ICNRG D. Oran

Internet-Draft Network Systems Research and Design

Intended status: Informational October 12, 2019

Expires: April 14, 2020

Considerations in the development of a QoS Architecture for CCNx-like

ICN protocols

draft-oran-icnrg-qosarch-02

Abstract

This is a position paper. It documents the author's personal views

on how Quality of Service (QoS) capabilities ought to be accommodated

in ICN protocols like CCNx or NDN which employ flow-balanced

Interest/Data exchanges and hop-by-hop forwarding state as their

fundamental machinery. It argues that such protocols demand a

substantially different approach to QoS from that taken in TCP/IP,

and proposes specific design patterns to achieve both classification

and differentiated QoS treatment on both a flow and aggregate basis.

It also considers the effect of caches as a resource in addition to

memory, CPU and link bandwidth that should be subject to explicitly

unfair resource allocation. The proposed methods are intended to

operate purely at the network layer, providing the primitives needed

to achieve both transport and higher layer QoS objectives. It

explicitly excludes any discussion of Quality of Experience (QoE)

which can only be assessed and controlled at the application layer or

above.

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 https://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 14, 2020.

D. Oran Expires April 14, 2020 [Page 1]

Internet-Draft ICN QoS Architecture October 2019

Copyright Notice

Copyright (c) 2019 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

(https://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. Requirements Language . . . . . . . . . . . . . . . . . . . . 4

3. Some background on the nature and properties of Quality of

Service in network protocols . . . . . . . . . . . . . . . . 4

3.1. Congestion Control basics relevant to ICN . . . . . . . . 5

4. What can we control to achieve QoS in ICN? . . . . . . . . . 6

5. How does this relate to QoS in TCP/IP? . . . . . . . . . . . 8

6. Why is ICN Different? Can we do Better? . . . . . . . . . . . 9

6.1. Equivalence class capabilities . . . . . . . . . . . . . 9

6.2. Topology interactions with QoS . . . . . . . . . . . . . 10

6.3. Specification of QoS treatments . . . . . . . . . . . . . 10

6.4. ICN forwarding semantics effect on QoS . . . . . . . . . 11

6.5. QoS interactions with Caching . . . . . . . . . . . . . . 11

7. A strawman set of principles to guide QoS architecture for

ICN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

8. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 18

9. Security Considerations . . . . . . . . . . . . . . . . . . . 18

10. References . . . . . . . . . . . . . . . . . . . . . . . . . 19

10.1. Normative References . . . . . . . . . . . . . . . . . . 19

10.2. Informative References . . . . . . . . . . . . . . . . . 19

Author's Address . . . . . . . . . . . . . . . . . . . . . . . . 24

1. Introduction

The TCP/IP protocol suite used on today's Internet has over 30 years

of accumulated research and engineering into the provision of Quality

of Service machinery, employed with varying success in different

environments. ICN protocols like Named Data Networking (NDN [NDN])

and Content-Centric Networking (CCNx [RFC8569],[RFC8609]) have an

accumulated 10 years of research and very little deployment. We

therefore have the opportunity to either recapitulate the approaches

D. Oran Expires April 14, 2020 [Page 2]

Internet-Draft ICN QoS Architecture October 2019

taken with TCP/IP (e.g. IntServ [RFC2998] and Diffserv [RFC2474]) or

design a new architecture and associated mechanisms aligned with the

properties of ICN protocols, which differ substantially from those of

TCP/IP. This position paper advocates the latter approach and

comprises the author's personal views on how Quality of Service (QoS)

capabilities ought to be accommodated in ICN protocols like CCNx or

NDN. Specifically, these protocols differ in fundamental ways from

TCP/IP. The important differences are summarized in the following

table:

+---------------------------------+---------------------------------+

| TCP/IP | CCNx or NDN |

+---------------------------------+---------------------------------+

| Stateless forwarding | Stateful forwarding |

| Simple Packets | Object model with optional |

| | caching |

| Pure datagram model | Request-response model |

| Asymmetric Routing | Symmetric Routing |

| Independent flow directions | Flow balance |

| Flows grouped by IP prefix and | Flows grouped by name prefix |

| port | |

| End-to-end congestion control | Hop-by-hop congestion control |

+---------------------------------+---------------------------------+

Table 1: Differences between TCP/IP and ICN relevant to QoS architecture

This document proposes specific design patterns to achieve both flow

classification and differentiated QoS treatment for ICN on both a

flow and aggregate basis. It also considers the effect of caches as

a resource in addition to memory, CPU and link bandwidth that should

be subject to explicitly unfair resource allocation. The proposed

methods are intended to operate purely at the network layer,

providing the primitives needed to achieve both transport and higher-layer QoS objectives. It does not propose detailed protocol

machinery to achieve these goals; it leaves these to supplementary

specifications, such as [I-D.moiseenko-icnrg-flowclass]. It

explicitly excludes any discussion of Quality of Experience (QoE)

which can only be assessed and controlled at the application layer or

above.

Much of this document is derived from presentations the author has

given at ICNRG meetings over the last few years that are available

through the IETF datatracker (see, for example [Oran2018QoSslides]).

D. Oran Expires April 14, 2020 [Page 3]

Internet-Draft ICN QoS Architecture October 2019

2. Requirements Language

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].

3. Some background on the nature and properties of Quality of Service

in network protocols

Much of this background material is tutorial and can be simply

skipped by readers familiar with the long and checkered history of

quality of service in packet networks. Other parts of it are

polemical yet serve to illuminate the author's personal biases and

technical views.

All networking systems provide some degree of "quality of service" in

that they exhibit non-zero utility when offered traffic to carry.

The term, therefore, is used to describe systems that control the

allocation of various resources in order to achieve \_managed

unfairness\_. Absent explicit mechanisms to decide what traffic to be

unfair to, most systems try to achieve some form of "fairness" in the

allocation of resources, optimizing the overall utility delivered to

all demands under the constraint of available resources. From this, it

should be obvious that you cannot use QoS mechanisms to create or

otherwise increase resource capacity! In fact, all known QoS schemes

have non-zero overhead and hence may (albeit slightly) decrease the

total resources available to carry user traffic.

Further, accumulated experience seems to indicate that QoS is helpful

in a fairly narrow range of network conditions:

o If your resources are lightly loaded, you don't need it, as

neither congestive loss nor substantial queueing delay occurs

o If your resources are heavily oversubscribed, it doesn't save you.

So many users will be unhappy that you are probably not delivering

a viable service

o Failures can rapidly shift your state from the first above to the

second, in which case either:

\* your QoS machinery cannot respond quickly enough to maintain

the advertised service quality continuously, or

\* resource allocations are sufficiently conservative to result in

substantial wasted capacity under non-failure conditions

D. Oran Expires April 14, 2020 [Page 4]

Internet-Draft ICN QoS Architecture October 2019

Nevertheless, though not universally deployed, QoS is advantageous at

least for some applications and some network environments. Some

examples include:

o applications with steep utility functions [Shenker2006], such as

real-time multimedia

o applications with safety-critical operational constraints, such as

avionics or industrial automation

o dedicated or tightly managed networks whose economics depend on

strict adherence to challenging service level agreements (SLAs)

Another factor in the design and deployment of QoS is the scalability

and scope over which the desired service can be achieved. Here there

are two major considerations, one technical, the other economic/

political:

o Some signaled QoS schemes, such as RSVP [RFC2205], maintain state

in routers for each flow, which scales linearly with the number of

flows. For core routers through which pass millions to billions

of flows, the memory required is infeasible to provide.

o The Internet is comprised of many minimally cooperating autonomous

systems [AS]. There are practically no successful examples of QoS

deployments crossing the AS boundaries of multiple service

providers. This in almost all cases limits the applicability of

QoS capabilities to be intra-domain.

Finally, the relationship between QoS and either accounting or

billing is murky. Some schemes can accurately account for resource

consumption and ascertain to which user to allocate the usage.

Others cannot. While the choice of mechanism may have important

practical, economic, and political consequences for cost and workable

business models, this document considers none of those things and

discusses QoS only in the context of providing managed unfairness.

Some further background on congestion control for ICN is below.

3.1. Congestion Control basics relevant to ICN

Congestion control is necessary in any packet network that

multiplexes traffic among multiple sources and destinations in order

to:

1. Prevent collapse of utility due to overload, where the total

offered service declines as load increases, perhaps

precipitously, rather than increasing or remaining flat.

D. Oran Expires April 14, 2020 [Page 5]

Internet-Draft ICN QoS Architecture October 2019

2. Avoid starvation of some traffic due to excessive demand by other

traffic.

3. Beyond the basic protections against starvation, achieve

"fairness" among competing traffic. Two common objective

functions are [minmaxfairness] and [proportionalfairness] both of

which have been implemented and deployed successfully on packet

networks for many years.

Before moving on to QoS, it is useful to consider how congestion

control works in NDN or CCNx. Unlike the IP protocol family, which

relies exclusively on end-to-end congestion control (e.g.

TCP[RFC0793], DCCP[RFC4340], SCTP[RFC4960],

QUIC[I-D.ietf-quic-transport]), CCNx and NDN can employ hop-by-hop

congestion control. There is per-Interest/Data state at every hop of

the path and therefore for each outstanding Interest, bandwidth for

the data returning on the inverse path can be allocated. In the current

designs, this allocation is often done using Interest counting. By

accepting one Interest packet from a downstream node, implicitly this

provides a guarantee (either hard or soft) that there is sufficient

bandwidth on the inverse direction of the link to send back one Data

packet. A number of congestion control schemes have been developed

for ICN that operate in this fashion, for example [Wang2013],

[Mahdian2016], [Song2018], [Carofiglio2012]. Other schemes, like

[Schneider2016] neither count nor police Interests, but instead

monitor queues using AQM (active queue management) to mark returning

Data packets that have experienced congestion. This later class of

schemes is similar to those used on IP in the sense that they depend

on consumers adequately reducing their rate of Interest injection to

avoid Data packet drops due to buffer overflow in forwarders. The

former class of schemes are (arguably) more robust against misbehavior by consumers.

4. What can we control to achieve QoS in ICN?

QoS is achieved through managed unfairness in the allocation of

resources in network elements, particularly in the routers doing

forwarding of ICN packets. So, a first-order question is, what

resources need to be allocated, and how to ascertain which traffic

gets what allocations. In the case of CCNx or NDN the important

network element resources are:

D. Oran Expires April 14, 2020 [Page 6]

Internet-Draft ICN QoS Architecture October 2019

+-----------------------------+-------------------------------------+

| Resource | ICN Usage |

+-----------------------------+-------------------------------------+

| Communication Link capacity | buffering for queued packets |

| Content Store capacity | to hold cached data |

| Forwarder memory | for the Pending Interest Table |

| | (PIT) |

| Compute capacity | for forwarding packets, including |

| | the cost of Forwarding Information |

| | Base (FIB) lookups. |

+-----------------------------+-------------------------------------+

Table 2: ICN-related Network Element Resources

For these resources, any QoS scheme has to specify two things:

1. How do you create \_equivalence classes\_ (a.k.a. flows) of traffic

to which different QoS treatments are applied?

2. What are the possible treatments and how are those mapped to the

resource allocation algorithms?

Two critical facts of life come into play when designing a QoS

scheme: First, the number of equivalence classes that can be

simultaneously tracked in a network element is bounded by both memory

and processing capacity to do the necessary lookups. One can allow

very fine-grained equivalence classes, but not be able to employ them

globally because of scaling limits of core routers. That means it is

wise to either restrict the range of equivalence classes, or allow

them to be \_aggregated\_, trading off the accuracy of policing the traffic.

Second, the flexibility of expressible treatments can be tightly

constrained by both protocol encoding and algorithmic limitations.

The ability to encode the treatment requests in the protocol can be

limited (as it is for IP - there are only 6 of the TOS bits available

for Diffserv treatments), but as or more important is whether there

are practical traffic policing, queuing, and pacing algorithms that

can be combined to support a rich set of QoS treatments.

The two considerations above in combination can easily be

substantially more expressive than what can be achieved in practice

with the available number of queues on real network interfaces or the

amount of per-packet computation needed to enqueue or dequeue a

packet.

D. Oran Expires April 14, 2020 [Page 7]

Internet-Draft ICN QoS Architecture October 2019

5. How does this relate to QoS in TCP/IP?

TCP/IP has fewer resource types to manage than ICN, and in some cases

the allocation methods are simpler, as shown in the following table:

+-----------------------------+-------------+-----------------------+

| Resource | IP Relevant | TCP/IP Usage |

+-----------------------------+-------------+-----------------------+

| Communication Link capacity | YES | buffering for queued |

| | | packets |

| Content Store capacity | NO | no content store in |

| | | IP |

| Forwarder memory | MAYBE | not needed for |

| | | output-buffered |

| | | designs |

| Compute capacity | YES | for forwarding |

| | | packets, but arguably |

| | | much cheaper than ICN |

+-----------------------------+-------------+-----------------------+

Table 3: IP-related Network Element Resources

For these resources, IP has specified three fundamental things, as

shown in the following table:

+----------------+--------------------------------------------------+

| What | How |

+----------------+--------------------------------------------------+

| \*Equivalence | subset+prefix match on IP 5-tuple |

| classes\* | {SA,DA,SP,DP,PT} |

| \*Diffserv | (very) small number of globally-agreed traffic |

| treatments\* | classes |

| \*Intserv | per-flow parameterized \_Controlled Load\_ and |

| treatments\* | \_Guaranteed\_ service classes |

+----------------+--------------------------------------------------+

Table 4: Fundamental protocol elements to achieve QoS for TCP/IP

Equivalence classes for IP can be pairwise, by matching against both

source and destination address+port, pure group using only

destination address+port, or source-specific multicast with source

adress+port and destination multicast address+port.

With Intserv, the signaling protocol RSVP [RFC2205] carries two data

structures, the FLOWSPEC and the TSPEC. The former fulfills the

requirement to identify the equivalence class to which the QoS being

signaled applies. The latter comprises the desired QoS treatment

along with a description of the dynamic character of the traffic

D. Oran Expires April 14, 2020 [Page 8]

Internet-Draft ICN QoS Architecture October 2019

(e.g. average bandwidth and delay, peak bandwidth, etc.). Both of

these encounter substantial scaling limits, which has meant that

Intserv has historically been limited to confined topologies, and/or

high-value usages, like traffic engineering.

With Diffserv, the protocol encoding (6 bits in the TOS field of the

IP header) artificially limits the number of classes one can specify.

These are documented in [RFC4594]. Nonetheless, when used with fine-

grained equivalence classes, one still runs into limits on the number

of queues required.

6. Why is ICN Different? Can we do Better?

While one could adopt an approach to QoS mirroring the extensive

experience with TCP/IP, this would, in the author's view, be a

mistake. The implementation and deployment of QoS in IP networks has

been spotty at best. There are of course economic and political

reasons as well as technical reasons for these mixed results, but

there are several architectural choices in ICN that make it a

potentially much better protocol base to enhance with QoS machinery.

This section discusses those differences and their consequences.

6.1. Equivalence class capabilities

First and foremost, hierarchical names are a much richer basis for

specifying equivalence classes than IP 5-tuples. The IP address (or

prefix) can only separate traffic by topology to the granularity of

hosts, and not express the actual computational instances nor sets of

data. Ports give some degree of per-instance demultiplexing, but

this tends to be both coarse and ephemeral, while confounding the

demultiplexing function with the assignment of QoS treatments to

particular subsets of the data. Some degree of finer granularity is

possible with IPv6 by exploiting the ability to use up to 64 bits of the

address for classifying traffic. In fact, the hICN project

([I-D.muscariello-intarea-hicn]), while adopting the request-response

model of CCNx, uses IPv6 addresses as the available namespace, and

IPv6 packets (plus "fake" TCP headers) as the wire format.

Nonetheless, the flexibility of tokenized, variable length,

hierarchical names allows one to directly associate classes of

traffic for QoS purposes with the structure of an application

namespace. The classification can be as coarse or fine-grained as

desired by the application. While not \_always\_ the case, there is

typically a straightforward association between how objects are

named, and how they are grouped together for common treatment.

Examples abound; a number can be conveniently found in

[I-D.moiseenko-icnrg-flowclass].

D. Oran Expires April 14, 2020 [Page 9]

Internet-Draft ICN QoS Architecture October 2019

6.2. Topology interactions with QoS

In ICN, QoS is not pre-bound to topology since names are non-

topological, unlike unicast IP addresses. This allows QoS to be

applied to multi-destination and multi-path environments in a

straightforward manner, rather than requiring either multicast with

coarse class-based scheduling or complex signaling like that in RSVP-

TE [RFC3209] that is needed to make point-to-multipoint MPLS work.

Because of IP's stateless forwarding model, complicated by the

ubiquity of asymmetric routes, any flow-based QoS requires state that

is decoupled from the actual arrival of traffic and hence must be

maintained, at least as soft-state, even during quiescent periods.

Intserv, for example, requires flow signaling with state O(#flows).

ICN, even worst case, requires state O(#active Interest/Data

exchanges), since state can be instantiated on arrival of an

Interest, and removed lazily once the data has been returned.

6.3. Specification of QoS treatments

Unlike Intserv, Difserv eschews signaling in favor of class-based

configuration of resources and queues in network elements. However,

Diffserv limits traffic treatments to a few bits taken from the ToS

field of IP. No such wire encoding limitations exist for NDN or

CCNx, as the protocol is completely TLV-based, and one (or even more

than one) new field can be easily defined to carry QoS treatment

information.

Therefore, there are greenfield possibilities for more powerful QoS

treatment options in ICN. For example, IP has no way to express a

QoS treatment like "try hard to deliver reliably, even at the expense

of delay or bandwidth". Such a QoS treatment for ICN could invoke

native ICN mechanisms, none of which are present in IP, such as:

o In-network retransmission in response to hop-by-hop errors

returned from upstream forwarders

o Trying multiple paths to multiple content sources either in

parallel or serially

o Higher precedence for short-term caching to recover from

downstream errors

Such mechanisms are typically described in NDN and CCNx as

\_forwarding strategies\_. However, little or no guidance is given for

what application actions or protocol machinery is used to decide

which forwarding strategy to use for which Interest that arrive at a

forwarder. See [BenAbraham2018] for an investigation of these

D. Oran Expires April 14, 2020 [Page 10]

Internet-Draft ICN QoS Architecture October 2019

issues. Associating forwarding strategies with the equivalence

classes and QoS treatments directly can make them more accessible and

useful to implement and deploy.

Stateless forwarding and asymmetric routing in IP limits available

state/feedback to manage link resources. In contrast, NDN or CCNx

forwarding allows all link resource allocation to occur as part of

Interest forwarding, potentially simplifying things considerably.

For example, with symmetric routing, producers have no control over

the paths their data packets traverse, and hence any QoS treatments

intended to influence routing paths from producer to consumer will

have no effect.

One complication in the handling of ICN QoS treatments is not present

in IP and hence worth mention. CCNx and NDN both perform \_Interest

aggregation\_ (See Section 2.3.2 of [RFC8569]). If an Interest

arrives matching an existing PIT entry, but with a different QoS

treatment from an Interest already forwarded, it can be tricky to

decide whether to aggregate the Interest or forward it, and

how to keep track of the differing QoS treatments for the two

Interests. Exploration of the details surrounding these situations

is beyond the scope of this document; further discussion can be found

for the general case of flow balance and congestion control in

[I-D.oran-icnrg-flowbalance], and specifically for QoS treatments in

[I-D.anilj-icnrg-dnc-qos-icn].

6.4. ICN forwarding semantics effect on QoS

IP has three forwarding semantics, with different QoS needs (Unicast,

Anycast, Multicast). ICN has the single forwarding semantic, so any

QoS machinery can be uniformly applied across any request/response

invocation, whether it employs dynamic destination routing, multi-

destination parallel requests, or even localized flooding (e.g.

directly on L2 multicast mechanisms). Additionally, the pull-based

model of ICN avoids a number of thorny multicast QoS problems that IP

has ([Wang2000], [RFC3170], [Tseng2003]).

The Multi-destination/multi-path forwarding model in ICN changes

resource allocation needs in a fairly deep way. IP treats all

endpoints as open-loop packet sources, whereas NDN and CCNx have

strong asymmetry between producers and consumers as packet sources.

6.5. QoS interactions with Caching

IP has no caching in routers, whereas ICN needs ways to allocate

cache resources. Treatments to control caching operation are

unlikely to look much like the treatments used to control link

D. Oran Expires April 14, 2020 [Page 11]

Internet-Draft ICN QoS Architecture October 2019

resources. NDN and CCNx already have useful cache control directives

associated with Data messages. The CCNx controls include:

ExpiryTime: time after which a cached Content Object is considered

expired and MUST no longer be used to respond to an Interest from

a cache.

Recommended Cache Time: time after which the publisher considers the

Content Object to be of low value to cache.

See [RFC8569] for the formal definitions s.

ICN flow classifiers, such as those in

[I-D.moiseenko-icnrg-flowclass] can be used to achieve soft or hard

partitioning of cache resources in the content store of an ICN

forwarder. For example, cached content for a given equivalence class

can be considered \_fate shared\_ in a cache whereby objects from the

same equivalence class are purged as a group rather than

individually. This can recover cache space more quickly and at lower

overhead than pure per-object replacement. In addition, since the

forwarder remembers the QoS treatment for each pending Interest in

its PIT, the above cache controls can be augmented by policy to

prefer retention of cached content for some equivalence classes as

part of the cache replacement algorithm.

7. A strawman set of principles to guide QoS architecture for ICN

Based on the observations made in the earlier sections, this summary

section captures the author's ideas for clear and actionable

architectural principles for how to incorporate QoS machinery into

ICN protocols like NDN and CCNx. Hopefully, they can guide further

work and focus effort on portions of the giant design space for QoS

that have the best tradeoffs in terms of flexibility, simplicity, and

deployability.

\*Define equivalence classes using the name hierarchy rather than

creating an independent traffic class definition\*. This directly

associates the specification of equivalence classes of traffic with

the structure of the application namespace. It can allow

hierarchical decomposition of equivalence classes in a natural way

because of the way hierarchical ICN names are constructed. Two

practical mechanisms are presented in [I-D.moiseenko-icnrg-flowclass]

with different tradeoffs between security and the ability to

aggregate flows. Either prefix-based (EC3) or explicit name

component-based (ECNT) or both could be adopted as part of the

QoS architecture for defining equivalence classes.

D. Oran Expires April 14, 2020 [Page 12]

Internet-Draft ICN QoS Architecture October 2019

\*Put consumers in control of Link and Forwarding resource

allocation\*. Do all link buffering and forwarding (both memory and

CPU) resource allocations based on Interest arrivals. This is

attractive because it provides early congestion feedback to

consumers, and allows scheduling the reverse link direction ahead of

time for carrying the matching data. It makes enforcement of QoS

treatments a single-ended rather than a double-ended problem and can

avoid wasting resources on fetching data that will wind up dropped

when it arrives at a bottleneck link.

\*Allow producers to influence the allocation of cache resources\*.

Producers want to affect caching decisions in order to:

o Shed load by having Interests served by content stores in

forwarders before reaching the producer itself.

o Survive transient outages of either the producer or links close to

the producer.

For caching to be effective, individual Data objects in an

equivalence class need to have similar treatment; otherwise well-

known cache thrashing pathologies due to self-interference emerge.

Producers have the most direct control over caching policies through

the caching directives in Data messages. It therefore makes sense to

put the producer, rather than the consumer or network operator in

charge of specifying these equivalence classes.

See [I-D.moiseenko-icnrg-flowclass] for specific mechanisms to

achieve this.

\*Allow consumers to influence the allocation of cache resources\*.

Consumers want to affect caching decisions in order to:

o Reduce latency for retrieving data

o Survive transient outages of either a producer or links close to

the consumer

Consumers can have indirect control over caching by specifying QoS

treatments in their Interests. Consider the following potential QoS

treatments by consumers that can drive caching policies:

o A QoS treatment requesting better robustness against transient

disconnection can be used by a forwarder close to the consumer (or

downstream of an unreliable link) to preferentially cache the

corresponding data.

D. Oran Expires April 14, 2020 [Page 13]

Internet-Draft ICN QoS Architecture October 2019

o Conversely a QoS treatment together with, or in addition to a

request for short latency, to indicate that new data will be

requested soon enough that caching the current data being

requested would be ineffective and hence to only pay attention to

the caching preferences of the producer.

o A QoS treatment indicating a mobile consumer likely to incur a

mobility event within an RTT (or a few RTTs). Such a treatment

would allow a mobile network operator to preferentially cache the

data at a forwarder positioned at a \_join point\_ or \_rendezvous

point\_ of their topology.

\*Give network operators the ability to match customer SLAs to cache

resource availability\*. Network operators, whether closely tied

administratively to producer or consumer, or constituting an

independent transit administration, provide the storage resources in

the ICN forwarders. Therefore, they are the ultimate arbiters of how

the cache resources are managed. In addition to any local policies

they may enforce, the cache behavior from the QoS standpoint emerges

from how the producer-specified equivalence classes map onto cache

space availability, including whether cache entries are treated

individually, or fate-shared. Forwarders also determine how the

consumer-specified QoS treatments map to the precedence used for

retaining Data objects in the cache.

Besides utilizing cache resources to meet the QoS goals of individual

producers and consumers, network operators also want to manage their

cache resources in order to:

o Ameliorate congestion hotspots by reducing load converging on

producers they host on their network.

o Improve Interest satisfaction rates by utilizing caches as short-

term retransmission buffers to recover from link errors or

outages.

o Improve both latency and reliability in environments when

consumers move in the operator's topology.

\*Re-think how to specify traffic treatments - don't just copy

Diffserv\*. Some of the Diffserv classes may form a good starting

point, as their mapping onto queuing algorithms for managing link

buffering is well understood. However, Diffserv alone does not

allow one to express latency versus reliability tradeoffs or other

useful QoS treatments. Nor does it permit "TSPEC"-style traffic

descriptions as are allowed in a signaled QoS scheme. Here are some

examples:

D. Oran Expires April 14, 2020 [Page 14]

Internet-Draft ICN QoS Architecture October 2019

o A "burst" treatment, where an initial Interest gives an aggregate

data size to request allocation of link capacity for a large burst

of Interest/Data exchanges. The Interest can be rejected at any

hop if the resources are not available. Such a treatment can also

accommodate Data implosion produced by the discovery procedures of

management protocols like [I-D.irtf-icnrg-ccninfo].

o A "reliable" treatment, which affects preference for allocation of

PIT space for the Interest and Content Store space for the Data in

order to improve the robustness of IoT data delivery in

constrained environment, as is described in

[I-D.gundogan-icnrg-iotqos].

o A "search" treatment, which, within the specified Interest

Lifetime, tries many paths either in parallel or serial to

potentially many content sources, to maximize the probability that

the requested item will be found. This is done at the expense of

the extra bandwidth of both forwarding Interests and receiving

multiple responses upstream of an aggregation point. The

treatment can encode a value expressing tradeoffs like breadth-

first versus depth-first search, and bounds on the total resource

expenditure. Such a treatment would be useful for instrumentation

protocols like [I-D.mastorakis-icnrg-icntraceroute].

As an aside, loose latency control can be achieved by bounding

Interest Lifetime as long as it is not also used as an application

mechanism to provide subscriptions or establish path traces for

producer mobility. See [Krol2018] for a discussion of the network

versus application timescale issues in ICN protocols.

\*What about the richer QoS semantics available with INTServ-like

traffic control?\*. Basic QoS treatments such as those summarized

above may not be adequate to cover the whole range of application

utility functions and deployment environments we expect for ICN.

While it is true that one does not necessarily need a separate

signaling protocol like RSVP given the state carried in the ICN data

plane by forwarders, there are some potentially important

capabilities not provided by just simple QoS treatments applied to

per- Interest/Data exchanges. INTserv's richer QoS capabilities may

be of value, especially if they can be provided in ICN at lower

complexity and protocol overhead than INTServ+RSVP.

There are three key capabilities missing from Diffserv-like QoS

treatments, no matter how sophisticated they may be in describing the

desired treatment for a given equivalence class of traffic. INTserv-

like QoS provides all of these:

D. Oran Expires April 14, 2020 [Page 15]

Internet-Draft ICN QoS Architecture October 2019

1. The ability to \*describe traffic flows\* in a mathematically

meaningful way. This is done through parameters like average

rate, peak rate, and maximum burst size. The parameters are

encapsulated in a data structure called a "TSPEC" which can be

placed in whatever protocol needs the information (in the case of

TCP/IP INTserv, this is RSVP).

2. The ability to perform \*admission control\*, where the element

requesting the QoS treatment can know \_before\_ introducing

traffic whether the network elements have agreed to provide the

requested traffic treatment. An important side-effect of

providing this assurance is that the network elements install

state that allows the forwarding and queuing machinery to police

and shape the traffic in a way that provides a sufficient degree

of \_isolation\_ from the dynamic behavior of other traffic.

Depending on the admission control mechanism, it may or may not

be possible to explicitly release that state when the application

no longer needs the QoS treatment.

3. The permissible \*degree of divergence\* in the actual traffic

handling from the requested handling. INTServ provided two

choices here, the \_controlled load\_ service and the \_guaranteed\_

service. The former allows stochastic deviation equivalent to

what one would experience on an unloaded path of a packet

network. The latter conforms to the TSPEC deterministically, at

the obvious expense of demanding extremely conservative resource

allocation.

Given the limited applicability of these capabilities in today's

Internet, the author does not take any position as to whether any of

these INTserv-like capabilities are needed for ICN to be successful.

However, a few things seem important to consider. The following

paragraphs speculate about the consequences to the CCNx or NDN

protocol architectures of incorporating these features.

Superficially, it would be quite straightforward to accommodate

INTserv-equivalent traffic descriptions in CCNx or NDN. One could

define a new TLV for the Interest message to carry a TSPEC. A

forwarder encountering this, together with a QoS treatment request

(e.g. as proposed in Section 6.3) could associate the traffic

specification with the corresponding equivalence class derived from

the name in the Interest. This would allow the forwarder to create

a state that not only would apply to the returning Data for that

Interest when being queued on the downstream interface, but be

maintained as a soft state across multiple Interest/Data exchanges to

drive policing and shaping algorithms at per-flow granularity. The

cost in Interest message overhead would be modest, however, the

complications associated with managing different traffic

D. Oran Expires April 14, 2020 [Page 16]

Internet-Draft ICN QoS Architecture October 2019

specifications in different Interests for the same equivalence class

might be substantial. Of course, all the scalability considerations

with maintaining per-flow state also come into play.

Similarly, it would be equally straightforward to have a way to

express the degree of divergence capability that INTserv provides

through its controlled load and guaranteed service definitions. This

could either be packaged with the traffic specification or

encoded separately.

In contrast to the above, performing admission control for ICN flows

is likely to be just as heavy-weight as it turned out to be with IP

using RSVP. The dynamic multi-path, multi-destination forwarding

model of ICN makes performing admission control particularly tricky.

Just to illustrate:

o Forwarding paths are not confined to single paths (or a few ECMP

equivalent paths) as they are with IP, making it difficult to know

where to install state in advance of the arrival of an Interest to

forward.

o As with point-to-multipoint complexities when using RSVP for MPLS-

TE, the state has to be installed to multiple producers over multiple

paths before an admission control algorithm can commit the

resources and say "yes" to a consumer needing admission control

capabilities

o Knowing when to remove admission control state is difficult in the

absence of a heavy-weight resource reservation protocol. Soft

state timeout may or may not be an adequate answer.

Despite the challenges above, it may be possible to craft an

admission control scheme for ICN that achieves the desired QoS goals

of applications without the invention and deployment of a complex

separate admission control signaling protocol. There have been

designs in earlier network architectures that were capable of

performing admission control piggybacked on packet transmission.

(The earliest example the author is aware of is [Autonet]).

Such a scheme might have the following general shape \*(warning:

serious hand waving follows!)\*:

o In addition to a QoS treatment and a traffic specification, an

Interest requesting admission for the corresponding equivalence

class would so indicate via a new TLV. It would also need to: (a)

indicate an expiration time after which any reserved resources can

be released, and (b) indicate that caches be bypassed, so that the

D. Oran Expires April 14, 2020 [Page 17]

Internet-Draft ICN QoS Architecture October 2019

admission control request arrives at a bone-fide producer (or

Repo).

o Each forwarder processing the Interest would check for resource

availability and if not available, or the requested service not

feasible, reject the Interest with an admission control

failure. If resources are available, the forwarder would record

the traffic specification as described above and forward the

Interest.

o If the Interest successfully arrives at a Producer, the producer

returns the requested Data.

o Each on-path forwarder, on receiving the matching Data message, if

the resources are still available, does the actual allocation, and

marks the admission control TLV as "provisionally approved".

Conversely, if the resource reservation fails, the admission

control is marked "failed", although the Data is still passed

downstream.

o Upon the Data message arriving, the consumer knows if admission

succeeded or not, and subsequent Interests can rely on the QoS

state being in place until either some failure occurs, or a

topology or other forwarding change alters the forwarding path.

To deal with this, additional machinery is needed to ensure

subsequent Interests for an admitted flow either follow that path

or an error is reported. One possibility (also useful in many

other contexts), is to employ a \_Path Steering\_ mechanism, such as

the one described in [Moiseenko2017].

8. IANA Considerations

This document does not require any IANA actions.

9. Security Considerations

There are a few ways in which QoS for ICN interacts with security and

privacy issues. Since QoS addresses relationships among traffic

rather than the inherent characteristics of the traffic, it neither

enhances nor degrades the security and privacy properties of the data

being carried, as long as the machinery does not alter or otherwise

compromise the basic security properties of the associated protocols.

The QoS approaches advocated here for ICN can serve to amplify

existing threats to network traffic, however:

o An attacker able to manipulate the QoS treatments of traffic can

mount a more focused (and potentially more effective) denial of

service attack by suppressing performance on traffic the attacker

D. Oran Expires April 14, 2020 [Page 18]

Internet-Draft ICN QoS Architecture October 2019

is targeting. Since the architecture here assumes QoS treatments

are manipulable hop-by-hop, any on-path adversary can wreak havoc.

Note however, that in basic ICN, an on-path attacker can do this

and more by dropping, delaying, or mis-routing traffic independent

of any particular QoS machinery in use.

o By explicitly revealing equivalence classes of traffic via either

names or other fields in packets, an attacker has yet one more

handle to use to discover linkability of multiple requests.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119,

DOI 10.17487/RFC2119, March 1997,

<https://www.rfc-editor.org/info/rfc2119>.

[RFC8569] Mosko, M., Solis, I., and C. Wood, "Content-Centric

Networking (CCNx) Semantics", RFC 8569,

DOI 10.17487/RFC8569, July 2019,

<https://www.rfc-editor.org/info/rfc8569>.

[RFC8609] Mosko, M., Solis, I., and C. Wood, "Content-Centric

Networking (CCNx) Messages in TLV Format", RFC 8609,

DOI 10.17487/RFC8609, July 2019,

<https://www.rfc-editor.org/info/rfc8609>.

10.2. Informative References

[AS] "Autonomous System (Internet)", no date,

<https://en.wikipedia.org/wiki/

Autonomous\_system\_(Internet)>.

[Autonet] Schroeder, M., Birrell, A., Burrows, M., Murray, H.,

Needham, R., Rodeheffer, T., Satterthwaite, E., and C.

Thacker, "Autonet: a High-speed, Self-configuring Local

Area Network Using Point-to-point Links", SRC Research

Reports 59, April 1990,

<https://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-

59.pdf>.

D. Oran Expires April 14, 2020 [Page 19]

Internet-Draft ICN QoS Architecture October 2019

[BenAbraham2018]

Ben Abraham, H., Parwatikar, J., DeHart, J., Dresher, A.,

and P. Crowley, "Decoupling Information and Connectivity

via Information-Centric Transport, in 5th ACM Conference

on Information-Centric Networking (ICN '18), September

21-23, 2018, Boston, MA, USA",

DOI 10.1145/3267955.3267963, September 2018,

<https://conferences.sigcomm.org/acm-icn/2018/proceedings/

icn18-final31.pdf>.

[Carofiglio2012]

Carofiglio, G., Gallo, M., and L. Muscariello, "Joint hop-

by-hop and receiver-driven interest control protocol for

content-centric networks, in ICN Workshop at SIGcomm

2012", DOI 10.1145/2377677.2377772, 2102,

<http://conferences.sigcomm.org/sigcomm/2012/paper/icn/

p37.pdf>.

[I-D.anilj-icnrg-dnc-qos-icn]

Jangam, A., suthar, P., and M. Stolic, "QoS Treatments in

ICN using Disaggregated Name Components", draft-anilj-

icnrg-dnc-qos-icn-01 (work in progress), September 2019.

[I-D.gundogan-icnrg-iotqos]

Gundogan, C., Schmidt, T., Waehlisch, M., Frey, M., Shzu-

Juraschek, F., and J. Pfender, "Quality of Service for ICN

in the IoT", draft-gundogan-icnrg-iotqos-01 (work in

progress), July 2019.

[I-D.ietf-quic-transport]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed

and Secure Transport", draft-ietf-quic-transport-23 (work

in progress), September 2019.

[I-D.irtf-icnrg-ccninfo]

Asaeda, H., Ooka, A., and X. Shao, "CCNinfo: Discovering

Content and Network Information in Content-Centric

Networks", draft-irtf-icnrg-ccninfo-02 (work in progress),

July 2019.

[I-D.mastorakis-icnrg-icntraceroute]

Mastorakis, S., Gibson, J., Moiseenko, I., Droms, R., and

D. Oran, "ICN Traceroute Protocol Specification", draft-

mastorakis-icnrg-icntraceroute-05 (work in progress),

August 2019.

D. Oran Expires April 14, 2020 [Page 20]

Internet-Draft ICN QoS Architecture October 2019

[I-D.moiseenko-icnrg-flowclass]

Moiseenko, I. and D. Oran, "Flow Classification in

Information Centric Networking", draft-moiseenko-icnrg-

flowclass-04 (work in progress), July 2019.

[I-D.muscariello-intarea-hicn]

Muscariello, L., Carofiglio, G., Auge, J., and M.

Papalini, "Hybrid Information-Centric Networking", draft-

muscariello-intarea-hicn-02 (work in progress), June 2019.

[I-D.oran-icnrg-flowbalance]

Oran, D., "Maintaining CCNx or NDN flow balance with

highly variable data object sizes", draft-oran-icnrg-

flowbalance-01 (work in progress), August 2019.

[Krol2018]

Krol, M., Habak, K., Oran, D., Kutscher, D., and I.

Psaras, "RICE: Remote Method Invocation in ICN, in

Proceedings of the 5th ACM Conference on Information-

Centric Networking - ICN '18",

DOI 10.1145/3267955.3267956, September 2018,

<https://conferences.sigcomm.org/acm-icn/2018/proceedings/

icn18-final9.pdf>.

[Mahdian2016]

Mahdian, M., Arianfar, S., Gibson, J., and D. Oran,

"MIRCC: Multipath-aware ICN Rate-based Congestion Control,

in Proceedings of the 3rd ACM Conference on Information-

Centric Networking", DOI 10.1145/2984356.2984365, 2016,

<http://conferences2.sigcomm.org/acm-icn/2016/proceedings/

p1-mahdian.pdf>.

[minmaxfairness]

"Max-min Fairness", no date,

<https://en.wikipedia.org/wiki/Max-min\_fairness>.

[Moiseenko2017]

Moiseenko, I. and D. Oran, "Path Switching in Content

Centric and Named Data Networks, in 4th ACM Conference on

Information-Centric Networking (ICN 2017)",

DOI 10.1145/3125719.3125721, September 2017,

<https://conferences.sigcomm.org/acm-icn/2017/proceedings/

icn17-2.pdf>.

[NDN] "Named Data Networking", various,

<https://named-data.net/project/execsummary/>.

D. Oran Expires April 14, 2020 [Page 21]

Internet-Draft ICN QoS Architecture October 2019

[Oran2018QoSslides]

Oran, D., "Thoughts on Quality of Service for NDN/CCN-

style ICN protocol architectures, presented at ICNRG

Interim Meeting, Cambridge MA", September 2018,

<https://datatracker.ietf.org/meeting/interim-2018-icnrg-

03/materials/slides-interim-2018-icnrg-03-sessa-thoughts-

on-qos-for-ndnccn-style-icn-protocol-architectures>.

[proportionalfairness]

"Proportionally Fair", no date,

<https://en.wikipedia.org/wiki/Proportionally\_fair>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7,

RFC 793, DOI 10.17487/RFC0793, September 1981,

<https://www.rfc-editor.org/info/rfc793>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S.

Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1

Functional Specification", RFC 2205, DOI 10.17487/RFC2205,

September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black,

"Definition of the Differentiated Services Field (DS

Field) in the IPv4 and IPv6 Headers", RFC 2474,

DOI 10.17487/RFC2474, December 1998,

<https://www.rfc-editor.org/info/rfc2474>.

[RFC2998] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L.,

Speer, M., Braden, R., Davie, B., Wroclawski, J., and E.

Felstaine, "A Framework for Integrated Services Operation

over Diffserv Networks", RFC 2998, DOI 10.17487/RFC2998,

November 2000, <https://www.rfc-editor.org/info/rfc2998>.

[RFC3170] Quinn, B. and K. Almeroth, "IP Multicast Applications:

Challenges and Solutions", RFC 3170, DOI 10.17487/RFC3170,

September 2001, <https://www.rfc-editor.org/info/rfc3170>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V.,

and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP

Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001,

<https://www.rfc-editor.org/info/rfc3209>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram

Congestion Control Protocol (DCCP)", RFC 4340,

DOI 10.17487/RFC4340, March 2006,

<https://www.rfc-editor.org/info/rfc4340>.

D. Oran Expires April 14, 2020 [Page 22]

Internet-Draft ICN QoS Architecture October 2019

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration

Guidelines for DiffServ Service Classes", RFC 4594,

DOI 10.17487/RFC4594, August 2006,

<https://www.rfc-editor.org/info/rfc4594>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol",

RFC 4960, DOI 10.17487/RFC4960, September 2007,

<https://www.rfc-editor.org/info/rfc4960>.

[Schneider2016]

Schneider, K., Yi, C., Zhang, B., and L. Zhang, "A

Practical Congestion Control Scheme for Named Data

Networking, in Proceedings of the 2016 conference on 3rd

ACM Conference on Information-Centric Networking - ACM-ICN

'16", DOI 10.1145/2984356.2984369, 2016,

<http://conferences2.sigcomm.org/acm-icn/2016/proceedings/

p21-schneider.pdf>.

[Shenker2006]

Shenker, S., "Fundamental Design Issues for the Future

Internet, in IEEE Journal on Selected Areas in

Communications", DOI 10.1109/49.414637, 2006,

<https://dl.acm.org/citation.cfm?id=2316898>.

[Song2018]

Song, J., Lee, M., and T. Kwon, "SMIC: Subflow-level

Multi-path Interest Control for Information Centric

Networking, in 5th ACM Conference on Information-Centric

Networking", DOI 10.1145/3267955.3267971, 2018,

<https://conferences.sigcomm.org/acm-icn/2018/proceedings/

icn18-final62.pdf>.

[Tseng2003]

Tseng, CH., "The performance of QoS-aware IP multicast

routing protocols, in Networks, Vol:42, No:2",

DOI 10.1002/net.10084, September 2003,

<https://onlinelibrary.wiley.com/doi/abs/10.1002/

net.10084>.

[Wang2000]

Wang, B. and J. Hou, "Multicast routing and its QoS

extension: problems, algorithms, and protocols, in IEEE

Network, Vol:14, No:1", DOI 10.1109/65.819168, Jan/Feb

2000, <https://ieeexplore.ieee.org/

document/819168?arnumber=819168>.

D. Oran Expires April 14, 2020 [Page 23]

Internet-Draft ICN QoS Architecture October 2019

[Wang2013]

Wang, Y., Rozhnova, N., Narayanan, A., Oran, D., and I.

Rhee, "An Improved Hop-by-hop Interest Shaper for

Congestion Control in Named Data Networking, in ACM

SIGCOMM Workshop on Information-Centric Networking",

DOI 10.1145/2534169.2491233, 2013,

<http://conferences.sigcomm.org/sigcomm/2013/papers/icn/

p55.pdf>.

Author's Address

Dave Oran

Network Systems Research and Design

4 Shady Hill Square

Cambridge, MA 02138

USA

Email: daveoran@orandom.net

D. Oran Expires April 14, 2020 [Page 24]