Name-based Quality for Name-based Networks

Ryo Yanagida, University of Glasgow
Jeremy Singer, University of Glasgow
Paul Harvey, University of Glasgow
Leon Wong, Rakuten Mobile, Inc.
Colin Perkins, University of Glasgow

***Abstract*—Name-based protocols give an opportunity to revisit aspects of content delivery including Quality of Service (QoS). Prior named-based approaches require additional signalling, distinct from forwarding, adding complexity. We propose a new name-based QoS mechanism, leveraging the simplicity of the name-prefix forwarding model. We demonstrate: (i) the effectiveness of our QoS prioritisation via a proof-of-concept implementation, (ii) scalability through analytical evaluation, and (iii) simplicity and clarity of the trust model through an analytical model.**

***Index Terms*—Content Delivery, ICN, QoS**

I. Introduction

The Internet uses an address-based model where clients retrieve content by connecting to servers by IP address. This was well-suited to the early Internet, where servers at known locations hosted data, but is increasingly poorly suited to today's virtualised, cloud-based, global infrastructure, where clients must locate the server with the 'best' replica out of servers hosting the copies.

Name-based protocols, NDN [1]/CCNx [2], shift the paradigm. Rather than find the address of an appropriate server, connect, then fetch the content, name-based approaches request content by name *without specifying a location*. The network routes the request to an appropriate replica, achieved by making names first-class identifiers.

These protocols improve efficiency and provide pervasive caching [3] but lack support for quality of service (QoS). We explore how QoS can be added to name-based protocols, avoiding many trust issues inherent in IP-based QoS, clarifying the point of control, and reducing complexity. Specifically, our proposal selects names to prioritise by a name-prefix and applying QoS policies to packets with matching names. We propose Forwarding Behaviours (FBs), based on the Per-hop Behaviours (PHBs) defined in the Differentiated Services (DiffServ) [4] to manage forwarders' scheduling/forwarding behaviours. We also describe how this mechanism can be implemented in one particular name-based network protocol, the Named Data Networking (NDN), as a new name-based QoS module in the forwarder.

We define a novel name-prefix based QoS mechanism for name-based protocols, realised within NDN [1] as proof-of-concept. We simulate this using ndnSIM [5] and compare its scalability with IP-based DiffServ. Finally, we perform a trust model analysis. We show that our proposed approach scales and avoids trust issues inherent in DiffServ.

This work was funded in part by Rakuten Mobile, Inc.

II. Background

Traditional IP Quality of Service requires a per-packet label to distinguish the required QoS treatment. This label (ToS field) is an optional header that is often subject to modification or removal ('bleaching') [6]. This header can be set arbitrarily, making it subject to abuse, see Section VII.

Name-based networking [7] changes the network architecture to enable clients to express interest in content by *name*, rather than sending request to a *location* denoted by an IP address. This moves the routing, forwarding, and addressing architecture of the network away from using *topological location* of a device in the network, to instead route requests based on the *name* of the data sought. The result is a more natural fit with the way the Internet is used today – many services' infrastructures are virtualised, cloud-based, deployed in locations across the globe.

Content Delivery Networks (CDNs) account for significant amounts of traffic (e.g., [8] reports 40% of traffic at an European IXP was due to CDNs in 2021) and require the client and CDN to jointly perform a complex name resolution to find the IP address of an server based on their location and the requested content, before they can connect to that server to retrieve the content.

Name-based protocols move the focus from the *hosts* to the *data*, removing the separate name resolution step and integrating it into the routing and forwarding model. Not only this is a more natural model for content distribution, it enables features at the network layer, such as on-path caching and request aggregation, that are difficult in IP.

Figure 1 shows an example of name-based networking. It comprises a consumer, which request content by sending Interest packets; a producer, which responds with Data packets; and forwarders that route them. Forwarders have a routing information base (RIB) holding name-prefix routing information; a forwarding information base (FIB), which holds name-prefix forwarding information; a pending interest table (PIT), which keeps track of Interest packets that have yet to be satisfied and which network interface they came from; and a content store (CS) that caches received Data packets in case it receives additional matching Interest packets.

III. Name-based Quality of Service

Motivated by efficiency and ease of deployment, we propose name-based QoS, selecting packets for priority forwarding based on name-prefixes as shown in Figure 2.

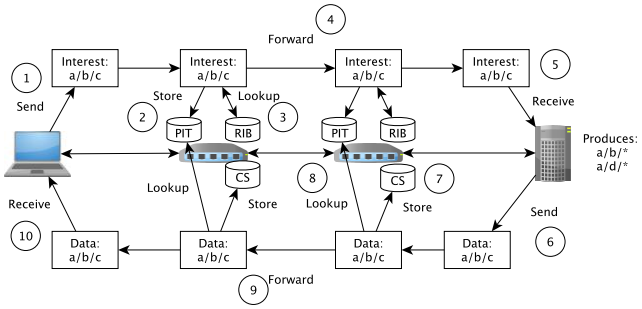


Figure 1: Example of name-based network communication.

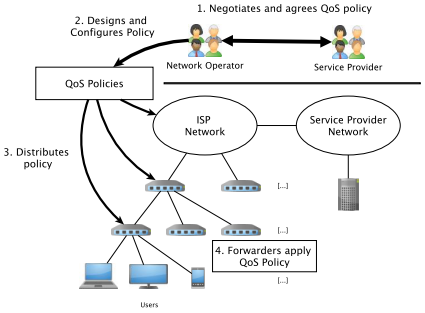


Figure 2: Diagram showing example QoS operation.

Per-Hop Forwarding Behaviours (FB): We adopt the per-hop behaviours defined for IP DiffServ: best-effort forwarding (BE-PHB) as the default policy [9]; expedited forwarding (EF-PHB) to reduce queueing delay [10]; assured forwarding (AF-PHB) to reduce the chance of being dropped when queued [11]; and lower-effort forwarding (LE-PHB) that deliberately selects packets to be dropped to protect default/best-effort packets [12]. We leverage DiffServ PHBs since current deployments in IP networks demonstrate they are effective in providing different treatments to different classes of packets at scale.

Name-based QoS Prioritisation: To align with name-based forwarding, we assign forwarding classes based on the name of the requested data. There are three possible approaches to name matching: *Prefix-based matching* selects packets to prioritise using a name prefix, assigned hierarchically, with the left-most component identifing the content provider and sub-components identifing increasingly specific data; *Suffix-based matching* selects names based on their final component, e.g., prioritising video content ending in `.mp4`, which may be challenging as the latter part of the name is often used to segment the data in name-based networks; and finally *hybrid matching* that combines both prefix- and suffix-based approaches with the prefix selecting the 'parent' name space while the suffix selects the type of content within that namespace. We use **prefix-based matching** to align with the name prefixes used by NLSR name-based routing [13] and to match hierarchical name assignments typically used by content providers.

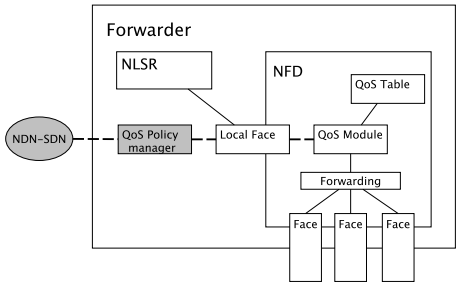


Figure 3: Schematic diagram of the modified forwarder. Grey shapes and dashed lines are proposed future work.

Packet Forwarding: QoS treatment can be applied to INTEREST packets, DATA packets, or both. We choose to apply QoS treatment to *both* packet types. In name-based networking, the DATA packet is tightly coupled to the INTEREST packet. In IP networking, protocols such as RTP can be used to send a stream of data without acknowledging each individual packet. In contrast, a consumer using a name-based protocol will not receive data unless the relevant interest reaches the producer. Therefore, unlike the protocols like RTP, where only the egress packets from the content distributor needs QoS marking, in name-based networks, both INTEREST/DATA packets must be prioritised for an effective QoS treatment.

We distribute QoS policy entries via an **out-of-band management plane**. Name-based forwarding is often combined with pervasive **on-path caching**. Due to space limitations, we do not consider the impact of caching.

We implemented name-based QoS within NDN [1]:

The NDN forwarder daemon (NFD) responsible for forwarding INTEREST and DATA packets, was modified to perform a QoS policy lookup and label packets to select the FB. A QoS module was added, with a *policy table* mapping name pattern to per-hop FB and a *policy decision mechanism* called to identify packets requiring QoS treatment. Internal packet representations were amended to identify the FB. Figure 4 summarises.

Packet scheduling: We use the mechanism built into the 'host system', for example the kernel `qdisc` if UDP is used as an underlay. In the following evaluation, Traffic Control Layer in `ns3` was used to enable priority scheduling of the packet. We provide a packet classification marking mechanism in NFD to append an appropriate marking to be interpreted at the lower layers, compatible with DiffServ.

IV. Experimental Evaluation

We use ndnSIM [5] to answer **three research questions**: is the expected QoS policy behaviour, in terms of its impact on latency, retransmissions, etc., observed **(RQ1)**; does the QoS mechanism introduce undesirable side-effects, e.g., jitter, to prioritised traffic **(RQ2)**; and does the QoS mechanism adapt to dynamic traffic load **(RQ3)**?

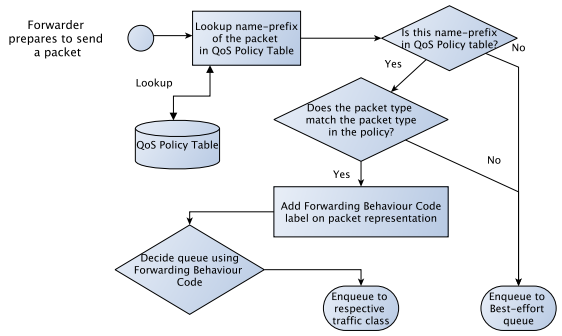


Figure 4: Flowchart of QoS processing within the forwarder

A. Experimental Design

Network Topology: We configure a dumbbell topology with two consumers (C1, C2), producers (P1, P2), and forwarders (F1, F2) The bottleneck connects the forwarders at 10Mbps with 20ms latency; 1Gbps links with 2ms latency connects the other nodes. Two outgoing FIFO queues for traffic classes at the forwarders' bottleneck link use the `ns3` traffic control layer PrioQueue Queueing discipline. Packets are queued using the label passed from NFD.

Network Traffic Profiles: We test fixed-, variable-rate, and intermittent traffic. Consumers are implemented using a mock consumer application with constant-rate requests; producers return fixed-size 1024 byte DATA packets. The following name prefixes are used:

`/streaming_service_A/live/*` – prioritised
`/streaming_service_B/live/*` – best-effort
`/social_network_A/resource/static/*` – best-effort

The forwarders use expedited FB policy prioritising the above prioritised prefix. The consumer C1 requests the prioritised prefix for all except the competing traffic scenario; C2 requests non-prioritised `/social_network_A/*` prefix. The consumers collects application-layer INTEREST-DATA packet measurements, consisting of timestamp, latency, node identifier, INTEREST packet transmission count, and state of interests (satisfied or not).

B. Results

Fixed-rate Traffic: We first validate the correct behaviour of the prioritisation mechanism. We simulate consumers sending a fixed-rate of INTEREST packets for 10 seconds at different packet rates. This scenario observes the behaviour in a fixed-rate load scenario to evaluate RQ1 and RQ2.

With 1024 byte DATA packets, the bottleneck saturates at ~1200pps. We start at 520pps per consumer (1040pps) and iteratively increase the rate to 675pps per consumer (1350pps) in 25pps increments. We observed an increase in non-prioritised traffic's latency. Figure 5a shows 575pps × 2 = 1150pps is approximately the rate the network began to saturate. Figure 5b shows the increased jitter for standard traffic, with no negative impact on the prioritised traffic. Figure 5c shows the distribution of number of interest packets; two or more transmissions indicates the interest timeout triggering retransmission at the consumer. This worsened with increased rate.

Prioritisation enables consistently lower latency profiles of specified name-based traffic, answering RQ1/RQ2.

Variable-rate Prioritised Traffic: Next, we evaluated a scenario where the non-prioritised application requests at a fixed 575pps rate, while the prioritised application increases the rate by 25pps / 2 seconds from 525pps throughout the run – gradually loading the network to observe the behaviour as the bottleneck capacity is exceeded.

Figure 6 shows the simulation results. The latency for the standard traffic increased significantly, while the prioritised traffic exhibited a smaller increase in latency as the sending rate increased. The bottleneck link saturates ~4 seconds, when the both prefixes reach 575pps each – consistent with the results in Figure 5a. The small size of latency boxplot for prioritised traffic (the 25th percentile and 75th percentile are indistinguishable), indicates the very small latency jitter, i.e., the latency is consistent.

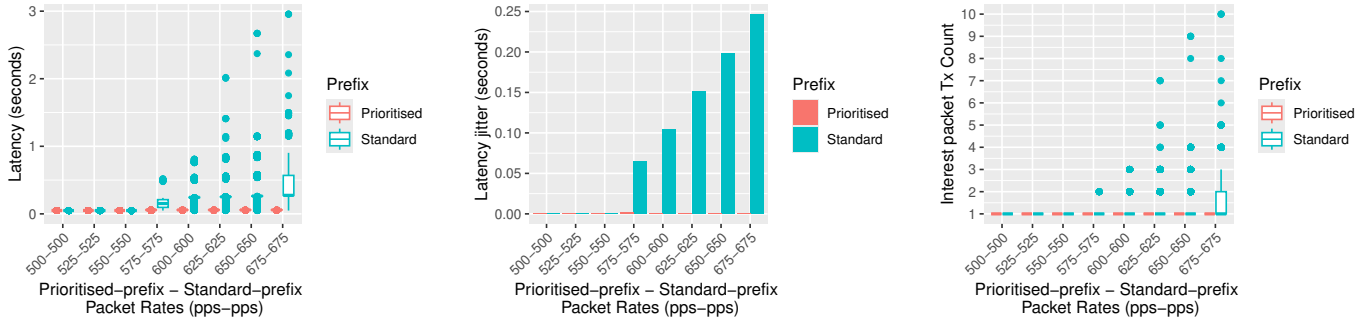
The results show that the network handles the gradual increase in demand and applies the expected QoS policy without any significant impact on the prioritised traffic, transitioning gracefully from the un-saturated to the saturated state, and positively answering RQ2 and RQ3.

Intermittent Background Traffic: Not all traffic in the network requests data at a fixed nor stepped variable rate. Many traffic, such as web browsing and social media feeds are intermittent. We ran tests for 30 seconds with the following traffic pattern: (i) the scenario begins with 500pps for the constant traffic; (ii) the periodic traffic application sends 2 seconds long 700pps interest packets followed by 2 seconds of rest; (iii) terminate the simulation. We ran 25 runs and observed the latency to evaluate how this mechanism copes with the intermittent traffic pattern.

Figure 7a shows the latency for the scenario with C2's constant 'standard' prefix at 500pps, and C1's periodic 'prioritised' prefix requests at 700pps. The vertical broken line indicates the start/end of the periodic traffic. Figure 7b shows the latency results of C1's constant 'prioritised' prefix, with C2's periodic 'standard' prefix traffic.

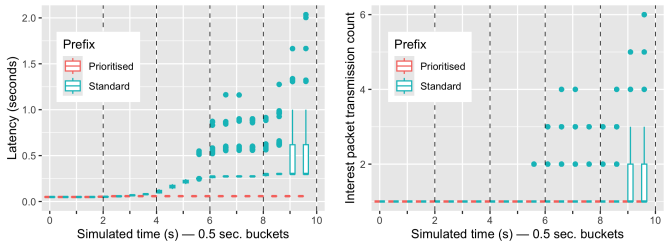
In both scenarios, the prioritised traffic received much lower latency, while the standard prefix traffic exhibited significantly higher, varying, and increasing latency. Demonstrating that our mechanism delivers the desired traffic characteristics of expedited FB, i.e. low latency characteristics dynamically adapting to the prioritised prefix, addressing RQ1 and RQ3.

Figure 7c shows a scenario where no prefixes are prioritised. Constant traffic for C1 in red as well as the periodic traffic for C2 are both experiencing a large increase in latency during the periodic 'standard' traffic demand. This demonstrates the need for prioritisation and QoS treatment in a resource constrained environment.



(a) Latency at varying INTEREST rates (b) Latency jitter at different packet rates. (c) INTEREST transmission counts

Figure 5: Fixed-rate experiment results – two consumers sent the INTEREST packets at the same rate with distinct prefixes. As the INTEREST packet rate increases, the DATA packets returned increases, saturating the bottleneck link. Leading to a higher, less stable latency characteristics, and time-outs for non-prioritised prefix.



(a) Latency Impact (b) INTEREST counts

Figure 6: Variable-rate results; increasing priority traffic

Metric	DiffServ	NB QoS
Fwd. states	RIB entry	RIB, PIT entry
QoS States	⟨Src / Dst IP, Src / Dst Pt⟩ QoS Policy Rule	QoS Policy Entry

Table I: State entities required for forwarding and QoS

The results demonstrates the effectiveness of name-based traffic filtering as well as its dynamic behaviour of the relative and non-explicit QoS prioritisation mechanism, adapting to the condition as it changes.

V. SCALABILITY ANALYSIS

We consider the following scenario: the network is a consumer ISP that uses QoS policy to improve the latency for specific real-time video streaming services; all routers in the network prioritise that real-time traffic with respective QoS mechanisms For QoS operations, DiffServ needs edge nodes to either classify the traffic for QoS treatment, or verify the flow's expected QoS label('policing'). They must track flows via a set of rules. In our name-based QoS approach, all forwarders use the QoS policy table to match the name in the packet to the name-prefix for any applicable policy. Therefore, these entries must exist to prioritise a name-prefix. Table I summarises the metrics considered.

Scaling with topology changes: We examine the effect of topology changes to the QoS states by keeping the number of content and users constant, while changing the number of routers/forwarders, n_r, and the layers in the ISP network, n_l, as shown in Figure 2.

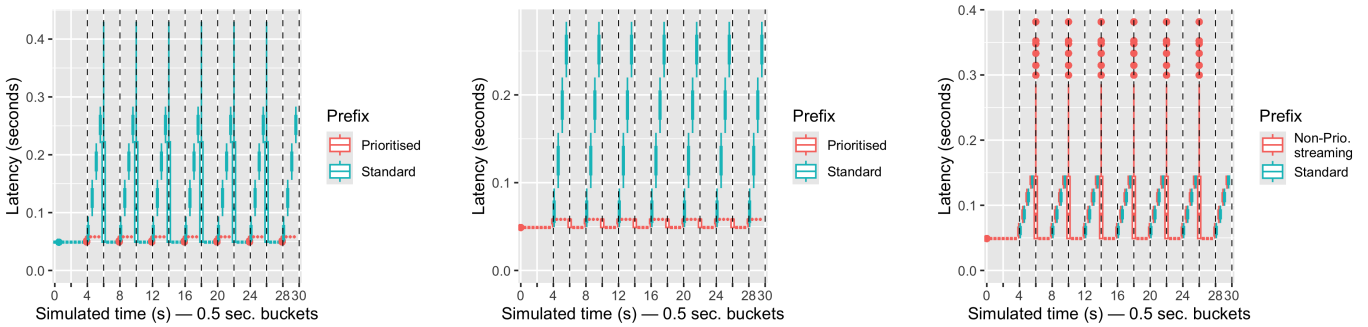
Impact of topology depth: For DiffServ, as the number of layers in the topology increases, RIB entries for lower-layer routers will be divided-up but not increase. The flow states at the edge of the network will also not change. Since the QoS policy only considers the service provider and the code points, the topology does not impact them.

For name-based QoS, regional and access routers will not have specific name-prefix entries for each service provider. Therefore, RIB entries will not grow. However, the PIT holds all names requested from the downstream, and increasing the number of layers means more hops on-path. Therefore, in the worst-case scenario when all hosts requests distinct names, PIT grows $O(n_l)$ up to the number of users n_u. If INTERESTs aggregate, because users request the same content in close time synchrony, PIT growth is slowed. Since the number of total forwarders do not increase the number of total QoS policy entries present remains the same.

Impact of number of routers: In DiffServ, an increase in the number of routers requires routing table growth across the network at $O(n_r)$. However this does not impact the number of flow-states held by the traffic classification-/QoS policing-node. Similarly, this does not affect the number of QoS policy rules.

For name-based QoS, RIB entries exist on every router so when number of routers grows, state grows $O(n_r)$. Unlike the number of layers, even for the worst case scenario, more forwarders mean simply dividing the number of content requested in the network and does not increase the number of PIT entries. Since the QoS policy must exist on every forwarder, the QoS policy entry will grow $O(n_r)$.

Due to the way forwarding works and how our approach requires a copy of QoS policy entry on every forwarder, DiffServ scales better as the topology scales.



(a) Prioritised and non-prioritised traffic latency as the *prioritised* traffic exhibits bursty demands.

(b) Prioritised and non-prioritised traffic latency as the *non-prioritised* traffic exhibits bursty demands.

(c) The latency of non-prioritised prefix from C1 and C2, as the C2's standard prefix application exhibits bursty demands.

Figure 7: Intermittent experiment results – one host requests at a 500 pps fixed-rate, while the other requests at 700pps for 2 seconds every 4 seconds. Vertical lines indicate when the periodic requests starts and ends.

Scaling with number of users: For DiffServ, as the number of users, n_u, grows, RIB at lower layers grows at $O(n_u)$. In addition, the flows tracked for traffic classification/QoS policing at the edge grows $O(n_u)$. For name-based QoS, growth of the PIT depends on INTEREST aggregation and caching. Assuming all users request distinct content, PIT entries grow as $O(n_u)$ as no caching can take place. However, if users request identical content in time synchronisation, PIT entries will aggregate and remain constant. This will not affect RIB entries nor QoS policy as both dependent on contents and service providers serving them. Here, even in the worst case scenario, the name-based QoS approach scale better as the number of users increases.

Scaling with content: We next examine how the changes in amount of content, n_c, and the number of services offered, n_s, affects the state held by the network, while keeping the topology and the number of users fixed.

For DiffServ, the main pressure is from the added service providers requiring more RIB entries, which grow at $O(n_s)$, as well as more rules needed for traffic classification / QoS policing that also grow as $O(n_s)$. However, more variety of content being requested will not lead to a state entity growth as the number of flow required will remain equal to the number of user, as they need a flow each.

For name-based QoS, growth in service providers requires additional RIB entries in the core-network, growing as $O(n_s)$, and QoS policy in forwarders that grows as $O(n_s)$. The growth of available content may increase burden on the PIT, depending whether INTEREST aggregation or caching occurs or not. While the growth against contents can be up to $O(n_c)$, the starting point would be much lower than the n_u unlike IP flow-state. Since the number of content each user consume is limited to one, the number of PIT entry is limited to the n_u at each layer at most, which is no worse than the IP flow-states also at n_u.

In the context of content and service provider growth, our name-based QoS mechanism scales no worse than DiffServ.

Summary: We summarise our analysis in Table II, noting that while the name-based mechanism scales worse in terms of topology growth, it scales better in terms of users, and no worse in terms of the content and service growth when assuming a worst case scenario for name-based approach.

VI. TRUST MODEL ANALYSIS

For IP DiffServ, from the network operator's perspective, QoS labels are difficult to trust unless the network is directly connected to the service provider since they are commonly removed or modified at network boundaries [6]. This encourages operators to either ignore the marking, or apply their own interpretation, e.g., through deep packet inspection or rule-based traffic matching. Although DiffServ has the potential to be deployed incrementally, any presence of third-party network components on-path introduces an uncertainty in terms of the legitimacy of markings. Similar concerns arise for the service provider: without full knowledge of exact path between them and the customer, it is hard to trust that the QoS markings are correct. Therefore, any service agreements of QoS policy for their traffic can only be made if they are connected directly. Even then, for each provider, a service provider may have to provide different markings depending on the agreement with each operator.

For name-based QoS, the trust model is more straightforward. Regardless of how the service provider and the network operator are connected, network operator can construct the policy entries based on the name structure with the desired characteristics. With an appropriate name structure that includes the service provider as a top-level name, a set of policy can be constructed for each service provider. As names are fundamental to the forwarding, on-path components will not be able to modify them. From the service provider's perspective, a name-based approach enables the following: First, this enables the agreement with the network provider to be feasible regardless of

Metric	Topology — Layer (n_l)	Topology — Routers (n_r)	Users (n_u)	Contents (n_c)	Services (n_s)
Constants & Limits	n_r, n_u, n_c, n_s	n_r, n_u, n_c, n_s	n_l, n_r, n_c, n_s	n_l, n_r, n_u, n_s	n_l, n_r, n_u, n_c
IP-RIB Entry	$O(1)$	$O(n_r)$	$O(n_u)$	$O(1)$	$O(n_s)$
IP-Flow	$O(1)$	$O(1)$	$O(n_u)$	$O(1)$	$O(1)$
IP-QoS Rule	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(n_s)$
NB-RIB Entry	$O(1)$	$O(n_r)$	$O(1)$	$O(1)$	$O(n_s)$
NB-PIT Entry	$O(n_l)$	$O(1)$	$O(n_u)$	$O(n_c)$	$O(1)$
NB-QoS Policy	$O(1)$	$O(n_r)$	$O(1)$	$O(1)$	$O(n_s)$

Table II: Table showing state entities required for forwarding and QoS mechanism.

how they are connected. Second, this allows more flexible agreement with the network operators where the treatment of the traffic can be different amongst the different network operators. The service provider does not need to mark packets at all, since QoS treatment is now inherent in the name. The new challenge is to measure and demonstrate that these QoS treatments are actually taking place and to show whether they are making a difference.

VII. Related Work

Many QoS management approaches for ICN require an explicit admission and packet marking [14] [15], and may require an additional signalling protocol [16] [17] [18]. The network needs to learn the specific name that the end-host is requesting for specific QoS treatment, and the topological direction of it. Meanwhile, the non-explicitly admitted QoS treatments can omit labelling of the individual packets by employing a purely name-based approach [19] – the unit of QoS treatment is a set of packets, which in the ICN context are derived from first-party identity at the network layer and cannot be altered. This is a crucial aspect, as this is a departure from IP-based QoS mechanism that requires labelling and has the issues of those markings being bleached at the domain boundaries, symptomatic of trust issues in these markings as discussed in Section VI.

Gündogen, et. al. [19] propose a purely name-based approach where a name prefix is used to specify QoS treatments, focusing on disaster recovery scenarios. This is similar to our approach, but signals custom parameters rather than use standard PHBs. Our approach provides more granular QoS by adopting standard PHBs. Oran [20] discusses QoS for named-based protocols and advocates a maximalist approach, with explicit flow state and resource reservation, to provide strong QoS guarantees.

VIII. Conclusion

In this paper, we propose a pure name-based QoS mechanism for name-based networks. Our approach leverages information encoded in the name of the data, which is a first-class identifier free from being tampered on-path. We present an NDN-based implementation, evaluated using ndnSIM with promising results. Our analysis shows the state entity required in our approach is no worse off than the IP-based approach while our approach has an advantage when it comes to the trust model of QoS mechanism, one of the hurdles of the IP-based QoS mechanisms today. Future work includes extending our name-based QoS approach to the caching mechanism, implementing and evaluating other FBs, developing a management protocol to distribute QoS policies, and conducting further empirical evaluation.

References

[1] L. Zhang *et al.*, "Named data networking," *SIGCOMM CCR*, vol. 44, no. 3, Jul. 2014.

[2] M. Mosko, I. Solis, and C. A. Wood, "Content-Centric Networking (CCNx) Semantics," RFC 8569, Jul. 2019.

[3] H. Asaeda, K. Matsuzono, Y. Hayamizu, H. Htet Hlaing, and A. Ooka, "A Survey of Information-Centric Networking: The Quest for Innovation," *IEICE Trans. Comm.*, Jan. 2024.

[4] D. Black, Z. Wang, M. Carlson, W. Weiss, E. Davies, and S. Blake, "An architecturefor diff. services," RFC 2475, Dec. 1998.

[5] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the Evolution of ndnSIM: An Open-Source Simulator for NDN Experimentation," *ACM SIGCOMM CCR*, vol. 47, no. 3, 2017.

[6] A. Custura *et al.*, "Exploring DSCP modification pathologies in mobile edge networks," in *IFIP TMA*, 2017.

[7] V. Jacobson *et al.*, "Networking named content," in *ACM CoNEXT*, Dec. 2009.

[8] V. Stocker, G. Knieps, and C. Dietzel, "The Rise and Evolution of Clouds and Private Networks – Internet Interconnection, Ecosystem Fragmentation," Rochester, NY, Aug. 2021.

[9] F. Baker, D. L. Black, K. Nichols, and S. L. Blake, "Definition of the DS field in IPv4 and IPv6 Headers," RFC 2474, Dec. 1998.

[10] J.-Y. L. Boudec, W. Courtney, J. Bennett, S. Davari, D. Stiliadis, K. Benson, V. Firoiu, B. S. Davie, and A. Charny, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246, Mar. 2002.

[11] W. Weiss, J. Heinanen, F. Baker, and J. T. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, Jun. 1999.

[12] R. Bless, "A Lower-Effort Per-Hop Behavior (LE PHB) for Differentiated Services," RFC 8622, Jun. 2019.

[13] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data link state routing protocol," in *Proc. SIGCOMM Workshop on ICN*, Aug. 2013.

[14] T. Endo, A. Yokotani, S. Ohzahata, R. Yamamoto, and T. Kato, "An Adaptive Bandwidth Reservation Method for a Content-Centric Network," in *IEEE CSAC*, vol. 02, Jul. 2018.

[15] A. Yokotani *et al.*, "A dynamic cache size assignment method with bandwidth reservation for CCN," in *Intl. Conference on Information Networking*, Jan. 2019.

[16] S. Shannigrahi, C. Fan, and C. Papadopoulos, "SCARI: A strategic caching and reservation protocol for ICN," in *Proc. ACM Conference on ICN*, Sep. 2018.

[17] T. Pan, C. Xu, J. Lv, Q. Shi, Q. Li, C. Jia, T. Huang, and X. Lin, "LD-ICN: Towards Latency Deterministic Information-Centric Networking," in *IEEE Intl. Conf. on Smart City*, Aug. 2019.

[18] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "MIRCC: Multipath-aware ICN Rate-based Congestion Control," in *Proc. ACM Conference on ICN*, Sep. 2016.

[19] C. Gündoğan *et al.*, "Gain More for Less: The Surprising Benefits of QoS Management in Constrained NDN Networks," in *Proc.' ACM Conference on ICN*, Sep. 2019.

[20] D. Oran, "Considerations in the development of a QoS architecture for CCNx-like ICN protocols," RFC 9064, Jun. 2021.