

# DICE: Dynamic Multi-RAT Selection in the ICN-enabled Wireless Edge<sup>†</sup>

Gaurav Panwar  
New Mexico State University  
gpanwar@cs.nmsu.edu

Reza Tourani  
New Mexico State University  
rtourani@cs.nmsu.edu

Travis Mick  
New Mexico State University  
tmick@cs.nmsu.edu

Abderrahmen Mtibaa  
New Mexico State University  
amtibaa@cs.nmsu.edu

Satyajayant Misra  
New Mexico State University  
misra@cs.nmsu.edu

## ABSTRACT

Coupled with the rapid increase in mobile device users and the bandwidth and latency demands are the continuous increase of devices' processing capabilities, storage, and wireless connectivity options. The multiple radio access technology (multi-RAT) is proposed to satisfy mobile users' increasing needs. The Information-Centric Networking (ICN) paradigm is better tuned (than the current Internet Protocol approach) to support multi-RAT communications. ICN eschews the connection-based content retrieval model used today and has desirable features such as data naming, in-network caching, and device mobility—a paradigm ripe for exploration.

We propose DICE, an ICN forwarding strategy that helps a device dynamically select a subset of its multi-RAT interfaces for communication. DICE assesses the state of edge links and network congestion to determine the minimum number of interfaces required to perform data delivery. We perform simulations to compare DICE's performance with *bestroute2* and *multicast* strategies (part of the named data networking simulator, ndnSIM). We show that DICE is the best of both worlds: providing a higher delivery ratio (0.2–2 times) and much lower overhead (by 2–8 times) for different packet rates.

## CCS CONCEPTS

• **Networks** → *Network protocol design; Network simulations; Wireless access networks;*

## KEYWORDS

ICN; NDN; multi-RAT; 5G; forwarding strategy.

## 1 INTRODUCTION

Untethered mobile devices, such as smartphones, tablets, smart watches, and low-capability wireless devices (forming the Internet of Things (IoT)) are fast becoming the major bandwidth consumers on the Internet. The demand for bandwidth from these devices continues to grow; global mobile traffic increased by over 60% in 2016 compared to 2015 [1]. This increase is largely attributable to streaming services (e.g., Netflix, Hulu, and Youtube), consisting of 54% of the total mobile traffic [1], but edge-generated content proportion is rising. With most of the devices being wirelessly connected, latency of data retrieval and reliability of delivery become important factors. Thus, there is an increasing need for networking

solutions to meet the throughput, latency, and reliability needs of the applications on these end-devices.

As most mobile devices today have two to three different wireless interfaces, concurrent use of multiple radio technologies (multi-RAT) has been proposed as a solution to the problem. This proposal forms one of the core principles of the 5G standard effort [2]. However, the traditional host-centric IP-based paradigm is incapable of efficiently scaling for a large number of devices using multi-RAT [2, 3]. The data name-based information-centric networking (ICN) paradigm, with its in-network caching, name-based routing and forwarding, and better mobility support, holds greater promise [4, 5]. However, there is a need to enhance ICN for effective usage of multi-RAT.

In this paper, we propose DICE, an optimized forwarding strategy which allows an end-device to choose a minimum subset of available multi-RAT interfaces for communication, aimed at achieving desired throughput, reliability, and latency goals. With DICE, the minimum subset selection is tuned dynamically to concurrently transfer data requests and receive data without incurring network congestion. Our strategy is used only at the last-hop (i.e., the wireless edge) and does not require any change to the network core as in [6].

**Contributions:** DICE dynamically selects the minimum set of interfaces required to concurrently transfer interests for (i) high data throughput and low delivery latency, (ii) resilient communication using multiple paths to overcome link or node failure, and (iii) efficient load balancing by distributing requests using an estimate of path capacity and congestion. Different from the “choose-one” interface or “choose-all” interfaces based approaches in the literature, DICE uses an objective function to select a subset of interfaces on which to transmit a request.

We choose NDN, an ICN architecture, due to its popularity and its widely used simulator, ndnSIM [7] (an NDN module for the ns-3 simulator). To validate DICE and demonstrate its efficacy for multi-RAT based communications at the mobile wireless edge we perform simulation analysis using ndnSIM.

The remainder of this paper is organized as follows. In Section 2, we present an overview of existing multi-RAT solutions in cellular networks and ICN. Section 3 details our ICN-based network model. Then, we present our DICE strategy, its objective function, and the DICE algorithm in Section 4. We discuss our implementation details and simulation results in Section 5. Section 6 concludes the paper and discusses our future directions.

## 2 RELATED WORK

Multi-RAT has been proposed to complement existing networks and achieve higher throughput [8], load balancing [9], or fault tolerance [10]. The multi-RAT selection problem has been largely explored in the context of cellular (e.g., WCDMA and LTE) and wireless (e.g., WLANs) access networks [8, 11]. Researchers have been mainly focusing on multi-RAT in heterogeneous cellular networks [12, 13]. Research goal include reducing handoff delays despite larger multi-RAT delays compared to intra-RAT delays [13], fault tolerance [10], load balancing [9], and scalability [2]. These proposed approaches underline the shortcomings of the host-centric IP-based networking paradigm in regard to multi-RAT, including scalability [2] and interoperability [3].

The ICN paradigm holds greater promise for successfully employing multi-RAT. NDN [14] is one of the most popular and well studied ICN architectures, though other exists. All ICN architectures agree on principles such as content naming, name-based routing, and in-network caching. Users' request (called interests in NDN) can be satisfied either by the provider or intermediate nodes that have previously cached the content. In NDN, each node is equipped with three fundamental structures: the *Forwarding Information Base* (FIB), which is the name-based routing table; the *Pending Interest Table* (PIT), which keeps track of outstanding requests; and the *Content Store* (CS) that is essentially a local cache.

NDN, in particular its modular strategy layer, inherently allows the concurrent utilization of multiple interfaces due to its connectionless communication model and individually addressable chunks (pieces of a content). Rossini *et al.* [15] proposed forwarding strategies that either select a single interface for interest forwarding (uniformly at random or in sequence) or utilize all interfaces to flood the network. Udugama *et al.* [16] proposed an on-demand multi-path forwarding strategy that concurrently utilizes all the disjoint paths towards the provider. Detti *et al.* [17] proposed the *fast pipeline filling* strategy, which uses the set of available interfaces with lower outstanding interests than the pipeline capacity. The strategy then selects the interface with the smallest latency to fully utilize the link capacity.

Schneider *et al.* in [18, 19] proposed a strategy in which an application dictates the number of interfaces to be selected by the strategy. The strategy then dynamically selects the desired number of interfaces. However, allowing network-agnostic applications to enforce the number of required interfaces limits the strategy's flexibility. Tourani *et al.* proposed an ICN-based mobile converged network architecture, which leverages multi-RAT communication for better bandwidth utilization [9]. However, this strategy only uses one interface per interest.

To the best of our knowledge, DICE is the first attempt in dynamically selecting an optimized subset of interfaces based on link and network statistics.

## 3 NETWORK MODEL AND ASSUMPTIONS

Our network is composed of a set of providers ( $\mathcal{P}$ ), routers ( $\mathcal{R}$ ), access points ( $\mathcal{AP}$ ), and users ( $\mathcal{U}$ ). We assume that each user's device ( $u \in \mathcal{U}$ ) is equipped with multiple wireless technologies  $\mathcal{I}_u$  (also representing different interfaces), such as WiFi, 5G, WiMax, and Bluetooth. The device is connected to the core networks through multiple  $\mathcal{AP}$ s as shown in Fig 1. To represent ICN, we select the

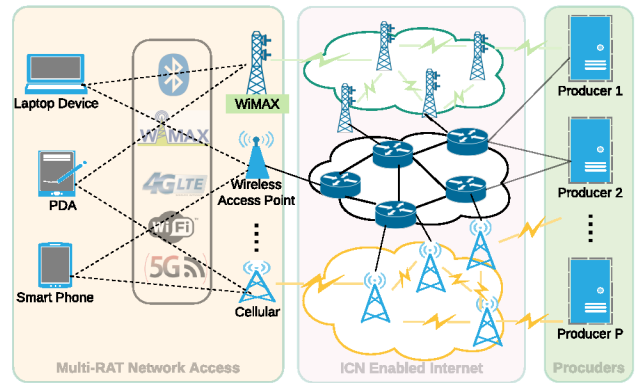


Figure 1: The network model and illustration of Multi-RAT devices.

NDN architecture on account of its popularity. Fig 1 illustrates our network setup and the multi-RAT access network model.

A device  $u$  can send its interests via a set  $\mathcal{I}'_u \subseteq \mathcal{I}_u$  of interfaces simultaneously using the corresponding technologies. Interests will travel through multiple paths that may converge at different levels in the core routers. The core routers aggregate interests when a new interest that matches an outstanding interest is received. While multi-RAT may introduce data redundancy in the network, it will also improve communication reliability and increase rate of successful delivery. We assume that each interest generated by an application has an associated desired latency  $\delta$  (assumed to be a system wide constant in this work) and a targeted probability of success  $TP$ . Parameter  $TP$  is an application-specific parameter chosen according to quality-of-experience (QoE) constraints. A higher  $TP$  value translates to stricter application QoE requirements (e.g., lower latency and higher reliability). Thus, DICE's strategy aims at achieving the desired latency  $\delta$  and targeted probability of success  $TP$  when it selects interfaces for interest forwarding.

## 4 DICE FORWARDING STRATEGY

In this section, we will discuss DICE in detail. We start with a brief overview of our approach, before presenting the detailed DICE objective function and algorithm. Lastly, we describe DICE's implementation.

### 4.1 Brief Overview

With DICE, an end-device  $u$  proactively collects link and network statistics to make an informed selection of the least number of suitable interfaces for interest forwarding. Each device  $u$  maintains a *congestion window* (in principle similar to TCP), *interest drop rate*, and *end-to-end latency* statistics per interface-producer pair  $(i_u, p)$ , for interface  $i_u$  and producer  $p$ . In the rest of the paper, as we use device  $u$  for illustration, we interchangeably use  $(i, p)$  and  $i$ , instead of  $(i_u, p)$  and  $i_u$ , respectively.

**Congestion window:** A Congestion window  $W_{(i,p)}$  is used by DICE to shape traffic load (in bytes) across available interfaces and avoid network congestion (more details in Section 4.3). Upon successful delivery of a packet DICE slowly increases this window, and rapidly decreases it when failures occur (i.e., timeout).

**Window increase:** When a packet  $k$  from producer  $p$  gets delivered to interface  $i$ , the congestion window  $W_{(i,p)}$  is increased by the size of data packet  $k$  (in bytes).

**Window decrease:** Packet loss is triggered when an interest times out. When interest  $k$  for producer  $p$ , sent on interface  $i$  times out, the congestion window  $W_{(i,p)}$  is decreased by a factor  $\rho$  as given by  $W_{(i,p)} = W_{(i,p)}(1 - \rho)$ . We experimented with multiple values of  $\rho$  and chose  $\rho = 0.1$  for all our experiments.

**Interest Drop:** The interest drop rate for a given interface  $i$  and producer  $p$ ,  $DR_{(i,p)}$ , is measured for packet  $k$  in a sequence of packets  $1, \dots, k, \dots, K$  by the following equation:

$$DR^k(i, p) = \gamma d(k) + (1 - \gamma)DR^{k-1}(i, p), \quad (1)$$

where  $d(k) = 0$  if packet  $k$  is successfully received by  $i$ , and  $d(k) = 1$  otherwise. In this paper, we use  $\gamma = 0.833$  [9].

**End-to-end latency:** In addition to loss, the latency statistics for a packet  $k$  are approximated after receiving the packet  $k-1$ . The estimated round trip time,  $RTT_e^k$ , and the deviation of the round trip time,  $RTT_d^k$  for the packet  $k$  are calculated according to an EWMA, as:

$$\begin{aligned} RTT_e^k(i, p) &= (1 - \alpha)RTT_e^{k-1}(i, p) + \alpha RTT_s^{k-1}(i, p) \\ RTT_d^k(i, p) &= (1 - \beta)RTT_d^{k-1}(i, p) + \\ &\quad \beta |RTT_s^{k-1}(i, p) - RTT_e^k(i, p)|, \end{aligned} \quad (2)$$

where  $RTT_s^{k-1}$  is the measured RTT upon the reception of the packet  $k-1$ . We choose values of  $\alpha = 0.125$ , and  $\beta = 0.25$  as in [20, 21].

DICE uses the RTT estimates from Eqn. 2 to calculate the probability that packet  $k$  will have  $RTT_s^k$  less than  $\delta$ , if requested over interface  $i$ , as follows:

$$P_{(i,p)}^k(x \leq \delta), \quad (3)$$

Eqn. 3 evaluates the probability that a random variable  $x$  such that  $x \sim \mathcal{N}(RTT_e^k(i, p), (RTT_d^k(i, p))^2)$  ([22]) takes a value less than or equal to  $\delta$ .

## 4.2 Interface Selection Objective Function

When an interest is received from an application, the DICE strategy computes an objective function which estimates the minimum number of suitable interfaces to forward current interest on for meeting the desired latency  $\delta$  and the targeted probability of success  $TP$ .

The strategy calculates the probability of the data reaching back to the device within time  $\delta$  when fetching data packet  $k$  from producer  $p$  using a given interface  $i$  as:

$$\mathbb{P}_{(i,p)}^k = (1 - DR_{(i,p)}^k)P_{(i,p)}^k(x \leq \delta) \quad \forall \quad i \in \mathcal{I}'_u, \quad (4)$$

where  $\mathcal{I}'_u$  is the set of interfaces available to reach producer  $p$ .

Therefore, the probability of meeting desired round-trip latency  $\delta$  using set of interfaces  $Z \subseteq \mathcal{I}'_u$  is:

$$CP_Z = 1 - \prod_{i \in Z} (1 - \mathbb{P}_{(i,p)}^k). \quad (5)$$

Thus, DICE selects the minimum subset  $Z$  which satisfies  $CP_Z \geq TP$ .

## 4.3 DICE Algorithm

In Algorithm 1, we describe the steps taken by DICE when the application sends an interest to be forwarded. On receiving an interest for packet  $k$  produced by  $p$ , the strategy layer in  $u$ 's NDN stack selects the interfaces ( $\mathcal{I}'_u$ ) available as next hops for the current interest (Line 1). The interfaces are checked for eligibility (*HasAvailableBW*), and added to set  $M$  (Lines 3-7). An interface  $i$  is considered eligible (has available bandwidth) if  $W_{(i,p)} > OI_{(i,p)}$ , where  $OI_{(i,p)}$  is the cumulative size of the current outstanding interests for  $(i, p)$ .

DICE calculates  $\mathbb{P}_{(i,p)}^k$  as in Eqn. 4 for all  $i \in M$  (Line 5). Then DICE returns  $Z$ , the smallest number of interfaces from  $M$  to meet  $TP$  (Lines 9-14) as shown in Eqn. 5. If  $CP_Z < TP$  even when  $Z = M$  (all available interfaces considered), then the interest is sent on all  $i \in Z$ , in order to maximize the possibility of successfully meeting the desired latency. However, if  $M = \phi$ , that is, no interface in  $\mathcal{I}'_u$  has adequate available bandwidth (Line 15). DICE iterates through all the interfaces in  $\mathcal{I}'_u$ , computes  $S_i$ , and chooses the most suitable interface  $i \in \mathcal{I}'_u$  such that score  $S_i$  (Line 18), is maximized (Line 20). We note that  $S_i$  is used to weigh the interfaces based on their load, where the load fraction on  $(i, p)$  is computed as  $L_{(i,p)} = \frac{OI_{(i,p)}}{W_{(i,p)}}$ .

By proactively selecting the least loaded interface, DICE distributes its interests across all available interfaces to help load-balancing while avoiding overloading a single interface and perhaps the path to the producer. Upon receiving a data packet, the loss and

---

### Algorithm 1 DICE Forwarding Strategy

---

**Input:** Interest  $k$  to be forwarded toward producer  $p$ .

**Output:**  $Z \subseteq \mathcal{I}_u$ , set of interfaces selected for forwarding.

```

1:  $\mathcal{I}'_u \leftarrow getNextHops(k)$ 
2:  $M = \phi$ 
3: for face  $i \in \mathcal{I}'_u$  do
4:   if HasAvailableBW( $i, p$ ) then
5:      $M = M \cup (i, \mathbb{P}_{(i,p)}^k)$ 
6:   end if
7: end for
8:  $Z = \phi, M' = M$ 
9: if  $M \neq \phi$  then
10:  while ( $CP_Z < TP$  &&  $M' \neq \phi$ ) do
11:     $j = \arg \max_i \{M' \mid \mathbb{P}_{(i,p)}^k\}$ 
12:     $M' = M' - \{j\}$ 
13:     $Z = Z \cup \{j\}$ 
14:  end while
15: else
16:  for  $i \in \mathcal{I}'_u$  do
17:     $L_{(i,p)} = \frac{OI_{(i,p)}}{W_{(i,p)}} \quad \backslash OI_{(i,p)}$ : outstanding requests count
18:     $S_i \leftarrow \mathbb{P}_{(i,p)}^k / L_{(i,p)}$ 
19:  end for
20:   $j = \arg \max_i (S_i)$ 
21:   $Z = \{j\}$ 
22: end if
23: return  $Z$ 

```

---

latency statistics are updated as shown in Eqns. 1 and 2 and the congestion window is increased by the size of the packet received. However, if DICE times-out due to unsuccessful delivery that may be caused by congestion, loss statistics for the corresponding pair are updated according to Eqn. 1 and the congestion window is reduced by  $\rho$ .

The running time complexity of Algorithm 1 is  $O(|\mathcal{I}'_u|)$

## 5 SIMULATION RESULTS

We evaluate the performance of DICE and quantify its gain under different network conditions using simulations. In this section, we first present our simulation setup, then we discuss the list of parameters and metrics utilized to measure the performance of DICE. Finally we evaluate DICE's performance in comparison with the state of the art.

### 5.1 Simulation Setup

**5.1.1 Strategy Implementation.** We compare DICE's performance with two other strategies available as part of ndnSIM: *multicast* (MCAST) and *bestroute2* (BEST) [7]. On receiving an interest, MCAST forwards it on all available interfaces simultaneously. Whereas, BEST chooses a single interface out of the available interfaces. This selection is based on the number of hops to the producer through a given interface, and only switches to an alternative interface (path) upon receipt of a negative acknowledgment from the upstream router(s).

In our simulations, all strategies are deployed only on the client nodes. The core only implements a stateful forwarding strategy, which supports interest aggregation. This stateful forwarding strategy propagates negative acknowledgments (NACKs) from the core all the way to the client. As we are studying the effect of strategy choice at the network edge, we do not deploy any dynamic routing in the core of the network, and do not simulate link failures nor losses due to signal noise or mobility. We *disable in-network caching*, as it would affect RTT statistics in this preliminary work. In future work, we plan to extend our strategy to handle effects of in-network caching.

**5.1.2 Network Topology.** We simulate ten different network topologies, each generated with a different random seed. Measurements are obtained for each topology, then averaged and presented in our simulation results. For each topology, we generate a scale-free network of 200 core nodes ( $\mathcal{R} = 200$ ). All the links within the core network have 1 ms propagation delay and 10 Mbps bandwidth. Core nodes implement a first-in-first-out queue length of 20 packets. We deploy  $|\mathcal{P}| = 5$  nodes as producers, connected randomly to five nodes of the network core. We randomly chose 30 nodes in the network core to represent 30 access points ( $|\mathcal{AP}| = 30$ ). The APs connect to the end user devices and are chosen such that the devices are at least five hops away from each producer (chosen to simulate average path lengths on the Internet).

We assign each of three wireless technologies (LTE, WiFi, WiMAX) to the ten APs. Each edge link is assigned a 10 ms propagation delay, while the link bandwidth were chosen as LTE = 0.5 Mbps, WiFi = 1 Mbps, and WiMAX = 2 Mbps.

We connect 40 client devices ( $C = 40$ ) to the 30 APs. Each device is connected to three APs in total (each access point is a different

technology). All clients are running a Constant Bit Rate (CBR) consumer application with interest transmission rate in packets per second (pps) chosen from: {80, 160, 240, 300}. Interests are generated from each client to two randomly chosen producers. We choose  $\delta = 65$  ms,  $TP = 90\%$ ,  $\alpha = 0.125$ ,  $\beta = 0.25$ ,  $\gamma = 0.833$  as in [9], and the interest timeout (PIT timeout) value as 1000 ms.

### 5.2 Metrics & Parameters

We compare the performance of DICE with BEST and MCAST using the following metrics: (i) the *packet delivery ratio* (the ratio between the number of data packets received by the client and the number of interests), (ii) the *end-to-end latency* (the time elapsed between sending an interest and receiving the corresponding data), (iii) the *normalized network interest overhead* (RTT), and (iv) the *average number of active interfaces*.

The **normalized overhead** is the ratio of the average number of interest packets propagated within the network and the total number of interests generated by all clients'. We define  $\mathcal{W} = \mathcal{P} \cup \mathcal{R}$  as the set of non-client nodes in the whole network. The overhead is computed as

$$\frac{\sum_{a \in \mathcal{W}} \text{IncomingInterests}(a)}{\sum_{u \in \mathcal{U}} \text{InterestsGenerated}(u)},$$

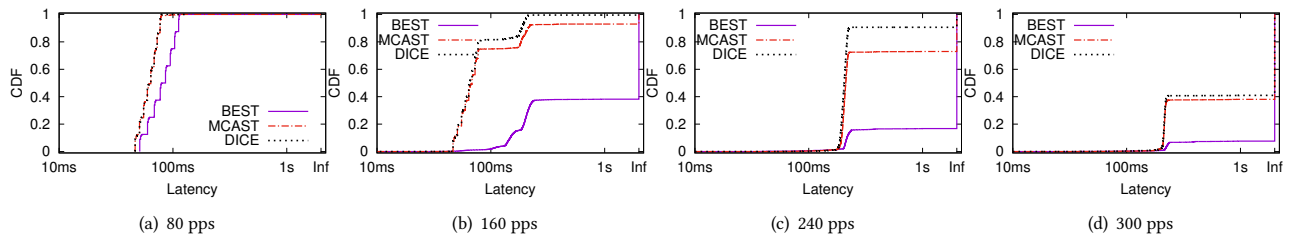
where  $\text{IncomingInterests}(a)$  is the number of interests received by a node  $a \in \mathcal{W}$ , and  $\text{InterestsGenerated}(u)$  measures the number interests generated by a client device  $u$ . For better illustrating **end-to-end latency** and packet loss, when interests are dropped, we define the end to end latency to be a large number (defined as  $\text{Inf} = 2$  s). The average end-to-end delay is measured as the average delivery latency of all successfully retrieved data packets.

### 5.3 Latency Measurements

We plot, in Fig. 2, the cumulative distribution function (CDF) of the end-to-end latencies observed over all topologies for varying data rates. We denote by "Inf" the latency for unsuccessful data delivery as described earlier.

Fig. 2(a) shows that, in an underloaded network, all three strategies have the same success rate, i.e. no "Inf" values are reported. However, DICE and MCAST register lower latencies compared to BEST, which can be attributed to their use of multiple interfaces. This increases their chances of finding a faster path whose routers have negligible queued packets. We note that even though DICE chooses only a subset of paths that MCAST does, its performance is as good or better than MCAST.

As the packet rates rise (160 pps-240 pps) (Fig. 2(b)-(c)), the delivery ratio of BEST decreases substantially due to congestion-driven packet losses in the network. MCAST has similar latencies as DICE for roughly 70% of the data requested, but then suffers from packet loss as it injects more packets into the network and ultimately overloads the paths. DICE takes a proactive approach to load balancing and reacts to congestion by reducing the number of copies of the packets sent, and hence performs more than 25-30% better. In Fig. 2(d), where the packet rate is 300 pps, packet loss is very high but DICE outperforms both MCAST (marginally) and BEST. Given that MCAST is a passive strategy that is agnostic to the network conditions, it floods the network without considering congestion.

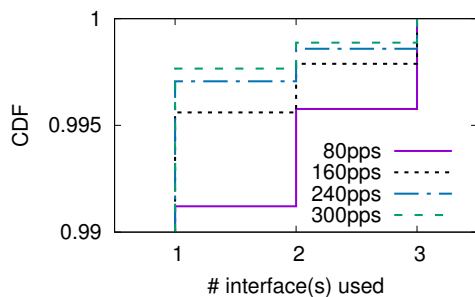


**Figure 2: Comparing DICE, MCAST, and BEST end-to-end latency distribution. Note that “Inf” indicates the interest timeout value.**

BEST, as a reactive strategy, does not switch forwarding routes often enough, as it relies on congestion NACKs or other control messages from routers upstream. This slow reactive strategy contributes to lower network overhead when the network is underloaded, but leads to significant congestion when packet rates are high. BEST also exhibits oscillating behavior on its interfaces when it congests one and goes to the next interface and back. MCAST also does exhibit oscillations, but does a better job at reducing overall congestion. At high congestion-level packet rates ( $> 240$  pps), DICE still performs better, but the high packet rates still causes congestion-driven packet losses.

#### 5.4 Interface Selection

Fig. 3 shows that DICE utilizes mostly (more than 99% of the time) one interface to send the interests throughout the entire simulation. However, we have shown DICE outperforming BEST, which also uses a sole “best” interface. This is attributable to DICE’s use of link and loss statistics to perform load balancing while maximizing the goodput and minimizing the end-to-end latency. DICE reduces the number of interfaces when it detects that the network is overloaded or congested. For instance, when the interest load doubles from 80 pps to 160 pps, DICE reduces multi-interface selection by half. In the higher load cases, DICE makes multiple concurrent interface selections less often, but spreads the requests across the interfaces.



**Figure 3: CDF of the number of interfaces used by DICE at different interest rates**

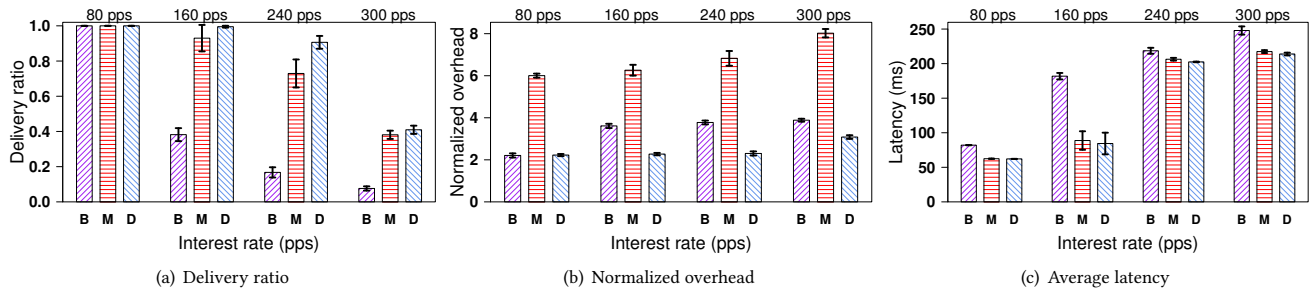
#### 5.5 Handling Network Congestion

In Fig. 4, we compare the average delivery ratio, normalized overhead, and average latency of DICE, BEST and MCAST; the subfigures also show error bars. DICE outperforms all other strategies with respect to delivery ratio as shown in Fig. 4(a). While all strategies achieve 100% success ratio at the lowest interest rate (80 pps), delivery failures increase when the interest rate increases due to network congestion at core links. DICE registers the lowest failure counts compared to both BEST and MCAST. It achieves 75% better success rate than BEST, and 20% better than MCAST when the interest rate is 240 pps. At very high interest load, that is, 300 pps, DICE outperforms MCAST by less than 4% due to the high network congestion.

As shown in Fig. 4(b), DICE has the least normalized overhead compared to all forwarding strategies. In fact, at 240 pps, DICE has as low as 2.3 interests in the network for every interest created by the application, when compared to BEST’s 3.77 and MCAST’s 6.82 ( $\sim 3\times$ ). At 80 pps interest rate, where the network is not congested, DICE performs as good as BEST, which typically selects the fastest route to the provider. At higher loads, DICE shows lower network overhead than BEST due to its lower packet loss, which results in less number of application interest retransmissions. Due to high packet losses in BEST, while some clients need to retransmit for the lost packets, other clients request subsequent data packets. This increases the number of unique interests in the network and hence decreases the chance of interest aggregation. MCAST has the highest network overhead due to its transmission over all interfaces. Thus with very little overhead, DICE performs as well as MCAST in delivery.

DICE’s load balancing helps achieve lower round-trip-time delays compared to BEST and even MCAST (as shown in Fig. 4(a)). DICE achieves 50% end-to-end delay reduction compared to BEST at 160 pps; reaching providers in less than 84 ms on average. Moreover, while MCAST and DICE achieve similar delay at low interest rates, DICE outperforms MCAST when the load increases in packet delivery while still maintaining the latency. We note that because we calculate round-trip-time delays, we only account for interest-data packet pairs that have been successful. In that context both MCAST and DICE have almost the same latencies as they end-up choosing the best path among all paths.

In general, DICE outperforms MCAST in underloaded network conditions while incurring a smaller overhead. It forwards interests



**Figure 4: Comparing BEST (B), MCAST (M), and DICE (D) performance: (a) average delivery ratio, (b) average number of interests in network per application request, and (c) the average content delivery latency.**

using the best face or a set of best faces to increase QoE at the network edge. However, the passive approach taken by MCAST in continuously sending interests on all interfaces contributes to network congestion. BEST adopts a reactive approach that shows many limitations due to its late detection of congestion, resulting in higher latencies and more overhead in the network.

## 6 CONCLUSION AND FUTURE WORK

We have proposed DICE, a fault tolerant, load balancing, and network-aware forwarding strategy, that leverages multi-RAT in the ICN wireless edge to improve delivery ratios, network overhead, and latencies. DICE takes a proactive approach that estimates the network load and selects the minimum set of interfaces to attempt to meet application specific QoS constraints. We compare the performance of DICE with two popular NDN strategies: bestroute2 and multicast. DICE outperforms the strategies by achieving up to two times more successful deliveries and injecting half to one-tenth the amount of overhead packets into the network.

To the best of our knowledge, DICE is the first step towards assessing the gain of using multi-RAT at the wireless edge in ICN. We believe that DICE can be further enhanced by an adaptive strategy at the core of the network also. We plan to investigate the impact of enabling caching and of mobility on efficiency of DICE's approximation and objective functions. Additionally, we plan to improve our optimization function by incorporating the energy consumption of different wireless technologies.

## ACKNOWLEDGEMENT

This work was supported in part by the US National Science Foundation Grants 1241809, 1345232, and 1719342 and the Intel Corporation. The information reported here does not reflect the position or the policy of the federal government.

## REFERENCES

- [1] D. Webster. Cisco visual networking index (VNI). *Global Forecast Update*, page 6, 2017.
- [2] O. Galinina, A. Pyattaev, S. Andreev, M. Dohler, and Y. Koucheryavy. 5G multi-RAT LTE-WiFi ultra-dense small cells: performance dynamics, architecture, and trends. *IEEE Journal on Selected Areas in Communications*, 33(6):1224–1240, 2015.
- [3] S. Cavalli and F. Meago. Common radio resource management method in a multi-rat cellular telephone network, 2007. US Patent 7,224,977.
- [4] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Brannard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.
- [5] S. Tarkoma, M. Ain, and K. Visala. The publish/subscribe internet routing paradigm (psirp): Designing the future internet architecture. *Towards the Future Internet*, page 102, 2009.
- [6] G. Carofoglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. Optimal multipath congestion control and request forwarding in information-centric networks. In *IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, 2013.
- [7] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2, NDN, November 2016.
- [8] V. Marques, R. Aguiar, C. Garcia, I. Moreno, C. Beaujean, E. Melin, and M. Liebisch. An ip-based qos architecture for 4g operator scenarios. *IEEE Wireless Communications*, 10(3):54–62, 2003.
- [9] R. Tourani, S. Misra, and T. Mick. IC-MCN: An architecture for an information-centric mobile converged network. *IEEE Communications Magazine*, 54(9):43–49, 2016.
- [10] A. Ravanshid, P. Rost, D. Michalopoulos, et al. Multi-connectivity functional architectures in 5G. In *IEEE International Conference on Communications Workshops (ICC)*, pages 187–192, 2016.
- [11] X. Gelabert, O. Sallent, J. Pérez-Romero, and R. Agustí. Performance evaluation of radio access selection strategies in constrained multi-access/multi-service wireless networks. *Computer Networks*, 55(1):173–192, 2011.
- [12] X. Gelabert, J. Pérez-Romero, O. Sallent, and R. Agustí. On the suitability of load balancing principles in heterogeneous wireless access networks. In *Wireless Personal Multimedia Communications Symposium*, 2005.
- [13] T.F.M. Hendriksen. UMTS and LTE/SAE handover solutions and their comparison. In *Proc. 11th Twente Student Conf. IT*, pages 1–9, 2009.
- [14] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, and K. Claffy. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.
- [15] G. Rossini and D. Rossi. Evaluating ccn multi-path interest forwarding strategies. *Computer Communications*, 36(7):771–778, 2013.
- [16] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg. An on-demand multi-path interest forwarding strategy for content retrievals in ccn. In *Network Operations and Management Symposium (NOMS)*, pages 1–6. IEEE, 2014.
- [17] A. Detti, C. Pisa, and N. B. Melazzi. Modeling multipath forwarding strategies in information centric networks. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 324–329, 2015.
- [18] K. M. Schneider and U. R. Krieger. Beyond network selection: Exploiting access network heterogeneity with named data networking. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 137–146. ACM, 2015.
- [19] K. M. Schneider, K. Mast, and U. R. Krieger. Ccn forwarding strategies for multihomed mobile terminals. In *International Conference and Workshops on Networked Systems (NetSys)*, pages 1–5. IEEE, 2015.
- [20] J. Chu, Y. Cheng, N. Dukkkipati, and M. Mathis. Increasing TCP's initial window, 2013. <https://tools.ietf.org/html/rfc6928>.
- [21] J. F. Kurose and K. W. Ross. Computer networking: A top-down approach. *Transport*, 9:26, 2001.
- [22] M. Mitzenmacher and R. Rajaraman. Towards more complete models of tcp latency and throughput. *The Journal of Supercomputing*, 20(2):137–160, 2001.