Network Working Group
Internet-Draft
Intended status: Experimental
Expires: June 26, 2017

B. Trammell
M. Kuehlewind
ETH Zurich
December 23, 2016

Path Layer UDP Substrate Specification
draft-trammell-plus-spec-00

Abstract

This document specifies a common Path Layer UDP Substrate (PLUS) wire image for encrypted transport protocols carried over UDP. The base PLUS header carries information for driving a minimal state machine at middleboxes described in [I-D.trammell-plus-statefulness], and provides optional exposure of additional information to devices along the path using the mechanism described in [I-D.trammell-plus-abstract-mech].

Table of Contents

1. Introduction

This document defines a wire image for a Path Layer UDP Substrate (PLUS), for limited exposure of information from encrypted, UDP-encapsulated transport protocols. The wire image implements signaling to drive the minimal state machine defined in [I-D.trammell-plus-statefulness] as well as optional exposure of additional information to devices along the path using the mechanism described in [I-D.trammell-plus-abstract-mech].

As discussed in [I-D.hardie-path-signals], basic information about flows currently exposed by TCP are missing from transport protocols that encrypt their headers. Given the ossification of protocol wire images due to the widespread deployment of stateful network devices that rely on header inspection, this header encryption is necessary to support transport protocol evolution. However, the loss of basic information for on-path state maintenance as well as network performance measurement, diagnostics, and troubleshooting via header encryption makes network management more difficult. The PLUS wire image defined by this document aims to mitigate this difficulty, allowing deployment of encrypted protocols without loss of essential in- network functionality.

This wire image is intended primarily to support state maintenance and measurement; the principles of measurement and primitives we aim

to support are drawn from recent work on explicit measurability in protocol design [IPIM].

2. State Maintenance and Measurement: Basic Header

Every packet in each direction of a flow using PLUS MUST carry either a PLUS basic or PLUS extended header. The PLUS basic header supports multiplexing using a connection token; basic state maintenance using association and confirmation signals, packet serial numbers, and a two-way stop signal; and basic measurability using packet serial number echo. The format of the basic header, together with the UDP header, is shown in Figure 1.

The extended header is defined in Section 3.

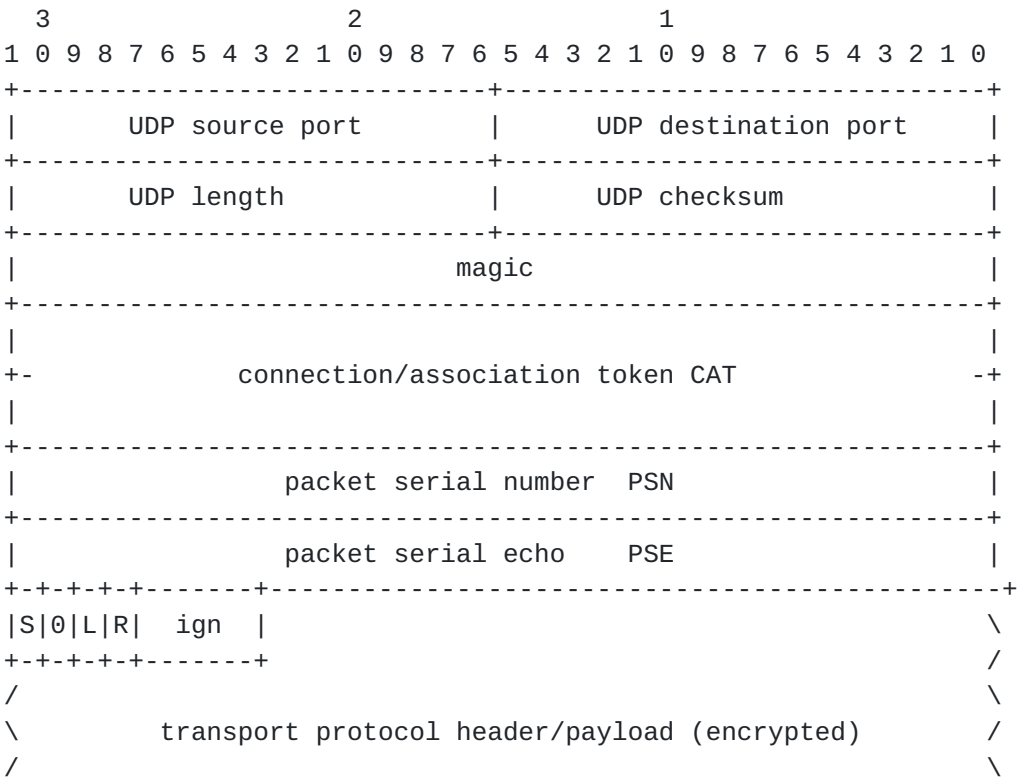

Figure 1: PLUS header with basic exposure

Fields are encoded in network byte order and are defined as follows:

- magic: A 32-bit number identifying this packet as carrying a PLUS header. This magic number is chosen to avoid collision with possible values of the first four bytes of widely deployed protocols on UDP. Should the QUIC [I-D.ietf-quic-transport] header be defined to place the version number in the first four bytes of the packet, this number should be compatible with the

QUIC version numbering scheme. The value 0xd8007ffe has been provisionally selected for the PLUS magic number based of experience with the SPUD prototype, and a cursory survey of common UDP protocols.

- Connection/Association Token (CAT): A 64-bit token identifying this association. The CAT should be chosen randomly by the connection initiator. The CAT performs two functions in the PLUS header:
  - Multiplexing: PLUS packets on the same 5-tuple with a different CAT value are taken to belong to a separate flow, with completely separate state.
  - Rebinding: A PLUS packet sharing one endpoint (source address/port pair, or destination address/port pair) and the CAT with an existing flow is taken to belong to that flow, since the other endpoint identifier has changed due to a mobility event or address translation change.
- Packet Serial Number (PSN): A 32-bit serial number for this packet. The first PSN for each direction in a flow is chosen randomly, and subsequent packets increment the PSN by one. The PSN wraps around.
- Packet Serial Echo (PSE): The most recent PSN seen by the sender in the opposite direction before this packet was sent.
- Flags byte: eight bits carrying additional flags:
  - Stop flag (S): Packet carries a stop or stop confirmation when set.
  - Extended Header bit: Flag bit 0x40 is set to zero in packets with a Basic Header.
  - LoLa flag (L): Packet is latency sensitive and prefers drop to delay when set.
  - RoI flag (R): Packet is not sensitive to reordering when set.
  - Ignored: Bits 0-3 are ignored, and available for use by the overlying transport.

Since PLUS is designed to be used for UDP-encapsulated, encrypted transport protocols, overlying transports are presumed to provide encryption and integrity protection for their own headers. For the

sake of efficiency, it is also assumed that this integrity protection can be extended to the bits in the PLUS Basic Header.

2.1. Sender Behavior

When a sender has a packet ready to send using PLUS, it determines the values in the Basic Header as follows:

- The magic number is set to the constant 0xd8007ffe.
- If the sender is the flow initiator, and the packet is the first packet in the flow, the sender selects a cryptographically random 64-bit number for the CAT. When multiplexing, it must ensure the PSN is not already in use for the 5-tuple. Otherwise, the sender uses the CAT associated with the flow.
- If the packet is the first packet in the flow in this direction, the sender selects a cryptographically random 32-bit number for the PSN. Otherwise, the sender adds one to the PSN on the last packet it sent in this flow, and uses that value for the PSN. If the last PSN is 0xffffffff, it wraps around, setting the PSN to 0x00000000.
- If the packet is the first packet in the flow in this direction, the sender sets the PSE to 0x00000000. Otherwise it sets the PSE to the PSN of the last packet seen in the opposite direction.
- If the overlying transport determines that this packet is the last to be sent in this direction, the sender sets the S flag; see Section 2.3.2 for details.
- If the overlying transport determines that this packet is loss-insensitive but latency-sensitive, the sender sets the L flag.
- If the overlying transport determines that this packet may be freely reordered, the sender sets the R flag.
- The overlying transport may freely use the four ignored flag bits for its own purposes.

2.2. Receiver Behavior

When a receiver receives a packet containing a PLUS Basic Header, it processes the values in the Basic Header as follows:

- It verifies that the magic number is the constant 0xd800fffe.

- It verifies the integrity of the information in the PLUS Basic Header, using information carried in the overlying transport. Packets failing integrity checks SHOULD be dropped, but MAY be further analyzed by the receiver to determine the likely cause of verification failure; reaction to the failure is transport and implementation specific.

- It stores the PSN to be sent as the PSE on the the next packet it sends in the opposite direction.

2.3. On-Path State Maintenance using the Basic Header

The basic header provides all the signals necessary to drive the transport- independent state machine described in [I-D.trammell-plus-statefulness], as shown in Figure 2.

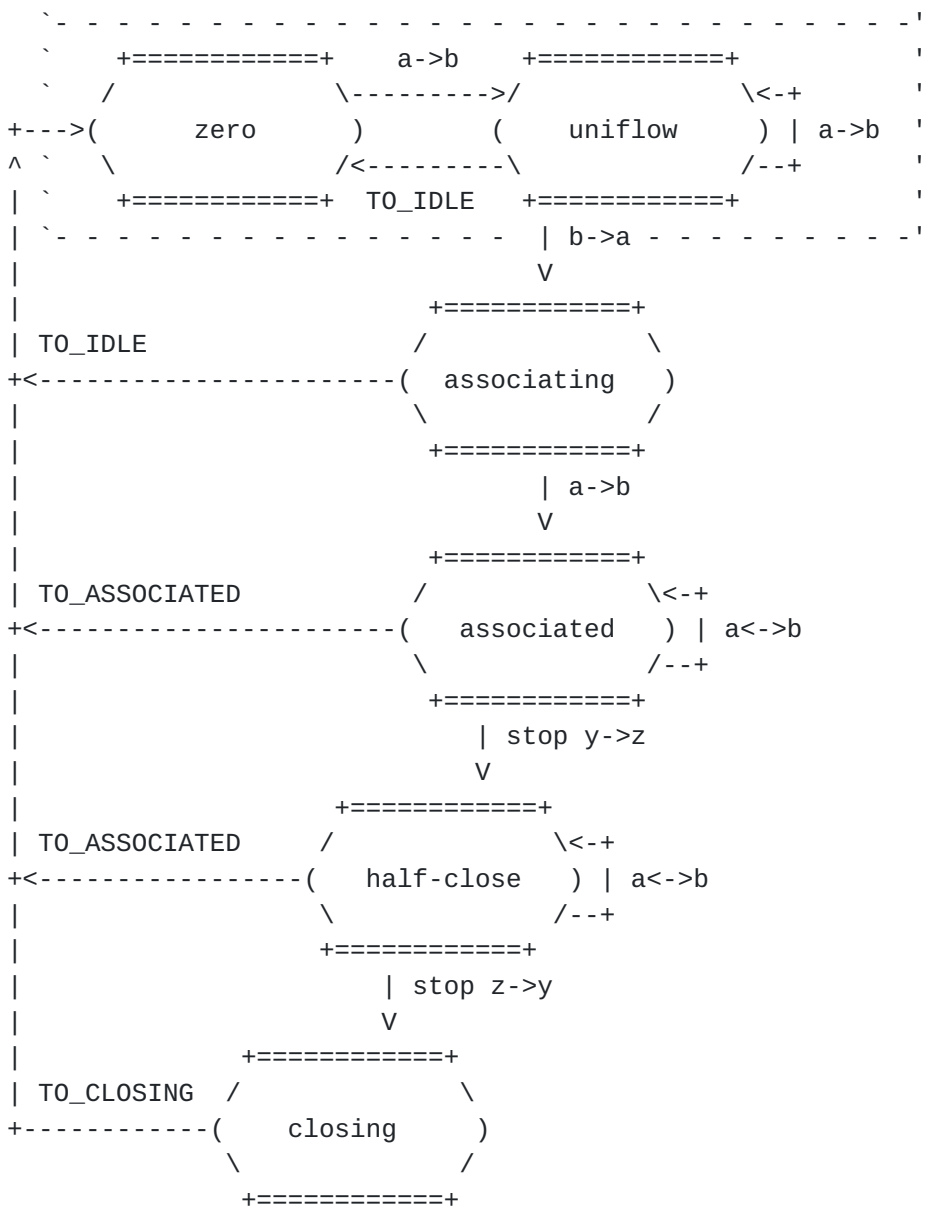

Figure 2: Transport-independent state machine as implemented by PLUS

2.3.1. State Establishment

A PLUS-aware on-path device forwarding a packet with a PLUS Basic Header with a 5-tuple and CAT it does not have state for moves that flow to the uniflow state. It will move the flow back to zero state after not seeing a packet on the same flow in the same direction with the same CAT within a timeout interval TO_IDLE, and continues forwarding packets in that direction (the a->b direction in Figure 2).

A PLUS-aware on-path device forwarding a packet with a PLUS Basic Header with a matching 5-tuple and CAT as a flow in the uniflow state, but in the opposite direction (the b->a direction in Figure 2), moves that flow to the associating state. It stores the PSN of the packet that caused this transition, and waits for a packet in the a->b direction containing a PSE indicating that that packet has been received. When it sees that packet, it transitions the flow to associated state. Otherwise, it drops state after a timeout interval TO_IDLE.

Once a flow has moved to the associated state, it will remain in that state for a timeout interval TO_ASSOCIATED. The on-path device forwards any packet with a PLUS Basic Header in either direction for this flow. It resets the TO_ASSOCIATED timer for every packet it forwards in this state.

2.3.2. Bidirectional Stop Signaling

A PLUS-aware on-path device forwarding a packet for a flow in the associated state with an S flag set moves that flow to half-close state. It stores the PSN on the packet causing the transition, and continues forwarding packets as if in associated state, dropping state on timeout interval TO_ASSOCIATED.

When it sees a packet in the opposite direction with the S flag set and the PSE set to exactly the stored PSN, it transitions the flow to closing state. The device will forward packets in both directions for flows in the closing state within a timeout interval TO_CLOSING; these packets will not reset the timer. See Section 2.3.2 for details.

Note that even though the S flag is integrity-protected end to end, a packet with the S flag set could be forged by one on-path device to drive the flow into half-close state on all downstream devices. However, this attack is of severely limited utility. First, it would require coordination between attackers on both sides of a given on-path device in order to forge a confirmation of the stop signal - a flag with the S bit set and a valid PSE corresponding to the PSN of the first stop signal to drive the flow into closing state. Second, the information in the Basic Header on each packet will drive the state machine into associated state even in the middle of a flow, enabling fast recovery even in the case of such a coordinated attack.

2.3.3. State Rebinding

A PLUS-aware on-path device forwarding a packet for a flow in the zero state, where one of the endpoint identifiers (address and port) and the CAT, but not the other endpoint identifier, match a flow in a

non-zero state, accounts that packet to the existing flow, updating the changed endpoint identifier. This allows fast rebinding of state in case of changes in network address translation or connectivity of the sender.

2.4. Measurement and Diagnosis using the Basic Header

The basic header trivially supports passive two-way delay measurement as well as partial loss estimation at a single observation point.

To calculate two-way delay, an observation point calculates the delay between seeing a PSN and a corresponding PSE in each direction, then adds the delays from each direction together. The fact that the PSN increments by one for every packet, including packets carrying retransmitted data or only control traffic, makes this measurement much simpler than the equivalent measurement using TCP sequence and acknowledgment numbers.

To calculate loss upstream from an observation point in each direction, the observation point simply counts skips in the PSN number space. Since PLUS does not expose information about retransmissions (and, indeed, may not even carry a transport that uses retransmission for loss recovery), loss downstream from the observation point cannot be observed.

3. Path Communication: Extended Header

Additional facilities for communicating with on-path devices under endpoint control are provided by the PLUS Extended Header. The extended header shares the layout of its first 21 bytes with the PLUS Basic Header, except the Extended Header bit (0x40 on byte 20) is set. As with the Basic Header, overlying transports are presumed to provide encryption and integrity protection for the PLUS Extended Header.

The Extended Header shown in Figure 3 provides for a single Sender to Path or Path to Receiver information element, as in [I-D.trammell-plus-abstract-mech], to appear on the packet, within a Path Communication Field. PCF Type information is carried in Byte 21 of the header, with the length of the PCF value to be determined by its type.

Further details of PCF encoding are not yet defined in this revision of the specification; the remainder of this section discusses the types of information elements to be supported. The exact encoding of PCF type and value information are to be derived from an analysis of the requirements of these Information Elements.

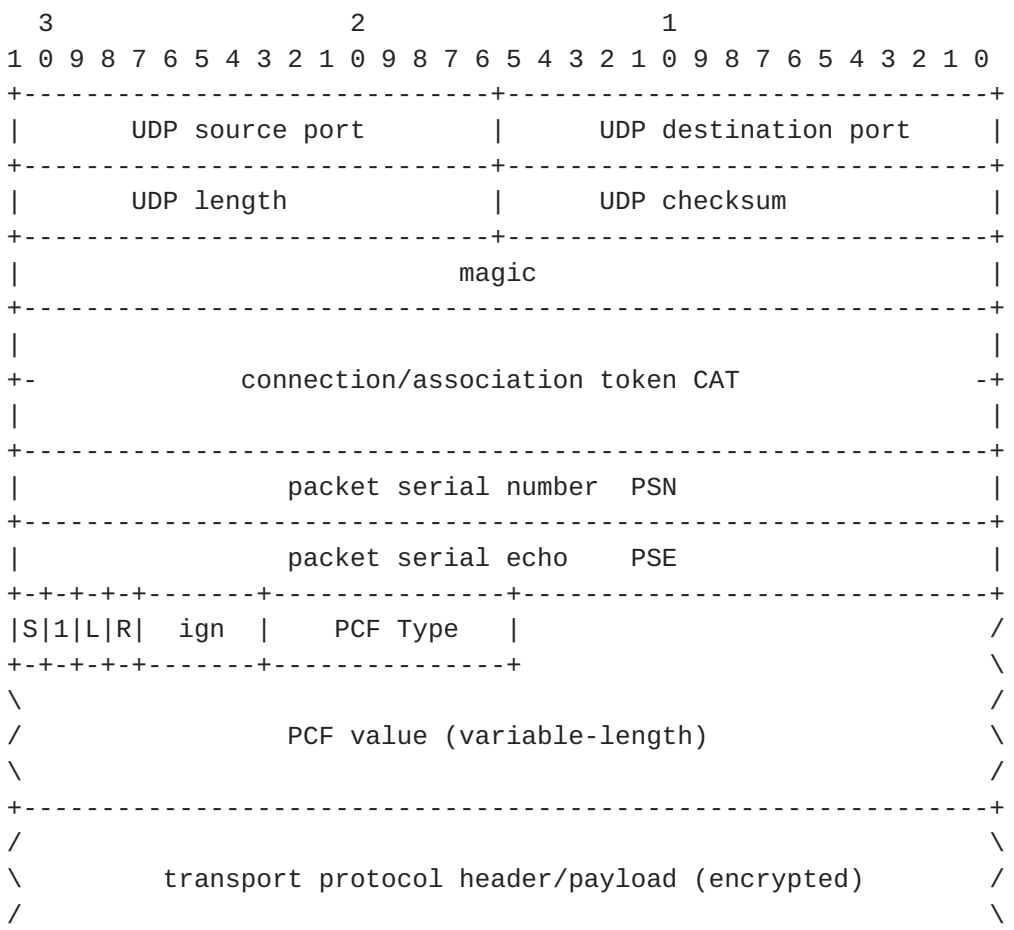

Figure 3: PLUS extended header (conceptual; details TBD)

As described in [I-D.trammell-plus-abstract-mech], there are two types of signals: Path to Receiver signals, which allow devices along the path to provide information about the path or its treatment of the flow to the receiver; and Sender to Path signals, which allow the sender to expose information about itself or the flow to the path. Path to Receiver signals are treated specially by header integrity protection, as their values, but not length or type, may be changed by devices on path: the value of a given path- to-receiver signal is assumed to be an appropriately sized array of zero bytes by the integrity protection facility.

Path to Receiver signals generally take the form of accumulators: initialized to some value by the sender, and subject to some aggregation function by each on-path device that understands them. Sender to Path signals are generally used to expose information about the traffic for measurement or diagnostic purposes. In any case, the information sent and received is to be treated as advisory only.

3.1. Measurement and Diagnostics using the Extended Header

We have identified the following sender to path signals as potentially useful for measurement and diagnostic purposes. These signals are advisory only, and should not be presumed by either the endpoints or devices along the path to affect forwarding behavior.

- Timestamp and timestamp echo. Similar to TCP timestamps in [RFC7323], also encoding a delta between receipt of last timestamp and transmission of echo as in section 4.1.2 of [IPIM]. Allows constant-rate clock exposure to devices on path. Note that this is less necessary for RTT measurement of one-sided flows than it is in TCP, due to the properties of the PSN and PSE values in the Basic Header.

We have identified the following path to receiver signals as potentially useful. Note that accumulated values for use at the sender must be fed back to the sender by the overlying transport, and that the presence of non-PLUS aware devices on path at breaks in MTU mean that the accumulated value can only be used as a hint to processes for measurement and discovery of the accumulated values at the sender.

- MTU accumulator. This signal allows measurement of MTU information from PLUS-aware devices. The sender sets the initial value to the sender's MTU. A PLUS-aware forwarding device on path receiving this value fills in the minimum of the received value and the MTU of the next hop into this field.

- State timeout accumulator. This signal allows measurement of timeouts from PLUS-aware devices. It is initialized to a maximum ("no information") value by the sender. A PLUS-aware forwarding device on path receiving this value fills in the minimum of the received value and the configured timeout for the flow's present state into this field.

- Rate limit accumulator. This signal allows exposure of rate limiting along the path. It is initialized to a maximum ("no information") value by the sender. A PLUS-aware forwarding device on path receiving this value fills in the minimum of the received value and the rate limit to which this flow is subject into this field.

- Trace accumulator. This signal allows exposure of a trace of PLUS-aware devices on path, similar to the Path Changes mechanism in section 4.3 of [IPIM]. The sender initializes the value to a value chosen randomly for the flow; all packets in the flow using path trace accumulator must use the same initial value. A PLUS-

aware forwarding device on path receiving this value fills in the result of XORing the received value with a randomly chosen device identifier, which it must use for all path trace accumulator signals it participates in. Packets traversing the same set of PLUS-aware forwarding devices in the same flow therefore arrive at the receiver with the same accumulated value, and changes to the set of devices on path can be detected by the receiver.

4. IANA Considerations

This document has no actions for IANA. Path communication field types and PLUS magic numbers may be moved to a Standards Action registry in a future revision.

5. Security Considerations

[EDITOR'S NOTE: write me]

6. Acknowledgments

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement.

7. Informative References

[I-D.hardie-path-signals]
Hardie, T., "Path signals", draft-hardie-path-signals-00 (work in progress), October 2016.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-00 (work in progress), November 2016.

[I-D.trammell-plus-abstract-mech]
Trammell, B., "Abstract Mechanisms for a Cooperative Path Layer under Endpoint Control", draft-trammell-plus-abstract-mech-00 (work in progress), September 2016.

[I-D.trammell-plus-statefulness]
Kuehlewind, M., Trammell, B., and J. Hildebrand, "Transport-Independent Path Layer State Management", draft-trammell-plus-statefulness-02 (work in progress), December 2016.

[IPIM] Allman, M., Beverly, R., and B. Trammell, "In-Protocol Internet Measurement (arXiv preprint 1612.02902)", December 2016.

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

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <http://www.rfc-editor.org/info/rfc2474>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <http://www.rfc-editor.org/info/rfc7323>.

[RFC7675] Perumal, M., Wing, D., Ravindranath, R., Reddy, T., and M. Thomson, "Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness", RFC 7675, DOI 10.17487/RFC7675, October 2015, <http://www.rfc-editor.org/info/rfc7675>.

Authors' Addresses

Brian Trammell
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: ietf@trammell.ch

Mirja Kuehlewind
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: mirja.kuehlewind@tik.ee.ethz.ch