

Internet Draft
Intended status: Informational
Expires: January 2014

S. Jobert
I. Hamchaoui
W. Diego
Orange
July 11, 2013

Packet-oriented QoS management model for a wireless Access Point
draft-jobert-iccr-g-ip-aware-ap-00.txt

Status of this Memo

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 11, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document introduces a packet-oriented QoS management model applicable to a wireless Access Point, referred to as "IP aware Access Point". This model may be used in addition, or as a substitute to the connection-oriented traditional QoS model defined for cellular mobile networks, based on the notion of bearer.

Table of Contents

1. Introduction	2
2. Conventions used in this document	4
3. Concepts and architecture	4
4. IP aware wireless Access Point behavior	6
5. Extension to upstream traffic	7
6. Examples of possible models	8
6.1. Model for intra-bearer arrangement	8
6.2. Models for inter-bearer arrangement	9
7. Security Considerations	9
8. IANA Considerations	9
9. References	9
9.1. Normative References	9
9.2. Informative References	10
10. Acknowledgments	10

1. Introduction

The management of QoS in cellular mobile networks remains connection-oriented in many cases: when QoS differentiation is targeted, the general solution proposed by most existing cellular mobile standards consists in establishing several tunnels (e.g. bearers), one per class of service, between the mobile terminal (e.g. UE) and the wireless Access Point (e.g. NodeB, eNB).

The packets are then positioned into the various tunnels, depending on various criteria, for instance on the required QoS of the corresponding application. Adapted QoS treatments are then applied on the overall tunnels, according to the characteristics of the established bearers, but not at the packet level. In other words: even if an IP packet is marked (e.g. using DSCP marking) as being of high priority, this information inside the packet is not taken into account to schedule the packet over the radio interface, only the bearer that the packet belongs to matters.

These tunnels are moreover in general extended up to an aggregation point in the network (e.g. GGSN, PDN-GW). Packet-oriented QoS management may however be applied over the fixed portion of the network, although tunnels are present.

This connection-oriented model was inherited from TDM-based transport, and it was well adapted to early mobile data services deployments when the amount of data exchanged was low. It now raises important questions considering the high volumes and variability of mobile data traffic to be handled by most of mobile network operators generated by smartphones applications.

In addition, managing the QoS of mobile networks using a connection-oriented model has clearly some drawbacks, such as: problems of scalability of the architecture due to the number of tunnels to be maintained simultaneously, additional signaling traffic required to establish or modify the parameters of the tunnels, time to establish or modify a tunnel, etc.

In this draft, we introduce the concepts of a packet-oriented QoS management model applicable to a wireless Access Point, referred to as "IP aware Access Point", which can be considered as an alternative approach to the existing connection-oriented model. Note however that both models could be simultaneously deployed: a connection-oriented model could be used for managed ISP services when signaling traffic is exchanged prior to starting the service (a typical example might be managed VoLTE), while a packet-oriented model could be suitable to differentiate among the other services (e.g. Internet services), thanks to proper DSCP marking/remarking (according to the QoS policy enforced in the ISP-operated DiffServ domain).

It is expected that this model be applicable to both licensed and unlicensed wireless networks (such as Wi-Fi).

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

AP: Access Point

CQI: Channel Quality Indicator

DSCP: Differentiated Services Code Point

eNB: evolved NodeB

GGSN: Gateway GPRS Support Node

Mifi: Mobile Wi-Fi

NB: NodeB

PDN-GW: Packet Data Network Gateway

PF: Proportional Fair

RB: Resource Block

TTI: Transmission Time Interval

UE: User Equipment

VoLTE: Voice over LTE

3. Concepts and architecture

Introducing a packet-oriented model in mobile networks is expected to simplify QoS management by leveraging the experience and current practice in fixed IP networks and permitting consistent operations over all network segments. Indeed, it simply reuses proven (and standard) mechanisms of fixed IP networks. It also solves most of the drawbacks of the connection-oriented approach. Finally, it

allows for global seamless offers over various wireline and wireless accesses.

The proposed packet-oriented QoS management model consists in the following main principles (the description below applies mainly to the downstream direction, i.e. from the wireless AP to the terminal, but can be extended to the upstream direction, i.e. from the terminal to the wireless AP):

- o There is no need for establishing several tunnels to allow differentiated QoS treatments; multiple classes of service are defined for this purpose independently of the tunnels, allowing differentiated QoS treatments inside the same tunnel, which may be maintained for mobility purposes if needed.
- o The packets SHOULD be marked or re-marked properly at the ingress of the DiffServ domain corresponding to the mobile network, according to the applicable criteria (which may depend on the type of service, agreement between parties, etc.). The DSCP field in the header of the IP packets MUST be used for this purpose. This marking SHOULD be maintained transparently up to the IP aware Access Point. Depending on the applicable scenario, the DSCP marking function MAY be located in any upstream point from the IP aware access point, including the traffic source.
- o The IP aware wireless Access Point MUST classify the packets based on the marking indicated in each packet and differentiated treatments are applied on a packet-per-packet basis. The packets with the highest priority SHOULD for instance be positioned in a high priority queue, which is served first when radio resources are allocated to the terminal.

Figure 1 below illustrates the proposed packet-oriented QoS management model, with an example where 3 classes have been configured in the IP aware Access Point (e.g. one for traffic requiring low delay/jitter, one for traffic requiring low packet loss, and one for Best Effort traffic). Note that the IP layer at the IP aware Access Point MAY NOT implement a full IP layer (e.g. only the DSCP marking in the IP header MUST be analyzed to classify the packets into the relevant queues, but the forwarding of the packets MAY be based on traditional relaying functions between the tunnel on fixed network and the radio layers).

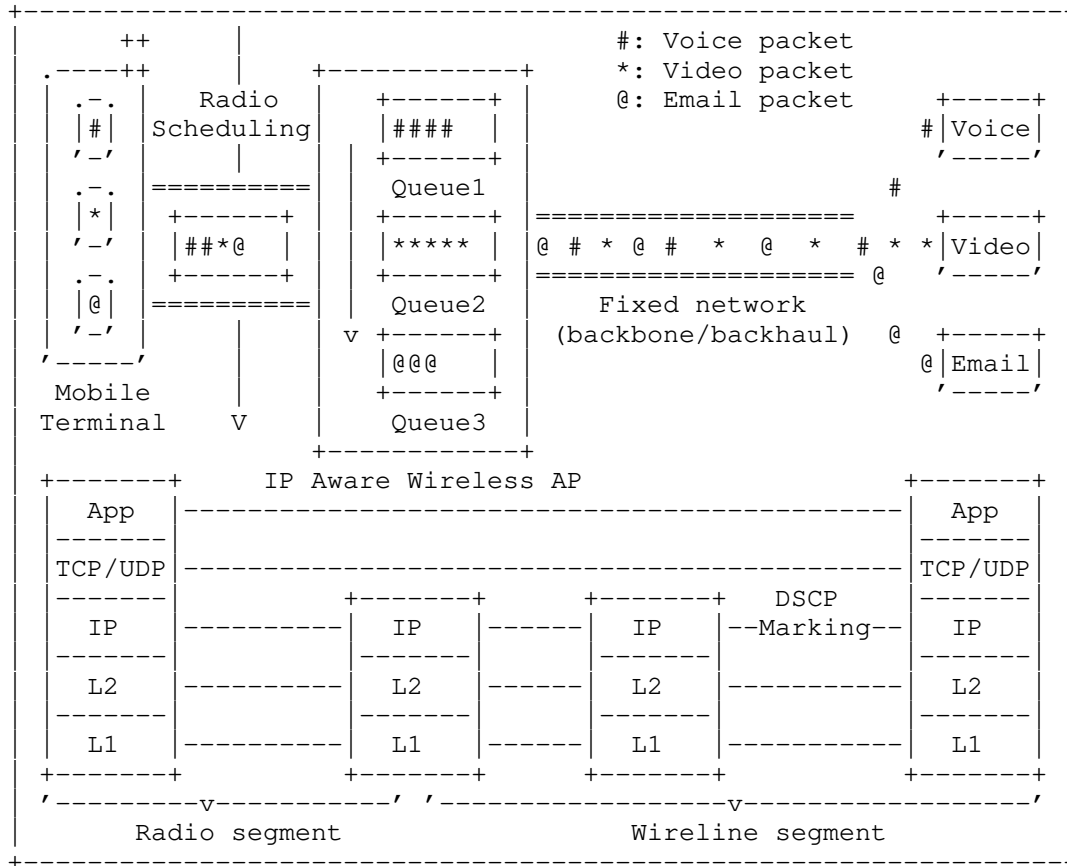


Figure 1 - Packet-oriented QoS management model in mobile networks

4. IP aware wireless Access Point behavior

The IP aware Access Point MUST take into account the DSCP/ToS field of each packet prior to the transmission over the radio segment. For this purpose, an IP multiplexing stage (i.e. an IP queuing system) is added before the radio scheduler. This two-level scheduling process can be implemented through various algorithms. In some cases, it is relevant to consider a model which associates one set of IP queues to each terminal (see for instance the model in section 6.1).

In this draft, we distinguish two main types of behavior for the IP aware Access Point:

- o Intra-bearer arrangements: the allocation of radio resources is independent of the traffic mix waiting for transmission; in this case, radio resources allocated to a given user are determined by a basic radio scheduler which generally takes into account the radio conditions (e.g. algorithm based on Proportional Fairness). The addition of an IP priority queuing system per user before the radio scheduler (without influencing it) allows prioritizing the sensitive flows of a given user against his own other flows when populating the radio frame.
- o Inter-bearer arrangements: the allocation of the radio resources depends on the traffic mix waiting for transmission; in this case, the radio resources allocated to one UE not only depend on the radio conditions of the UE, but also on the traffic mix offered to the IP queuing system. This traffic mix is defined through the DSCP of its constitutive packets. Several approaches are possible (e.g. weighting the allocation according to the prioritized traffic volume / priority queue backlog, ensuring a maximum latency for certain classes, etc.)

The first case (intra-bearer arrangement) is expected to be useful for multi-tasking users (e.g. user running several applications simultaneously, for instance in case of tethering terminals or acting as Mifi access point). This model is further detailed in section 6.1 of this document.

The second case (inter-bearer arrangement) targets real time applications with stringent QoS constraints. This model is further detailed in section 6.2 of this document.

5. Extension to upstream traffic

A future version of this draft will provide some details on the usage of the mechanism for upstream traffic.

Note that some solutions, such as the model described in section 6.1 of this document, are already applicable to upstream traffic without changes in the mobile terminal.

6. Examples of possible models

The following sections provide examples of applicable models for intra-bearer and inter-bearer arrangements. It should be noted that these models are considered as examples only (mainly as references for simulation activity), and that actual implementations may differ from them.

6.1. Model for intra-bearer arrangement

In this section, we describe the intra-bearer arrangement model. The aim of this model is to prioritize the sensitive flows of one UE against its own other flows without modifying the radio resources allocated to this UE. As a consequence, the good properties of the scheduling algorithm are preserved (e.g. trade-off between UE fairness and cell throughput), thus preserving the cell throughput capacity.

As mentioned before, the "IP aware Access Point" MUST take into account the DSCP/ToS field, therefore an IP layer MUST be added on the eNB side, which is not the current practice in most existing cellular networks. In the present model, it is assumed that the marking procedure is performed upstream from the eNB.

On the "IP aware Access Point" side, a two-level scheduler is implemented when considering an intra-bearer arrangement:

The first level consists in an IP priority queuing system composed of "n" finite queues per UE (DiffServ model - one queue per level of QoS). These queues are operated according to a non-preemptive service policy. This means that if one or more high priority packets arrive when a packet of lower priority is served, the high priority packets will be served only after the current service of a low priority packet is complete. For the sake of simplicity, we propose the use of a strict priority policy between queues, but other policies are also possible, such as weighted fair queuing/weighted round robin. The service rate of this IP queuing system is constant during each TTI (Transmission Time Interval). It is calculated for each UE at each TTI by the second level described hereafter.

The second level is a radio scheduler which shares the available radio resources (e.g. RB - Resources Block in LTE) between the UEs.

Different scheduling algorithms are possible for the allocation of radio resources depending on the optimization criteria (delay, throughput, etc., or trade-off between some of these criteria). On the present model, Proportional Fair (PF) scheduler, without any modification, is suggested because it offers a good trade-off between cell throughput and fairness between UEs, and because it is the basis of many scheduler implementations in base stations. PF assigns the available radio resources to UEs every TTI (Transmission Time Interval), regardless of the first level queuing process described above. The PF scheduler algorithm takes into account the radio conditions (e.g. CQI - Channel Quality Indicator in LTE) provided by each UE.

6.2. Models for inter-bearer arrangement

A future version of this draft will provide some details on the applicable models for inter-bearer arrangements.

7. Security Considerations

<Add any security considerations>

8. IANA Considerations

<Add any IANA considerations>

9. References

9.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

9.2. Informative References

- [3] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.
- [4] Hamchaoui, I., Jobert, S., Boufelja, S., "IP aware radio scheduling - Introducing IP QoS management in LTE networks", IEEE ICC'13, June 2013

10. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Sébastien Jobert
Orange
2 avenue Pierre Marzin
22300 LANNION, FRANCE

Email: sebastien.jobert@orange.com

Isabelle Hamchaoui
Orange
2 avenue Pierre Marzin
22300 LANNION, FRANCE

Email: isabelle.hamchaoui@orange.com

William Diego
Orange
2 avenue Pierre Marzin
22300 LANNION, FRANCE

Email: william.diego@orange.com

Transport Area
Internet-Draft
Intended status: Informational
Expires: April 21, 2014

T. Moncaster, Ed.
University of Cambridge
M. Welzl
University of Oslo
D. Ros
Telecom Bretagne
October 18, 2013

Problem Statement: Why the IETF Needs Defined Transport Services
draft-moncaster-tsvwg-transport-services-00

Abstract

The IETF has defined a wide range of transport protocols over the past three decades. However, the majority of these have failed to find traction within the Internet. This has left developers with little choice but to use TCP and UDP for most applications. In many cases the developer isn't interested in which transport protocol they should use. Rather they are interested in the set of services that the protocol provides to their application. TCP provides a very rich set of transport services, but offers no flexibility over which services can be used. By contrast, UDP provides a minimal set of services.

As a consequence many developers have begun to write application-level transport protocols that operate on top of UDP and offer them some of the flexibility they are looking for. We believe that this highlights a real problem: applications would like to be able to specify the services they receive from the transport protocol, but currently transport protocols are not defined in this fashion. There is an additional problem relating to how to ensure new protocols are able to be adopted within the Internet, but that is beyond the scope of this problem statement.

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 <http://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 April 21, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Transport Services	3
2.1. Identifying Transport Services	3
2.2. Exposing Transport Services	4
3. Why Now?	5
4. Security Considerations	5
5. IANA Considerations	6
6. Conclusions	6
7. Contributors and Acknowledgements	6
8. Comments Solicited	6
9. References	7
9.1. Normative References	7
9.2. Informative References	7

1. Introduction

The IETF has defined a wide array of transport protocols including UDP [RFC0768], TCP [RFC0793], SCTP [RFC4960], UDP-Lite [RFC3828], DCCP [RFC4340] and MPTCP [RFC6824]. In most cases new protocols have been defined because the IETF has established that there is a need for a set of behaviours than cannot be offered by any existing transport protocol.

However, for an application programmer, using protocols other than TCP or UDP can be hard: not all protocols are available everywhere, hence a fall-back solution to TCP or UDP must be implemented. Some protocols provide the same services in different ways. Layering decisions must be made (e.g. should a protocol be used natively or over UDP?). Because of these complications, programmers often resort to either using TCP (even if there is a mismatch between the services provided by TCP and the services needed by the application) or implementing their own customised solution over UDP, and the opportunity of benefiting from other transport protocols is lost. Since all these protocols were developed to provide services that solve particular problems, the inability of applications to make use of them is in itself a problem.

We believe this mismatch between the application layer and transport layer can be addressed in a simple fashion. If the socket interface provided a way for applications to request transport services without specifying the protocol, a transport system underneath the socket API could automatically try to make the best of its available resources. It could use available transport protocols in a way that is most beneficial for applications and without the application needing to worry about problems with middlebox traversal. Adopting this approach could give more freedom for diversification to designers of Operating Systems.

2. Transport Services

The transport layer provides many services both to the end application (e.g. multiplexing, flow control, ordering, reliability) and to the network (e.g. congestion control). For the purposes of this document we define Transport Services as follows:

- o A Transport Service is any service provided by the transport layer that can only be correctly implemented with information from the application.

The key word here is "information" -- many existing transport protocols function perfectly adequately because the choice of protocol implicitly includes information about the desired transport capabilities. For instance the choice of TCP implies a desire for reliable, in-order data delivery. However we think that such implicit information is not always sufficient. The rest of this section explains how we propose to identify Transport Services and how those services might then be exposed to the application.

2.1. Identifying Transport Services

One of the key aspects of this work is how to actually identify which Transport Services should be supported. The top-down approach to this would be to identify every possible service that popular applications might need. The problem with this method is that every potential service becomes an item for debate, and it is likely that such an approach would grind on indefinitely. Instead we intend to use a bottom-up approach where we establish the set of services that have already been published in RFCs coming from the Transport Area. This way, much of the discussion about the need to specify these services has already taken place, and it is unnecessary to re-visit those discussions. It is our hope that this approach will lead to identifying a set of service primitives that can be combined to offer a rich set of services to the application.

2.2. Exposing Transport Services

These Transport Services would be exposed to the application via an API. The definition of such an API and the functionality underneath the API are strictly beyond the scope of this problem statement. However in order to show that this is not just an abstract idea we briefly describe three possible approaches.

One approach could be to develop a transport system that fully operates inside the Operating System. This transport system would provide all the defined services for which it can use TCP as a fallback at the expense of efficiency (e.g., TCP's reliable in-order delivery is a special case of reliable unordered delivery, but it may be less efficient). To test whether a particular transport is available it could take the Happy Eyeballs [I-D.wing-tsvwg-happy-eyeballs-sctp] approach proposed for SCTP -- if the SCTP response arrives too late then the connection just uses TCP and the SCTP association information could be cached so that a future connection request to the same destination IP address can automatically use it.

Polyversal TCP [PVTCP] offers another possible approach. This starts by opening a TCP connection and then attempts to establish other paths using different transports. The TCP connection ensures there's always a stable fallback. Having established the initial connection, PVTCP can then use service requests coming through `setsockopt()` to select the most appropriate transport from the available set.

Another approach could be to always rely on UDP only, and develop a whole new transport protocol above UDP which provides all the services, using a single UDP port. Instead of falling back to TCP, this transport system could return an error in case there is no other instance of the transport system available on the other side; the first packets could be used to signal which service is being

requested to the other side (e.g., unordered delivery requires the receiving end to be aware of it).

3. Why Now?

So why do we need to deal with this issue now? There are several answers. Firstly, after several decades of dominance by various flavours of TCP and UDP (plus limited deployment of SCTP [RFC4960]), transport protocols are undergoing significant changes. Recent standards allow for parallel usage of multiple paths (MPTCP [RFC6824] and CMT-SCTP [I-D.tuexen-tsvwg-sctp-multipath]) while other standards allow for scavenger-type traffic LEDBAT [RFC6817]. What sets these apart from e.g. DCCP [RFC4340] is that they have already seen deployment in the wild -- one of the Internet's most popular applications, BitTorrent, uses LEDBAT and MPTCP is already seeing deployment in major operating systems [Bonaventure-Blog]. Meanwhile there is a trend towards tunnelling transports inside UDP -- SCTP over DTLS over UDP is now being shipped with a popular browser in order to support WebRTC [RFC6951][I-D.ietf-tsvwg-sctp-dtls-encaps] while RTMFP [I-D.thornburgh-adobe-rtmfp] and QUIC [QUIC] are recent examples of transport protocols that are implemented over UDP in user space. In a similar vane, Minion [I-D.iyengar-minion-protocol] is a proposal to realise some SCTP-like services with a downwards-compatible extension to TCP.

All of a sudden, application developers are faced with a heterogeneous, complex set of protocols to choose from. Every protocol has its pro's and con's, but often the reasons for making a particular choice depend not on the application's preferences but on the environment (e.g., the choice of Minion vs. SCTP would depend on whether SCTP could successfully be used on a given network path). Choosing a protocol that isn't guaranteed to work requires implementing a fall-back method to e.g. TCP, and making the best possible choice at all times may require sophisticated network measurement techniques. The process could be improved by using a cache to learn which protocols previously worked on a path, but this wouldn't always work in a cloud environment where virtual machines can and do migrate between physical nodes.

We therefore argue that it is necessary to provide mechanisms that automate the choice and usage of the transport protocol underneath the API that is exposed to applications. As a first step towards such automation, we need to define the services that the transport layer should expose to an application (as opposed to today's typical choice of TCP and UDP).

4. Security Considerations

{ToDo} While security could be seen as a Transport Service, we prefer to view it as an intrinsic function of the transport layer. In many cases it is essential for the transport connection to be secure (for instance where confidential data is being transferred across the connection). Even where data security is not essential, connection-level security is desirable in all but fully trusted environments. So unless connections actively choose not to be secure, we would expect them to use TLS [RFC5246].

5. IANA Considerations

This document makes no request to IANA although in future an IANA register of Transport Services may be required.

6. Conclusions

After decades of relative stagnation the last few years have seen many new transport protocols being developed and adopted in the wild. This evolution has been driven by the changing needs of application developers and has been enabled by moving transport services into the application or by tunnelling over an underlying UDP connection.

Application developers are now faced with a genuine choice of different protocols with no clear mechanism for choosing between them. At the same time, the still-limited deployment of some protocols means that the developer must always provide a fall-back to an alternative transport if they want to guarantee the connection will work. This is not a sustainable state of affairs and we believe that in future a new transport API will be needed that provides the mechanisms to facilitate the choice of transport protocol. The first step towards this is to identify the set of Transport Services that a transport protocol is able to expose to the application. We propose doing this in a bottom-up fashion, starting from the list of services available in transport protocols that are specified in RFCs.

7. Contributors and Acknowledgements

Many thanks to the many people that have contributed to this effort so far including Michael Tuexen, Arjuna Sathiaselalan, Jon Crowcroft, Marwan Fayed and Bernd Reuther.

8. Comments Solicited

To be removed by RFC Editor: This draft is the first step towards an IETF BoF on Transport Services. Comments and questions are encouraged and very welcome. They can be addressed to the current mailing list <transport-services@ifi.uio.no> and/or to the authors. We also have a website at <<https://sites.google.com/site/transportprotocolservices/>>

9. References

9.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, December 2012.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.
- [RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication", RFC 6951, May 2013.

9.2. Informative References

[Bonaventure-Blog]

Bonaventure, O., "Blog Entry: MPTCP used in iOS 7",
September 2013.

[I-D.dreibholz-tsvwg-sctpsocket-multipath]

Dreibholz, T., Becke, M., and H. Adhari, "SCTP Socket API
Extensions for Concurrent Multipath Transfer", draft-
dreibholz-tsvwg-sctpsocket-multipath-06 (work in
progress), July 2013.

[I-D.ietf-tsvwg-sctp-dtls-encaps]

Jesup, R., Loreto, S., Stewart, R., and M. Tuexen, "DTLS
Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-
dtls-encaps-01 (work in progress), July 2013.

[I-D.iyengar-minion-protocol]

Jana, J., Cheshire, S., and J. Graessley, "Minion - Wire
Protocol", draft-iyengar-minion-protocol-01 (work in
progress), July 2013.

[I-D.thornburgh-adobe-rtmfp]

Thornburgh, M., "Adobe's Secure Real-Time Media Flow
Protocol", draft-thornburgh-adobe-rtmfp-10 (work in
progress), July 2013.

[I-D.tuexen-tsvwg-sctp-multipath]

Amer, P., Becke, M., Dreibholz, T., Ekiz, N., Jana, J.,
Natarajan, P., Stewart, R., and M. Tuexen, "Load Sharing
for the Stream Control Transmission Protocol (SCTP)",
draft-tuexen-tsvwg-sctp-multipath-07 (work in progress),
October 2013.

[I-D.wing-tsvwg-happy-eyeballs-sctp]

Wing, D. and P. Natarajan, "Happy Eyeballs: Trending
Towards Success with SCTP", draft-wing-tsvwg-happy-
eyeballs-sctp-02 (work in progress), October 2010.

[PVTCP]

Nabi, Z., Moncaster, T., Madhavapeddy, A., Hand, S., and
J. Crowcroft, "Evolving TCP: how hard can it be?",
Proceedings of ACM CoNEXT 2012, December 2012.

[QUIC]

Roskind, J., "Quick UDP Internet Connections", June 2013.

Authors' Addresses

Toby Moncaster (editor)
University of Cambridge
Computer Laboratory
J.J. Thomson Avenue
Cambridge CB3 0FD
UK

Phone: +44 1223 763654
EMail: toby.moncaster@cl.cam.ac.uk

Michael Welzl
University of Oslo
PO Box 1080 Blindern
Oslo N-0316
Norway

Phone: +47 22 85 24 20
EMail: michawe@ifi.uio.no

David Ros
Telecom Bretagne
Rue de la Chataigneraie, CS 17607
35576 Cesson Sevigne cedex
France

Phone: +33 2 99 12 70 46
EMail: david.ros@telecom-bretagne.eu