

ICNRG  
Internet-Draft  
Intended status: Informational  
Expires: September 10, 2020

Anil Jangam, Ed.  
Prakash Suthar  
Milan Stolic  
Cisco Systems  
March 09, 2020

QoS Treatments in ICN using Disaggregated Name Components  
draft-anilj-icnrg-dnc-qos-icn-02

Abstract

Mechanisms for specifying and implementing end-to-end Quality of service (QoS) treatments in Information-Centric Networks (ICN) has not been explored so far. There has been some work towards implementing QoS in ICN; however, the discussions are mainly centered around strategies used in efficient forwarding of Interest packets. Moreover, as ICN has been tested mainly as an IP overlay, its QoS is still governed by the IP QoS framework and there is a need for implementing QoS in ICN natively. This document describes mechanisms to classify and provide associated "name-based" extensions to identify QoS within the framework of ICN's core design principles. The name-based design provides a mechanism to carry QoS information and implement the treatments as ICN packets travel across different routers in the ICN network. Detailed discussion is provided for QoS specific procedures in each of the ICN network elements i.e. consumer, producer, and forwarder.

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 September 10, 2020.

## Copyright Notice

Copyright (c) 2020 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 . . . . .	3
2. Conventions and Terminology . . . . .	3
3. Prior Work and Motivation . . . . .	3
4. A Perspective on QoS in ICN . . . . .	4
4.1. Network Resources and QoS Treatments in ICN . . . . .	5
4.2. Mutation of QoS Marker . . . . .	6
4.3. Nameless Object . . . . .	7
4.4. QoS Marker Inside Content Name . . . . .	7
4.4.1. Name-based QoS Encoding Scheme . . . . .	8
4.5. QoS Marker Inside Hop-by-Hop Header . . . . .	8
4.5.1. Modified Interest Message . . . . .	9
5. Network Procedures . . . . .	10
5.1. Consumer Procedure . . . . .	10
5.2. Forwarder Procedure . . . . .	11
5.2.1. QoS and Multipath Forwarding . . . . .	12
5.3. Producer Procedure . . . . .	12
6. QoS-Aware Forwarder Design . . . . .	12
6.1. Maintaining QoS State in PIT . . . . .	12
6.2. QoS-Aware Interest Aggregation in PIT . . . . .	14
6.3. Handling of QoS and PIT Aggregation . . . . .	14
6.4. Data Delivery at PIT . . . . .	15
7. Security Considerations . . . . .	16
8. Summary . . . . .	16
9. Acknowledgements . . . . .	17
10. IANA Considerations . . . . .	17
11. References . . . . .	17
11.1. Normative References . . . . .	17
11.2. Informative References . . . . .	18
Authors' Addresses . . . . .	19

## 1. Introduction

Information Centric Networking (ICN) is rapidly emerging as an alternative networking mechanism for the TCP/IP based host-centric networking paradigm. Use cases of video conferencing [MPVCICN] and real-time streaming [NDNRTC] signify the value of ICN architecture and has been studied in detail. Also, a number of studies on routing of Interest and flow classification [ICNFLOW] have been published; however, relatively less work has been done on end-to-end quality of service (QoS) architecture for ICN. QoS is important to deliver preferential service to a variety of traffic flows resulting in better user experience. Evaluation and study of prior work done in this area is provided in Section 3. The current QoS implementation is based on either Layer-3 TOS or DiffServ, which is applicable only for ICN as an overlay. The QoS mechanisms described in this draft are applicable to the native ICN architecture. A detailed discussion related to the design of name-based QoS encoding is provided in Section 4.4.

## 2. Conventions and Terminology

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 [RFC2119].

This document uses the terminology of [CCNXSEMANTICS] and [CCNXMESSAGES] for CCNx entities.

## 3. Prior Work and Motivation

Among the work related to the quality of service (QoS) requirements in ICN network, larger emphasis is given to an optimized and efficient routing of Interest packets towards its content.

M.F. Al-Naday et.al. in [NADAY] argues that information awareness of ICN network would help build scalable QoS model. In the context of CCN/NDN network design, the authors point to the possibility of using the QoS aware name prefixes, potentially limiting the third parties (e.g. network operators) from defining alternative QoS enforcement mechanisms. Moreover, the QoS solution is developed around the PURSUIT architecture, which may not be applicable to CCN/NDN.

Weibo Chu et.al. in [WEIBO] present a QoS model for ICN network based on the popularity ranking of the content and its placement/location in the network. They present a classification of the content into three categories - locally cached, remotely cached, and uncached contents, hence the network delay is modeled as a function of the distance of the content from the requester. Essentially, the QoS

problem is seen in the perspective of faster routing of Interest request towards its content.

In [XINGWEI] authors present a QoS mechanism, which is applicable to the routing of Interest requests in ICN network. The basis of the proposal is to decide the suitability of the forwarding link to make the process more energy-efficient. They use the same Data forwarding algorithm specified in the original NDN design [JACOBSON].

In [CHRISTOS] Christos et.al. argue about a need for a differentiated routing and forwarding mechanisms based on not only the name of the content but also specifying the nature of the traffic. They further emphasize that this differentiation is better handled at the network level rather than leaving it for the upper layer.

In all the above use cases, the QoS related discussions are mainly focused on the forwarding of the Interest requests. The Data packets are forwarded in the downstream direction according to the pending Interest table (PIT). There is little or no discussions provided about how preferential treatment can be implemented and enforced in the Data packet path.

In [YAOGONG], authors present a scheme for hop-by-hop Interest shaper algorithm and demonstrates how by shaping of Interest results the returning Data packets are controlled and shaped. In this algorithm, when coupled with the consumer-driven QoS treatment of Interest, automatically achieves preferential treatment of returning Data.

#### 4. A Perspective on QoS in ICN

In general, QoS marking is used to fine-tune (set and change) certain attributes for the traffic belonging to a specific class. The marking helps segregate the traffic that requires special treatment, thus helps achieve optimal network performance. The in-network treatment is determined based on how these attributes are set. Some of the possible network treatments are:

- o Set the precedence of the traffic entering a network e.g. selecting specific queue for the real-time (voice and/or video) traffic.
- o Identify traffic for any class-based QoS feature implementation.
- o Marking of the QoS for packets traversing the network layer boundary (e.g. from L3 to L2 and vice versa) as well as the Autonomous System (AS) boundaries of different service providers.

From the ICN perspective, while the producer decides the classification of the data flow packets, it is consumer's prerogative what QoS treatment is actually provided to them by the network. Consumer application itself or the network on behalf of the consumer can perform the QoS marking in the Interest messages. The following factors govern the type of QoS markers we may require.

- o Consumer's subscription: The type of service user subscribes with the service provider e.g. subscription with high-speed data plan vs. low-speed data plan.
- o Service type: The type of service or the application consumer is running. We may refer to service classes as described in [RFC4594] (see section 2.0).

#### 4.1. Network Resources and QoS Treatments in ICN

An effective QoS management is achieved through managed unfairness in the allocation of resources including the resources on individual network elements (e.g. router) and the network links connecting them. In [QOSARCH], author lists the resources on a ICN network element.

Resource Type	Use in ICN
Link Capacity	Packet priority queues
Content Store Capacity	Cache the content data chunks
Forwarder Memory	Pending Interest Table (PIT) storage
Compute Capacity	CPU cycles for FIB, PIT, and CS lookups

Figure 1: ICN Network Element Resources

Two tradeoffs are discussed, which are important in the modeling of QoS treatments. The first points to the ability to track of the number of traffic classes given the memory (to implement packet queues) and processing capacity available for various lookups. Second points to a trade-off between the flexibility of expressing the QoS treatment to the ability of protocol encoding and limitations of the implementation of the algorithms.

In another work [IOTQOS], authors model the QoS treatments as a tradeoff between two service attributes - reliability and latency. While the proposed treatment model is useful for QoS in ICN network in general, the design mainly focuses on meeting the requirements of a constrained NDN (Named-data network) IoT (Internet of Things) network.

Following table list the potential QoS treatments depending on the type of network element resource they influence. It also lists the native ICN features used in the realization of the treatment. Note that the table only provides guidance and a particular implementation of QoS treatment may utilize one or more native ICN construct.

QoS Treatment Type	Type of Resource and Influence
Reliable delivery	++ CPU - utilization to handle errors ++ Queues - for multi-path forwarding ++ Cache - utilization for short term
Low Latency delivery	++ CPU - utilization to handle errors ++ Queues - for multi-path forwarding ++ Cache - replace cache entries ++ PIT - replace low priority PIT entries in saturated PIT
Mobility event	++ Cache - update cache at next forwarder
Bursty data	++ Queues - allocation of link capacity
Search data	++ Queues - for multi-path forwarding ++ CPU - utilization to handle errors

Figure 2: QoS Treatment and its Influence on Network Element Resource

It is important to note that the description of QoS treatment and their influence can be quite expressive as compared to flat DSCP codes defined in IP QoS design. In addition, it would be beneficial to specify the attributes of influence the treatment is going to have on the network resource. For instance, when specifying 'search' QoS treatment, number of forwarding paths to be attempted in parallel can be specified.

#### 4.2. Mutation of QoS Marker

Changing the QoS marking (a.k.a. QoS remaking) is a feature often used by the network routers. While QoS remarking is a useful feature, it can potentially cause an inconsistent end-to-end QoS treatment handling. From per-hop-behavior (PHB) perspective when QoS is remarked, the initial QoS marker added to the packet is lost and upstream router has no clue of what treatment the consumer intended to receive from the network.

While IP network allows for QoS marking and remarking, it suffers from this inconsistent end-to-end QoS treatment as it (DiffServ) allows only one QoS marker field. ICN QoS design can improve over this QoS inconsistency in following ways.

One of the mutation schemes provide the space for two QoS markers - the first preserves the original QoS marker added by consumer and second provides a running marker to be mutated by set by the intermediate forwarder in the network. This provides an opportunity to the network router node to meet the QoS as per the consumer's expectation rather than the treatment set by the predecesing router based on its resource conditions. In an alternate mechanism, a stack of QoS markers can be used where remarked treatment is pushed/popped by the node that performs the QoS-based forwarding.

In both the two-marker based design, the original QoS marker needs to be encoded such that it is immutable and is always present in the packet, hence it is proposed to encode it into a mandatory hop-by-hop header. Encoding original QoS marker into an optional hop-by-hop header may not be a good option.

#### 4.3. Nameless Object

The optional content name field in the content object makes it a nameless object. As an example, the nameless content objects are used inside a Manifest. So, one could put a QoS marking in an Interest name (to be used inside manifest), but it would not be used in the content object. It is for further study to find an encoding scheme to put the QoS marker in a nameless content object.

In rest of the document, we discuss the design of name-based encoding of QoS marker.

#### 4.4. QoS Marker Inside Content Name

In this scheme, the consumer encode the QoS markers by appending them as a non-routable suffix to the content name. The idea and distinction of routable vs non-routable component are that in general QoS marking is the consumer-initiated activity and content publishing is the producer's task. The routing protocol only advertises the name or the prefixes (without any QoS marker suffix in it as producer never publishes the QoS markers) to eventually update the FIB entries.

It is important to note the distinction between the content name and the QoS marker as the content and content name are published by the content producer whereas QoS marker suffix is appended to the content name by the consumer before requesting the content. Figure 3 shows a

conceptual design of the QoS marker encoding into content name. Note that discovery of content name by the consumer is out of the scope of this draft.

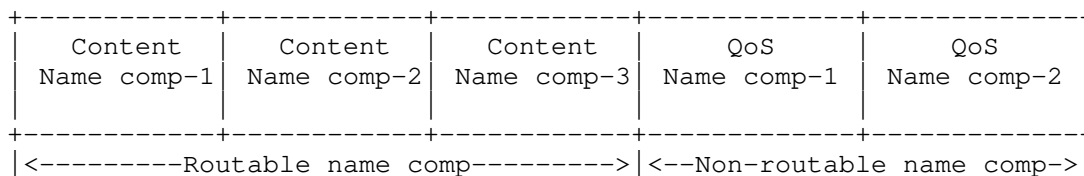


Figure 3: Disaggregate Content and QoS Name Components

As QoS marker is appended as non-routable suffix to the content name, the content name matching algorithm at the PIT, CS are extended to ignore QoS markers. The suffix-based design of QoS markers does not affect FIB's prefix-based matching, as the FIB table contains the only name prefix advertised by the routing protocol. The QoS marker, however, will be used to implement the QoS-aware forwarding strategy for both Interest and Data packets. All name components are potentially routable, in the sense that if they (or their prefix) are in a FIB they will be matched.

In this approach, however the name-based encoding of QoS marker (in both Interest and Data packet) makes it immutable as it is inside the authentication signature and routers along a path cannot change it.

#### 4.4.1. Name-based QoS Encoding Scheme

Figure 3 shows a reference model for name component-based QoS marker scheme. The number of QoS name components required depends on the type of QoS encoding uses as well as the total number of markers required. QoS marker design can either be hierarchical or based on a flat naming scheme. The exact requirements and design specification of the QoS marker is the subject of further study. Following are the potential mechanisms that can be used for encoding of QoS marking into the content name:

- o Using the 'application payload name segments' approach defined in CCNX [CCNXMESSAGES] (see section-3.6.1.1).

#### 4.5. QoS Marker Inside Hop-by-Hop Header

In this design, the QoS marker is encoded in a mandatory hop-by-hop header. The mandatory header ensures that QoS marker is available to each forwarding node in the network Interest packet path and allows it to save the QoS state in PIT and remark the QoS marker when



required. We propose to add a new QoS marker TLV in the CCNx Interest message as shown in Figure 6.

While we have proposed two QoS markers (see Section 4.2), we are showing encoding for single QoS marker only. We will evolve the two-marker scheme and provide an update based on the community feedback.

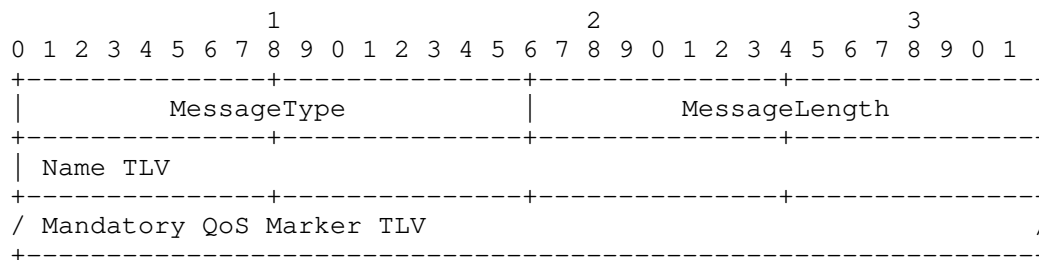


Figure 4: QoS Marker TLV

Abbrev	Name	Description
T_QOS_MARKER	QoSMarker	representation of the QoS Marker TLV

Table 1: QoS Marker TLV

The bit-wise structure of the QoS Marker TLV is shown in Figure 5. We propose to use this TLV to encode the QoS treatment identifiers listed in Section 4.1.

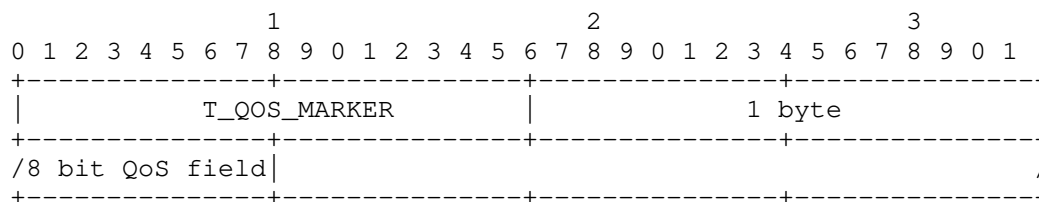


Figure 5: Binary Encoding of QoS Marker TLV

#### 4.5.1. Modified Interest Message

Figure 6 shows the Interest message structure with addition of the QoS Marker TLV.

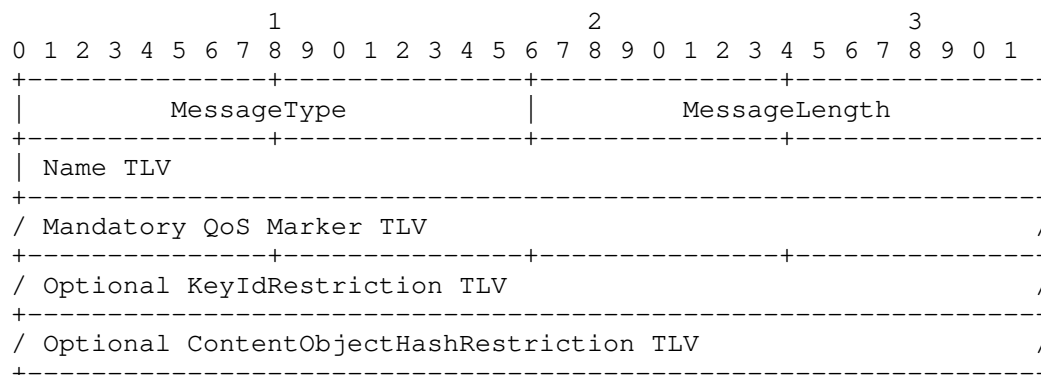


Figure 6: Modified Interest Message with QoS Marker TLV

## 5. Network Procedures

Along with the traffic treatment, network policy configuration decides how the Interest and Data traffic is carried optimally. In this section, we discuss how various ICN network nodes behave to support the QoS in the handling of Interest and Data traffic. Important changes in the behavior of each of the ICN network elements are discussed.

### 5.1. Consumer Procedure

Consumer sends out the Interest packet into the network adding the QoS marker as per its service subscription and/or quality requirements. Consumer performs the QoS marking and adds it as non-routable name component, as shown in Figure 3.

Consumer, the initiator of the Interest is the most appropriate network entity to perform the QoS marking for the following reasons:

- o ICN is a pull-based, consumer driven design and consumer directly influences the resource allocation in the network in terms of timing and rate of Interest traffic.
- o Consumer is aware of the context of the application initiating the Interest. For instance, an application starting a simple video download compared to initiating a video conferencing.
- o Consumer at least partially in control of the traffic paths in both directions [MIRCC].

As an alternative to consumer adding the QoS marker in the Interest, the network (i.e. forwarder) can be allowed to amend the content name with the QoS marker. It should be possible since the QoS marker is encoded as a non-routable component of the content name. The network adds the QoS marker based on the information such as user's service or subscription authorization.

## 5.2. Forwarder Procedure

In addition to preserving the Interest state (i.e. the mapping between content name and the interface) in the PIT, forwarder also preserves and maps the QoS marker information to the interface it receives the Interest on. Unlike PIT, there is no change in the structure of FIB table; however, forwarder forwards to the upstream ICN router both content name and QoS marker, as they are received from its predecessor.

With enhanced QoS-aware content name design, forwarder performs the content store (CS) lookup by ignoring the QoS markers in the name. The Interest aggregation at the PIT uses both content name and QoS marker during the PIT lookup, which makes it a QoS-aware Interest aggregation. Section 6.1 provide more details about the QoS state in the PIT and related procedures.

While there are no changes in the FIB table, the conventional name prefix based multipath forwarding path selection can use the QoS marker to make the QoS-aware forwarding decision. In other words, the QoS markers can be used to implement the forwarding strategies. For example, QoS marking can be used to select a low latency interface over a high latency interface or it can be used to select a high bandwidth path over a low bandwidth path or vice versa. However, note that how forwarder maintains or knows about current operation state of forwarding interface is beyond the scope of this design.

The process of mapping the QoS marker and the forwarding queue is not very different than other packet-switched forwarding mechanisms. The significance of QoS treatment design is that it allows the implementation of a queuing algorithm that can accomplish the intended QoS treatment.

In the context of QoS remarking by the network, it may also become important to let the consumer know, what network is doing with their QoS marking. From the network behavior perspective, the following are the possibilities:

- o QoS marking is preserved and obeyed by the router in the current hop

- o QoS marking is preserved but not obeyed
- o QoS marking is remarked and obeyed

#### 5.2.1. QoS and Multipath Forwarding

QoS marking in the Interest packet does not change the multipath forwarding capability of ICN forwarders. In Section 6.2, more details are provided about specific QoS behavior related to multipath forwarding.

#### 5.3. Producer Procedure

The producer is aware of the disaggregation between routable name and the non-routable QoS marker. It looks up the content in the content store (CS) only using a routable name component. An intermediate router acts in a similar manner.

Depending on the what kind of QoS marking is done in the Interest packet, the producer can have the following response behaviors:

- o The QoS aware response to provide information to the consumer about how much distance (e.g. number of hops) Interest has travelled into the network before it is satisfied.
- o QoS-aware response does not change the original requested content.

#### 6. QoS-Aware Forwarder Design

Towards supporting end-to-end QoS and handling of Interest and Data traffic in the network, there are some important design changes in the way PIT maintains the pending Interests and the way forwarding decisions are made. This section discusses in detail each of the changes.

##### 6.1. Maintaining QoS State in PIT

To support the QoS treatment processing we leverage the Interest state mechanism provided by the PIT. When an Interest arrives on an interface and is aggregated in the PIT, its QoS attribute is preserved and mapped to the interface. The specifics of the implementation are beyond the scope of this draft but a generic, conceptual model is provided here. As shown in Figure 7, the interface data structure is enhanced to maintain the QoS marker received in the Interest.

Content Name	Interface Id	QoS Marker
/yt/vid1/ch1	-----> face1	
		+-----> /qosmrk1
/yt/vid2/ch1	-----> face2	
		+-----> /qosmrk1

Figure 7: Enhanced PIT Design with QoS Marker

A special QoS handling is required in forwarder for couple of scenarios when more than one Interests are received with same content name but with different QoS markers. The new aspects are discussed in Section 6.2.

- a. Interests with same content name with different QoS markers are received on the same interface. In this representation, Interest #1 is having a lower priority than Interest #2, which is a higher QoS priority Interest.

Int#	Content name	Face Id	QoS Marker
Int1	/yt/vid1/ch1	face1	qosmrk1
Int1	/yt/vid1/ch1	face1	qosmrk2

Figure 8: Duplicate Interest on Same Face with Different QoS Marker

- b. Interests with same content name with different QoS markers are received on different interfaces. In this representation, Interest #1 is having a lower priority than Interest #2, which is a higher QoS priority Interest.

Int#	Content name	Face Id	QoS Marker
Int1	/yt/vid1/ch1	face1	qosmrk1
Int2	/yt/vid1/ch1	face2	qosmrk2

Figure 9: Duplicate Interest on two Faces with Different QoS Marker

## 6.2. QoS-Aware Interest Aggregation in PIT

In scenarios shown in Figure 8 and Figure 9, since Interests are received with same content name, the PIT aggregation decision has to be done based on the QoS marker. In both cases, if forwarder decides to forward both the Interests to the upstream router, it is going to violate the conventional PIT aggregation behavior.

In order to support QoS-aware forwarding, the conventional PIT aggregation needs to be loosened up proportional to the number of QoS markers without which the forwarder would not get an opportunity to enforce all the QoS treatments. As a result, the theoretical upper bound on the number of Interests with the same content name will be bound to the number of QoS markers. However, in a practical case, it can safely be assumed that not all QoS markers are utilized all the time using the same content name. Section 6.3 discusses an optimization in QoS-aware Interest aggregation handling.

The impact on the PIT aggregation can be mitigated by the following methods:

- o By keeping the number of QoS markers limited
- o By having a hierarchy or an order among the QoS markers.

## 6.3. Handling of QoS and PIT Aggregation

The forwarder can avoid forwarding the duplicate Interest if it receives it with a lower QoS marking than the one already pending in the PIT. This achieves the Interest aggregation based on the higher QoS marker for given content name. Conversely if the duplicate Interest is received with a higher QoS marking, then forwarder forwards the Interest and updates the related interface entry with the higher QoS marking. Also, note that forwarder updates the PIT entry irrespective of the interface the higher QoS marked Interest is received on.

The resulting state of the PIT after handling the Interest scenarios depicted in Figure 8 and Figure 9 are shown in Figure 10 and Figure 11 respectively.

Content Name	Interface Id	QoS Marker
/yt/vid1/ch1	face1	
	+	
		/qosmrk2
		/qosmrk1

Figure 10: PIT Status after Handling Duplicate Interest with different QoS Received on Same Interface

Content Name	Interface Id	QoS Marker
/yt/vid1/ch1	face1	
	+	
		/qosmrk1
	face2	
	+	
		/qosmrk2

Figure 11: PIT Status after Handling Duplicate Interest with different QoS Received on Different Interfaces

In an another case where the duplicate Interest is received but with lower priority than the pending one, the second interest with lower QoS marker will not be forwarded.

It is important to note that forwarding of Interest with higher QoS marker while having a pending Interest with a lower QoS marker is a breach of conventional Interest aggregation in the PIT; however, it provides an opportunity to the router to enforce the higher priority QoS treatment to the Interest as well as the resulting Data traffic.

#### 6.4. Data Delivery at PIT

Assuming the router has forwarded more than one Interest in the network for the same content, there is no certainty which of the Interests (i.e. one with higher QoS priority or with lower QoS priority) would be satisfied first. This depends on various network conditions as well as the distance of the content cache having the requested content. In this case, it is very likely that arrival of Data packet for either Interest is going to satisfy all pending Interest marked with different QoS marking.

The PIT behavior of Data handling does not change with the addition of the QoS marker mainly because the content in the Data packet does not change with the QoS marker. Depending on the implementation of the PIT aggregation, two Data forwarding scenarios are possible. In both cases, it also determines how the data packet is queued on the downstream interface.

- o If the Interest are aggregated as shown in Figure 10, Data packet to the downstream interface is forwarded with the higher QoS marking recorded at the interface in the PIT.
- o If the Interest are aggregated as shown in Figure 11, Data packet to the downstream interface is forwarded with its original QoS marking recorded at the interface in the PIT.

With the QoS handling in the PIT described in Section 6.3, it is possible to satisfy a pending Interest with a lower QoS marking with the arrival of a Data packet having higher QoS marker. As a result, a user with lower QoS subscription may experience an improved latency from the network. Note that this is a legitimate behavior as it is ICN's one of the design goals i.e. to optimize the network round-trip time providing better user experience.

## 7. Security Considerations

ICN being name-based networking opens up new security and privacy considerations which have to be studied in the context of name-based QoS framework.

Depending on where the QoS marker is encoded in the Interest, certain security attack scenarios against QoS treatment are possible. If the QoS marker is located inside the security envelope, it can be read, but not changed. Conversely, if the QoS marker is placed outside of the security envelope, it can be added as hop-by-hop message header and, therefore, can be modified by the ICN router nodes in the transit.

Consumer procedure discussed in Section 5.1 and forwarder procedure discussed in Section 5.2 shall decide the security requirements related to implementing QoS treatments in ICN.

## 8. Summary

This draft provides an architecture to implement end-to-end QoS treatments in the ICN network using a name-based non-routable QoS marker suffix. Mechanics for mutation of the QoS marker is discussed along with schemes to preserve the original QoS marker added by the



consumer. The independence between content name and QoS marking makes their evolution easier.

An enhancement to the PIT to store the per-interface QoS state is presented. An optimization to deal with the QoS-aware Interest aggregation is discussed where a number of pending Interests requests in the PIT for the same content to be normalized around the highest QoS marking.

Security requirements are dependent on whether the QoS marker is encoded inside a security envelope or outside of it. Consumer and forwarder procedure requirements shall also govern the security features.

A detailed analysis and evaluation is required, either through a prototype using [VICN] or a simulation using [NDNSIM], to assess the effect of QoS-aware PIT aggregation. The details on the design of a naming scheme for QoS marking in the content name is required. It is also necessary to test and measure the effectiveness of the QoS framework by deploying it using a multimedia streaming application.

## 9. Acknowledgements

We thank all contributors, reviewers and the chairs for their valuable time in providing the comments and feedback, which has helped to improve this draft. We would like to thank Marc Mosko who provided useful feedback on proposed name-based encoding scheme and nameless content objects.

## 10. IANA Considerations

TBD

## 11. References

### 11.1. Normative References

[CCNXMESSAGES]

"Marc Mosko et.al., CCNx Messages in TLV Format, ICNRG Internet-Draft 2019", <<https://tools.ietf.org/html/draft-irtf-icnrg-ccnxmessages-09#section-3.6.1.1>>.

[CCNXSEMANTICS]

"Marc Mosko et.al., CCNx Semantics, ICNRG Internet-Draft 2018", <<https://datatracker.ietf.org/doc/draft-irtf-icnrg-ccnxsemantics/>>.

- [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>>.

## 11.2. Informative References

- [CHRISTOS] "Christos Tsilopoulos et.al., Supporting Diverse Traffic Types in Information Centric Networks, Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, Pages 13-19, ICN 2011", <<https://dl.acm.org/citation.cfm?id=2018588>>.
- [ICNFLOW] "Moiseenko et.al., Flow Classification in Information Centric Networking, ICNRG Internet-Draft, February 2017, <https://datatracker.ietf.org/doc/draft-moiseenko-icnrg-flowclass/>".
- [IOTQOS] "Cenk et.al., Quality of Service for ICN in the IoT, ICNRG Internet-Draft, July 2019, <https://datatracker.ietf.org/doc/html/draft-gundogan-icnrg-iotqos-01>".
- [JACOBSON] Jacobson, Van et.al, "Networking Named Content, 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pp. 1-12, 2009".
- [MIRCC] "Milad Mahdian et.al., MIRCC: Multipath-aware ICN Rate-based Congestion Control, Proceedings of the 3rd ACM Conference on Information-Centric Networking Pages 1-10, ICN 2016", <<https://dl.acm.org/citation.cfm?id=2984365>>.
- [MPVCICN] Jangam, A., Ravindran, R., Chakraborti, A., Wan, X., and G. Wang, "Realtime multi-party video conferencing service over information centric network", IEEE International Conference on Multimedia and Expo Workshops (ICMEW) Turin, Italy, pp. 1-6, June 2015, <<https://ieeexplore.ieee.org/document/7169810>>.
- [NADAY] "M. F. Al-Naday et.al., Quality of service in an information-centric network, 2014 IEEE Global Communications Conference GLOCOM.2014, pp. 1861-1866, Dec 2014".

- [NDNRTC] Gusev, P., Wang, Z., Burke, J., Zhang, L., Yoneda, T., Ohnishi, R., and E. Muramoto, "Real-time Streaming Data Delivery over Named Data Networking," IEICE Transactions on Communications vol. E99.B, pp. 974-991, May 2016, <<https://doi.org/10.1587/transcom.2015AMI0002>>.
- [NDNSIM] "ndnSIM: NS-3 based Named Data Networking (NDN) simulator", <<http://ndnsim.net/2.2/>>.
- [QOSARCH] "Dave Oran, Considerations in the development of a QoS Architecture for CCNx-like ICN protocols, ICNrg Internet-Draft, February 2020, <https://datatracker.ietf.org/doc/draft-oran-icnrg-qosarch/>".
- [RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", August 2006, <<https://tools.ietf.org/html/rfc4594#section-2>>.
- [VICN] "Mauro Sardara et.al., Virtualized ICN (vICN): towards a unified network virtualization framework for ICN experimentation, ICN'17 Proceedings of the 4th ACM Conference on Information-Centric Networking Pages 109-115", <<https://wiki.fd.io/view/Vicn>>.
- [WEIBO] "Weibo Chu et.al., Network delay guarantee for differentiated services in content-centric networking, Journal of Computer Communications Volume 76, Pages 54-66, February 2016".
- [XINGWEI] "Xingwei Wang et.al., Energy-efficient ICN routing mechanism with QoS support, Journal of Computer Communications Volume 131, Pages 38-51, 2018".
- [YAOGONG] "Wang, Yaogong et.al., An Improved Hop-by-Hop Interest Shaper for Congestion Control in Named Data Networking, ACM SIGCOMM Computer Communication Review, August 2013", <<https://dl.acm.org/citation.cfm?id=2491233>>.

## Authors' Addresses

Anil Jangam (editor)  
Cisco Systems  
San Jose, California 95134  
USA

Email: [anjangam@cisco.com](mailto:anjangam@cisco.com)

Prakash Suthar  
Cisco Systems  
Rosemont, Illinois 60018  
USA

Email: [psuthar@cisco.com](mailto:psuthar@cisco.com)

Milan Stolic  
Cisco Systems  
Rosemont, Illinois 60018  
USA

Email: [mistolic@cisco.com](mailto:mistolic@cisco.com)

ICNRG  
Internet-Draft  
Intended status: Informational  
Expires: July 17, 2021

I. Moiseenko  
Apple Computer  
D. Oran  
Network Systems Research and Design  
January 13, 2021

Flow Classification in Information Centric Networking  
draft-moiseenko-icnrg-flowclass-07

Abstract

For the ubiquitous and highly important Internet protocols (TCP, UDP, IP), flows are conventionally identified by the "5-tuple" of source and destination IP addresses, source and destination port, and protocol type in an IP packet. Information Centric Networking (ICN) is a new paradigm where network communications are accomplished by requesting named content, instead of sending packets to destination addresses. This document describes mechanisms allowing ICN forwarders, consumers, producers and other ICN nodes to encode, decode, and process equivalence class identifiers (flows) at any desired granularity of a routable name prefix and beyond the routable name prefix. This document is a product of the IRTF Information-Centric Networking Research Group (ICNRG).

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 July 17, 2021.

Copyright Notice

Copyright (c) 2021 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
1.1. Requirements Language . . . . .	3
2. Flow Identification Challenges and Opportunities in ICN . . . .	3
3. Flow Encoding Schemes . . . . .	4
3.1. Equivalence class component count (EC3) . . . . .	5
3.2. Equivalence class name component type (ECNCT) . . . . .	6
4. Producer operation . . . . .	7
5. Consumer operation . . . . .	8
6. Forwarder operation . . . . .	8
7. IANA Considerations . . . . .	9
8. Security Considerations . . . . .	9
9. Normative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

The problem of identifying groups of packets that get consistent treatment in a network and allowing that treatment to be independent and isolated from the treatment of other groups of packets, is ubiquitous and long-standing. The purposes to which this identification can be put is highly varied, including such functions are providing differentiated quality of service, traffic engineering, traffic filtering for security functions like intrusion detection and firewalling, etc.

Providing the capability to apply different functions to groupings (formally equivalence classes) of packets is generally known as the "flow identification problem" where the definition of what constitutes a "flow" is highly dependent on the particular protocol or protocols carrying the packets. Some of the above uses of flows also bring a mechanism requirement that the flow identification technique be useful to have not just equivalence classes, but the ability to apply some useful notion of fairness among the instances of each equivalence class. There are many possible flow identification techniques that are either too granular (spatially or

temporally) to establish fairness, or conversely too coarse and cannot separate traffic a fine enough level to have useful fairness.

For the ubiquitous and highly important Internet protocols (TCP, UDP, IP), flows are conventionally identified by the "5-tuple" of source and destination IP addresses, source and destination port, and protocol type in an IP packet. Some systems augment this by further distinguishing equivalence classes by the TOS/DSCP field, but this is secondary to the 5-tuple methods. 2-party flows are present where the source and destination addresses are unicast IP addresses. Multi-party flows can exist when the destination IP address is a multicast address. One key common characteristic is that the identification of flows depends in a very deep way on the presence of source addresses in the packets, and the limited richness of IP addresses is correspondingly constraining as a means to classify traffic in a semantically meaningful way.

The purpose of this document is to devise a mechanism allowing ICN forwarders, consumers, producers and other ICN nodes to encode, decode, and process equivalence class identifiers (flows) at any desired granularity of a routable name prefix and beyond the routable name prefix.

### 1.1. 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].

## 2. Flow Identification Challenges and Opportunities in ICN

ICN systems differ from IP-based designs in a number of ways, three of which are quite fundamental.

1. The packets are addressed to a rich namespace of packets, which is hierarchical and carry semantic information that can be useful for classification of flows.
2. Conversely, the packets do not contain source addresses of any kind, which means that identifying flows as groups of packets between a single pair of endpoints (in the unicast case) is not possible for intermediate forwarders (other than possibly the first-hop forwarder if it serves a single consumer per interface).
3. Instead of group-based multicast, ICN systems use multi-destination delivery semantics. This allows a different way to map packets to flows, and in fact in the IP world multicast has

been difficult to use partly because there is no good way to make use of flow identification for multicast flows (for a variety of reasons).

These differences lead to a need to find a different method to identify flows than used in the IP protocol suite. Ideally, the method would provide semantics that map well with the expected uses of ICN to build applications. It would also use native capabilities in the ICN protocols rather than having to change the protocol architecture in ways that affect the semantics or utility of an ICN approach to networking.

In NDN and CCN protocols, Interest and Data names are the only identifiers in the network; neither source addresses nor destination addresses are employed. Each Interest packet is responded by exactly one Data packet, producing a useful property known as "flow balance". This means that flow identification can be tied directly to the Interest/Data exchanges. The key to having useful flow identification is for the equivalence classes to be associated with the names in the corresponding Interest and Data packets, and to be stable over multiple exchanges using different names that share some common "handle" that can be used to separate the names into equivalence classes. As mentioned above, simply using the routing state that maps name prefixes to routes does not provide a useful set of equivalence classes, because:

- o in general, routing prefixes are too coarse; many equivalence classes of packets are generally covered by a single routing prefix because they are present at the same set of destinations from a routing perspective;
- o practical, scalable routing needs to do route aggregation, which further blurs the discrimination of the equivalence classes.

Therefore, NDN and CCN protocols need to have something that both relates to the name structure but provides finer granularity for flow classification purposes. This document describes two alternative mechanisms addressing these issues.

### 3. Flow Encoding Schemes

Flow encoding schemes described in this document allow ICN systems to perform flow identification at any desired granularity of a routable name prefix and beyond the routable name prefix. Techniques described herein permit both consumer nodes and forwarders to use equivalence classes to perform per-flow functions. The encoding to achieve the flow classification is lightweight and does not require changes to the protocol architecture in ways that affect the



semantics or utility of an ICN approach to networking. Furthermore, equivalence classes can be specified by the data producer, in contrast to IP protocols in which the data producer can only control the destination port as an equivalence-class discriminator.

No matter what method is used to identify equivalence classes that can be treated as flows, there is the independent but critically important issue of how to scale any state that is kept on a per-flow basis when the flow count is very high. For consumers and producers, this state scales naturally with the number of applications and application interactions are going on simultaneously. Therefore the scaling limit is not likely to be in the producers or consumers. For ICN forwarders this state could scale quadratically or worse if the forwarders need classic prior resource reservation to deterministically partition resources on a producer/consumer pair basis. This need not be the case however. Practical resource control algorithms exist that keep state only for "active" flows (those with packets either currently or recently moving through the network). Further state reduction is also possible with some loss of accuracy using approximate techniques, like stochastic fair queuing (SFQ). If the ICN forwarder cannot keep all the state due to memory or processing limitations, it faces the common problem of which flows to remember and which to forget. This problem is fundamental, and is mostly independent of the choice of flow identification method in the protocol. Flow encoding schemes described in this document provide a method for identifying equivalence classes using protocol machinery that already has to scale (e.g. name parsing and lookup) and hence does not introduce a new class of problems not inherently present.

### 3.1. Equivalence class component count (EC3)

For this encoding scheme a new field called equivalence class component count (EC3) is introduced into the Data packets. It is set by a producer and counts the number of name components in the corresponding name that are to be considered, when grouped together under the same prefix part of the name, to be one equivalence class instance. This allows either finer (or coarser) granularity than provided by routing prefixes. Because the EC3 is a separate field of the packet (Figure 1), producers can "regroup" equivalence classes dynamically by including more or fewer levels of the name hierarchy when they respond to Interests for the corresponding Data packets. This brings a set of clear advantages and disadvantages. The primary advantage is flexibility in re-grouping equivalence classes, especially in aggregating flows at different granularities. The main disadvantage is that the binding of the equivalence class into the namespace is not explicit, and hence it is harder to enforce consistent interpretation among producers, consumers and forwarders.

An additional consideration with the EC3 encoding scheme is whether or not the field is inside or outside the security envelope that provides cryptographic packet integrity to the name and data in the data packet. Either approach is possible; however having the field outside the security envelope would allow ICN forwarders to modify it, allowing the aggregation/disaggregation of flows to be performed by the forwarders as well as the consumers. Conversely, leaving the field outside the security envelope may enhance certain attack scenarios against flow classification when employed for quality of service differentiation or firewall filtering.

/youtube	/<mediaID>	/video OR /audio	<frameID>	<segment#>
Name component type	Name component type	Name component type	Name component type	Segment component type
Equivalence Class Component Count = 2 (up to MediaID stream) OR Equivalence Class Component Count = 3 (video or audio substream)				

An example of EC3 encoding of flow information.

Figure 1

### 3.2. Equivalence class name component type (ECNCT)

For this scheme the equivalence class information is encoded directly in the name, by adding a name component to the name of the Interest and Data packets. This new typed named component is called equivalence class name component type (ECNCT). It is set by the producer as part of constructing all Data packets in the desired equivalence class and is therefore immutable for the lifetime of the associated named data. A consequence of this is that the ECNCT is present in Interest packets as well, and hence may affect both PIT matching and FIB matching. The Equivalence Class name component both names the equivalence class explicitly, and implicitly makes all Data packets named below it in the hierarchy part of that equivalence class. In other words, the name can have multiple equivalence class (e.g. flow and subflows) markings using this scheme (Figure 2). As in EC3 encoding scheme, depending where in the name component hierarchy the ECNCT is placed, one can have either finer or coarser granularity than provided by routing prefixes.

The exact details of how to encode the ECNCT name component may differ among ICN architectures. The CCN design has explicitly typed name components, so for that protocol an explicit name component type can be assigned straightforwardly. The NDN design eschews typed name components and instead uses textual naming conventions for name components. In that case an architectural constant string would be chosen to distinguish ECNCT from other name component semantics.

/youtube	/<mediaID>	/video OR /audio	<frameID>	<segment#>
Name component type	Flow component type	Flow component type	Name component type	Segment component type

An example of ECNCT encoding of flow information.

Figure 2

When an ICN forwarder receives a packet with a name carrying ECNCT(s), it can be processed on a component-by-component basis, and substreams can be identified according to name prefixes indicated by the equivalence class identifiers. The identification of substreams enables special treatment of selected substreams. For example, video substreams can be discriminated from other substreams, such as audio substreams. In the example in Figure 2, two name components include equivalence class identifiers to define a hierarchy of flows (or substreams). Specifically, two flow components are encoded to define the following hierarchy of flows:

First level name prefix: /youtube/<mediaID>

Second level name prefix: /youtube/<mediaID>/video

Second level name prefix: /youtube/<mediaID>/audio

#### 4. Producer operation

In ECNCT encoding scheme, an ICN producer receives an Interest packet carrying equivalence class identifiers in the name. A producer might use the equivalence class identifiers for demultiplexing, load sharding and other purposes, and reply with a Data packet matching the Interest name.

In EC3 encoding scheme, an ICN producer receives an Interest packet that might not carry an equivalence class identifier. In such case,

the producer may refer to the name schemas used in a particular application to dynamically determine the equivalence class identifier for Interest demultiplexing, load sharding and other purposes, and for replying with a Data packet carrying the equivalence class identifier in EC3 field.

## 5. Consumer operation

An ICN consumer may also use the knowledge of equivalence classes of packets to take certain actions. For example, when a Data packet with a name specifying a particular equivalence class arrives at a consumer in response to a previously sent Interest packet, the consumer can associate the data packet with the correct equivalence class. Consequently, the consumer can manage subsequent Interest/Data exchanges with the same name prefix and equivalence class identifier (e.g., EC3 or ECNCT) as one flow. Associated measurements such as round trip time (RTT) or marginal delay can be leveraged to perform flow and congestion management for the equivalence class as a whole.

## 6. Forwarder operation

A flow table may be provisioned in ICN node to enable the node to make decisions about performing actions on Interest and/or Data packets based on one or more equivalence classes. The flow table can include name prefixes mapped to equivalence class identifiers obtained from previous Interest-Data exchanges. In ECNCT encoding scheme, Interest packets carry the equivalence class identifier, therefore flow table may only include name prefixes. Typically, name prefixes in flow table are more granular than prefixes in the FIB, but less granular than names in the PIT. Flow table could be separate from other elements of ICN node or could be integrated with FIB or PIT.

Flow management logic can be configured to treat flows having the same equivalence class similarly. Actions taken that are related to flows or objects having a similar equivalence class can include, but are not limited to, dropping a packet, using a particular interface for a packet, security related actions (e.g., filtering traffic for security functions like intrusion detection and firewalling), quality of service (QoS) related actions (e.g., types of resources to allocate to the packets, moving a packet up in the queue for forwarding purposes, etc.), and/or traffic engineering (e.g., selecting one path over another path). Flow management logic can enable such actions to be taken on a particular flow based on the equivalence class associated with the flow or object and policies related to the equivalence class.

Specific examples of how ICN node can use the knowledge of equivalence classes of packets include, but are not limited to, the following:

1. Enforce rate control for the equivalence class as a whole (e.g., dropping packets, queuing packets, etc.);
2. Estimate the number of simultaneous flows traversing a bottleneck link, which can improve the performance of many congestion control schemes; and
3. Make more intelligent selections of which packets to cache at the ICN forwarder, for example, to prefer to cache many packets of the same equivalence class.

#### 7. IANA Considerations

This memo includes no request to IANA.

#### 8. Security Considerations

Certain attack scenarios against flow classification for quality of service or firewall filtering may be prevented if the EC3 field located inside the security envelope. ICN forwarders can read, but not change, the EC3 value, because the EC3 field is covered by a security signature and not encrypted.

If the EC3 field is outside of the security envelope, it can be placed in the hop-by-hop headers and, therefore, be modified by the transit ICN forwarders. This allows the transit ICN forwarders to override the flow definitions set by the producer applications, but opens the system to various attack scenarios.

Modification of equivalence class identifiers in ECNCT encoding scheme effectively modifies the packet name, and therefore, ECNCT does not introduce any additional security threats.

#### 9. 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>>.

Authors' Addresses

Ilya Moiseenko  
Apple Computer  
USA

Email: imoiseenko@apple.com

Dave Oran  
Network Systems Research and Design  
4 Shady Hill Square  
Cambridge, MA 02138  
USA

Email: daveoran@orandom.net