         Low Latency Applications and the Internet Architecture
                  draft-arkko-arch-low-latency-02

Abstract

   Some recent Internet technology developments relate to improvements
   in communications latency. For instance, improvements in radio
   communications or the recent work in IETF transport, security, and
   web protocols. There are also potential applications where latency
   would play a more significant role than it has traditionally been in
   the Internet communications. Modern networking systems offer many
   tools for building low-latency networks, from highly optimised
   individual protocol components to software controlled, virtualised
   and tailored network functions. This memo views the developments
   from a system viewpoint, and considers the potential future stresses
   that the strive for low-latency support for applications may bring.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Some recent Internet technology developments relate to improvements
in communications latency. For instance, improvements in radio
communications or the recent work in IETF transport, security, and
web protocols.

There are also potential applications where latency would play a more
significant role than it has traditionally been in the Internet
communications.

New applications or technologies do not necessarily imply that
latency should be the main driving concern, or that any further
efforts are needed, beyond those already ongoing. Indeed, modern
networking systems offer many tools for building low-latency
networks, across the stack. At the IETF, for instance, there has
been a recent increase in work related to transport, security, and
web application protocols, in part to make significant improvements
in latency and connection set-up times. Similar efforts for other

components of communications technology exist in 3GPP, IEEE, and other standards organisations.

Despite a large number of specific developments, it may be interesting to view the developments from a system viewpoint, and to consider the potential future stresses that the strive for low-latency support applications may bring.

The rest of this memo is organised as follows: Section 2 discusses potential applications for low-latency communications. Section 4 reviews some of the recent work across the stack, related to latency improvements. Finally, Section 5 discusses some of the implications (and non-implications) from an architectural perspective.

2. Applications with Special Focus on Low Latency

Most Internet applications enjoy significant benefits from low-latency communications in the form of faster setup and response times as well as higher bandwidth communications enabled by transport protocol behaviour [RFC7323].

There are also potential applications where latency would play an even more significant role. For instance, embedding communications technology in automation or traffic systems, or consumer applications such as augmented or virtual reality where due to the human brain's perceptual limits variability in latency may not be feasible, i.e., render the service unusable due to motion sickness caused.

Many of the Internet-of-Things and critical services use cases in 5G, for instance, have been listed as requiring low latency and high reliability for communications [ER2015] [HU2015] [NGMN2015] [NO2015] [QU2016] [IMT2020].

Some example use cases include optimisation of utility services such as electricity networks, connected automation systems in factories, remote control of machinery such as mining equipment, or embedded technology in road or railway traffic.

The different applications vary in terms of their needs. Some may be very focused on high-speed local area communication, others need to connect at optimal speed over a wide-area network, and yet others need to find the right ways to provide global services without incurring unreasonable delays.

For these reasons it is difficult to specify what "low latency" means in terms of specific delays. Applications and network scenarios differ. Reaching a 50ms latency may be enough for some applications while others may require 50us. Obviously, latency is ultimately

limited by physics, location, and topology. Individual link
characteristics are important, but system level communication needs
both in terms of what is being communicated and between what parties
matter more.

Note that when we say "low-latency capabilities", there is no intent
to imply any specific implementation of those capabilities. In
particular, we look at the low-latency requirements from a broader
perspective than Quality-of-Service guarantees or separating traffic
onto different classes. Indeed, while today's virtualisation and
software-driven technologies give us more tools to deal with those
kinds of arrangements as well, past experience on deploying Quality-
of-Service mechanisms in the Internet should give us a pause
[CC2015].

It is not the purpose of this memo to analyse the application
requirements for low-latency applications much further; for our
purposes it suffices to note that there are applications that are
enabled by low-latency capabilities of the underlying network
infrastructure.

3. Role of Low-Latency vs. Other Communications

There are some limited applications that rely solely on local
communication. One example of such an application is vehicles
communicating braking status to nearby ones.

Also, while many applications run in the global Internet, some are
designed for specialised networks that may not have full or even any
Internet connectivity, but yet use IP technology.

However, many applications will include also wide-area communication.
If the factory automation machines are not talking other than with
themselves, at least their control systems are doing so in order to
ensure parts orders, monitoring and maintenance by equipment
manufacturers, and so on. This does not imply that these perhaps
critical applications are openly accessible from the Internet, but
many of them are likely to communicate outside their immediate
surroundings.

Many applications also rely on wide-area connectivity for software
updates.

As a result, this document recommends that when building
architectures for low-latency applications it is important to take
into account that these applications can also benefit from
communications elsewhere. Or at the very least, the existence of a

specialised communications link or network should not be immediately
taken to mean that no other communications are needed.

[4](#). Selected Improvements to Communications Latency

It should be noted that latency is a very broad topic in
communications protocol design, almost as broad as "security", or
even "correctness".

Implementation techniques to satisfy these requirements vary, some
applications can be built with sufficient fast local networking
capabilities, others may require, for instance, building world-wide,
distributed content delivery mechanisms.

Modern networking systems offer many tools for building low-latency
networks, from highly optimised individual protocol components
[I-D.ietf-tls-tls13] [I-D.ietf-quic-transport] [RFC7413] [RFC7540] to
software controlled, virtualised and tailored network functions
[NFV2012] [RFC7665] [I-D.ietf-sfc-nsh] [OF2008]. Data- and software-
driven network management and orchestration tools enable networks to
be built to serve particular needs as well as to optimize workload
placement in a way low-latency requirements could be met.

Obviously, low-latency communications are not merely a matter of
protocols and their optimisation. Implementation techniques matter,
from the placement of network functions and nodes in the right
places, to the quality of individual function implementations.
Today's technology allows much freedom on placement of (virtual)
functions at chosen locations and many options for all functions
ranging from load balancing to storage to packet processing to
management tools. Good design can provide significant gains by
reducing latency in and between network components, reducing the
necessary control traffic and state synchronization, and so on.

Across the stack there are also many other protocol tools, as well as
tools being in development, e.g., a new transport design [L4S] at the
IETF.

On the lower layers, improvements in radio communications are being
made. For instance, the IEEE 802.1 Time-Sensitive Networking Task
Group [TSN8021] has worked to define precise time synchronization
mechanisms for a local area network, and scheduling mechanisms to
enable different classes of traffic to use the same network while
minimising jitter and latency. At the IETF, the DETNET working group
is taking these capabilities and applying them for layer 3 networking
[DETNET].

The 3GPP 5G requirements for next-generation access technology are
stringent, and are leading to the optimization of the radio
interfaces. The requirements specify a one-way latency limit of
0.5ms for ultra-reliable low-latency communications [TS38913]. But
again, mere latency numbers mean very little without the context of a
system and what an application needs to communicate and with whom.

5. Architectural Considerations

Despite a large number of specific developments, it may be
interesting to view the developments from a system viewpoint, and to
consider the potential future stresses that the strive for low-
latency support for applications may bring.

5.1. Background

To begin with, it may be useful to observe that the requirements and
developments outlined above do not necessarily imply that any
specific new technology is needed or that the nature of
communications in the Internet would somehow fundamentally change.
And certainly not that latency should be the only or primary concern
in technology development.

With the drive for a new class of applications, there is often an
expectation that this means significant changes. However, all
changes need to stand on their own, be justifiable and deployable on
a global network. For instance, the discussion around the
introduction of the newest 4K or 8K high-definition video streaming
applications is reminiscent of the discussions about the introduction
of VoIP applications in the Internet. At the time, there was some
expectation that special arrangements and Quality-of-Service
mechanisms might be needed to support this new traffic class. This
turned out to be not true, at least not in general networks.

Experience tells us, for instance, that deploying Quality-of-Service
mechanisms in the Internet is hard, not so much because of the
technology itself, but due to lack of forces that would be able to
drive the necessary business changes in the ecosystem for the
technology to be feasibly deployable [CC2015]. As claffy and Clark
note:

   "Although the Internet has a standards body (the IETF) to resolve
   technical issues, it lacks any similar forum to discuss business
   issues such as how to allocate revenues among competing ISPs
   offering enhanced services. In the U.S., ISPs feared such
   discussions would risk anti-trust scrutiny. Thus, lacking a way
   to negotiate the business implications of QoS, it was considered a
   cost rather than a potential source of revenue. Yet, the

relentless growth of a diversity of applications with widely
varying performance requirements continued on the public Internet,
with ISPs using relatively primitive, and not always completely
benign, mechanisms for handling them."

These difficulties should not be read as prohibiting all changes. Of
course, change can also seem unlikely even in cases where it becomes
absolutely necessary or the forces necessary to make a change have
actually built up. As a result, statements regarding change in the
Internet should be carefully evaluated on their merits from both
technical and ecosystem perspective.

Secondly, we often consider characteristics from a too narrow
viewpoint. In the case of latency, it is easy to focus on a
particular protocol or link, whereas from the user perspective
latency is a property of the system, not a property of an individual
component.

For instance, improvements on the performance of one link on a
communications path can be insignificant, if the other parts make up
a significant fraction of the system-level latency. That may seem
obvious, but many applications are highly dependent on communications
between a number of different parties which may reside in different
places. For instance, a third party may perform authentication for a
cloud-based service that also interacts with user's devices and a
number of different sensors and actuators.

We cannot change the speed of light, and a single exchange with
another part of the world may result in a 100ms delay, or about 200
times longer than the expected 5G radio link delay for critical
applications. It is clear that designing applications from a system
perspective is very important.

5.2. Implications

This section discusses a selected set of architectural effects and
design choices within applications that desire low latency
communications.

5.2.1. Service Distribution

As noted above, low-latency applications need to pay particular
attention to the placement of services in the global network.
Operations that are on the critical path for the low-latency aspects
of an application are unlikely to work well if those communications
need to traverse half of the Internet.

Many widely used services are already distributed and replicated
throughout the world, to minimise communications latency. But many
other services are not distributed in this manner. For low-latency
applications such distribution becomes necessary. Hosting a global
service in one location is not feasible due to latency, even when
from a scale perspective a single server might otherwise suffice for
the service. All major public cloud providers offer CDN services to
their customers - AWS's CloudFront, Google's Cloud CDN and Azure's
CDN to mention a few.

Content-Delivery Networks (CDNs) and similar arrangements are likely
to flourish because of this. These arrangements can bring content
close to end-users, and have a significant impact on latency.
Typical CDN arrangements provide services that are on a global scale
nearby, e.g., in the same country or even at the ISP's datacenter.

Today's CDNs are of course just one form of distributed service
implementation. Previous generations, such as web caching, have
existed as well, and it is likely that the current arrangements will
evolve in the future. CDN evolution is also naturally affected not
only by the need to provide services closer to the user, but also
through the fine-grained control and visibility mechanisms that it
gives to the content owners. Such factors continue to affect also
future evolution, e.g., any information-centric networking solutions
that might emerge.

5.2.2. Edge Computing

Recent advances in "edge computing" take the more traditional type
service like CDN as well as a new class of services that require
"local compute" capabilities placement even further by providing
services near the users. This would enable more extreme uses cases
where latency from, say, ISP datacenter to the users is considered
too high. An important consideration is what is considered an edge,
however. From Internet perspective edge usually refers to the IP
point of presence or the first IP hop. But given the centralised
nature of many access networks, some of the discussions around the
use of edge computing also involve components at the edge that are
much closer to user than the first IP hop. Special arrangements are
needed to enable direct IP connectivity from the user equipment to
these components.

5.2.3. Routing and tunnels

How the communications are routed also matters. For instance,
architectures based on tunneling to a central point may incur extra
delay. One way to address this pressure is to use SDN- and
virtualisation-based networks that can be provisioned in the desired

manner, so that, for instance, instances of tunnel servers can be
placed in the topologically optimal place for a particular
application.

5.2.4. Alternative Paths and Control Tension

Recent developments in multipath transport protocols [RFC6824] also
provide application- and service-level control of some of the
networking behaviour. Similar choices among alternative paths also
exist in simpler techniques, ranging from server selection algorithms
to IPv6 "Happy Eyeballs" algorithms [RFC6555]. In all of these cases
an application makes some observations of the environment and decides
to use an alternative path or target that is perceived to be best
suited for the application's needs.

In all of these multipath and alternative selection techniques there
is tension between application control (often by content providers)
and network control (often by network operators).

One special case where that tension has appeared in the past is
whether there should be ways to provide information from applications
to networks on how packets should be treated. This was extensively
discussed during the discussion stemming from implications of
increased use of encryption in the Internet, and how that affects
operators [I-D.nrooney-marnew-report].

Another case where there is tension is between mechanisms designed
for a single link or network vs. end-to-end mechanisms. Many of the
stated requirements for low-latency applications are explicitly about
end-to-end characteristics and capabilities. Yet, the two mechanisms
are very different, and most of the deployment difficulties reported
in [CC2015] relate to end-to-end mechanisms.

Note that some of the multipath techniques can be used either by
endpoints or by the network. Proxy-based Multipath TCP is one
example of this [I-D.boucadair-mptcp-plain-mode].

5.2.5. Quality-of-Service

Existing approaches have not necessarily proven to be technical
deficient in any way, but it seems that it is reasonable to draw some
conclusions about the lack of feasibility for deploying mechanisms
that require a high degree of coordination among multiple parties.
Without significant changes in the marketplace or other conditions,
these types of solutions do not seem likely to get traction.

The other observation that may be worth noting is that many networks
have focused on providing highly tuned services for a relatively

small fractions of traffic. While this may be justifiable from the
perspective of special applications needing that support, this does
not seem to produce a good bang-for-the-buck ratio. There's
relatively small amount of work on mechanisms that would help a large
fraction of applications. For instance, in many access networks the
over-the-top content is by far the biggest source of traffic.
Solutions that are business-wise deployable for such traffic would
seem preferrable.

5.2.6. Cross-Layer Optimisations

In the search for even faster connection setup times one obvious
technique is cross-layer optimisation. We have seen some of this in
the IETF in the rethinking of the layers for transport, transport
layer security, and application framework protocols. By taking into
account the protocol layer interactions or even bundling the protocol
design together, it is relatively easy to optimise the connection
setup time, as evidenced by recent efforts to look for "0-RTT"
designs in various protocols.

But while cross-layer optimisation can bring benefits, it also has
downsides. In particular, it connects different parts of the stack
in additional ways. This can lead to difficulties in further
evolution of the technology, if done wrong.

   In the case of the IETF transport protocol evolution, significant
   improvements were made to ensure better evolvability of the
   protocols than what we've experienced with TCP, starting from an
   ability to implement the new protocols in applications rather than
   in the kernel.

While the connection setup is an obvious example, cross-layer
optimisations are not limited to them. Interfaces between
application, transport, networking, and link layers can provide
information and set parameters that improve latency. For instance,
setting DSCP values or requesting a specialised L2 service for a
particular application. Cross-Layer optimisations between lower
layers will be discussed in the upcoming versions of the draft.

The effects of badly designed cross-layer optimisation are a
particular form of Internet ossification. The general networking
trend, however, is for greater flexibility and programmability.
Arguably, the ease at which networks can evolve is probably even more
important than their specific characteristics.

These comments about cross-layer optimisation should not be
interpreted to mean that protocol design should not take into account
how other layers behave. The IETF has a long tradition of discussing

link layer design implications for Internet communications (see,
e.g., the results of the PILC working group [RFC3819]).

5.3. Recommendations for Further Work

Low-latency applications continue to be a hot topic in networking.
The following topics in particular deserve further work from an
architectural point of view:

o Application architectures for globally connected but low-latency
   services.

o What are the issues with inter-domain Quality-of-Service
   mechanisms? Are there approaches that would offer progress on
   this field? Work on Quality-of-Service mechanisms that are
   deployable for common cases, and without excessive need for
   technical and non-technical coordination across the ecosystem.

o Network architectures that employ tunneling, and mitigations
   against the delay impacts of tunnels (such as tunnel server
   placement or "local breakout" techniques). Low latency often
   implies high reliability, special care is to be taken of network
   convergence, and other, relevant characteristics of the underlying
   infrastructure.

o The emergence of cross-layer optimisations and how that affects
   the Internet architecture and its future evolution.

o Inter-organisatorial matters, e.g., to what extent different
   standards organisations need to talk about low latency effects and
   ongoing work, to promote system-level understanding?

Overall, this memo stresses the importance of the system-level
understanding of Internet applications and their latency issues.
Efforts to address specific sub-issues are unlikely to be fruitful
without a holistic plan.

In the authors' opinion, the most extreme use cases (e.g., 1ms or
smaller latencies) are not worth building general-purpose networks
for. But having the necessary tools so that networks can be flexible
for the more general cases is very useful, as there are many
applications that can benefit from the added flexibility. The key
tools for this include ability to manage network function placement
and topology.

6. Acknowledgements

   The author would like to thank Brian Trammell, Mirja Kuehlewind,
   Linda Dunbar, Goran Rune, Ari Keranen, James Kempf, Stephen Farrell,
   Mohamed Boucadair, Kumar Balachandran, Jani-Pekka Kainulainen, Goran
   AP Eriksson, and many others for interesting discussions in this
   problem space.

   The author would also like to acknowledge the important contribution
   that [I-D.dunbar-e2e-latency-arch-view-and-gaps] made in this topic.

7. Informative References

   [CC2015]   claffy, kc. and D. Clark, "Adding Enhanced Services to the
              Internet: Lessons from History", September 2015
              (https://www.caida.org/publications/papers/2015/
              adding_enhanced_services_internet/
              adding_enhanced_services_internet.pdf).

   [DETNET]   "Deterministic Networking (DETNET) Working Group", March
              2016 (https://tools.ietf.org/wg/detnet/charters).

   [ER2015]   Yilmaz, O., "5G Radio Access for Ultra-Reliable and Low-
              Latency Communications", Ericsson Research Blog, May 2015
              (https://www.ericsson.com/research-blog/5g/5g-radio-
              access-for-ultra-reliable-and-low-latency-
              communications/).

   [HU2015]   "5G Vision: 100 Billion connections, 1 ms Latency, and 10
              Gbps Throughput", Huawei 2015
              (http://www.huawei.com/minisite/5g/en/defining-5g.html).

   [I-D.boucadair-mptcp-plain-mode]
              Boucadair, M., Jacquenet, C., Bonaventure, O., Behaghel,
              D., stefano.secci@lip6.fr, s., Henderickx, W., Skog, R.,
              Vinapamula, S., Seo, S., Cloetens, W., Meyer, U.,
              Contreras, L., and B. Peirens, "Extensions for Network-
              Assisted MPTCP Deployment Models", draft-boucadair-mptcp-
              plain-mode-10 (work in progress), March 2017.

   [I-D.dunbar-e2e-latency-arch-view-and-gaps]
              Dunbar, L., "Architectural View of E2E Latency and Gaps",
              draft-dunbar-e2e-latency-arch-view-and-gaps-01 (work in
              progress), March 2017.

[I-D.ietf-quic-transport]
          Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
          and Secure Transport", draft-ietf-quic-transport-07 (work
          in progress), October 2017.

[I-D.ietf-sfc-nsh]
          Quinn, P., Elzur, U., and C. Pignataro, "Network Service
          Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress),
          October 2017.

[I-D.ietf-tls-tls13]
          Rescorla, E., "The Transport Layer Security (TLS) Protocol
          Version 1.3", draft-ietf-tls-tls13-21 (work in progress),
          July 2017.

[I-D.nrooney-marnew-report]
          Rooney, N., "IAB Workshop on Managing Radio Networks in an
          Encrypted World (MaRNEW) Report", draft-nrooney-marnew-
          report-03 (work in progress), June 2017.

[IMT2020] "Framework and overall objectives of the future
          development of IMT for 2020 and beyond", ITU
          Recommendation M.2083-0, September 2015
          (http://www.itu.int/rec/R-REC-M.2083-0-201509-I/en).

[L4S]   "Low Latency Low Loss Scalable throughput (L4S) Birds-of-
          Feather Session", July 2016
          (https://datatracker.ietf.org/wg/l4s/charter/).

[NFV2012] "Network Functions Virtualisation - Introductory White
          Paper", ETSI,
          http://portal.etsi.org/NFV/NFV_White_Paper.pdf, October
          2012.

[NGMN2015]
          "5G White Paper", NGMN Alliance, February 2015
          (https://www.ngmn.org/fileadmin/ngmn/content/downloads/
          Technical/2015/NGMN_5G_White_Paper_V1_0.pdf).

[NO2015]   Doppler, K., "5G the next major wireless standard", DREAMS
          Seminar, January 2015
          (https://chess.eecs.berkeley.edu/pubs/1084/
          doppler-DREAMS_5G_jan15.pdf).

   [OF2008]   McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G.,
               Peterson, L., Rexford, J., Shenker, S., and J. Turner,
               "OpenFlow: Enabling Innovation in Campus Networks", ACM
               SIGCOMM Computer Communication Review, Volume 38, Issue 2,
               pp. 69-74 2008.

   [QU2016]   "Leading the world to 5G", Qualcomm, February 2016
               (https://www.qualcomm.com/media/documents/files/
               qualcomm-5g-vision-presentation.pdf).

   [RFC3819]   Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D.,
               Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L.
               Wood, "Advice for Internet Subnetwork Designers", BCP 89,
               RFC 3819, DOI 10.17487/RFC3819, July 2004,
               <https://www.rfc-editor.org/info/rfc3819>.

   [RFC6555]   Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with
               Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April
               2012, <https://www.rfc-editor.org/info/rfc6555>.

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

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

   [RFC7413]   Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP
               Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014,
               <https://www.rfc-editor.org/info/rfc7413>.

   [RFC7540]   Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext
               Transfer Protocol Version 2 (HTTP/2)", RFC 7540,
               DOI 10.17487/RFC7540, May 2015,
               <https://www.rfc-editor.org/info/rfc7540>.

   [RFC7665]   Halpern, J., Ed. and C. Pignataro, Ed., "Service Function
               Chaining (SFC) Architecture", RFC 7665,
               DOI 10.17487/RFC7665, October 2015,
               <https://www.rfc-editor.org/info/rfc7665>.

   [TS38913]   "3rd Generation Partnership Project; Technical
               Specification Group Radio Access Network; Study on
               Scenarios and Requirements for Next Generation Access
               Technologies; (Release 14)", 3GPP Technical Report TR
               38.913 V14.2.0, March 2017
               (https://portal.3gpp.org/desktopmodules/Specifications/
               SpecificationDetails.aspx?specificationId=2996).

   [TSN8021]   "Time-Sensitive Networking Task Group", IEEE
               (http://www.ieee802.org/1/pages/tsn.html).

Authors' Addresses

   Jari Arkko
   Ericsson
   Kauniainen 02700
   Finland

   Email: jari.arkko@piuha.net

   Jeff Tantsura
   Futurewei, Future Networks

   Email: jefftant.ietf@gmail.com