A   M. Amend   D. von Hugo
         uDT   DT
         t
         h
         o
         r
         s
         :

Multipath sequence maintenance

Abstract

   This document discusses the issue of packet reordering 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

Copyright Notice

Table of Contents

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.

```
                      ------------
                     /            \
        +---------| Access Net 1 |--+
        |           \            /  |
        |            ------------   |
        |            ------------   |
        |           /            \  |
        | +------| Access Net 2 |-+ |
        | |        \            /  | |
        | |         ------------   | |
        | |                        | |
        | |                        | |
        | |              +-------+-+---+
     +--+-+-+            |         |          +------+
     |End   |            | Aggregation +----/.../--| Host |
     |User  |            |   Node      |          +------+
     |Device|            |             |
     +--+-+-+            +-------+-+---+
        | |                      | |
        | |      --------------  | |
        | |     /              \ | |
        | +---| Access Net N-1 |--+ |
        |      \              /    |
        |       --------------     |
        |                          |
        |       ------------       |
        |      /            \      |
        +--------| Access Net N |---+
                  \            /
                   ------------
```
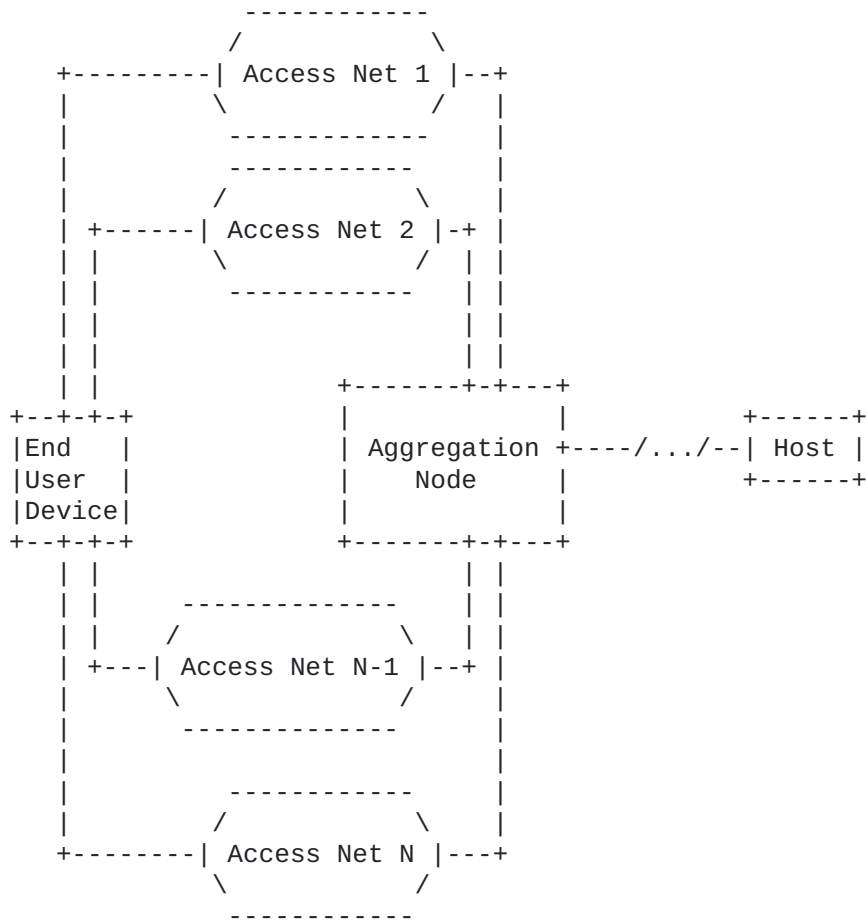
                Figure 1: Reference Architecture for multi-path reordering

   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 reordering
   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 [RFC8684] 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 reordering as required in Multipath TCP [RFC8684].
Although [I-D.liu-multipath-quic] 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 reordering 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.

```
APP                   UE                 Aggregation Node                Host
 |   SN1 ... SN8   |                       |                               |
 |---------------->|path 1 SN1 SN3 SN5 SN7...|                             |
 |                 |------------------------>|                             |
 |                 |path 2 SN2 SN4 SN6 SN8...|                             |
 |                 |------------------------>|                             |
 |                 |                       |SN1 SN3 SN2 SN5 SN4 SN7|        |
 |                 |                       |======================>|        |
 |                 |                       |                               |
```
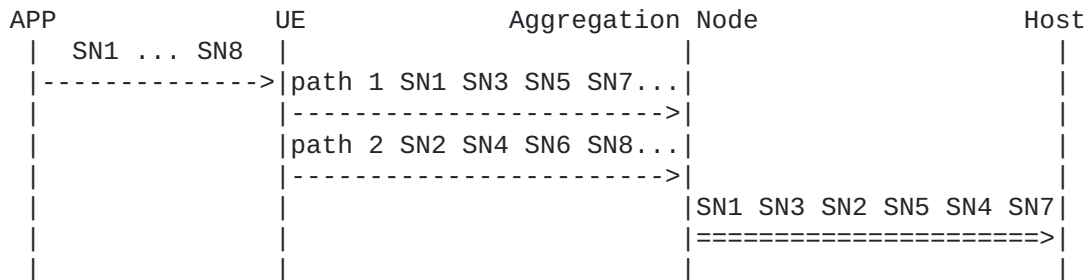
Figure 2: Exemplary data transmission for a dual-path scenario

In such a case reordering 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 put 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.

[AOPS] proposes to use a predictive Adaptive Order Prediction
Scheduling (AOPS) mechanism considering both the anticipated time of
packet delivery and the reliability of each path to optimize the
traffic scheduling for MP-DCCP, thus coping with reordering and
achieving in-order delivery.

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 mechanisms 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 any
other amount 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. A possible re-transmission - either in the multipath system internally or based on the piggybacked protocol/service - will possibly not be requested before this threshold is exceeded. Thus an additional delay in the overall latency budget will occur so that 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 the required time for decision and feedback on the delay, some re-sequencing still remains to be executed. Similar to Section 5.4, explicit knowledge of 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. After arrival at the aggregation node the lowest packet sequence number at each of the multiple paths is compared the that of the last correctly received packet. When the numbers are not consecutive (i.e., when on all paths a higher number is received than the next expected in-order packet), an overall packet loss is detected. While only a single comparison of packet numbers has to be performed and the out-of-order arrival on a single path can be partly compensated this scheme does not allow for immediate detection of where reordering happens.

5.6.2. Per-path sequencing

Per-path sequencing is a path inherent sequencing mechanism valid in the particular path domain only. In this case the packets are marked by path-specific sequence numbers at the sender side and at each interface of the aggregation node the sequence numbers of arriving packets are compared on per-path level. When a higher sequence number is received than the one which is waited for (next expected in-order packet), a packet loss for this specific path is declared. This may prevent partly misinterpretation of out-of-order arrival as packet loss and allow for path specific countermeasures towards overall performance improvement, as, e.g., chosing a more robust transmission technique for this path.

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. Since the out-of-order arrival is detected on per-path basis, the path specific out-of-order delivery rate can be used as a criterion to choose repair parameters
on a per-path basis (which thus may work more efficiently). In addition the decision on the path selection and weighting can be made based on this criterion. Thus an improved overall performance can be achieved in this case. [to be checked/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 [COPE] drawbacks 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 LNC. TNC is therefore advantageous for implementation on small devices mobile phones and wireless sensors with limited processing capability and power supply [TNC].

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. On the other hand for sake of resource efficiency the amount of unnecessary retransmissions should be limited to not degrade the overall throughput of the connection.

7.1. Signaling

In case of detected packet loss the receiver has to send a corresponding signalling message to the sender to re-transmit a missing packet. This is the traditional way of negative acknowledgement in case of missing the correct reception of packets

within a time window and sending a repeat-request. This approach
requires a send buffer which keeps information for a reasonable
time, thus allowing the beneficial use of this mechanism. On the
other
hand the additional delay in terms of at least once the RTT until
the
lost packet is correctly received results in performance degradation

for time-critical applications. ... [to be continued?]

7.2. Anticipated

To speed up the induced re-transmission delay a pro-active or
anticipated approach would allow to trigger the sender to re-
transmit data without needing to wait for notification from the
receiver. This method can be applied when the assumed packet loss
can be estimated based on other data, e.g., from lower layer, such
as information on path or
link quality degradation derived from, e.g., an increased raw BER
detected by FEC mechanism (see Section 6.1).
[to be continued/extended?]

7.3. Flow-selection

Repeating data on the same path is not always useful. In some
scenarios it makes sense to re-transmit data on another path, e.g.,
when the original path is broken or another path is known to provide
higher throughput or lower packet loss. To apply such a selection of
the flow for re-transmission.

  *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
detection mechanisms as described in Section 7.1 or Section 7.2.

7.4. Other re-transmission issues

In certain scenarios data to be re-transmitted can be duplicated
across paths (either in advance or after loss detection) to increase
reliability and reduce potential overall transmission delay.
However, such approaches decrease the resource efficiency and reduce

the overall user throughput. A more pro-active measure would
be to encode multiple packets either on per-path or on per-
connection
basis in a single 'repair packet' in 'XOR style' to be injected
after
a set of packets (similarly as described in [COPE]. This would
allow
to recreate exactly one lost packet out of the set in case the
others
have been correctly received. Depending on the anticipated loss rate

the amount of packets within a set is chosen to more efficiently use

the transmission resources. [to be continued]

8. Security Considerations

This document does not add any additional security considerations in

addition to the ones introduced by multipath extensions to other
transmission protocols as, e.g., described for MPTCP in [RFC8684].
Also the described issues for GMA [I-D.zhu-intarea-gma], MP-DCCP
[I-D.amend-tsvwg-multipath-dccp], and MP-QUIC
[I-D.liu-multipath-quic] may apply here.

9. Informative References

[AOPS]   Huang, C., Chen, Y., and S. Linn, "Packet scheduling and
            congestion control schemes for Multipath Datagram
            Congestion Control Protocol", The Computer Journal 58,
            no. 2, pp. 188--203] , February 2015.

[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>.

[COPE]   Katti, S., Rahul, H., Katabi, W.H.D., Medard, M., and J.
            Crowcroft, "XORs in The Air: Practical Wireless Network
            Coding", September 2006, <http://nms.csail.mit.edu/
            ~sachin/papers/copesc.pdf>.

[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., Hugo, D. V., Brunstrom, A., Kassler, A.,
            Rakocevic, V., and S. Johnson, "DCCP Extensions for
            Multipath Operation with Multiple Addresses", Work in
            Progress, Internet-Draft, draft-amend-tsvwg-multipath-
            dccp-05, 12 July 2021, <https://www.ietf.org/archive/id/
            draft-amend-tsvwg-multipath-dccp-05.txt>.

[I-D.bonaventure-iccrg-schedulers]
            Bonaventure, O., Piraux, M., Coninck, Q. D., Baerts, M.,
            Paasch, C., and M. Amend, "Multipath schedulers", Work in
            Progress, Internet-Draft, draft-bonaventure-iccrg-
            schedulers-02, 25 October 2021, <https://www.ietf.org/
            archive/id/draft-bonaventure-iccrg-schedulers-02.txt>.

[I-D.deconinck-quic-multipath] Coninck, Q. D. and O. Bonaventure,
            "Multipath Extensions for QUIC (MP-QUIC)", Work in
            Progress, Internet-Draft, draft-deconinck-quic-
            multipath-07, 3 May 2021, <https://www.ietf.org/archive/
            id/draft-deconinck-quic-multipath-07.txt>.

[I-D.huitema-quic-ts]
          Huitema, C., "Quic Timestamps For Measuring One-Way
          Delays", Work in Progress, Internet-Draft, draft-huitema-
          quic-ts-06, 12 September 2021, <https://www.ietf.org/
          archive/id/draft-huitema-quic-ts-06.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-06, 5
          October 2021, <https://www.ietf.org/archive/id/draft-
          ietf-quic-datagram-06.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-34, 14 January
          2021, <https://www.ietf.org/archive/id/draft-ietf-quic-
          recovery-34.txt>.

[I-D.liu-multipath-quic] Liu, Y., Ma, Y., Huitema, C., An, Q., and
          Z. Li, "Multipath Extension for QUIC", Work in Progress,
          Internet-Draft, draft-liu-multipath-quic-04, 5 September
          2021, <https://www.ietf.org/archive/id/draft-liu-
          multipath-quic-04.txt>.

[I-D.song-mptcp-owl] Song, F., Zhang, H., Chan, H. A., and A. Wei,
          "One Way Latency Considerations for MPTCP", Work in
          Progress, Internet-Draft, draft-song-mptcp-owl-06, 18
          June 2019, <https://www.ietf.org/archive/id/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-12, 21 October
          2021, <https://www.ietf.org/archive/id/draft-zhu-intarea-
          gma-12.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,
          <https://www.ietf.org/archive/id/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>.

[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>.

[TNC]
Qureshi, J., Foh, C.H., and J. Cai, "Optimal solution for the index coding problem using network coding over GF(2)", June 2012, <https://doi.org/10.1109/SECON. 2012.6275780>.

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