Improving the Freshness of NDN Forwarding States

Jianxun Cao*, Dan Pei*, Zhelun Wu*, Xiaoping Zhang*, Beichuan Zhang†, Lan Wang‡, Youjian Zhao*
*Tsinghua University †University of Arizona ‡University of Memphis
*Tsinghua National Laboratory for Information Science and Technology (TNList)

***Abstract*—Named Data Networking (NDN) is a new Internet architecture that replaces today's focus on where – addresses and hosts – with what – the content that users and applications care about. A unique advantage of NDN over IP is the *adaptive* forwarding plane, which, by observing the performance of Interest/Data exchange, can dynamically select the best performing forwarding path, detect and recover from failures, load-balance across multiple paths, and mitigate attacks such as prefix hijacking and DDoS. A key component of adaptive forwarding is *interface ranking*, namely when and how to update the interfaces' metrics and rank them.**

As we will point out in this paper, however, the existing interface ranking scheme suffers from the problem of outdated forwarding states. Using two concrete problems, *SRTT slow-convergence* and *probing oscillation*, we illustrate how outdated forwarding states can impact the forwarding performance. We propose new forwarding strategies with *Adaptive SRTT Update (ASU)* and *Proactive Probing* to *achieve up-to-date forwarding states*, and evaluate how these strategies are able to address the two problems. Both theoretical analysis and simulation results show that the new strategies can reduce SRTT convergence time by 37.9% and the loss rate by 75% to 94.75%, compared to the existing interface ranking strategies.

I. Introduction

Named Data Networking (NDN) [1] is a new Internet architecture that emphasizes the content itself rather than its container (*e.g.*, host) or channel (*e.g.*, connection). In NDN, the consumer sends an Interest packet into the network to request a Data packet. The Interest carries the name of the data being requested instead of the destination address, and the matching Data packet can be retrieved from anywhere. By shifting the network service abstraction from "delivering a packet to the destination" to "retrieving a named data", NDN brings benefits such as in-network caching, native multicast, data-centric security and many others.

A unique feature of NDN is its *adaptive forwarding plane* [2], [3]. When an NDN router forwards an Interest packet out via a particular interface, it records this Interest in the Pending Interest Table (PIT) and starts waiting for a matching Data packet to return on the same interface. If the Data does return, the content retrieval is a success and the round-trip time (RTT) can be recorded to reflect the performance of using that interface. If the Data does not return in time or a Negative Acknowledgement (NACK) packet is received, the router knows that this interface does not work in retrieving this content and can record this information as well. This forms a forwarding-plane feedback loop that allows the router to detect any network fault, e.g., link failures and congestion, and take an alternative path to resolve the problem without relying on the control plane, i.e., routing convergence. Thus compared with IP's forwarding plane, NDN's is more intelligent, more resilient to network faults, and more effective in using multiple paths.

ISBN 978-3-901882-83-8

The decision process at the adaptive forwarding plane is called a "forwarding strategy," which considers the recorded performance metrics of multiple interfaces and chooses the most suitable one to forward an incoming Interest packet. A key component of forwarding strategies is the **interface ranking, *which updates the interfaces' metrics and ranks all the outgoing interfaces***, so that the strategy can choose the best interface to use. Without a good interface ranking scheme, forwarding states may be outdated, leading to reduced forwarding efficiency. In this paper, we focus on designing interface ranking schemes that can keep forwarding states up to date. We analyze two key parts of interface ranking: *periodical measurement* and *color classification*. The existing interface ranking schemes suffer from two problems: *SRTT slow-convergence* and *Probing oscillation*, which illustrates the impact of outdated forwarding states on forwarding performance. We propose new schemes for interface ranking to achieve up-to-date forwarding states, and show the effectiveness of our proposed solution via theoretical analysis and simulations.

The contributions of our work are twofolds. First, to the best of our knowledge, this paper is the first to study in depth interface ranking strategies and report two specific problems, SRTT slow-convergence and probing oscillation, which affect the accuracy of forwarding states. Second, we propose two new schemes, the *Adaptive SRTT Update (ASU)* algorithm and *Proactive Probing* approach, to achieve up-to-date forwarding states. Both theoretical analysis and NDNSim simulation results show that our new schemes reduce SRTT convergence time by 37.9% and the loss rate by 75% to 94.75%, compared to existing interface ranking strategies.

The remainder of the paper is organized as follows. Section II introduces basic concepts in NDN's adaptive forwarding, especially interface ranking. Limitation of current adaptive forwarding strategies is shown in Section III. SRTT slow-convergence and probing oscillation problems are described in Section IV and Section V, respectively. We introduce our new strategies and carry out the analysis theoretically in Section VI and Section VII, respectively. In Section VIII, we evaluate our new strategies using ndnSIM 2.0 simulator and analyze the results. Section IX briefly reviews related work. Finally, Section X presents our conclusion.

II. Adaptive Forwarding in NDN

In NDN, routers maintain state information of pending Interests, which brings adaptive forwarding plane to observe data retrieval performance and explore multiple forwarding paths. **Interface ranking** [3] is the key component of adaptive forwarding[1] to help routers find the current best outgoing interface to fetch data. *Color classification* and *periodical measurement* are introduced to help implement interface ranking.

Based on interface ranking, there are two major adaptive forwarding strategies in the literature[2]: Best-Route [3] and NCC [5]. These two approaches are similar in interface ranking and only differ slightly in the number of probed interfaces in color classification.

A. Name prefix

In NDN, Data names instead of IP addresses are hierarchically structured. For routing aggregation, router's FIB (Forwarding Information Base) contains *name prefixes*, and network routing protocols will distribute name prefixes in a way similar to distributing IP prefixes in todays Internet. A FIB entry records the working status of each interface. Please note that *interface status is per-name-prefix-per-interface*.

B. Periodical measurement and color classification

Detailed interface metrics (*e.g.* SRTT) are used for ranking interfaces. For a given prefix, to maintain the ranking metrics for interface ranking, the router periodically sends a copy of the Interest packet to all the interfaces to measure various metrics (such as SRTT) used in forwarding policies, and we name this probing as periodical measurement.

Besides ranking with interface metrics, color classification [3] is used to record an outgoing interface's working status for each prefix: *GREEN* means that the interface can bring the data back, *YELLOW* means that it is unknown whether the interface can bring the data back, and *RED* shows that the interface cannot bring the data back. The color classification are updated based on various feedbacks (whether to get the data successfully or not) from the network. Please note the periodical measurement can only change the ranking metrics, and it does not change the color of the interfaces regardless of the measurement results.

With periodical measurement and color classification, the interface ranking for a given prefix works as follows.

[1]Adaptive forwarding consists of NACK, interface ranking, congestion control, and so on. In this paper, we just focus on interface ranking, and our study does NOT influence the strategy of NACK and congestion control.

[2]In this paper, we omit the impractical broadcast approach [1], which floods the Interest packet to all available interfaces when a router receives any incoming Interest packet. Obviously, for each Interest packet, broadcast approach can guarantee fetching the data from the optimal path, but it will tremendously add extra overhead. Furthermore, broadcast strategy breeds the Interest flooding attack [4] and causes trouble for network security in NDN. Thus, broadcast strategy has little practical application value except in some extreme cases.

1) Ranking rules: For choosing the best path, GREEN interfaces are preferred over YELLOW ones. RED interfaces are not used in forwarding, and can change color only by the routing protocol. When ranking interfaces of the same color, NDN supports a wide variety of forwarding policies (or ranking metrics), such as "follow routing" based on OSPF cost, "the sooner the better" based on SRTT, and so on.

2) Forwarding based on ranking: For a specific Interest prefix, *all* the Interest packets destined to the same prefix will be transmitted through the best GREEN interface until the router receives a *NACK* or *Timeout*. The router will then degrade the currently used GREEN interface to YELLOW. If there still exists any GREEN interface, the router will switch to the highest ranked GREEN interface. Otherwise, the router will probe the YELLOW outgoing interfaces in the order of their ranking. We call the process of probing the YELLOW interfaces as **triggered probing**. We will explain the triggered probing in detail in the following.

C. Triggered Top-N probing

In Best-Route, once the probing process is triggered, the router will keep forwarding Interests along the current forwarding interface, which was just degraded from GREEN to YELLOW, and send a **probing** Interest to the highest ranked YELLOW interface to test whether it can bring Data back. When it receives NACK or Timeout for the probing packet, it will start trying the second-ranked YELLOW interfaces, and so on. On the other hand, if a Data comes back from the probed YELLOW interface, the router will change the color of this interface from YELLOW to GREEN, and this GREEN interface will be used as the best path, by now the triggered probing process is finished.

NCC is implemented in CCNx 0.7.2 as the default strategy. In NCC strategy, once the probing process is triggered, the router probes the top **2** YELLOW interfaces together if there are more than one YELLOW interfaces. Otherwise, it probes the only YELLOW interface. All other mechanisms are the same as in Best-Route. We unify these two approaches as **Triggered Top-N probing approaches**, where $N = 1$ in best-route, and $N = 2$ in NCC.

D. Benefits of interface ranking in adaptive forwarding plane

The periodical measurement and color classification with triggered probing in NDN's adaptive forwarding plane help a router maintain the information for all its interfaces. It allows the forwarding plane to quickly detect link fault through NACK mechanism and path probing. When a NACK arrives or any Interest packet is timed out, the router can freely explore the alternative path to find an available path to continue the forwarding work.

Compared with the routing plane, the forwarding plane can handle the link failure more quickly and solve the problem more flexibly. The forwarding plane can deal with the link failure without too much additional overhead as soon as possible when discovering any network fault.

III. Limitations of Interface Ranking

Interface ranking can help routers detect link fault more quickly, handle the link failure more promptly and solve the problem more flexibly. In ideal conditions, to choose outgoing interface, routers should make the decision to forward packets according to *the global network status*, because, for routers, the global network status can provide the optimal decision in real time to help routers optimize the outgoing packets flow. Specially, when the routers detect link fault, the global real-time network status can help routers ensure the essential network problem and figure out the most efficient solution. However, with the network measurement in the forwarding plane, the routers can just obtain *the local information*. Unlike the global information in routing plane, the local information in the forwarding plane limits the efficiency of forwarding packets and dealing with the link failure.

To approximate the global network status as far as possible, the network measurement for interface ranking metrics in forwarding plane and the ranking rules should be adaptive enough. However, the existing simple periodical measurement and color classification cannot work well enough.

A. Periodical measurement

With the in-network cache, the uncertainty of location where Interest packets are satisfied leads to the frequent changing of network metrics. Thus, in NDN, the simple periodical measurement will lead to some outdated network metrics. *In this paper, we focus on the important network metric for interface ranking: SRTT (Smoothed Round-Trip Time).* In Section IV, we will introduce a new problem caused by SRTT measurement: *SRTT slow-convergence*.

B. Triggered Top-N probing approach for color classfication

From Section II, we note that *Triggered Top-N probing approach is reactive*, which is illustrated in the following two key details. First, The probing is triggered after the NACK or Timeout, and YELLOW interfaces are probed in the order of their rankings. However, the ranking is based on periodical measurement, which can be quite outdated when there are **bursty congestions** along the path. Second, the change of interface color is done at the **packet-level** feedback for each prefix: NACK or Timeout for GREEN interfaces, and Data for YELLOW interfaces. When faced with burst congestions, the colors might frequently change between GREEN and YELLOW back and forth. In fact, later in Section V, we will show that, with the outdated ranking information under burst congestions, color oscillation will result from the packet-level action and will cause a new problem in NDN forwarding plane: *probing oscillation*.

IV. SRTT Slow-Convergence

In this section, we will focus on the measurement of SRTT and introduce a new problem: SRTT slow-convergence. SRTT reflects the network situation and is used for calculating RTO (Retransmission Time-Out) value. The reason why SRTT is used instead of RTT, is to eliminate as much as possible the impact of the network jitter which is caused by the instable network state such as the queue buffer congestion. In this paper, we adopt the typical algorithm: Exponential weighted moving average in TCP/IP to update the SRTT:

$$SRTT_i = \alpha \cdot SRTT_{i-1} + (1-\alpha) \cdot RTT_i \quad (1)$$

Where α is the constant weighting factor ($0 < \alpha < 1$). Usually, α is between 0.8 and 0.9. In this paper, we let $\alpha = 0.8$ for the analysis and evaluation.

TABLE I
THE FIB ENTRY OF R_c

prefix	interface				
	ID	OSPF	RTT	COST	COLOR
/prefix	R_2	0.55	0.7	1.27	GREEN
	R_3	0.67	0.73	1.4	YELLOW
	R_1	0.69	0.75	1.44	YELLOW
	R_4	0.75	0.8	1.55	YELLOW

TABLE II
SOME NOTATIONS

Con	The consumer
Pro	The producer
R_i	The routers
r	The loss rate of R_p
k	The number of sent Interest packets per second
t_{data}	The RTT of each outgoing interface
t_{out}	The timeout value

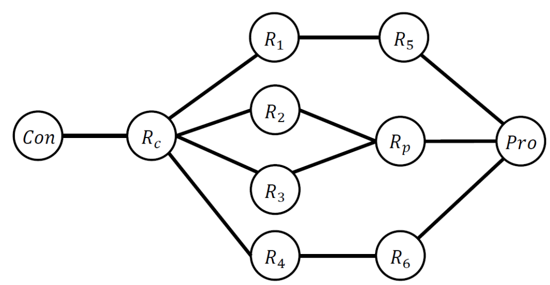


Fig. 1. A sub-topology of Tsinghua University Campus Network

Here is an example to describe SRTT slow-convergence. We use the sub-topology of Tsinghua University as shown in Fig 1 and give the COST[3] for interface ranking as shown in Table I. Some notations are defined in Table II.

We assume that at some time the consumer fetches the data along the path $Con - R_c - R_2 - R_p - Pro$. Here is a simple simulation to illustrate the problem. At the start, there is no cache in R_2 and R_p, so the consumer gets the data from the producer, which needs $150ms$. From $5s$ to $10s$, we assume that R_2 caches the needed data[4] and the consumer can just get the

[3] We specify COST to be $OSPF + RTT$ and the value of OSPF and RTT are both normalized linearly.

[4] For example, another consumer connected to R_2 fetches the same data from 5s to 10s.

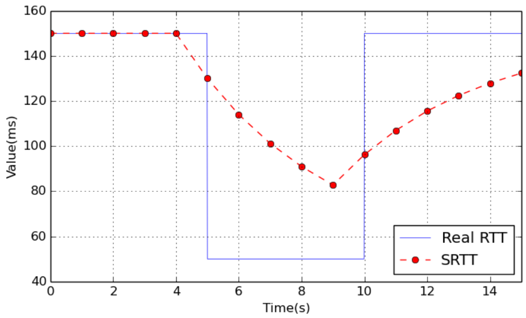


Fig. 2. A case to describe SRTT Slow-Convergence. In this case, the initial RTT is $150ms$. From $5s$, the RTT is reduced to $50ms$ and is recovered to $150ms$ after $10s$.

data from R_2 directly during this $5s$. We assume this RTT equals $50ms$. If the first consumer sends 100 Interest packets per second and the interval between two adjacent probing equals 100, then the RTT and SRTT perform as shown in Fig. 2. We can see that after $5s$, SRTT cannot reflect the real RTT. We call this problem as *SRTT Slow-Convergence*.

SRTT slow-convergence just exists in NDN while not in IP. In IP routing plane, the network metrics are nearly constant when the network works stably. Thus, SRTT filters the variation of RTT that are just caused by network jitter. However, in NDN, the variation of RTT cannot completely represent the network faults because of **in-network cache**. With the additional overhead, the frequency of periodical measurement cannot be very high, while the low measurement frequency will cause the SRTT value to not converge to the rapidly variational RTT timely, which makes SRTT often outdated. The outdated SRTT will deeply influence the judgement and lead to the above-mentioned SRTT Slow-Convergence.

V. Probing Oscillation

In this section, we introduce a new problem existing in NDN forwarding plane but not in IP: *Probing Oscillation*.

A. Problem description

Here is a simple example to illustrate what probing oscillation is. In Fig 1, the router R_c has four next hops R_1, R_2, R_3 and R_4. According to the interface ranking in FIB Table I, we see that firstly R_c will choose R_2 as the top choice to forward all the Interest packets.

Now we assume that $R_p - Pro$ congests, then, for R_c, the next actions of choosing the forwarding interface oscillate between the next two situations:

1) Situation 1: $R_2 \to R_3$*:* When congestion occurs, some Interest packets along $R_c - R_2 - R_p - Pro$ cannot fetch the corresponding data before the timer expires. But R_c just considers that R_2 now is not suitable to fetch data, so according to interface ranking, R_c will probe the second highest ranked interface R_3 to forward Interest packets along $R_c - R_3 - R_p - Pro$. It is worth noting that the link $R_p - Pro$ is the shared common sub-path of both paths ($R_c - R_2 - R_p - Pro$ and $R_c - R_3 - R_p - Pro$), thus for R_c, R_3 is still not the good choice. However, because congestion just causes packet loss instead of failure, even if the loss rate of $R_p - Pro$ reaches up to 10%, the success rate of probing is as high as 90%. Thus, *there is a very low possibility that a simple probing packet detects congestion, while for a given period of time, there is a very high possibility that congestion leads to timer expiration of some Interest packets.* So, if the router probes R_3 successfully, R_3 then becomes GREEN to forward Interest packets and R_2 changes to YELLOW.

2) Situation 2: $R_3 \to R_2$*:* Once the choice for R_c changes from R_2 to R_3, this choice is still instable and may change from R_3 to R_2. There are two possible reasons. First, the path $R_c - R_3 - Rp - Pro$ also congests (including the congested $pathRp - Pro$), then R_c will check the highest ranked YELLOW interface, namely R_2, and may switch back to R_2. Second, as illustrated in Fig. 3, we suppose at time t_0, R_c sends an Interest packet I_1 to R_2 which is lost later. So after t_{out} time, R_c does not receive the corresponding Data packet and mark the interface as YELLOW (Situation 1). But during this t_{out} tentative-sending time period, R_c has sent out $k \cdot t_{out}$ Interest packets to R_2 and some of these Interest packets, such as I_2, I_3, are satisfied after $t_0 + t_{out}$. This will cause R_2 to become GREEN again and R_2 replaces R_3 as the forwarding interface.

In conclusion, R_c will switch its outgoing interface back and forth between R_2 and R_3, and we name this phenomenon as *probing oscillation*.

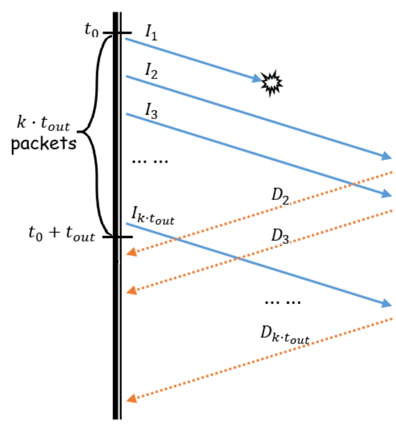


Fig. 3. A simple illustration of Probing Oscillation

The essential reason why probing oscillation occurs is the high-frequency color variation between GREEN and YELLOW caused by multipath forwarding with a shared congested common link and packet-level triggered probing, which makes that probing oscillation is unique in NDN while not in TCP/IP. In NDN, probing oscillation commonly happens and greatly influences the benefit of multipath forwarding according to our experiments. *When the router detects the network failure, with the influence of probing oscillation, it cannot recover from the network failure, which may bring a high loss of packets.* Although we just chose a simple topology to explain the problem, in a more large and complex network, probing oscillation still continues to appear as long as there exists a shared common link, especially when there is a bottleneck link. Experiments in Section VIII prove the above claims.

VI. Adaptive SRTT Update (ASU) algorithm for periodical measurement

In this section and the next section, we propose some new strategies to improve the freshness of forwarding states to essentially solve the SRTT slow-convergence and probing oscillation. In this section, for periodical measurement, we firstly propose an **Adaptive SRTT Update (ASU) algorithm** to calculate SRTT more accurately with dynamic sample frequency instead of the simple periodical measurement. Then we will analyze the theoretical efficiency of solving the probing oscillation.

A. Adaptive SRTT Update (ASU) algorithm

Unlike IP, the forwarding plane in NDN can gain the dynamic network congestion status in real time, so that any router can update the cost of each outgoing interface per prefix according to the RTT of the pending Interest packets. Thus, all the interfaces need to update the SRTT value to reflect the network situation more exactly, which means routers need to flood a probing Interest packet periodically to all the interfaces. However, the probing frequency is hard to be set. On the one hand, low probing frequency will cause SRTT slow-convergence as described in Section IV. On the other hand, high probing frequency will cause too much extra overhead. In fact, in the most situation, SRTT tends to be constant except when the current RTT varies widely, *e.g*, cache hit. We hope to adopt *dynamical sample frequency* to make sure that the probing frequency can get lower when the RTT is approximately constant and get higher with the great variation of RTT.

Thus, we propose a new **Adaptive SRTT Update (ASU) algorithm** to calculate SRTT more accurately with dynamical sample frequency and improve SRTT slow-convergence. We use Δn to denote the number of Interest packets between two adjacent probing. The initial value of Δn is denoted as Δn_0 and the minimum or the maximum of Δn is denoted as Δn_{min} or Δn_{max}. We define the Changing Factor of RTT to be η which is calculated as follows:

$$\eta = \left| \frac{RTT - SRTT}{RTT + SRTT} \right| \tag{2}$$

From Equation 2, we can see that:

$$0 \leq \eta < 1 \tag{3}$$

Obviously, η is a normalized value to describe the distance between the real RTT and SRTT. The greater RTT changes, the bigger η will be. Based on the Changing Factor η, we propose the stretch Δn:

$$\Delta n_i = \begin{cases} \max(\Delta n_{min}, (1-\eta_i)\Delta n_{i-1}) & \eta \geq \eta_{threshold} \\ \min(\Delta n_{max}, (1+\beta)\Delta n_{i-1}) & \eta < \eta_{threshold} \end{cases} \tag{4}$$

In the above formula, $\eta_{threshold}$ denotes the critical value for distinguishing RTT changing greatly from RTT tending to SRTT. If η is larger than $\eta_{threshold}$, Δn should become smaller to increase the probing frequency. Otherwise, Δn should become bigger to decrease the probing frequency.

B. Analysis for solving SRTT slow-convergence

There are two cases when calculating SRTT: 1) When RTT changes little and 2) When RTT changes greatly. In case 1, no matter whether the ASU algorithm is used or not, SRTT will still be close to the real RTT, that is, all the strategies perform perfectly. However, by using the ASU algorithm, the Extra Overhead Ratio (EOR) will be lower:

$$EOR = \frac{1}{\Delta n_{max}} \tag{5}$$

In case 2, according to the definition of η in Equation 2, great variation of RTT means $\eta \geq \eta_{threshold}$. We assume that at 0 time, RTT changes from one value to another. Let $SRTT_0$ denote the initial SRTT at 0 time. We define m as:

$$m = \frac{RTT}{SRTT_0} \tag{6}$$

Obviously, when $\eta_i < \eta_{threshold}$, $SRTT_i$ approximately converges to RTT. From Equation 1, we can get the general term formula of SRTT:

$$SRTT_i = RTT \cdot \left[1 - \left(1 - \frac{1}{m}\alpha^i \right) \right] \tag{7}$$

Let $\eta_i < \eta_{threshold}$, then the probing number of quitting the SRTT-convergence i_c equals:

$$\begin{cases} i_c = \left\lceil \lg_\alpha \frac{2\eta_{threshold}}{(1-\eta_{threshold})(\frac{1}{m}-1)} \right\rceil, m < 1 \\ i_c = \left\lceil \lg_\alpha \frac{2\eta_{threshold}}{(1+\eta_{threshold})(1-\frac{1}{m})} \right\rceil, m > 1 \end{cases} \tag{8}$$

Assume that $m = 3$, $\eta_{threshold} = 0.1$, $\alpha = 0.8$, we can work out $i_c = 6$, which means, SRTT will converge to RTT after sampling for only 6 times. Thus, ASU algorithm is efficient for SRTT slow-convergence.

VII. Proactive Probing approach for color classification

In this section, for color classification, we introduce **Proactive Probing approach** as a new triggered probing approach, including introducing path backup and probability-based forwarding, to improve interface ranking and analyze the theoretical efficiency of solving probing oscillation.

A. Proactive Probing approach

The Proactive Probing approach contains two sides: 1) **constant Multi-GREEN-Interface strategy** to maintain several path backups, and 2) **probing based on *Probing Probability Function (PPF)* algorithm** for YELLOW interfaces to avoid probing strictly by interface ranking.

1) Constant Multi-GREEN-Interface strategy: The basic idea of constant Multi-GREEN-Interface strategy is that, for all the routers of which outgoing interfaces number is larger than 2, we set more than one GREEN interface to ensure some **backup paths**. The router will keep the number of GREEN interfaces as a constant **Minimal Number of GREEN Interfaces (MNGI)**. Constant Multi-GREEN-Interface strategy will start when an Interest packet comes to check the number of GREEN interfaces. While the number of GREEN interfaces is less than MNGI, the probing based on PPF mentioned in the following part will be triggered to choose one YELLOW interface to explore an available path. When this trying succeeds, the router will mark it as a GREEN interface. In order to prevent an extra cost, the value of MNGI should not be too large. We suggest that MNGI value does not exceed half the number of total interfaces.

2) Probing based on Probing Probability Function algorithm: As the above description, probing will be triggered when constant Multi-GREEN-Interface strategy discovers that the GREEN interfaces number is less than MNGI. However, how to probe will affect the performance directly. It is not smart enough that the router chooses the probing interface *only* based on the Interface Ranking. There are two reasons for this: 1) For each YELLOW outgoing interface, with the introduction of router cache, the OSPF value from the routing algorithm cannot reflect the actual time of fetching the data. 2) Because SRTT will not be updated since this interface was remarked as YELLOW, the SRTT similarly cannot reflect the actual efficiency of fetching the data, which means that the SRTT may be outdated. For example, in Fig. 1, the OSPF value of R_4 is the highest, but when the requested data is cached in R_4, choosing R_4 to fetch data is the optimal choice.

Thus, at this time, the COST attribute is not effective enough to rank all the YELLOW interfaces. But we can assume that *the smaller the COST is, the higher the probability of the optimal choice will be*. Based on this thought, we come up with the *Probing Probability Function (PPF)* to help **probability-based probing**. PPF should satisfy the decreasing monotonicity:

$$COST_i > COST_j \Leftrightarrow PPF(i) < PPF(j) \tag{9}$$

In this paper, we choose the following formula as PPF:

$$PPF(i) = \frac{1/COST_i}{\sum(1/COST_i)} \tag{10}$$

PPF ensures that each outgoing interface has a chance to be selected for probing. The probability is due to the history information of the COST calculated by both the routing algorithm and SRTT. Based on PPF, the router has a high probability of choosing the interface which performed well in the past, which guarantees the fairness for all interfaces.

B. Analysis of solving probing oscillation

The topology in Fig. 1 is selected for analysis. With Proactive Probing strategy, we choose the Best-Route (Triggered Top-1 probing) strategy for theoretical analysis. We try to analyze how many probing times the loss rate of R_c will converge to 0.

We assume that, at first, R_c chooses R_2 to fetch the data. When the first packet loss occurs, only if R_c doesn't choose R_3 as its next outgoing interface, then R_c jumps out of the probing oscillation. Thus, the probability of loss rate convergence equals $1 - PPF(3)$. Similarly, when the second packet loss occurs, the probability of quitting the probing oscillation equals $1 - PPF(2)$. Now we assume that the i^{th} packet loss occurs, then the probability of quitting the probing oscillation equals:

$$p = \begin{cases} 1 - PPF(2) & if \quad i\%2 = 1 \\ 1 - PPF(3) & if \quad i\%2 = 0 \end{cases} \tag{11}$$

We can approximatively consider it as Geometric distribution $G(p)$ while $p \simeq [(1 - PPF(2)) + (1 - PPF(3))]/2$. Thus, the expectation of the number of quitting probing oscillation (n_q) equals:

$$E(n_q) = \frac{1}{p} = \frac{2}{2 - PPF(2) - PPF(3)} \tag{12}$$

Even if we assume that $PPF(2)$ and $PPF(3)$ are as high as 0.8, $E(n_q)$ just equals 5. We can see that with Proactive Probing approach, limited probing times is enough to theoretically help routers jump out of probing oscillation.

VIII. Performance Evaluation

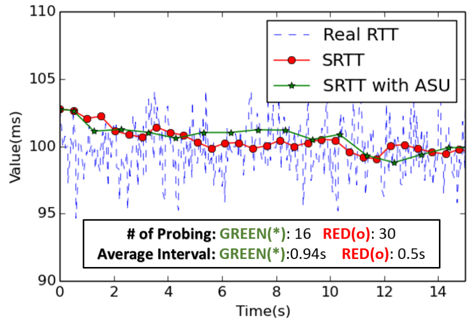
In this section, we evaluate the performance of both ASU algorithm and Proactive Probing approach for interface ranking in NDN using modified NDNSim 2.0 simulator [6]. The new version of NDNSIM integrates ndn-cxx library (NDN C++ library with eXperimental eXtensions) and the NDN Forwarding Daemon (NFD) to enable experiments with real code in a simulation environment. Our evaluation contains two parts. All the network parameters are set as follow:

- $t_{data} = 100ms$, $t_{out} = 120ms$
- $k = 100$
- $\eta_{threshold} = 0.1$

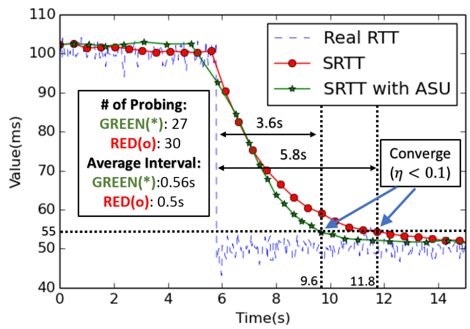
A. Evaluation on sub-topology of TUNET

We use the sub-topology of Tsinghua University Campus Network shown in Fig. 1 to verify the effectiveness of solving SRTT slow-convergence and probing oscillation.

1) Adaptive SRTT update algorithm for slow-convergence: Given the two different variations of generated Gaussian distributed RTT data, we use SRTT calculated by Equation 1 to see how the ASU algorithm has ameliorated the performance of RTT detection. **Firstly**: As shown in the Fig. 4(a), we set the distribution of RTT to be $N(100, 2)$ designed as a variable with very small variation. We see that no matter whether using ASU algorithm, SRTT can converge to real RTT as well. But paying attention to the overhead, we can find that in 15 seconds, with the advent of ASU algorithm, the total number of probing decreases from 30 to 16 and the average interval increases from 0.5s to 0.94s, which means that the Extra Overhead Ratio (EOR) is decreased by 46.7%.

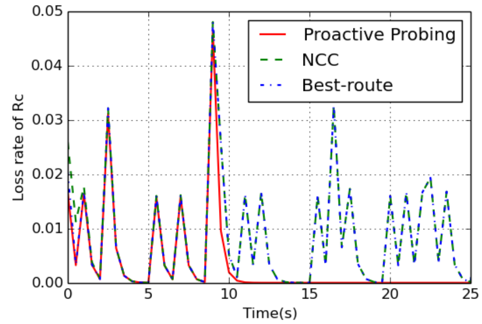


(a) SRTT comparison when RTT variation is small

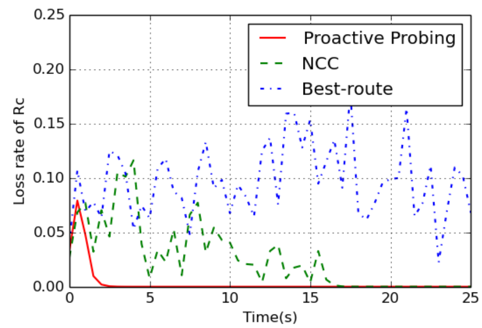


(b) SRTT comparison when RTT is suddenly reduced by half

Fig. 4. SRTT comparison when RTT variation is small or s suddenly reduced by half.



(a) The loss rate of R_c when $r = 0.01$



(b) The loss rate of R_c when $r = 0.1$

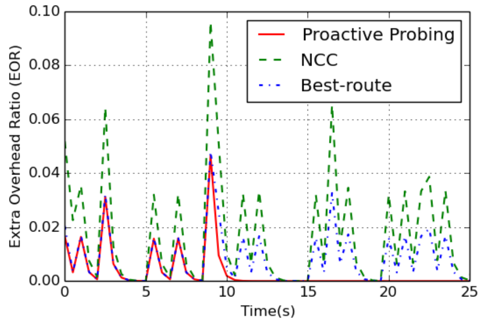
Fig. 5. The loss rate of R_c when $r = 0.01$ or $r = 0.1$. In a), the loss rate with NCC strategy and Best-route strategy nearly coincide while the loss rate with Proactive Probing strategy decreases to 0 at about 10 seconds. In b), the loss rate with Best-route strategy still performs the worst. The loss rate with NCC strategy converges to 0 after about 18s while the loss rate with Proactive Probing strategy decreases to 0 after only about 3 seconds.

So this proves that *in the flat phase of RTT, probing with ASU algorithm performs as well as the periodical measurement but brings a lower extra overhead.* **Secondly**: In this case, SRTT slow-convergence occurs. In the Fig. 4(b), we assume that there is a striking decline at around the time 6s. Before that, the distribution of RTT is $N(100, 2)$ and later is $N(50, 2)$. The figure illustrates that the SRTTs with ASU algorithm are more sensitive to the change. With ASU algorithm, the time from when RTT declines to it's convergence ($\eta < \eta_{threshold}$) is just 3.6 seconds, while on the other hand without ASU algorithm it's time is about 5.8 seconds. Thus, the response time is decreased by 37.9%. With ASU algorithm, although a little extra probing is generated during [6,10]s, the total overhead (average probing interval) in 15 seconds is also close to the plain probing. Thus, we conclude that *ASU algorithm can alleviate SRTT slow-convergence with little extra overhead as compared to the simple periodical measurement*.

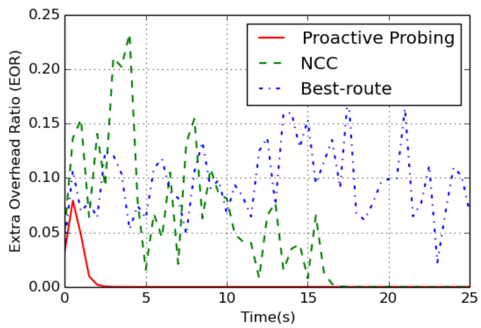
2) Proactive Probing approach for probing oscillation: To verify the performance of solving probing oscillation, we investigate two parameters: the loss rate of R_c and the extra overhead.

First, we analyze the loss rate of R_c in two occasions – one when the loss rate of R_p is 0.01 and the other is when the loss rate of R_p is 0.1. Both of the two cases will cause the probing oscillation of R_c. As shown in Fig. 5(a), when the loss rate of R_p is 0.01, we see that due to the very limited probability of packet loss, these three approaches perform closely to each other except that Proactive Probing approach slumps down to 0 at around 12s, while the other two keep on oscillating and do not reveal any trend to converge to 0. But as shown in Fig. 5(b), when the loss rate R_p is 0.1, things become more obvious. Proactive Probing approach leads to zero loss rate in only 3 seconds, which is about $1/5$ of the time as NCC falls to 0 loss rate. Best-route, however, does not show any trend of converging to 0. Therefore, we conclude from the experiments that Proactive Probing approach performs the best in solving the probing oscillation as compared to both the Best-Route and NCC.

Second, we see that the overhead of R_c is strongly related to the loss rate of R_c, because when the current path fails, each of the three strategies requires spontaneous probing of other pathes, and inevitably this incurs a detection overhead. Accordingly, from Fig. 6(a), we observe that when $R_p = 0.01$, the overhead of NCC is approximately 2 times as large as the loss rate of Best-route since NCC needs to find the first 2 available GREEN paths with the lowest cost. Best-Route and Proactive Probing approach, however, only need to find the cheapest GREEN path. And Proactive Probing approach performs as well as the Best-Route before it drops to 0 at



(a) Extra Overhead Ratio (EOR) when $r = 0.01$



(b) Extra Overhead Ratio (EOR) when $r = 0.1$

Fig. 6. Extra Overhead Ratio (EOR) when $r = 0.01$ or $r = 0.1$.

around the same time as the loss rate drops to 0. When $R_p = 0.1$, from Fig. 6(b) we still see that they almost copy the shape of the loss rate except for the 2-time-magnification of NCC.

B. Evaluation on a replica of the actual NDN testbed topology

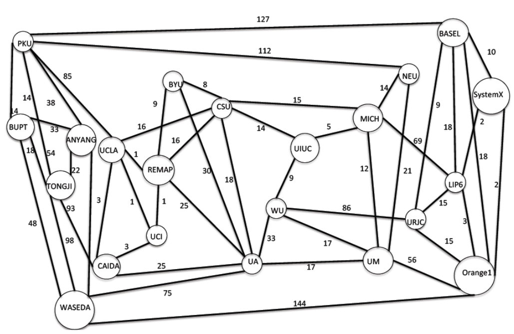


Fig. 7. A replica of the actual NDN testbed topology which contains 22 nodes and 50 links.

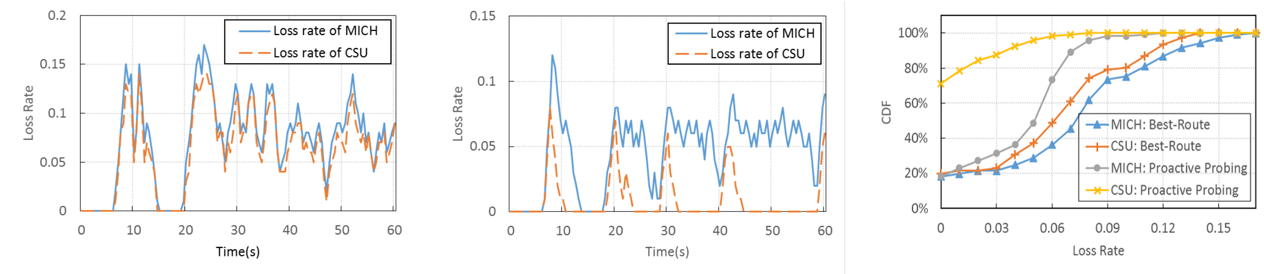
We use a replica of the actual NDN testbed topology which contains 22 nodes and 50 links shown in Fig. 7. The node "SystemX" is set as a server/producer and the other nodes are set as clients/consumers. The size of an Interest packet and a Data packet equals 25 Bytes and 1KB respectively. Each client initializes the sending rate of Interest packets as 10Kpps and randomly increases the sending rate by 5%-15% per second until the client detects the congestion which is defined in this section when the loss rate rises to 5%. Then the client will decrease the sending rate by half and repeat the increasing method. All the link bandwidth equals 1Gbps except for the bandwidth of MICH-LIP6 it is set as 100Mbps. *Because of the significantly lower bandwidth, the link MICH-LIP6 may become the bottleneck of this topology.*

We analyze the loss rate of the node MICH and the node CSU with the Best-Route and Proactive Probing approach respectively shown in Fig. 8. Because the link MICH-LIP6 is the bottleneck, it will get congested easily. Thus, it will cause one of its downstream nodes, CSU, to have probing oscillation. Fig. 8(a) shows that since the link MICH-LIP6 becomes congested, the node CSU begins to fall into probing oscillation and cannot handle this problem effectively with the Best-Route. Hence, the loss rate of CSU is almost as high as MICH's. While, in Fig. 8(b), with our Proactive Probing approach, although the congestion of MICH-LIP6 occurs, we can see that the loss rate of CSU decreases to 0 quickly. It means that CSU chooses another path to get data bypassing the link MICH-LIP6. We notice that after several seconds, the loss rate of CSU appears again and still returns to 0 in seconds. This is because that when the loss rate of MICH becomes low, CSU will reselect the path with MICH-LIP6 to get data until MICH-LIP6 seriously congests again. It's also worth noting that with Proactive Probing approach, the congestion level of MICH-LIP6 is significantly lower than that of Best-Route. With Proactive Probing approach, when CSU detects probing oscillation, it will finally unselect the congestion link MICH-LIP6, which will reduce the network flow via MICH-LIP6 and bring down the congestion level. Fig. 8(c) illustrates CDF of the loss rate with two approaches.

IX. Related Work

In TCP/IP, the stateless forwarding plane limits the network forwarding ability. Thus, multipath forwarding with which networks can provide end-hosts with multiple path choices are argued in [7] and [8] to perform better based on their experiments. Path Splicing [9], Pathlet routing [10] and Routing deflection [11] are designed to implement this idea. Multipath TCP [12] are introduced to set up multiple sub-TCP connections between the two ends. But all these improvement for TCP/IP forwarding plane is limited to the end-to-end transmission mode. Therefore, the increased adaptability of the forwarding plane are very limited. In TCP/IP, based on Fast reroute (FRR) mechanism [13], improved mechanisms such as MPLS FRR [14] and IPFRR [15] cannot efficiently handle multiple concurrent failures. For handling congestion control, Active Queue Management (AQM) mechanisms, such as Random Early Detection (RED) [16], have been introduced.

As an architecture of Information-Centric Networking (ICN), NDN supports multipath forwarding intrinsically. Adaptive forwarding plane [3] provide great performance in handling link failures and network congestion. Also, other congestion control mechanisms are studied. Hop-by-hop congestion control such as HR-ICP [17] and HIS [18] can take full advantage



(a) With Best-route strategy. (b) With Proactive Probing strategy. (c) The CDF of loss rate.

Fig. 8. The loss rate comparison between the node MICH and CSU in network testbed when using different strategies respectively.

of NDN adaptive forwarding plane, while all these mechanisms cannot effectively handle the dynamics in returning Data traffic. With AIMD algorithm liked in TCP/IP, Interest window size is presented to avoid excessive Interest packets [19], [20].

X. Conclusion

Adaptive forwarding plane is a key component for NDN architecture, but its existing strategies can suffer from outdated forwarding states, namely *interface coloring and ranking*. For the first time in the literature, in this paper we illustrate the impact of outdated forwarding states in the existing interface ranking mechanism through two concrete problems, *SRTT slow-convergence* and *probing oscillation*.

In order to efficiently achieve up-to-date forwarding states, the key challenge is to design *metric measurement* and *color classification* mechanism to balance the measurement overhead and the freshness of the forwarding states. Towards this goal, we propose *adaptive SRTT update* approach and *proactive probing* approach to improve the freshness of the forwarding states in NDN. Both theoretical analysis and NDNSim simulation results show that our new strategies reduce SRTT convergence time by 37.9% and the loss rate by 75% to 94.75% with little extra overhead, compared to the existing interface ranking strategies. We believe that this paper is an important first step towards up-to-date NDN adaptive forwarding plane.

Acknowledgement

This work has been supported by two National Natural Science Foundations of China (NSFC) under Grant No.61472210 and No.61472214, the Sate Key Program of National Science of China under Grant No.61233007, the National Key Basic Research Program of China (973 program) under Grant No.2013CB329105, the Tsinghua National Laboratory for Information Science and Technology Key Projects.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc cllaffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 44, no. 3, July 2014.

[2] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the role of routing in named data networking," *ICN(14): Proceedings of the 1st international conference on Information-centric networking*, 2014.

[3] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, April 2013.

[4] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," *IFIP Networking Conference*, May 2013.

[5] J. Shi, "ccnd 0.7.2 forwarding strategy," *http://redmine.named-data.net/projects/nfd/wiki/CcndStrategy, University of Arizona*, 2014.

[6] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim 2.0: A new version of the ndn simulator for ns-3," *NDN Technical Report*, vol. NDN, no. 0028, January 2015.

[7] D. Wendlandt, I. Avramopoulos, D. G. Andersen, and J. Rexford, "Dont secure routing protocols, secure data delivery," 2006.

[8] M. Caesar, M. Casado, T. Koponen, J. Rexford, and S. Shenker, "Dynamic route recomputation considered harmful," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 66–71, 2010.

[9] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala, "Path splicing," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 27–38.

[10] P. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet routing," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 111–122, 2009.

[11] X. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 159–170.

[12] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp." in *NSDI*, vol. 11, 2011, pp. 8–8.

[13] P. Pan, G. Swallow, A. Atlas *et al.*, "Fast reroute extensions to rsvp-te for lsp tunnels," 2005.

[14] T. Nadeau, K. Koushik, and R. Cetin, "Multiprotocol label switching (mpls) traffic engineering management information base for fast reroute," RFC 6445, Tech. Rep., 2011.

[15] A. K. Atlas and A. Zinin, "Basic specification for ip fast-reroute: loop-free alternates," 2008.

[16] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *Networking, IEEE/ACM Transactions on*, vol. 1, no. 4, pp. 397–413, 1993.

[17] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012, pp. 37–42.

[18] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An improved hop-by-hop interest shaper for congestion control in named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 55–60, 2013.

[19] S. Braun, M. Monti, M. Sifalakis, and C. Tschudin, "An empirical study of receiver-based aimd flow-control strategies for ccn," in *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*. IEEE, 2013, pp. 1–8.

[20] L. Saino, C. Cocora, and G. Pavlou, "Cctcp: A scalable receiver-driven congestion control protocol for content centric networking," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3775–3780.