Investigating the Synergy between Routing and Forwarding Strategy in NDN Networks

Lan Wang, Alexander Lane
University of Memphis
{lanwang,awlane}@memphis.edu

Constantin Serban, Jesse Elwell
Peraton Labs
{cserban,jelwell}@peratonlabs.com

Alex Afanasyev
Florida International University
aa@cs.fiu.edu

Lixia Zhang
UCLA
lixia@cs.ucla.edu

ABSTRACT

Although multiple routing protocols and forwarding strategies have been proposed for NDN, there is a lack of understanding about the synergy between them. In this work, we investigate the effectiveness of NDN forwarding strategy, routing, and the combination of the two in maximizing data fetching success in the face of network failures. Through emulation experiments, we first evaluate the ASF (Adaptive SRTT-based Forwarding) strategy and NLSR (Named-data Link State Routing) protocol separately. Our results show that ASF with static routing outperforms NLSR in most cases, although its data delivery performance exhibits a bias toward popular producers. We then conduct experiments that combine ASF and NLSR. The results show that the combination of ASF with frequent probing and NLSR with slow routing adaptation leads to better data delivery performance than using either ASF or NLSR alone. Our results provide insights into the future design and deployment of routing and forwarding strategies in NDN networks.

CCS CONCEPTS

• **Networks** → **Network protocol design**; **Network design principles**; **Network layer protocols**.

KEYWORDS

Named Data Networking, Routing, Forwarding Strategy

ACM Reference Format:
Lan Wang, Alexander Lane, Constantin Serban, Jesse Elwell, Alex Afanasyev, and Lixia Zhang. 2023. Investigating the Synergy between Routing and Forwarding Strategy in NDN Networks. In *Proceedings of 10th ACM Conference on Information-Centric Networking (ICN '23).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A unique feature of the Named Data Networking (NDN) architecture is the *stateful forwarding* in its data plane [1, 6]. This feature enables NDN to support multi-path forwarding, i.e., providing multiple options in next-hop selection towards the requested data, while avoiding forwarding loops. NDN forwarding strategy performs next-hop selection based on application-defined policy and observed data-plane performance. However, there has been no comprehensive research investigating the integrated usage of forwarding strategies and routing protocols to optimize data retrieval success.

There is a spectrum of methods to discover best forwarding paths in NDN. At one end of the spectrum is the use of adaptive forwarding strategies, which discovers reachability to *specific* name prefixes by sending probes (i.e., replicating NDN Interest packets to send them to alternative faces) [17], in the absence of a routing protocol. At the other end of the spectrum is the use of routing updates to disseminate reachability information for *all* name prefixes without adaptive forwarding. Unfortunately, quick adaption to dynamic connectivity changes requires frequent routing updates, which can lead to high routing overhead. To address this issue, a middle ground can be reducing the frequency of routing updates and employing probing for faster discovery of new paths. This middle ground has a potential of both reducing routing overhead and achieving fast new path discovery upon network failures.

In this paper, we aim to gain an initial understanding of the spectrum between adaptive forwarding strategy and routing through experimentation. Specifically, we use ASF (Adaptive SRTT-based forwarding) [9] and NLSR (Named-data Link State Routing protocol) [5, 19] in this study, as both have been deployed in the NDN testbed for over seven years. It is important to note that many other routing protocols [7] and forwarding strategies [18] have been proposed for NDN. Instead of doing a comprehensive survey, our goal in this paper is to use ASF and NLSR as representative designs to investigate the synergy between routing and forwarding strategies.

Our work reported in this paper makes three contributions. **First**, we examine the advantages and disadvantages of using ASF and NLSR independently. We found that ASF combined with static routing can lead to more data delivery than NLSR combined with the best-route strategy in most cases and the former has lower overhead than the latter. **Second**, we investigate the impact of various network scenarios, such as producer popularity, content popularity, and link failures, on the effectiveness of ASF and NLSR. One major finding is that, with ASF, when the traffic distribution is highly skewed toward certain producers, the popular producers tend to experience improved data delivery performance but less popular producers tend to receive poorer performance, compared to their performance when traffic is uniformly distributed. However, NLSR does not exhibit such bias, although overall NLSR's performance is

---

This work was supported in part by DARPA. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.
ICN '23, October 8-10, 2023, Reykjavik, Iceland

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

worse than ASF. **Third**, we evaluate the performance of integrating NLSR and ASF as well as the resulting tradeoffs among multiple performance metrics. Our results show that the combination of ASF with frequent probing and NLSR with slow adaptation leads to better data delivery performance than using ASF or NLSR alone. We hope that our results can help guide network operators to configure routing protocols and forwarding strategies to satisfy different application requirements.

While our experiments provided new insights into the distinct roles of NLSR and ASF that may apply to similar routing protocols and forwarding strategies, this work represents just the initial step in quantifying the combined effects of NDN routing and forwarding strategies. Our approach focused on a specific routing protocol, forwarding strategy, and static topology with limited scale. Future investigations are required for different routing protocol and forwarding strategy designs, as well as various topology and traffic settings.

2 BACKGROUND

Named Data Networking (NDN) is a data-centric Internet architecture that fetches secured data by names [1, 6]. Each NDN *Interest* packet carries the name of the requested data, and fetches one *Data* packet back via the reverse path of the *Interest*. This two-way, symmetric packet flow enables routers to measure the data fetching performance at every hop, which helps them detect failures and select best forwarding paths adaptively. As a result, NDN relaxes the stringent requirements of *fast routing convergence* and *optimal path selection* for IP routing protocols. Nevertheless, routing protocols can still play an important role in an NDN network by helping forwarding strategies bootstrap forwarding paths and improving probing efficiency [21].

2.1 NLSR Routing Protocol

The Named-data Link State Routing protocol (NLSR) [5, 19] was developed in 2013, deployed on the NDN testbed in August 2014, and has been in operational use since then. NLSR adopts a *hierarchical semantic naming scheme* for routers, routing updates, and associated cryptographic keys. It uses a *hierarchical trust model* for validating routing updates in a single administrative domain, and uses NDN Sync [12] for fast and resilient routing information dissemination. NLSR supports multipath forwarding by calculating and ranking multiple forwarding options for each name prefix.

2.1.1 Adjacency Establishment and Maintenance. Neighbor NLSR routers, R_A and R_B, need to establish adjacency between them before they exchange routing information. R_A sends periodic *Hello Interests* with the name "`/<neighbor>/nlsr/INFO/<this-router>`" to each neighboring node to detect its status. If the neighbor router R_B responds to the Interest with a *Hello Data packet* signed using R_B's NLSR process key based on the configured trust model, R_B is considered up. If a *Hello* Interest to R_B times out without a response, R_A will retransmit the Interest rapidly for up to three times in case the Interest was lost. If there is no response from R_B during this period, the adjacency with R_B is considered down. R_A continues to send *Hello Interests* to R_B at a regular Hello interval to detect further changes in the adjacency status. Note that if the lower layer has the capability to detect and report link failure and recovery events, NLSR can use this information to update its adjacency status and recompute its routing table, which is usually faster and more efficient than using the Hello messages.

2.1.2 Dissemination of LSAs. Each router running NLSR collects connectivity to neighbors and reachability to name prefixes from received Link State Advertisements (LSA). Whenever an NLSR router establishes or removes an adjacency with a neighboring router, it disseminates to the entire network a new version of its *Adjacency LSA*, which contains all its *active* links, each associated with a neighbor router's name and a link cost. In addition, the router advertises locally reachable name prefixes from both statically configured and dynamic registered name prefixes. Whenever any name prefix is added or deleted, the router also disseminates a new *Name LSA* that contains all the locally reachable name prefixes. The latest versions of collected LSAs are stored in a Link State Database (LSDB) at each node. Upon receiving any new LSA, each router recalculates its routes and updates the forwarding table (FIB) if needed.

NLSR treats the LSA dissemination as data synchronization of the LSDBs among routers; its current implementation uses the *PSync* protocol [23] to synchronize the routers' LSDBs. PSync maintains the names of all the latest LSAs (including Adjacency LSAs and Name LSAs) in each LSDB as a name set and uses an Invertible Bloom Filter (IBF) of the name set as a compact expression of the set. Routers run PSync to detect and resolve any differences in the LSA name sets quickly and reliably. While PSync typically multicasts its Sync Interests to the entire network, NLSR limits the propagation of Sync Interests to direct neighbors only by prepending /localhop to the Sync name prefix. This approach allows PSync to aggregate state changes carried in Sync Interests thus reducing the number of Sync Interests in the network.

2.1.3 Multipath Support. Based on the information available in the Adjacency LSAs, each NLSR node builds a complete network topology. It then runs a simple extension of Dijkstra's algorithm to produce multiple next hops to reach each node. lNote that we require only the first hop to be disjoint among the computed paths, which is different from computing k-shortest paths with disjoint edges [20]. Based on the mapping from name prefixes to associated routers provided by the Name LSAs, the router can then obtain a ranked list of next hops to reach each name prefix.

2.2 ASF Forwarding Strategy

ASF (Adaptive SRTT-based Forwarding) was originally designed to mitigate the sub-optimal routes produced by Hyperbolic Routing [9]. It can also work with other routing protocols to adjust data forwarding paths in reaction to connectivity changes more promptly than routing protocols can do alone, with proper parameter settings as described below.

2.2.1 Path Measurements. Utilizing NDN's Interest-Data packet exchange, ASF measures the round trip time of Data packet retrievals in the following way: every time a Data packet is received, it takes a sample of the RTT and computes the Smoothed RTT (SRTT), an exponential moving average of the RTT samples, in the same way as TCP. ASF maintains one SRTT for each face in a FIB entry of a specific name prefix. If the Interests to a name prefix sent to a face experience timeouts for a configured number of times or they

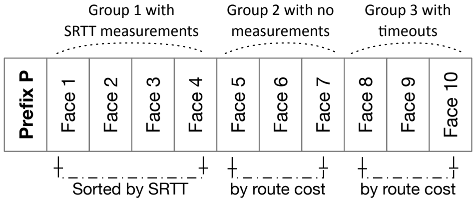


Figure 1: An example of how ASF ranks faces (i.e., faces) when forwarding an Interest

trigger a NACK in return, the face is marked with the *timeout* status for that name prefix. The Interest timeout value is calculated in the same way as the RTO in TCP. To avoid using stale information, ASF removes any SRTT estimate or timeout status that has not been updated for a configured period of time.

2.2.2 Ranking Faces for Interest Forwarding. ASF categorizes the faces in each FIB entry (per name prefix) into three groups (as illustrated in Figure 1): (a) Group 1 is comprised of faces that have recorded delay measurements, indicating successful data retrieval from them; (b) Group 2 is comprised of faces that have no measurements yet; and (c) Group 3 is comprised of faces that have been experiencing Interest timeouts. Upon receiving an Interest, ASF selects the face with the lowest SRTT in Group 1 to forward the Interest. If two faces have the same SRTT, routing costs serve as the tie breaker. If Group 1 is empty, ASF chooses the face with the lowest routing cost from Group 2 or, if Group 2 is also empty, the face with the lowest routing cost from Group 3. Group 2 takes precedence over Group 3, because faces that have not been tried may have a higher chance to work than faces that have experienced timeouts. It is important to note that the above describes the algorithm for choosing a face for *forwarding an Interest*; it is not the algorithm for choosing a face *to probe* (see Section 2.2.4).

2.2.3 When to Send Probes. With fluctuations in network conditions over time, the shortest path can also undergo changes, the shortest path may change, i.e., an alternate face (H') may deliver better performance than the face currently in use (H). However, it is essential to try H' in order to assess its performance. To this end, ASF periodically probes the faces that are not currently in use for each active name prefix with Interests for forwarding. A probing Interest is a copy of the original Interest carrying a different nonce. It is sent to an alternate face (H') that is not the primary face (H) used to forward the original Interest; carrying a different nonce ensures that it will not be dropped by an intermediate node as a duplicate (the loop detection mechanism will drop an Interest if it has the same nonce as one that has been forwarded before). If the probed path works, data will be returned on this path, perhaps in addition to the primary path where the original Interest is sent, and ASF will take an RTT sample of the probed path. To control the probing overhead, ASF sets a minimum time interval between adjacent probes; this defaults to 60 seconds, a value chosen for stable infrastructure network environments such as the NDN testbed. Networks with higher connectivity dynamics, e.g., wireless mobile networks, may desire to choose shorter probing intervals. We adjust the ASF probing interval in our experiments to evaluate its effect on data retrieval performance (see Section 3.2).

Parameter	Values
Producer Popularity	Uniform, Zipf(1), Zipf(2)
Content Popularity (measured by Content Request Overlap)	0%, 50%, 100%
Consumer Traffic Rate	10 Interests per second
Link Failure Model	Pareto / 1000s up / 120s down, Exponential / 30s up / 10s down

Table 1: Network Environment Parameters

Whenever an Interest is received by a node, the Interest is first sent to the primary face selected using the ranking algorithm described in Section 2.2.2. The node then checks if probing is due, and if so, a probe is sent to an alternate face selected by the ranking algorithm described in Section 2.2.4. As long as Interests for a given name prefix continue to arrive, probing will continue for that name prefix even if all the faces have been probed previously. Probing stops when Interests stop arriving for that name prefix.

2.2.4 Ranking Faces for Probing. Recall that ASF divides the faces of each FIB entry into three groups (see Section 2.2.2). In order to quickly acquire information about a face lacking measurements, ASF probing starts with the face F_{probe} in Group 2 with the lowest routing cost. If F_{probe} returns data, it is moved to Group 1. Otherwise, if the Interest sent to F_{probe} times out for a pre-configured number of times or if it returns a NACK, it is moved to Group 3. When Group 2 becomes empty, ASF will probabilistically choose a face from Group 1 and Group 3 to probe. The faces in Group 1 and 3 are ranked by SRTT and routing cost, respectively, while Group 1 has higher ranking than Group 3. The probing probability of the face that has rank $i = 1, \ldots, N$ in the sorted list (1 being the highest ranking) is

$$P(i) = 2 \times \frac{N + 1 - i}{N(N + 1)}. \tag{1}$$

In this way, faces that performed better previously are more likely to be probed, thus allowing the strategy to revert back to a better performing path.

Note that when the routing protocol updates the routing cost of a face for a name prefix in the FIB, the forwarding ranking and probing ranking in ASF are updated accordingly.

3 EXPERIMENT DESIGN

We divide our experimental study into two steps. The first step measures the performance of NLSR and ASF individually under various network conditions to identify their respective strengths and limitations. The second step investigates how to integrate NLSR and ASF to achieve the best possible performance by leveraging their strengths. In the rest of this section, we first identify a list of experimental environment parameters related to network and application traffic characteristics, then examine the protocol tuning knobs in NLSR and ASF that affect their adaptivity and protocol overhead, and finally describe our experiments setup and performance metrics.

3.1 Network Environment Parameters

Table 1 summarizes the network environment parameters in our experiments. Below we describe them in detail.

Producer Popularity Traffic from consumers may be uniformly distributed among all the producers or more skewed toward some popular producers. Previous measurement studies have shown that web traffic follows the Zipf distribution [8]. Under the Zipf distribution with an exponent s for N producers, a producer with the popularity rank of k receives a portion of the total traffic equal to $(1/k^s)/(\sum_{i=1}^{N} 1/i^s)$. To study the impact of this factor, we experiment with three distributions for producer popularity: (a) Uniform distribution, (b) Zipf(1), i.e., Zipf with $s = 1$, representing a typical traffic distribution, and (c) Zipf(2), i.e., Zipf with $s = 2$, representing an extremely skewed traffic distribution.

Content Popularity In addition to provider popularity, in some applications, multiple consumers may fetch the same content generated by the same producer. The exact degree of overlapping between consumer requests depends on the composition of consumers, occurrence of major news/events, and other factors. Since NDN supports in-network caching, different degrees of consumer request overlapping have a direct impact on the observable data fetching performance. Our experiments measures the performance of NLSR and ASF with 0%, 50%, and 100% **consumer request overlaps**.

Traffic Volume Consumers may fetch data at different rates based on the applications' needs. Since a router running ASF probes alternative paths for a name prefix N whenever it sees NDN Interests toward prefix N, the number of ASF probes to N is directly influenced by the Interests arrival rate (up to a limit, as described in 2.2.3). If Interests heading to prefix N arrive far apart, for a name prefix N, then few ASF probes are sent for that name prefix N. When Interests arrive at a rate higher than the probing rate, the probes sent will be limited by the probing rate. In our experiments, each consumer sends 10 Interests per second distributed randomly either via uniform or Zipf distribution to all the producers, emulating applications of a moderate packet rate which is faster than regular GPS location updates but slower than typical video applications.

Link Failure Model Both NLSR and ASF can adjust forwarding paths upon detecting link failures. To measure their performance under different frequency and duration of link failures, we experiment with the following two link failure models: (a) a low frequency failure model with a 1000sec mean-time-to-fail (uptime) and 120sec mean-time-to-recover (downtime), both following a shifted Pareto distribution. This model is based on a previous measurement study [10]. We set the scale parameter of the shifted Pareto distribution to 208 so that 50% of the failures last less than one minute, and (b) a high frequency failure model with 30sec mean-time-to-fail (uptime) and 10sec mean-time-to-recover (downtime), both following an exponential distribution. Note that these two sample failure models are used to examine NLSR and ASF performance; they are not necessarily representatives of all possible failure patterns.

Network Topology We use a network topology with 20 nodes and 41 links (Figure 2) based on a snapshot of the early NDN testbed [13] so that we may have a realistic network topology. Each link has no upper-bound on its bandwidth, and its propagation delay is set to the average delay between the corresponding testbed nodes measured using `ping`. Together with the link failure models described above, our experimental setting models *stationary but failure-prone networks*. We will study networks with mobility in our future efforts.

Failure Model	ASF Probing Interval and NLSR Hello Interval
Pareto/1000s up/120s down	30s, 60s, 120s, 180s, 210s
Exponential/30s up/10s down	2.5s, 5s, 10s, 15s, 17.5s

Table 2: Protocol Tuning Knob Values

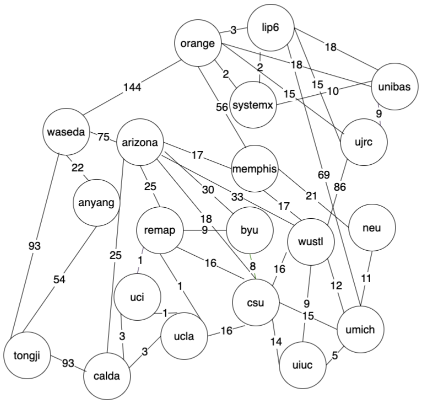


Figure 2: 20-Node Network Topology

3.2 Protocol Tuning Knobs

ASF Probing Interval As explained in Section 2, for each name prefix that has active Interest traffic, ASF selects an alternative face to probe periodically. The probing interval determines the delay in detecting a link failure or recovery, as well as the overhead of ASF. Our experiments tries five probing interval values – 25%, 50%, 100%, 150%, and 175% of the mean-time-to-recover which is 120s in the more stable model and 10s in the more unstable model (Table 2). We use the same ASF strategy parameters for all name prefixes.

NLSR Hello Interval Each NLSR router sends periodic Hello Interests to its neighbors to monitor the status of its links and neighbors. To compare NLSR and ASF's performance fairly, they should have the same reactiveness to link failures and recoveries. Therefore, we set the Hello interval to the same five values as those for the ASF probing interval (Table 2).

3.3 Experiment Setup

We utilize Mini-NDN [11] to conduct our experiments. Mini-NDN is a lightweight networking emulation tool that supports NFD, NDN libraries, NLSR, and NDN applications without modification. We run NFD and NLSR in applicable cases on each node,[1] and use the consumer and producer programs from `ndn-traffic-generator` [14] to drive the evaluation. Every producer serves data under a single name prefix, while every consumer has a preconfigured starting sequence number for every producer and sends a stream of NDN Interests at a preconfigured rate. Each Interest is sent to a randomly chosen producer based on producer popularity, and the Interest name is a concatenation of the producer's name prefix and the next sequence number that the consumer has not used for the particular producer. Two consumers' Interests and Data packets "overlap" if

---

[1] NLSR uses a patch of PSync [4] that fixes a few identified performance issues.

they have the same starting sequence number for the same producer. All the consumers are started at the same time. The Interest size ranges from 44 to 54 bytes. The Data packet size is either 1206 bytes or 1208 bytes depending on the data name length. Note that we do not set link bandwidth in our experiments, so the packet size does not affect transmission delay.

Our experiments start collecting measurement results 60 seconds after the NLSR processes on all the nodes start to ensure that all routers have an established FIB. Afterwards, each experiment lasts 3000 seconds for the Pareto/1000s/120s failure model and 1000 seconds for the Exponential/30s/10s failure model. For ASF experiments without NLSR, we use a centralized routing module in Mini-NDN to calculate the routes based on the network topology and statically configure the FIB entries at the beginning of the experiments.

3.4 Performance Metrics

We use the following three performance metrics:

- **Interest Satisfaction Ratio:** the percentage of Interests from consumers that successfully retrieve the matching Data packets.
- **Overhead:** ASF overhead includes probe Interest packets; NLSR overhead includes Hello, PSync, and LSA Interest and Data packets.
- **RTT:** the duration from when a consumer sends an Interest to when it receives the matching Data packet.

We run each experiment five times with difference random seeds and calculate the mean of the Interest satisfaction ratio and overhead from the five runs. The RTT measurements from the five runs are summarized using their quantiles.

4 RESULTS

In this section, we report the network forwarding performance in two steps. First, we present a baseline data delivery performance achieved by static routing without dynamic routing or adaptive forwarding (Section 4.1). Second, we measure (i) the performance of ASF in combination with static routing, and (ii) the performance of NLSR dynamic routing in combination with the Best-Route forwarding strategy which picks the highest-ranked face in each FIB entry to forward Interests. Here we measure the performance and overhead with ASF and NLSR operating separately, and discuss the impacts of traffic patterns and link failures (Section 4.2). Finally, we present the performance of combining NLSR and ASF, along with the tradeoffs between different protocol parameter settings (Section 4.3).

4.1 Baseline: Static Routing Performance

In this set of experiments, we use static routing along with the Best-Route forwarding strategy to obtain a baseline performance for later comparison. We use Dijkstra's algorithm to compute the shortest paths and pre-fill the FIBs of all routers before starting each experiment, which remain static throughout the emulation run. Table 3 summarizes the eight scenarios that combine two producer popularity distributions (Uniform and Zipf(2)), two consumer request overlap settings (0% and 100%) which measure the percentage of overlap in sequence number range among the consumers, and two link failure models (Pareto/1000sec Up/120sec Down, and Exponential/30sec Up/10sec Down) as described in Section 3.1.

No	Failure Model (Distribution/Uptime/Downtime)	Producer Popularity	Consumer Request Overlap
1	Pareto/1000s/120s	Uniform	0%
2	Pareto/1000s/120s	Zipf(2)	0%
3	Exponential/30s/10s	Uniform	0%
4	Exponential/30s/10s	Zipf(2)	0%
5	Pareto/1000s/120s	Uniform	100%
6	Pareto/1000s/120s	Zipf(2)	100%
7	Exponential/30s/10s	Uniform	100%
8	Exponential/30s/10s	Zipf(2)	100%

Table 3: Parameters for Scenarios in Section 4.1 and 4.2.

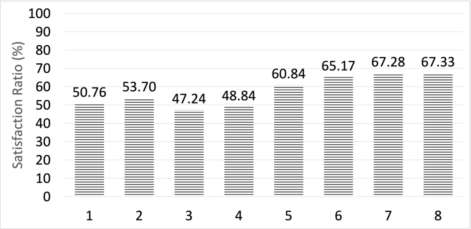


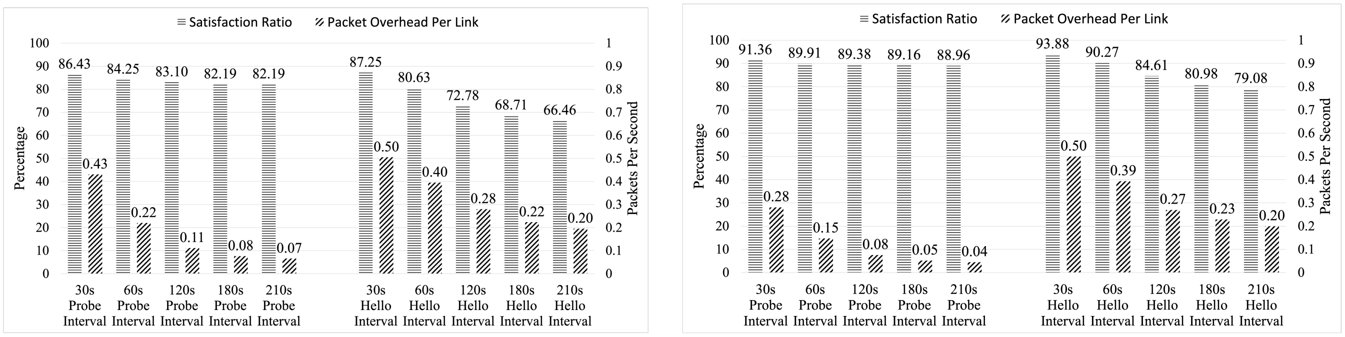
Figure 3: Interest Satisfaction Ratio from Static Routing and Best-Route Strategy for Traffic Scenarios in Table 3

Figure 3 shows that the Interest satisfaction ratio ranges from 47.24% to 67.33% with static routing in the eight scenarios. Moreover, the Interest satisfaction ratio is higher with 100% overlap among consumer requests (scenarios 5-8) compared with the cases of no overlap (scenarios 1-4), due to consumers' Interests fetching data from routers' content stores in the former case. **The benefit from caching is more prominent in the scenarios with higher dynamics**. When the link instability goes up, the absolute increase of Interest satisfaction ratio is 19.27% on average when comparing scenarios 7 and 8 with scenarios 3 and 4. On the other hand, comparing scenarios 5 and 6 with scenarios 1 and 2 shows that the absolute increase is only 10.77% when the links are more stable.

In all the above scenarios, **the Interest satisfaction ratio is inadequate for most applications**. Next we examine how much dynamic routing and adaptive forwarding can help improve the performance.

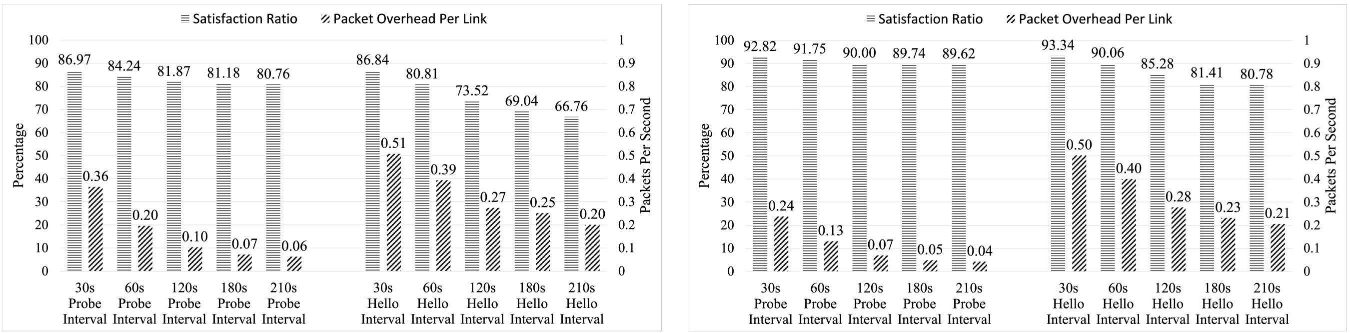
4.2 Running ASF and NLSR Separately

This section reports the performance evaluation from running ASF and NLSR separately. In the ASF case, all the router FIBs are configured with static routes at the beginning of each experiment. ASF uses both the FIB ranking and its own measurements to choose the best face in forwarding each Interest (Section 2.2). The NLSR evaluation chooses the highest ranked faces (in the FIB) to forward Interests, i.e. the Best-Route forwarding strategy. To understand the tradeoffs with different protocol parameters, we use five ASF probe interval values and five NLSR Hello interval values ranging from 25% to 175% of the mean-time-to-recover (i.e., link downtime). Figure 4 and 5 show our results.



(a) Uniform Producer Popularity and No Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)

(b) Uniform Producer Popularity and 100% Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)



(c) Zipf(2) Producer Popularity and No Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)

(d) Zipf(2) Producer Popularity and 100% Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)

Figure 4: ASF vs NLSR Performance under "Pareto/1000s Up/120s Down" Link Failure Model

4.2.1 Main Observations. **First**, in all the eight scenarios, as the ASF probe interval and NLSR Hello interval increase, both the corresponding Interest satisfaction ratio and overhead decrease. Comparing the performance between the highest and lowest probe/Hello frequency, the relative difference between the highest and lowest Interest satisfaction ratio is 12.2% for ASF and 20.9% for NLSR. Longer intervals for ASF probes and NLSR Hellos lead to slower detection of link failures/recoveries, which in turn increases the likelihood of selecting routes that are no longer functional, leading to decreased satisfaction ratio.

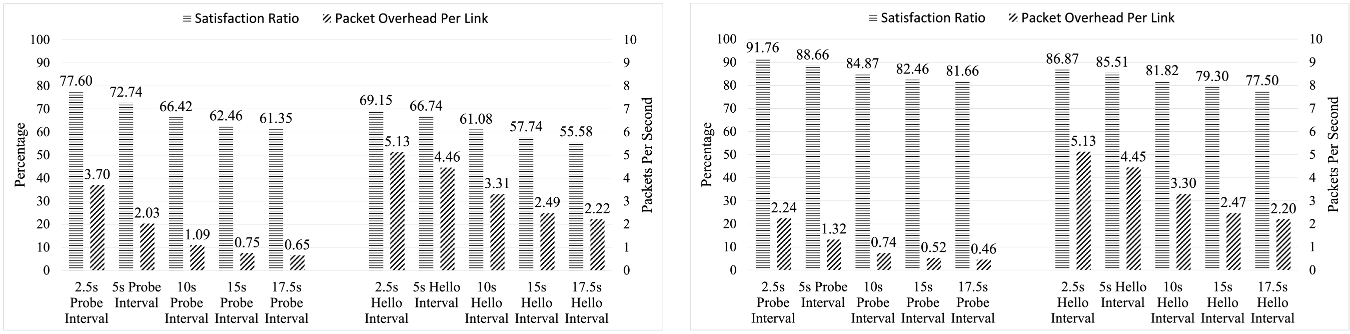
We note that when the probe/Hello interval is reduced by a factor of seven, the overhead is increased by 440.1% for ASF and 141.5% for NLSR, respectively. We can use ASF's design to explain its big overhead increase: *each router* probes one alternative path for every active name prefix, with the frequency of probing Interest upper-bounded by the probe interval. Thus, the total overhead is the product of probe interval, the number of active name prefixes, and path lengths. On the other hand, NLSR's overhead includes Hello, PSync, and LSA packets. Hello packets are sent periodically over every link, so the Hello overhead grows linearly with the Hello interval. The LSA overhead depends on how often links fail/recover and how fast Hellos can detect link failures/recoveries. The PSync overhead depends on the frequency of routing update publications (i.e., new LSAs generated by NLSR), with a lower-bound determined by the Sync Interest interval (30sec by default in NLSR).

Second, Interest satisfaction ratio shows considerable improvement over static routing. Averaging over all the eight scenarios, the least absolute improvement over static routing is 20.9% for ASF and 12.3% for NLSR, when both ASF probe and Hello intervals are the longest (175% of the mean downtime). Conversely, the biggest absolute improvement is 29.5% for ASF and 26.5% for NLSR, when the probe and Hello intervals take their lowest value (25% of the mean downtime).

Third, when the ASF probe interval and NLSR Hello interval are equal, NLSR has 246.9% higher overhead than ASF on average and ASF has 7.9% higher Interest satisfaction ratio than NLSR on average. In three of the four cases when the Hello/probe interval is 30s, NLSR has higher Interest satisfaction ratio than ASF, with considerably higher overhead than that of ASF. Further investigations are needed to understand NLSR's higher interest satisfaction in these cases.

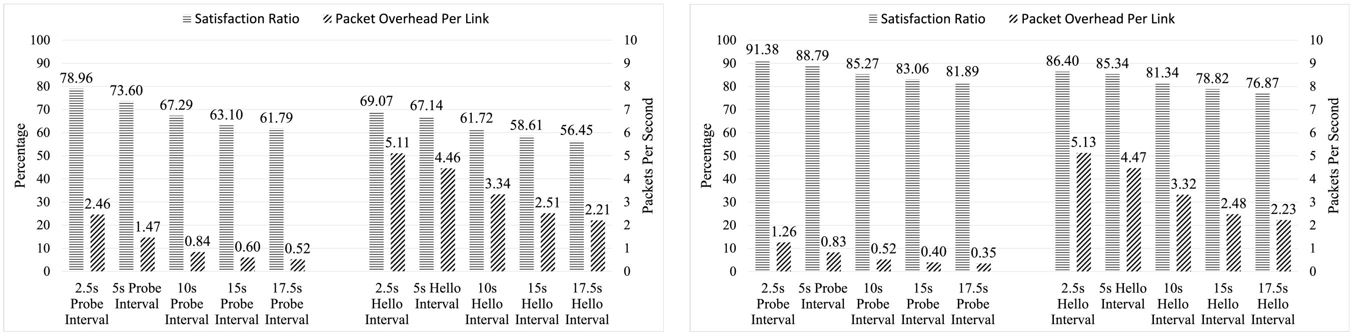
4.2.2 Effect of Network Environment Factors.

Effect of Consumer Request Overlap For both ASF and NLSR, the Interest satisfaction ratio increases as the degree of consumer request overlap increases, because an increasing number of consumers' Interests are satisfied by router caches instead of by the producers. When Interests hit caches, both the Interests and Data packets travel fewer hops, making them less likely to be dropped due to link failures. In addition, shortened Interest paths also reduce the number of ASF probes triggered by the Interests, leading to lower ASF overhead. In contrast, NLSR packets are not triggered by data plane traffic flows, so caching of consumer data has no impact on its overhead.



(a) Uniform Producer Popularity and No Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)

(b) Uniform Producer Popularity and 100% Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)



(c) Zipf(2) Producer Popularity and No Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)

(d) Zipf(2) Producer Popularity and 100% Consumer Request Overlap Scenario (ASF: left half; NLSR: right half)

Figure 5: ASF vs NLSR Performance under "Exponential/30s Up/10s Down" Link Failure Model

Effect of Producer Popularity Distribution Across all scenarios, ASF demonstrates a slightly lower average probe overhead when the producer popularity follows the Zipf(2) distribution as opposed to the Uniform distribution. This is because ASF probes are triggered by consumer Interests. With the Zipf(2) distribution, the number of probes to less popular prefixes is lower as they are triggered less often, while the number of probes to the popular prefixes is still limited by the configured probing interval. Another observation is that the average Interest satisfaction ratio of ASF increased slightly in most cases from the Uniform to the Zipf(2) popularity distribution. When the traffic is concentrated on a few popular producers, these producers' name prefixes get probed more frequently, making their reachability status more up to date as compared to the Uniform traffic distribution scenario. At the same time, less popular name prefixes get fewer probes, resulting in fewer updates to their reachability status, hence lowering their satisfaction ratio. The increase in satisfaction ratio of the popular producers is offset by the decrease in satisfaction ratio of the unpopular prefixes, leading to only moderate or no increase in the overall Interest satisfaction ratio. On the other hand, NLSR propagates reachability information for every name prefix equally, so its satisfaction ratio and overhead are not affected by the producer popularity distribution. To validate the the aforementioned reasoning, we computed the difference in Interest satisfaction ratio for individual producers between the Zipf(2) and Uniform distributions. With ASF and the Zipf(2) distribution, the most popular producers exhibited a higher satisfaction ratio and the majority of the less popular producers experienced a lower satisfaction ratio. In contrast, there is no such trend with NLSR. We also discovered two more findings. First, a few of the less popular producers did not experience a reduced satisfaction ratio with the Zipf(2) distribution. This is attributed to their infrequent transmission of Interests. When they do send Interests, these Interests trigger probes along the route, thereby increasing the likelihood of successful data retrieval. Second, the Zipf(2) producer popularity distribution amplifies the effect of caching as more Interests are fetching the same content. More specifically, when the consumer request overlap is 100%, the popular producers experienced an even larger increase in Interest satisfaction ratio compared to when there is no consumer request overlap.

Effect of Link Stability We have two link failure models: the Pareto model with 1000s uptime and 120s downtime, and Exponential model with 30s uptime and 10s downtime. Comparing Figure 4 and 5, we can see that the ASF/NLSR overhead is much higher in the more dynamic network, because a much higher probe/Hello frequency proportional to the failure/recovery frequency is needed to detect failures and recoveries in the more dynamic network. Even though the overhead is an order of magnitude higher, the Interest satisfaction ratio is still lower in the more dynamic network compared with the more stable network.

4.3 Combining NLSR and ASF

The previous section showed that, when ASF and NLSR are used separately, ASF can achieve a higher Interest satisfaction ratio than

NLSR in most cases and it has lower overhead than NLSR. Moreover, when we compared the performance under the uniform traffic distribution with that under the Zipf(2) distribution, we found that ASF can improve performance for traffic to popular producers but may lead to worse performance for less popular ones, whereas NLSR does not exhibit such bias (see "Effect of of Producer Popularity Distribution" in Section 4.2). We also remind the reader that the ASF evaluation starts with statically configured FIBs at all routers with ranked routes, so that ASF can use this information to bootstrap its forwarding and probing decisions before it obtains its own data plane measurements. In real networks, statically configuring FIBs is usually impractical and cannot handle addition and removal of network nodes and links. Therefore, we need to run dynamic routing in conjunction with adaptive forwarding strategies. In this section, we combine NLSR and ASF to leverage their strengths, and evaluate the combined performance.

4.3.1 Network and Protocol Parameter Settings. For this set of experiments, we use 50% consumer request overlap and Zipf(1) producer popularity distribution to emulate more realistic network scenarios instead of the more extreme scenarios in the previous experiments. To make 50% of consumer requests overlap, we divide the 20 nodes into two groups of 10 consumers each (the group assignment is generated based on the experiment seed, so it is different in runs with different seeds). We let the consumers in each group start with the same data sequence number for each producer, so the content they fetch from each producer overlaps. However, there is no overlap in the fetched content between the two groups. *Since ASF achieves relatively good performance with static routing, we can run NLSR with longer Hello intervals (100%, 150% and 175% of the mean downtime) to supply ASF with ranked routes, and at the same time avoid the high overhead associated with shorter Hello intervals.* ASF can use shorter probe intervals (25% and 50% of the mean downtime) to adapt to high network dynamics. This results in six combinations of protocol parameter settings for each failure model.

4.3.2 Observations. We can make the following observations from Figure 6 and 7.

First, in most cases, combining NLSR and ASF leads to higher Interest satisfaction ratio and lower median delay compared to using NLSR with Best-Route strategy or ASF with static routing. However, the range of RTTs increases in some cases, which may be due to the combined NLSR/ASF finding longer paths to satisfy those Interests that are not satisfied using either NLSR or ASF.

Second, Figure 6 shows that, while keeping the NLSR Hello interval constant, shorter ASF probe intervals increase the Interest satisfaction ratio, ASF overhead, and overall overhead. Similarly, while keeping the ASF probe interval constant, shorter NLSR Hello intervals increase the Interest satisfaction ratio, NLSR overhead, and overall overhead.

Third, Figure 7 shows that shorter ASF probing intervals result in lower median and range of the RTTs. This effect is more prominent in the Exponential failure case compared with the Pareto failure case. Shorter NLSR Hello intervals also result in slightly lower median RTTs in the Exponential failure case.

Fourth, the difference between the highest and lowest Interest satisfaction ratio is small - 2.73% in the Pareto failure case (Figure 6a), and 4.57% in the Exponential failure case (Figure 6b).

Finally, the difference between the highest and lowest overhead of the NLSR-ASF combinations is small in absolute terms, but high in relative terms - 0.22 packets per second or 61.1% in the Pareto failure case (Figure 6a) and 2.09 packets per second or 58.2% in the Exponential failure case (Figure 6b).

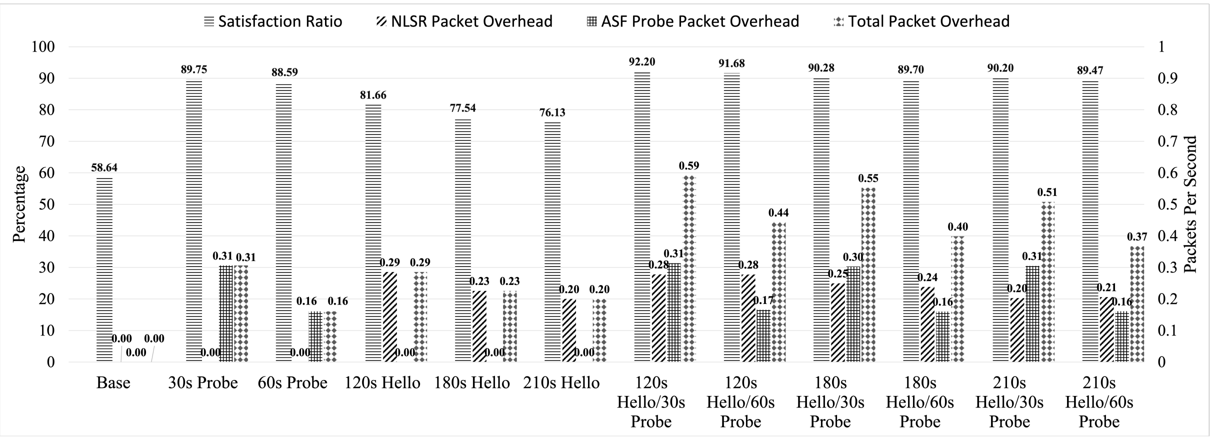
4.3.3 Implications. The above results suggest that getting a small increase in the Interest satisfaction ratio may require substantial increase in the NLSR/ASF overhead. However, this may be a necessary tradeoff for loss-sensitive applications. In addition, reducing the ASF probe interval can lower the RTTs considerably, this may be desirable if the application is delay sensitive. On the other hand, if the application is not sensitive to loss or delay, then a parameter combination with lower overhead may be a good choice. For example, when the NLSR Hello interval is 175% of the mean downtime and the ASF probe interval is 50% of the mean downtime, NLSR/ASF can achieve an Interest satisfaction ratio of 89.47% in the more stable network and 84.42% in the higher dynamic network. In other words, one may use application requirements to guide the selection of the right parameter combination for NLSR and ASF.

As a network routing protocol, NLSR should tailor its Hello interval for specific applications. However, ASF parameters can be configured for individual name prefixes. Therefore, the network operator can choose an NLSR Hello interval to satisfy the majority of applications, and set the ASF parameters for specific name prefixes based on the requirements of corresponding applications. In this paper, we experimented with a relatively long Hello interval for NLSR, combined with a relatively short probe interval for ASF due to two reasons. First, NLSR incurs much higher overhead than ASF while its satisfaction ratio is lower than that of ASF. Second, as it is currently designed, ASF still requires the guidance of a routing protocol to bootstrap its face ranking. We have shown that, in a stationary network, the combination of ASF and static routing works reasonably well, so NLSR can simply play the role of a static routing protocol to assist ASF. In other network scenarios, such as wireless ad-hoc networks, with distinct application requirements, we may need a different combination, e.g., the routing protocol may need to react to topological changes more quickly and the forwarding strategy can be less adaptive at least for those name prefixes with less stringent delay and loss requirements. Of course, such customization is only feasible when we have a good understanding of the combined routing and adaptive forwarding performance in various network environments.

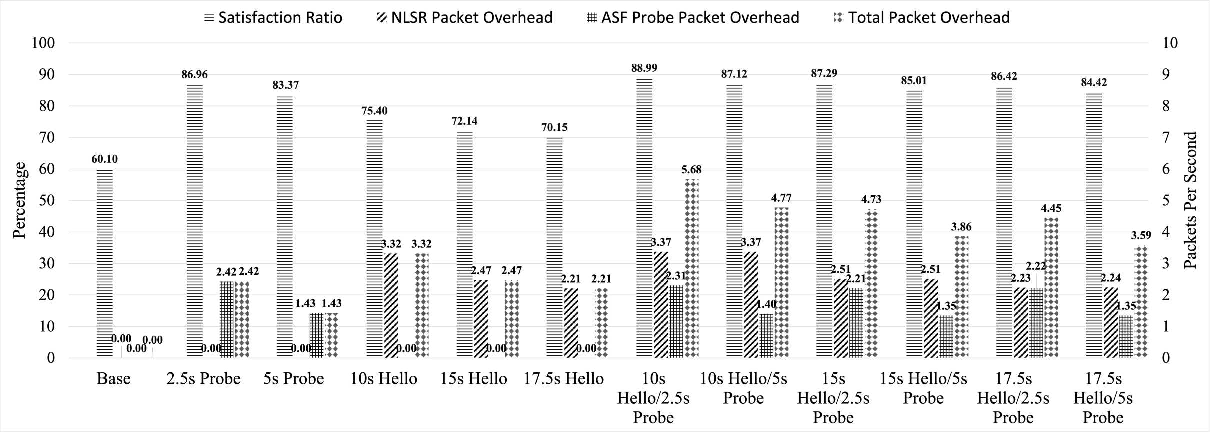
5 RELATED WORK

There have been a number of efforts contributing to the routing and forwarding area of NDN networking research [2, 7, 18]. Here we review several representative pieces most relevant to our work.

One early effort by Yi et al. [22] describes an initial design of NDN's forwarding plane that takes input from routing and measured data plane performance. Different from ASF which *proactively* probes alternative paths, the forwarding strategy in [22] tries alternative paths only *after* detecting failures of the paths in use. Yi et al. [21] examined the relationship between the stateful forwarding plane and routing protocols in NDN networks, and showed evidences that, while the former enables individual NDN routers to detect and recover from network failures without reliance on



(a) "Pareto/1000s Up/120s Down" Link Failure Model



(b) "Exponential/30s Up/10s Down" Link Failure Model

Figure 6: Interest Satisfaction Ratio and Overhead in Baseline, ASF-Only, NLSR-Only, NLSR/ASF Cases

external inputs, the latter remains advantageous as it disseminates the initial topological connectivity and policy information, as well as their *long-term* changes, to guide the overall forwarding process. Our work took a second step in the same direction by conducting experimentations to *quantitatively* evaluate the performance gains from combining NLSR and ASF under different parameter settings.

Another approach to making forwarding decisions is to apply various probabilistic techniques. Posch et al. [15] proposed Stochastic Adaptive Forwarding (SAF) that borrows concepts from a self-adjusting water pipe system to optimize forwarding performance. Specifically, it uses unsatisfied interests as an implicit feedback mechanism to detect "overpressure", and probabilistically drops Interests that are likely to be unsatisfied. Qian et al. [16] introduced another probability-based adaptive forwarding strategy (PAF) that applies ant colony principles to compute forwarding path selection probabilities. PAF sends both probes triggered by incoming Interests and probes independent of incoming Interests. It then uses recorded measurements as input to path selection probabilities. The probability-based approaches may forward Interests for similar prefixes over different routes, which may be considered positive (load-balancing) or negative (increased jitter) depending on the application scenario.

An alternative to conventional routing/forwarding approaches is to apply various machine learning (ML) techniques to optimize forwarding paths, an approach explored in DQN-AF [3], AFSndn [24], and IFS-RL [25]. ML approaches may provide better forwarding decisions, but can require significantly higher processing and storage capabilities, thus potentially limiting their applicability.

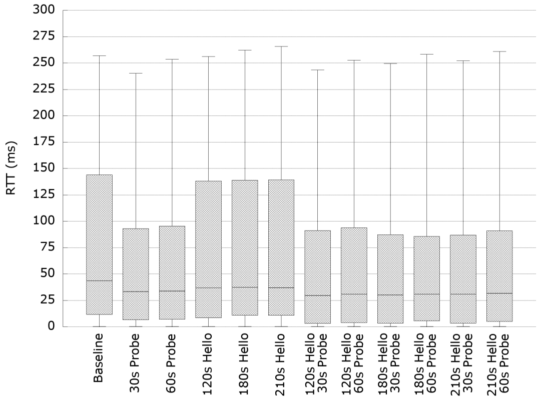
Our objective is to study the integrated usage of routing and forwarding strategies, which differs from the previous work that investigates individual routing protocols or forwarding strategies separately.

6 DISCUSSIONS

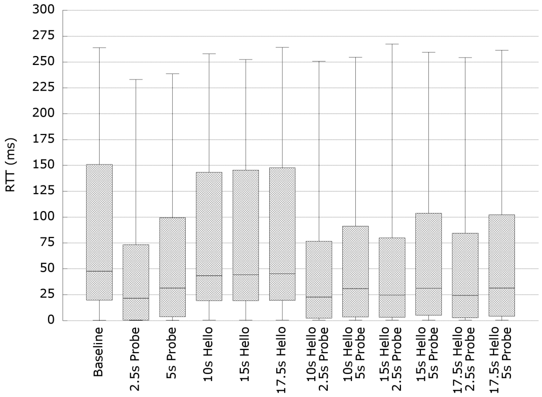
Having presented our experimental results, we first clarify the fundamental differences in network failure recovery between an IP and an NDN network, and then discuss the limitations of our work.

6.1 Failure Detection and Recovery: IP vs NDN

As a node-centric network design, IP relies on routing protocols to direct packets to their destinations. An IP node N_{ip} relies on the routing protocol to detect the failure of the next hop neighbor and compute new paths, so that it can forward future packets along the new paths to their destinations. Even if the lower layer hardware



(a) "Pareto/1000s Up/120s Down" Failure Model



(b) "Exponential/30s Up/10s Down" Failure Model

Figure 7: RTT in Baseline, ASF-Only, NLSR-Only, NLSR/ASF Cases (each boxplot shows 5th percentile, 1st quartile, median, 3rd quartile, and 95th percentile)

can inform N of an interface failure, N still relied on routing to find new paths after the failure. Current practice often pre-computes backup paths for fast switch over.

Equipped with a stateful forwarding plane, NDN handles failures in a fundamentally different way. Each NDN node N_{ndn} measures the fetching result of *every data packet*. When equipped with the ASF strategy, (i) N_{ndn} probes proactively to identify alternative working paths for each active name prefix; (ii) N_{ndn} ranks all working faces based on the results from both data fetching and probing; and (iii)N_{ndn} chooses the best face for every outgoing Interest, which could be different from the face used for the previous Interest under the same name prefix. In short, NDN's forwarding plane can detect and mitigate failures on its own effectively, as opposed to IP's sole reliance on routing protocols.

6.2 Limitations of This Work

Our experimentations exposed new insights into the different roles played by NLSR and ASF strategy, such as the results reported in Section 4.3, which show that ASF's probing discovers alternative working paths for active prefixes and improves performance for *popular* prefixes, but NLSR covers *all* prefixes and offers the same performance whether a prefix becomes popular or not. This observation should be applicable to similar routing protocols and forwarding strategies. Moreover, our methodology can be used to investigate other routing protocols, forwarding strategies, and other scenarios such as mobile networking.

However, we emphasize that this work represents only the initial step in quantifying the combined effects of NDN routing and forwarding strategies. To scope the effort of this first step, we take a "slow start" approach, by using a specific routing protocol, a specific forwarding strategy, and a static topology model with a relatively small size without bandwidth limitation. Significant future investigations are needed to evaluate the results from other routing and strategy designs, and from different topology settings.

In particular, the scalability of probing deserves more investigation. At this time, the ASF strategy proactively probes active prefixes that have traffic flowing. If the traffic distribution is skewed toward popular prefixes, probing active prefixes only can be more efficient than routing in brining performance gains, as shown in the paper; however we need to further quantify the effectiveness by the degree of traffic distribution skewness. More work is also needed to evaluate the cost and gain of proactive probing in various scenarios, for example whether proactive probing is necessary when the current face in use is working, i.e., to understand the cost of identifying optimal paths and the gain from doing it.

Furthermore, this work also took NLSR as given, even though its design has known options for further improvement. One of these options is to utilize forwarding plane traffic to reduce NLSR overhead: since each Interest-Data packet exchange provides definitive information on the link's status, this information can be used to set the link status, removing the need for NLSR Hello exchange (which is needed only in the absence of actual user traffic).

7 CONCLUSION

This work is an initial attempt to understand the performance of NDN routing and adaptive forwarding both independently and together. We found that, in most cases, ASF with static routing outperforms NLSR with the best-route strategy but it has a bias toward more popular name prefixes. We also evaluated the performance of combined NLSR/ASF, and showed the tradeoffs among Interest satisfaction ratio, RTT, and overhead with different parameter settings. Our results can help operators configure NLSR and ASF for their networks based on application requirements.

Our work is by no means comprehensive. For the immediate next step, we plan to evaluate NLSR/ASF performance using network topologies with different sizes, connectivity characteristics, mobility models, and bandwidth limits. In particular, we want to find out whether ASF can avoid synchronous path changes and oscillations when some links get congested, which is a well known problem when routing protocols use traffic load in path selections. We also plan to use analytical modeling to validate our experimental results. Furthermore, we will leverage our gained understanding to improve NLSR and ASF. For example, we plan to use data plane information to reduce NLSR hello protocol overhead and investigate different approaches to scaling ASF probing without reducing its effectiveness. Finally, we will compare NDN and IP's forwarding performance under the same link failure and traffic scenarios.

REFERENCES

[1] Alexander Afanasyev, Tamer Refaei, Lan Wang, and Lixia Zhang. 2018. A Brief Introduction to Named Data Networking. In *Proc. of MILCOM*.

[2] Narjes Aloulou, Mouna Ayari, Mohamed Faten Zhani, Leila Saidane, and Guy Pujolle. 2017. Taxonomy and comparative study of NDN forwarding strategies. In *2017 sixth international conference on communications and networking (ComNet)*.

[3] Ygor Amaral B. L. de Sena, Kelvin Lopes Dias, and Cleber Zanchettin. 2020. DQN-AF: Deep Q-Network based Adaptive Forwarding Strategy for Named Data Networking. In *2020 IEEE Latin-American Conference on Communications (LATINCOM)*.

[4] Ashlesh Gawande. [n.d.]. PSync patch: Replace PSync FullSync algorithm and fix bugs to lower delay and overhead. https://gerrit.named-data.net/c/PSync/+/6220/9.

[5] AKM Mahmudul Hoque, Syed Obaid Amin, Adam Alyyan, Beichuan Zhang, Lixia Zhang, and Lan Wang. 2013. NLSR: Named-data link state routing protocol. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. 15–20.

[6] Van Jacobson, Diana K Smetters, JD Thronton, Michael F Plass, Nicholas H Briggs, and RL Braynard. 2009. Network Named Content. *CoNEXT* (2009).

[7] Farhan Ahmed Karim, Azana Hafizah Mohd Aman, Rosilah Hassan, Kashif Nisar, and Mueen Uddin. 2022. Named Data Networking: A Survey on Routing Strategies. *IEEE Access* 10 (2022), 90254–90270.

[8] Serge A Krashakov, Anton B Teslyuk, and Lev N Shchur. 2006. On the universality of rank distributions of website popularity. *Computer Networks* 50, 11 (2006), 1769–1780.

[9] Vince Lehman, Ashlesh Gawande, Beichuan Zhang, Lixia Zhang, Rodrigo Aldecoa, Dmitri Krioukov, and Lan Wang. 2016. An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[10] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Yashar Ganjali, and Christophe Diot. 2008. Characterization of failures in an operational IP backbone network. *IEEE/ACM transactions on networking* 16, 4 (2008), 749–762.

[11] Mini-NDN Website [n.d.]. Mini-NDN Website. https://minindn.memphis.edu.

[12] Philipp Moll, Varun Patil, Lan Wang, and Lixia Zhang. 2022. SoK: The evolution of distributed dataset synchronization solutions in NDN. In *Proceedings of the 9th ACM Conference on Information-Centric Networking*. 33–44.

[13] NDN Testbed Website [n.d.]. NDN Testbed Website. https://named-data.net/ndn-testbed.

[14] NDN Traffic Generator [n.d.]. NDN Traffic Generator. https://github.com/named-data/ndn-traffic-generator.

[15] Daniel Posch, Benjamin Rainer, and Hermann Hellwagner. 2016. SAF: Stochastic adaptive forwarding in Named Data Networking. *IEEE/ACM Transactions on Networking* 25, 2 (2016), 1089–1102.

[16] Haiyang Qian, Ravishankar Ravindran, Guo-Qiang Wang, and Deep Medhi. 2013. Probability-based adaptive forwarding strategy in Named Data Networking. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. 1094–1101.

[17] Junxiao Shi, Eric Newberry, and Beichuan Zhang. 2017. On broadcast-based self-learning in named data networking. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. 1–9. https://doi.org/10.23919/IFIPNetworking.2017.8264832

[18] Asadullah Tariq, Rana Asif Rehman, and Byung-Seo Kim. 2020. Forwarding strategies in NDN-based wireless networks: A survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2020), 68–95.

[19] Lan Wang, Vince Lehman, AKM Mahmudul Hoque, Beichuan Zhang, Yingdi Yu, and Lixia Zhang. 2018. A secure link state routing protocol for NDN. *IEEE Access* 6 (2018), 10470–10482.

[20] Jin Y Yen. 1970. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of applied mathematics* 27, 4 (1970), 526–530.

[21] Cheng Yi, Jerald Abraham, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2014. On the role of routing in named data networking. In *Proceedings of the 1st ACM conference on information-centric networking*. 27–36.

[22] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2013. A case for stateful forwarding plane. *Computer Communications* 36, 7 (2013), 779–791.

[23] Minsheng Zhang, Vince Lehman, and Lan Wang. 2017. Scalable Name-based Data Synchronization for Named Data Networking. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*.

[24] Mingchuan Zhang, Xin Wang, Tingting Liu, Junlong Zhu, and Qingtao Wu. 2020. AFSndn: A novel adaptive forwarding strategy in named data networking based on Q-learning. *Peer-to-Peer Networking and Applications* 13 (2020), 1176–1184.

[25] Yi Zhang, Bo Bai, Kuai Xu, and Kai Lei. 2018. IFS-RL: An intelligent forwarding strategy based on reinforcement learning in named-data networking. In *Proceedings of the 2018 Workshop on Network Meets AI & ML*. 54–59.