RFC8684]. Moreover, careful design of buffer sizes and interpretation of sequence numbers to distinguish between (delayed) out-of-order packets and completely lost ones has to be considered.

[I-D.bonaventure-iccrg-schedulers] already reflects on proper packet scheduling schemes (at the sender side) to reduce the effort for re-assembly or even make such (time consuming) treatment unnecessary.

MP-QUIC [I-D.deconinck-quic-multipath] introduces the concept of uniflows with own IDs claiming to get rid of additional sequence numbers for re-ordering as required in Multipath TCP [RFC6824]. Although [] admits that statistical performance information should help a host in deciding on optimum packet scheduling and flow control a dedicated packet scheduling policy is out of scope of that document. A further improvement versus MPTCP can be achieved by decoupling paths used for data transmission from those for sending acknowledgments (ACKs) or claiming for re-transmission by NACKs to not introduce further latency.

[I-D.ietf-quic-recovery] specifies algorithms for QUIC Loss Detection and Congestion Control by using measurement of Round Trip Time (RTT) to determine when packets should be retransmitted. Draft [I-D.huitema-quic-ts] proposes to enable one way delay (1WD) measurements in QUIC by defining a TIME_STAMP frame to carry the time at which a packet is sent and combine the ACKs sent with a timestamp field and thus allow for more precise estimation of the (one-way) delay of each uniflow, assisting proper scheduling decisions.

Also other protocols as Multi-Access Management Services (MAMS) [RFC8743] consider the need for re-ordering on User Plane level which may be done at network and client level by introducing a new Multi-Access (MX) Convergence Layer. [I-D.zhu-intarea-mams-user-protocol] introduces accordingly Traffic Splitting Update (TSU) messages and Packet Loss Report (PLR) messages including beside

others Traffic Splitting Parameters and an expected next (in-order) sequence number, respectively.

[I-D.zhu-intarea-gma] on Generic Multi-Access (GMA) Convergence Encapsulation Protocols introduces a trailer-based encapsulation which carries one or multiple IP packets or fragments thereof in a Protocol Data Unit (PDU). At receiver side PDUs with identical Sequence Numbers (in the trailer) are to be placed in the relative order indicated by a so-called Fragment Offset.

3. Problem Statement

Assuming for simplicity the minimum multipath scenario with two separate paths for transmission of a flow of packets with sequence numbers (SN) SN1 ... SM. In case the scheduling of packets is done equally to both paths and path 2 exhibits a delay of the duration of transmission time required for e.g. two packets (assuming fixed packet size and same constant data for both paths) for an exemplary App-originated sequence of packets as SN1 SN2 SN3 SN4 SN5 SN6 SN7 SN8 ... the resulting sequence of packets could look as depicted in Figure 2 which of course depends on the queue processing and buffering at the Aggregation Proxy.

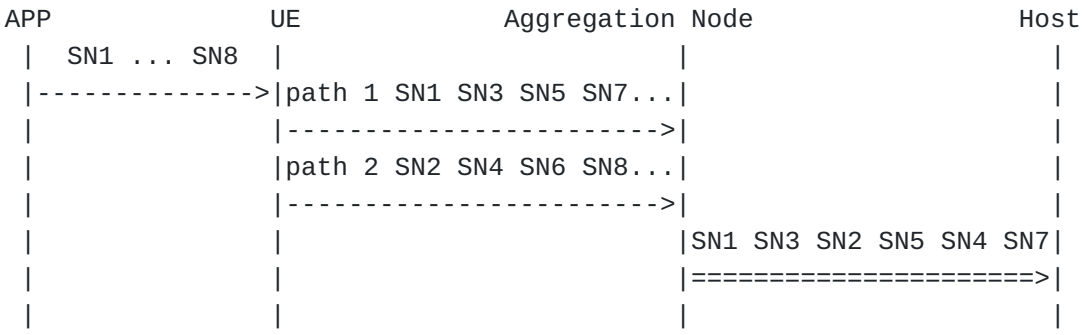

Figure 2: Exemplary data transmission for a dual-path scenario

In such a case re-ordering at the Aggregation Node would be simple and straight forward. It even could be avoided if the scheduling would already take the expected different delays into account (e.g. by pre-delaying the traffic on path 1 thus of course not leveraging the lower delay). Different from this simplistic scenario in general the data rate on both paths will vary in time and be not equal, also different and variable latency (jitter) per path will be introduced and in addition loss of packets as well as potential duplication may occur making the situation much more complicated. In case of loss detection after a threshold waiting time a retransmission could be initiated by the Host or if possible already by the Aggregation Node. Alternatively the UE could send redundant packets in advance coded in such a way that it allows for derivation of e.g. one lost

packet per M correctly received ones or by a (real-time) application able to survive singular lost packets.

Holding multiple queues and a large enough buffer both at UE and at the Aggregation Node would be required to apply proper scheduling at UE and reordering during re-assembly at Aggregation Node to mitigate the sketched impact of multiple paths variable characteristics in terms of transmission performance.

...

4. Scheduling mechanisms

Scheduling mechanisms decide on sender side how traffic is distributed over the paths of a multipath-setup. [I-D.bonaventure-iccrg-schedulers] gives an overview of possible distribution schemes. For this document it is assumed, that schedulers are used, which simultaneously distribute traffic over more than one path, whereas path characteristics differ between those multiple paths (e.g. a latency difference exists). While on the one hand, the traffic scheduling causes out-of-order multipath delivery when simultaneously utilize heterogeneous paths, it can also be used to mitigate this problem. Pre-delaying data on a fast path, according to the latency difference of the slowest path is aimed e.g. by OTIAS [OTIAS], DAPS [DAPS] and BLEST [BLEST]. However, the success is much dependent on the accuracy of path information like path latency, throughput and packet loss rate. In heterogeneous and volatile environments most often such information have to be estimated, e.g. using congestion control. That means, it takes at least one RTT to gain first indications and probably several RTTs to converge to a worthwhile accuracy. Changes of path characteristics in sub-RTT time frames puts such a system to test. Dependent on the demand on in-order delivery and/or the accuracy of the relevant path information, scheduling might be an exclusive alternative or can be applied in conjunction with other discussed mechanisms in this document.

Scheduling will not help to overcome any degree of out-of-order delivery, when the scheduling goal is different to this. For example a strict cost prioritization of Wi-Fi over cellular access in a mobile phone might be assumed counterproductive.

5. Resequencing mechanisms

Resequencing mechanism are responsible to modify the sequence of received data split over multiple paths according to a sequencing scheme. The degree of resequencing can reach from no measure up to re-generating the exact order.

Typically at least one sequencing scheme, describing the order of how data was generated on sender side is prerequisite. This is

referred to as "connection sequencing". Under certain circumstances an additional sequencing scheme per path of the multi-path setup can be leveraged, to optimize packet loss detection and is further elaborated in Section 5.6. For most multipath protocols both sequencing schemes are already available. Packet loss detection becomes important when multipath protocols are applied which do not guarantee successful transmission as TCP achieves by acknowledgement of successful reception. For example [I-D.amend-tsvwg-multipath-dccp] or the combination of [I-D.deconinck-quic-multipath] and [I-D.ietf-quic-datagram] are unreliable protocols in that sense.

For simplicity all the mechanism described in the following are explained based on two paths but in principle would work with an unlimited number though.

5.1. Passive

This approach includes no active change or reordering at the transport level and purely re-combines the packet flows incoming from both paths as is. All modification of the resulting sequence of packets is left to the application at the end node. Here no processing delay is added due to the resequencing but since no early packet loss detection with subsequent re-transmission request on transport level is possible the risk of a larger delay due to late loss detection at the application will arise in case of lossy connections.

5.2. Exact

This approach covers all mechanisms which attempt to re-generate the original order of packets in the flow exactly, independent of the expected or resulting delay due to waiting time for all packets on all paths to arrive. In case of unreliable transport protocols this may result in a large delay due to Head-of-Line blocking and for actual packet loss in a remaining packet gap which causes a stand still without an option to recover. For applications demanding near real-time delivery of packets it should not be applied.

5.3. Static Expiration

This method to detect and decide on packet loss assumes a certain fixed time threshold for the gap between packets within a sequence after re-combination of both paths. Since a possible re-transmission - either in the multipath system internal or based on the piggybacked protocol/service - will possibly not be requested before this threshold an additional delay in the overall latency budget this simple approach is only recommended for non-time critical applications. Every packet loss or simultaneous transmission of data

over the short and long latency path will cause spikes in the service perceived latency.

5.4. Adaptive Expiration

Here the packet gap is assumed as packet loss after exceeding a flexibly decided on time threshold which may be derived dynamically from the differences between latencies both paths exhibit. As the latency may vary due to propagation conditions or routing paths this latency difference has to be monitored and statistically evaluated (smoothed) which introduces additional effort. A possible solution for this is the determination of the the one way latency as described in [I-D.song-mptcp-owl] or sending available RTT information from the sender from which the receiver can calculate the latency difference.

5.5. Delay Equalization

This is an ordering mechanism which delays data forwarding on the faster path by the latency difference to the slower path. Ideally the resequencing effort on the aggregated packet flow can be greatly reduced up to no resequencing at all. Due to time variation in path delays (jitter) and delay differences and required time for decision and feedback on the delay some re-sequencing still remains to be executed. Similar to Section 5.4, the latency difference is required. Strictly speaking this method allows to avoid resequencing based on sequencing information. However, the overall delay may be larger since the advantage of the short-delay path is not exploited. In combination with Section 5.3 or Section 5.4 resequencing can be added with a presumably lower resequencing effort to scenarios without delay equalization. The essence is a in-order stream with a unified latency across the multiple paths.

5.6. Fast Packet Loss Detection

The following sections describe methods to achieve unambiguous detection of packet loss independent from thresholds in Section 5.3 or Section 5.4. Furthermore, packet loss can be differentiated from delayed delivery. The benefit is a much faster decision plane based on monitoring the sequence space of consecutive packets. For that, the sequencing coming along with the receiver based re-sequencing is further leveraged. Two sequencing schemes are considered here, the connection and the per-path sequencing.

5.6.1. Connection sequencing

Connection sequencing marks the outgoing packets in the order they enter the multipath system and is independent from a particular selected path for transmission. [to be continued...]

5.6.2. Per-path sequencing

Per-path sequencing is a path inherent sequencing mechanism valid in the particular path domain only. [to be continued...]

5.6.3. Combination connection and per-path sequencing

While the benefits from the individual sequencing schemes above can be combined, a further benefit crystallizes. [to be continued...]

6. Recovery mechanisms

Recovering packets, in particular lost packets or assumed lost packets on receiver side avoids re-transmission and potentially mitigates the resequencing process in respect to detecting packet loss. Shorter latencies will be an expected outcome. Discussing the complexity, computation overhead and reachable benefit is subject of this section.

6.1. FEC (Forward Error Correction)

This approach is based on introduction of redundancy to user data to detect errors and subsequently reconstruct data in case of a limited number of bit or Byte errors. As such packet with corrupted data can be recovered up to a certain degree but in case of a too high bit error rate (BER) a packet is completely lost. However, in combination with scrambling, i.e. the sequence of original data stream is distributed over multiple packets and re-compiled afterwards also data from lost packets could be recovered. As such methods introduce additional delay and overhead it is mainly applied in case of long re-transmission delays as e.g. is typical for satellite transmission. FEC can be applied to each path separately (e.g., if they exhibit deviating performance characteristics to not degrade the 'better one') or in an overall FEC fashion before split and recombination which would support scrambling and facilitate recovery of completely lost packets on the 'worse path'. Unsuccessful application of FEC may enable quick detection of unrecoverable errors in a packet and thus trigger re-transmission from the sender side before time-out.

6.2. Network Coding

In linear network coding (LNC) network nodes (or interfaces of a device) do not simply relay the packets of information they receive, but combine several packets together for transmission. After reception of combined and separate packets the maximum possible information flow in a network can be detected and throughput, efficiency and scalability, as well as resilience to attacks and eavesdropping can be improved. The method in general improves with the number of paths in excess of two. According to [wikipedia]

drawback of LNC are high decoding computational complexity, high transmission overhead, and linear dependency among coefficients vectors for en- and decoding. Triangular network coding (TNC) addresses the high encoding and decoding computational complexity without degrading the throughput performance, with code rate comparable to that of linear network coding.

7. Retransmission mechanisms

Re-transmission becomes interesting when it can help to reduce the time spent on waiting for outstanding packets for re-sequencing. In particular scenarios when for example a known path RTT (Round Trip Time) lets expect a shorter time to re-transmit than wait for packet loss detection, a likely scenario in e.g. Figure 1. It could also avoid a potential late triggering of re-transmission by the end-to-end service.

7.1. Signalling

Signal to the sender to re-transmit a missing packet. This requires a send buffer which keeps information for a reasonable time, which allows the beneficial use of this mechanism. [to be continued]

7.2. Anticipated

Re-transmit data without being notified from the receiver when packet loss can be assumed, e.g. from lower layer information. [to be continued]

7.3. Flow-selection

Repeating data on the same path is not always useful. In some scenarios it make sense to re-transmit data on another path, e.g. when the original path is broken or another path is known to be faster.

- *Requires path independent identification of data, e.g. the connection sequencing

- *Has to consider MTU discrepancies between paths

Flow selection for re-transmitting data can be combined with Section 7.1 or Section 7.2. In certain scenarios data to be re-transmitted can be duplicated across paths to increase reliability. [to be continued]

8. Informative References

[BLEST] Ferlin, S., Alay, Oe., Mehani, O., and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for

heterogeneous networks", IFIP Networking Conference , May 2014, <https://doi.org/10.1109/IFIPNetworking.2016.7497206>.

[DAPS] Kuhn, N., Lochin, E., Mifdaoui, A., Sarwar, G., Mehani, O., and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport", ICC IEEE International Conference on Communications, June 2014, <https://doi.org/10.1109/ICC.2014.6883488>.

[I-D.amend-tsvwg-multipath-dccp] Amend, M., Bogenfeld, E., Brunstrom, A., Kassler, A., and V. Rakocevic, "DCCP Extensions for Multipath Operation with Multiple Addresses", Work in Progress, Internet-Draft, draft-amend-tsvwg-multipath-dccp-03, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-amend-tsvwg-multipath-dccp-03.txt>.

[I-D.bonaventure-iccrg-schedulers] Bonaventure, O., Piraux, M., Coninck, Q., Baerts, M., Paasch, C., and M. Amend, "Multipath schedulers", Work in Progress, Internet-Draft, draft-bonaventure-iccrg-schedulers-01, 9 September 2020, <http://www.ietf.org/internet-drafts/draft-bonaventure-iccrg-schedulers-01.txt>.

[I-D.deconinck-quic-multipath] Coninck, Q. and O. Bonaventure, "Multipath Extensions for QUIC (MP-QUIC)", Work in Progress, Internet-Draft, draft-deconinck-quic-multipath-05, 20 August 2020, <http://www.ietf.org/internet-drafts/draft-deconinck-quic-multipath-05.txt>.

[I-D.huitema-quic-ts] Huitema, C., "Quic Timestamps For Measuring One-Way Delays", Work in Progress, Internet-Draft, draft-huitema-quic-ts-03, 10 August 2020, <http://www.ietf.org/internet-drafts/draft-huitema-quic-ts-03.txt>.

[I-D.ietf-quic-datagram] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-datagram-01, 24 August 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-datagram-01.txt>.

[I-D.ietf-quic-recovery] Iyengar, J. and I. Swett, "QUIC Loss Detection and Congestion Control", Work in Progress, Internet-Draft, draft-ietf-quic-recovery-32, 20 October 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-recovery-32.txt>.

[I-D.song-mptcp-owl]
Song, F., Zhang, H., Chan, A., and A. Wei, "One Way Latency Considerations for MPTCP", Work in Progress, Internet-Draft, draft-song-mptcp-owl-06, 18 June 2019, <http://www.ietf.org/internet-drafts/draft-song-mptcp-owl-06.txt>.

[I-D.zhu-intarea-gma] Zhu, J. and S. Kanugovi, "Generic Multi-Access (GMA) Encapsulation Protocol", Work in Progress, Internet-Draft, draft-zhu-intarea-gma-07, 14 May 2020, <http://www.ietf.org/internet-drafts/draft-zhu-intarea-gma-07.txt>.

[I-D.zhu-intarea-mams-user-protocol]
Zhu, J., Seo, S., Kanugovi, S., and S. Peng, "User-Plane Protocols for Multiple Access Management Service", Work in Progress, Internet-Draft, draft-zhu-intarea-mams-user-protocol-09, 4 March 2020, <http://www.ietf.org/internet-drafts/draft-zhu-intarea-mams-user-protocol-09.txt>.

[OTIAS] Yang, F., Wang, Q., and P.D. Amer, "Out-of-Order Transmission for In-Order Arrival Scheduling for Multipath TCP", AINAW 28th International Conference on Advanced Information Networking and Applications Workshops, May 2014, <https://doi.org/10.1109/WAINA.2014.122>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <https://www.rfc-editor.org/info/rfc6824>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/info/rfc8684>.

[RFC8743] Kanugovi, S., Baboescu, F., Zhu, J., and S. Seo, "Multiple Access Management Services Multi-Access Management Services (MAMS)", RFC 8743, DOI 10.17487/RFC8743, March 2020, <https://www.rfc-editor.org/info/rfc8743>.

Authors' Addresses

Markus Amend

Deutsche Telekom
Deutsche-Telekom-Allee 9
64295 Darmstadt
Germany

Email: Markus.Amend@telekom.de

Dirk von Hugo
Deutsche Telekom
Deutsche-Telekom-Allee 9
64295 Darmstadt
Germany

Email: dirk.von-Hugo@telekom.de