Driving innovation together

Technology Scouting 2020

Author(s):	Marijke Kaat and Ronald van der Pol
Version:	1.2
Date:	2021-03-08

Moreelsepark 48
3511 EP Utrecht
The Netherlands

PO Box 19035
3501 DA Utrecht
The Netherlands

+31 (0) 88 - 787 30 00
info@surf.nl
www.surf.nl

Contents

1 Introduction

We constantly look for new trends and emerging technologies in the network area. We call this *scouting for new technology*. This is done in various ways: we look at what other ISPs are doing, we look at new protocols being designed in standard organisations, we follow open source project in the area of networking, we follow mailing lists, blogs, webinars, recordings of conferences, etc. And in our Next Generation Networking (NGN) Lab we familiarise ourselves with these new technologies by getting hands-on experience with them. This report is the short summary of some of the most interesting trends and technologies during 2020. We think that these have a potential use in our network or the networks of our connected institutes. The technologies are described at a high level. The authors of this report can be contacted for a more detailed discussion.

2 IPv6 Segment Routing

Segment Routing is a kind of source routing which gives control of the complete (or partial) forwarding path through a network. It can be used for traffic engineering to steer traffic over arbitrary paths without the use of hop-by-hop signalling protocols. The Segment Routing Architecture describes two data planes that can be used. The first one is MPLS based (SR-MPLS) where the SR header is instantiated through an MPLS label stack, defined in RFC 8660 which was published in December 2019. The other is IPv6 based called SRv6, where a new type of Routing Extension Header is used to encode IPv6 segments. The SRv6 Segment Routing Header (SRH) has been standardized in RFC 8754 which was published in March 2020.

In a Segment Routing domain (SR-domain) the SRH with the correct segment list must be constructed and added to the incoming packets on each ingress node. The segment lists can be determined by a central controller, for example a node connected to the network running a Path Computation Element process. The PCE has information about all SIDs in the SR-domain and can compute paths and determine the appropriate Segment Identifiers (SIDs) that need to be added to incoming packets by the ingress nodes.

Segment Routing can also be used to forward traffic to Virtual Network Functions (VNFs) or physical service appliances for special packet processing, for example a firewall function. Such a service can be made available in the network and can be associated with a Segment Identifier (SID). These service SIDs can then be used as part of the SID list in the packet headers to steer the traffic through the corresponding service. This can be programmed in the network using an SDN controller for specific dataflows using the Path Computation Element Protocol (PCEP). PCEP is used between a Path Computation Element (PCE), the controller, and a Path Computation Client (PCC), a head-end (ingress) router. A PCE can be stateless, which means it keeps no information about previously computed and established paths. It just performs a path computation in response to a request received from a PCC. A stateful PCE also considers the set of earlier computed paths and reserved resources in the network when processing a new request. This allows for more optimal path computation, but also requires reliable state synchronization mechanisms which may increase control plane overhead. Especially when there are multiple PCEs in a network, to distribute the work load and for redundancy, synchronization state communication becomes quickly very complex.

Extensions to the PCEP protocol to support SRv6 are being defined in the IETF PCE working group (draft-ietf-pce-segment-routing-ipv6)[1]. For SRv6 a new subobject *SRv6-ERO*, ERO being the Explicit Route Object, and a new subobject *SRv6-RRO*, the Route Record Object, and new PCEP error codes are defined. A Path Computation Client (PCC) that requests a path uses the SRv6-ERO sent by the PCE to build the ordered list of segments. It converts the SRv6-ERO to an SRv6 Segment Routing Header (SRH) plus the next hop. The SRv6-RRO can be used by the PCC to report to the PCE the actual SID list that was applied by the PCC.

A controller (PCE) will need information from the network in order to compute the requested paths. A BGP-LS session to one or more routers can provide the functionality to discover the topology and needed SR information from the SR domain. BGP-LS is an extension to BGP. A new BGP Network Layer Reachability Information (NLRI) encoding format was defined to carry Link-State and TE information from the network's IGP (the Link-State Database or LSDB and the Traffic Engineering Database or TED). BGP-LS can also be used for topological visibility across Autonomous Systems for multi-domain paths. The work on extensions for SRv6 to

[1] PCEP extensions for Segment Routing for an MPLS forwarding plane were already defined in RFC 8664.

BGP-LS is in the last stage before publication as an RFC (draft-ietf-idr-bgpls-srv6-ext). BGP-LS can then also be used by a PCE to discover the SRv6 capabilities of nodes and the mapping of SRv6 segments to the nodes.

There was a lot of activity in 2020 the IETF SPRING (Source Packet Routing in Networking) working group that works on Segment Routing, and in the 6MAN (IPv6 Maintenance) working group for SRv6 specifically. Most documents are in a draft stage however and still being discussed. This includes work on OAM (operations, Administration and Maintenance), Yang models and performance measurement (TWAMP), which are all important for the management of an SR domain. Also, a lot of drafts deal with specific use-cases and requirements, such as SRv6 for the mobile user plane (GTP-U) and using SRv6 as a building block for network slicing. Network slicing is the technique to create end-to-end partitioned network infrastructures to be used for differentiated behaviour to fulfil requirements of diverse services and is one of the requirements in 5G.

2.1 SRv6 versus SRm6

The discussions on compression of the SRv6 information in the packet headers is not concluded yet. In July 2020 a special compression design team “SRCOMP” within the SPRING working group was formed, with their own mailing list and work items. This is a small team whose members are working at vendors like Juniper, Cisco, Huawei, ZTE and Nokia and some early adopters of SRv6. The task of the team is to first define the requirements and next evaluate existing proposed solutions according to those requirements.

Early January 2021 the third version of the requirements draft was published (draft-srcompdt-spring-compression-requirement). The different requirements are divided into several sections, such as:

- SRv6 SID list Compressions requirements which include: data plane efficiency and performance requirements, forwarding efficiency and state efficiency;
- SRv6 specific requirements:
    - SRv6 based - the solution may be based on a different data and control plane, this is not preferred;
    - functional requirements include: heterogeneous SID lists (combination of compressed and non-compressed segments must be supported), SID list length (must be up to 16 segments);
    - Operational requirements: lossless compression;
    - Scalability requirements that the proposal must be capable of representing: adjacency segment scale (65000 adjacency segments per node), prefix segment scale (1 million prefix segments per SID numbering space), service scale (1 million services per node);
    - Protocol Design requirements: SRv6 base coexistence;
    - Security requirements: security mechanisms and SR domain protection.

In a few months the team did a lot of work to gather the requirements and describe the rationale and metrics for each. However, the draft is far from finished, additional requirements are still under review, requirements may be changed and added or removed. Some of the requirements also mean that vendors with current implementations may have to add code or delete and replace code.

At the moment there are four drafts that describe different mechanisms that will be included in the analysis and evaluation of existing compression solutions:

- Compressed SRv6 Segment List Encoding in SRH (draft-filsfilscheng-spring-srv6-srh-comp-sl-enc-02)
- SRv6 vSID: Network Programming extension for variable length SIDs (draft-decraene-spring-srv6-vlsid-04)
- The IPv6 Compact Routing Header (CRH) (draft-bonica-6man-comp-rtg-hdr-24)
- Unified Identifier in IPv6 Segment Routing Networks (draft-mirsky-6man-unified-id-sr-08)

A first version of the analysis and evaluation of these four proposals draft is planned for the March 2021 IETF meeting. Which one will be the preferred proposal for compression of the SRv6 header cannot be easily predicted, there will probably be a lot of discussion in the SPRING working group in 2021.

2.2 SRv6 Network Programming

The draft on SRv6 Network Programming has been worked on for a few years in the SPRING working group, and was published as an RFC in February 2021: RFC 8986. The document defines a framework to specify a packet processing program by encoding a sequence of instructions in the packet header. Such an instruction can be implemented on one or several nodes in the network. The instruction is identified by an SRv6 Segment Identifier in the packet. The function can be simple or complex to achieve some kind of networking objective. A number of basic SRv6 behaviours has been specified in the document, but in the future more could be defined. The processing of an SRv6 Segment Identifier (SID) was already defined in RFC8754. An SR segment endpoint node creates Forwarding Information Base (FIB) entries for its local SIDs, and for each incoming IPv6 packet it will perform a longest-prefix-match lookup on the packet's destination to determine how to process that packet.

The format of the SID, which looks like an IPv6 address, is defined as consisting of:

LOC:FUNCT:ARG

```
where a locator (LOC) is encoded in the L most significant bits of the SID,
followed by F bits of function (FUNCT) and A bits of arguments (ARG).


L, the locator length, is flexible, and an operator is free to use the
locator length of their choice. F and A may be any value as long as L+F+A
<= 128. When L+F+A is less than 128 then the remaining bits of the SID MUST
be zero.
```

The FUNCT is an identification of local behaviour bound to the SRv6 SID (SRv6 Segment Endpoint Behaviour). If additional information is necessary for the processing of the behaviour, it may be encoded in the ARG bits of the SID.

The document defines the following fifteen behaviours and their associated algorithms in pseudocode, that can be associated with a SID:

```
End                Endpoint function
                   The SRv6 instantiation of a Prefix SID [RFC8402]
End.X              Endpoint with Layer-3 cross-connect
                   The SRv6 instantiation of an Adj SID [RFC8402]
End.T              Endpoint with specific IPv6 table lookup
End.DX6            Endpoint with decapsulation and IPv6 cross-connect
                   e.g. IPv6-L3VPN (equivalent to per-CE VPN label)
End.DX4            Endpoint with decaps and IPv4 cross-connect
                   e.g. IPv4-L3VPN (equivalent to per-CE VPN label)
End.DT6            Endpoint with decapsulation and IPv6 table lookup
```

```
                       e.g. IPv6-L3VPN (equivalent to per-VRF VPN label)
End.DT4                Endpoint with decapsulation and IPv4 table lookup
                       e.g. IPv4-L3VPN (equivalent to per-VRF VPN label)
End.DT46               Endpoint with decapsulation and IP table lookup
                       e.g. IP-L3VPN (equivalent to per-VRF VPN label)
End.DX2                Endpoint with decapsulation and L2 cross-connect
                       e.g. L2VPN use-case
End.DX2V               Endpoint with decaps and VLAN L2 table lookup
                       e.g. EVPN Flexible cross-connect use-case
End.DT2U               Endpoint with decaps and unicast MAC L2 table lookup
                       e.g. EVPN Bridging unicast use-case
End.DT2M               Endpoint with decapsulation and L2 table flooding
                       e.g. EVPN Bridging BUM use-case with ESI filtering
End.B6.Encaps          Endpoint bound to an SRv6 policy with encapsulation
                       SRv6 instantiation of a Binding SID
End.B6.Encaps.Red      End.B6.Encaps with reduced SRH
                       SRv6 instantiation of a Binding SID
End.BM                 Endpoint bound to an SR-MPLS Policy
                       SRv6 instantiation of an SR-MPLS Binding SID
```

The top twelve of these behaviours describe how the last router in the path should decapsulate a packet. Some define in which routing table that router should do a lookup after the decapsulation. The last three behaviours are about binding SIDs that can be used for inter-domain paths.

The behaviours for L3VPN's and L2VPN's could be interesting for use in the SURF network to build services using SRv6.

This document should be ready for publications as an RFC, however, an appeal was submitted to the IESG against the declaration of working group consensus. A number of participants of the SPRING WG believe that major concerns that were raised before and during the last call, have not been addressed satisfactorily. The main technical issue is about the so-called Penultimate Segment Pop (PSP). Processing in the penultimate node (of the path) may include removing the routing header when it finds that the "Segments Left" is zero. This behaviour is in violation of RFC8200 which states that en-route header insertion or removal is not allowed. The main procedural concern is that there is a possible conflict of interest and that the process of WG last call has not been transparent. The IESG answered they see no basis for returning the document to the SPRING WG for a second last call. The discussions continued and several new versions of the draft have been published. It seems the main technical issue has not been resolved, but the IESG approved the last version from 29th of December 2020 ready for publication.

2.3 Impact on the SURF Community

The SURFnet8 network uses Segment Routing over MPLS (SR-MPLS), but there are no plans to implement SRv6 in the short term. The writing of the standards for SRv6 and related work is still going on in the IETF, see for example the discussions on the compression of the SRv6 header. Each year there are interoperability tests performed by the European Advanced Networking Test Center (EANTC) between implementations from different vendors, including Juniper, and open source implementations. The EANTC publishes white papers about the devices and technologies tested and the results[2]. We keep following the discussions in the IETF and the progress in implementations by Juniper.

[2] https://eantc.de/fileadmin/eantc/downloads/events/MPLS2020/EANTC-MPLSSDNNFV2020-WhitePaper.pdf

3 The QUIC Transport Protocol

QUIC is a new secure general-purpose connection-oriented transport protocol. It has several improvements over TCP. QUIC has built-in encryption and traffic is always encrypted. QUIC has stream multiplexing so that an application can use multiple streams in the same QUIC connection. Each stream has its own flow control. This minimises head of line blocking. When one stream stalls, for example due to packet loss and retransmissions, other streams are not impacted and can continue independently. Finally, QUIC has low-latency connection establishment by using the 0-RTT (zero-RTT) support in TLS 1.3. This is also known as “zero roundtrip time connection resumption”. This is done by cashing information from a previously established TLS connection at both the client and server side. In this case a client can start sending data already in the first part of the TLS handshake.

3.1 HTTP/3

HTTP has seen several versions. HTTP/0.9, HTTP/1.0 and HTTP/1.1 were quite similar with minor changes. In 2010 Google published the SPDY protocol. It used the semantics of the HTTP/1.1 protocol, but it used a binary format instead of a string format. SPDY became the basis of the HTTP/2.0 protocol and resulted in RFC 7540 (May 2015). Meanwhile, starting in 2014 Google was experimenting with a next version of SPDY and a new transport protocol, which is now known as gQUIC (Google QUIC). Based on this work, the QUIC Working Group was started in October 2016 to standardise the QUIC transport protocol. HTTP/3 is HTTP/2 over the QUIC protocol. It is defined in draft-ietf-quic-http and is currently in the RFC editor queue to be published as an RFC. The QUIC transport protocol itself is defined in draft-ietf-quic-transport and is also in the RFC editor queue. QUIC runs on top of UDP and is implemented in user space. Figure 1 shows how the HTTP over TCP stack differs from the HTTP over QUIC stack.

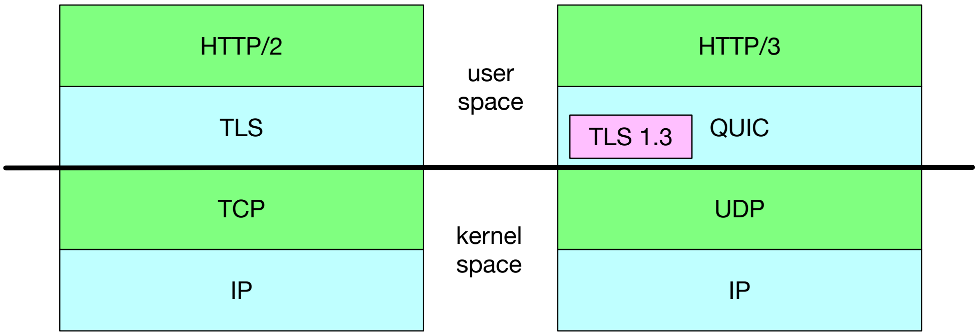

Figure 1: HTTP/2 versus HTTP/3

TLS 1.3 is defined in RFC 8446 (August 2018). Its use in combination with QUIC is defined in draft-ietf-quic-tls and is also in the RFC editor queue. So, in early 2021 it is expected that a series of HTTP over QUIC RFCs will be published as proposed standards. Section 3.4 gives an overview of current implementations and the companies behind them. Some of them will probably soon introduce QUIC in production on web sites and in browsers.

3.2 QUIC Frame Types

In TCP the payload is just a stream of bytes and all payload data belongs to this one stream. In QUIC the payload is structured and consists of one or more frames. Each frame starts with a frame type. The currently defined frame types are: PADDING, PING, ACK, RESET_STREAM, STOP_SENDING, CRYPTO, NEW_TOKEN, STREAM, MAX_DATA, MAX_STREAM_DATA, MAX_STREAMS, DATA_BLOCKED, STREAM_DATA_BLOCKED, STREAMS_BLOCKED, NEW_CONNECTION_ID, RETIRE_CONNECTION_ID, PATH_CHALLENGE, PATH_RESPONSE, CONNECTION_CLOSE, HANDSHAKE_DONE.

QUIC has an elaborate acknowledgement mechanism. An ACK frame contains one or more ranges. These ranges contain the packet numbers that are acknowledged. It also contains an *intentional delay* value. An endpoint intentionally waits for additional packets before sending an ACK. By including this value in the ACK frame, the peer endpoint can make a better estimation of the roundtrip time.

The CRYPTO frame is used for cryptographic handshake information. By using a dedicated frame type it is possible to switch from the current TLS1.3 protocol to another cryptographic protocol in the future.

STREAM is used for application data. Each STREAM frame contains a stream identifier so that multiple streams can be used simultaneously.

QUIC uses per stream flow control. MAX_DATA is used to inform the peer of the maximum amount of data that can be sent on the connection as a whole. MAX_STREAM_DATA is the maximum amount of data that can be sent on one stream.

QUIC offers the concept of connection migration. An endpoint can change its IP address and/or port number and it can tell its peer endpoint about this change. This could happen when an endpoint moves from WIFI to a mobile network. QUIC uses a connection ID to identify a session. The NEW_CONNECTION_ID, PATH_CHALLENGE and PATH_RESPONSE are used to validate the new address/port.

3.3 Monitoring QUIC

QUIC encrypts packets for at least three reasons. The first is security by encrypting the payload. The second and third are privacy preventing ossification[3]. QUIC encrypts more of the header than TCP/IP with TLS. QUIC uses a long header format during handshake and a short header for the rest of the session. The short header contains a packet number, destination connection ID (which may be zero) and some flags. This minimal information in the header enhances the privacy of the user by not leaking meta-information. It also prevents ossification.

The enhanced privacy and preventing of ossification make it more difficult to monitor QUIC sessions. A network administrator snooping packets on the wire sees mostly short headers with minimal information. Recent versions of *wireshark* recognise the QUIC protocol, although this is currently dependent on the version of the QUIC transport draft. This will change when QUIC is published as an RFC. The spin bit in the header is an experimental feature to enable roundtrip measurements by looking only at packets on the wire. The IETF QUIC WG is also working on a standard logging format for QUIC. It is currently a personal draft: draft-marx-qlog-event-definitions-quic-h3. Endpoints like browsers and web servers can write QUIC session data in a JSON like format to a log file. This includes information about the QUIC headers (not the payload in order to respect the privacy of users) and QUIC flow control parameters that are not sent over the wire.

[3] A protocol is ossified when it is difficult to change or enhance because middle boxes like firewalls, load balancers, NATs, etc. made (incorrect) assumptions about fields in the protocol header.

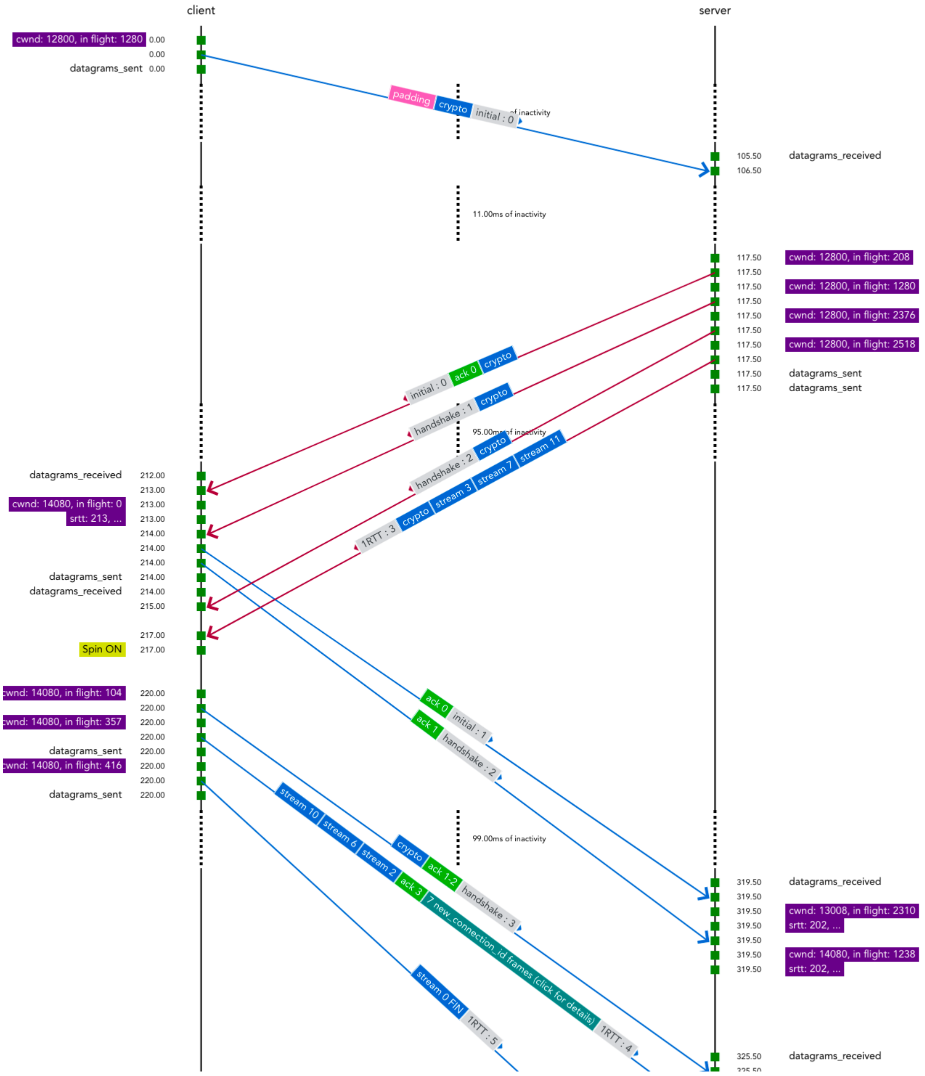

Figure 2 QUIC Stream Visualisation

The University of Hasselt in Belgium is working on tooling around the qlog format. Figure 2 shows and example of a visualisation of QUIC packets exchanged between a client on the left and a server on the right. The type of packets is shown in combination with the timing between packets.

3.4 QUIC Implementations

Table 1 shows known QUIC implementations[4] as of February 2021. New implementations (compared to the list of last year) have a light grey background. Other implementations (struck out in Table 1) have disappeared from last year's list.

[4] https://github.com/quicwg/base-drafts/wiki/Implementations

Table 1: QUIC Implementations (source: IETF QUIC Working Group Wiki)

Name	Language	Role	License	Company
Aioquic	Python	Client, Server, Library	3-clause BSD	Independent
Akamai QUIC	unknown	Server	unknown	Akamai
AppleQUIC	C, Objective-C	Client, Server	Closed	Apple
Ats	C++	Client, Server	Apache 2.0	Apache
Chromium	C, C++	Client, Server, Library	3-clause BSD	Chromium Project
F5	C	Client, Server	Closed	F5
Haskell Quic	Haskell	Client, Server, Library	3-clause BSD	Independent
Kwik	Java	Client	LGPL	Independent
LiteSpeed QUIC	C	Client, Server, Library	MIT	LiteSpeed Technologies
Microsoft QUIC	C	Client, Server	Closed	Microsoft
Mvfst (move fast)	C++	Client, Server, Library	MIT	Facebook
Neqo	Rust	Client, Server, Library	Apache2/MIT	Mozilla
Ngtcp2	C	Client, Server, Library	MIT	Independent
NGINX QUIC	C	Server	2-clause BSD	Cloudflare
NGINX-Cloudflare	C	Server	2-clause BSD	Cloudflare
~~Node.js QUIC~~	~~C++, Javascript~~	~~Client, Server~~	~~Node.js~~	~~Node.js~~
~~Pandora~~	~~C~~	~~Library~~	~~Not yet public~~	~~Aalto and TUM University~~
Picoquic	C	Library	MIT	Christian Huitema
Pluginized QUIC	C, BPF	Client, Server, Library	MIT	Christian Huitema, et al
Quant	C	Client, Server, Library	2-clause BSD	NetApp (Lars Eggert)
Quiche	Rust	Client, Server, Library	2-clause BSD	Cloudflare
~~QUICker~~	~~Typescript~~	~~Client, Server, Library~~	~~Unknown~~	~~Independent~~
Quicly	C	Client, Server	MIT	Fastly
~~Quincy~~	~~Java~~	~~Client, Server, Library~~	~~Unknown~~	~~Netty Project~~
Quinn	Rust	Client, Server, Library	Apache2/MIT	Independent
~~Sora_quic~~	~~Erlang/OTP~~	~~Server, Library~~	~~Unknown~~	~~Independent~~
Quic-go	Go	Client, Server, Library	MIT	Independent

3.5 Impact on the SURF Community

QUIC will be published as a proposed standard soon. It is expected that many cloud providers will enable QUIC on their web servers soon after. This will increase the amount of QUIC traffic on the network. Especially network managers and those involved in security should be able to recognise QUIC and have a basic understanding of it. They should also consider what it means for their procedures and monitoring capabilities when dealing with a protocol where most of the packets are encrypted.

4 New IP Proposal

There have been various proposals on a “new Internet” that redesign the basic architecture of the Internet technology over the years, usually called “clean slate’ proposals. They come and go, and mostly you don’t hear much more about it, but last year there was a “New IP framework” proposed to the ITU which generated quite a lot of reactions.

The proposal was presented at an ITU meeting and a similar presentation was given at a side meeting of the IETF 106 meeting in Singapore. The presentation[5] was given by Richard Li from Huawei, and it basically proposes to set up a new “Focus Group on Network 2030[6]” within a study group of the ITU-T to “help shape a new internet”. The presentation explains new applications that should be possible in the future, like holograms, tactile internet, integrated terrestrial and space networks, remote surgery and a trustworthy, intrinsically secure infrastructure. Several such new applications and services require very low latency (super ultralow <1ms), better protocol efficiency, high precision, guaranteed and even absolute delivery times. The “New IP” technology should make such new services and new architectures possible in the (long) future. What it exactly would look like is not explained and very vague, and it is by no means a new standard of any sort. Reactions on the proposal came from several of the SDOs and organizations involved in defining Internet standards and governance.

The Internet Society (ISOC) posted a long discussion paper on their website: “An analysis of the “New IP” proposal to the ITU-T”. The paper’s main conclusions are that the standardization work is already being done in other SDO’s such as the IETF, IEEE and 3GPP, and in the ITU-T’s study group 15, and therefore duplicative work is not needed and costly. And that a new proposal would require an expensive migration effort, that the embedded base of equipment and operational systems should be taken into account. And lastly, a new protocol system is likely to create multiple non-interoperable networks, something the New IP proposal states is one of the concerns of the current network system.

On the RIPE Labs website Marco Hogewoning, manager Public Policy and Internet Governance, posted a long blog about this “New IP” proposal, warning that this proposal might have as a long-term vision for this architecture to supersede TCP/IP and replace the Internet. He states that this New IP proposal is not needed and that technical challenges and needs should be developed by the SDOs that have control over the current standards, mainly the IETF, and that it should be done in an open and transparent model. RIPE NCC sent a formal response to the ITU[7] explaining the concerns with the proposal. Especially with the possibility that it might open up an opportunity to change the bottom-up decision-making model for both creating the Internet standards as well as the governance of resources. Other concerns raised were that RIPE NCC thinks the technical rationale of the proposal is flawed, and the alternative designs are unrealistic, and that more work on it would overlap significantly with ongoing activities in other SDOs. The recommendation to the ITU is to not pursue this work within the ITU-T.

Geoff Huston, Chief Scientist at APNIC, writes in an article (https://www.potaroo.net/ispcol/2020-05/futuretech.html) he thinks the proposal is not going anywhere useful, that it is just yet another proposal to try to add “value” to the commodity service of packet transmission that users are not willing to pay for. He sees things rather going the other way with new protocols like QUIC and BBR, and resolverless DNS, where the

[5] https://www.itu.int/en/ITU-T/Workshops-and-Seminars/201807/Documents/3_Richard%20Li.pdf

[6] https://www.itu.int/en/ITU-T/focusgroups/net2030/Pages/default.aspx

[7] https://www.ripe.net/participate/internet-governance/multi-stakeholder-engagement/ripe-ncc_tsag_new-ip.pdf

applications themselves are given more control over their data flows to optimize the behaviour, and not rely on the common set of operating system functions and techniques in the network. Moreover, he states applications are no longer managed by the network, but instead seem to be hiding their control mechanisms from the network, and make fewer assumptions about the characteristics of the network.

Many more reactions, from ICANN for example, and articles were written about this New IP proposal. The main fear seems to be that this proposal was brought forward by Chinese companies and organizations and that their goal is to somehow take over control of the Internet and make it possible to shut down parts of the network at will from a central point. The proposal, or at least the presentation is still vague, but also points at certain issues that do exist in the current infrastructure, and nobody will deny there are (security) issues in the current Internet. For something that may or may not become a reality in the far future the reactions were quite excessive and perhaps somewhat premature. Richard Li wrote a long article[8] in response to the ISOC discussion paper addressing all the concerns in detail and states that “New IP” is used as an umbrella term for multiple independent efforts to improve the Internet and connecting more (industrial) networks with more stringent performance requirements.

4.1 Impact on the SURF Community

As most of the proposals and research on new architectures for the Internet take years to develop before any new standards and technical specifications are written, if ever, this has no short-term impact on current infrastructures and it would take years before widespread adoption of such new technologies. But it is important to follow these developments and discussions to know where the industry might be heading. Some projects on Future Internet or clean slate Internet research do produces valuable results, for example software designed networking could be seen as such an outcome of a research project.

However, most people agree that for a new technology to become successful it should be backward compatible with current technologies and that introduction should be done incrementally and the costs should not be too high. Even though a new technology is superior to the current system does not mean it will be a success.

As there are serious shortcomings and issues in the current Internet it is certainly necessary to follow new developments and changing requirements, even if they are long-term proposals that might one day become real technologies or services. We will keep following such proposals and discussions and report on them if applicable to our environment. We do this also within the 2STiC project (www.2stic.nl), which is a joint research programme with several universities, research organisations, developers and operators to evaluate and experiment with new mechanisms or technologies that can improve the security, stability and transparency in inter-network communication.

[8] https://internet4future.wordpress.com/