



Towards Knowledge-Driven Mobility Support

Zhongda Xia and Yu Zhang(✉)

Harbin Institute of Technology, Harbin 150001, Heilongjiang, China
{xiazhongda,yuzhang}@hit.edu.cn

Abstract. Mobility refers to the ability to conduct “seamless” communication with network entities whose network location constantly changes. This paper examines the mobility support problem in IP and Named Data Networking (NDN), and identifies two dimensions in the mobility support solution space: the host dimension and data dimension. Existing host dimension solutions have exhausted the available design choices, and have not been able to achieve new breakthroughs in performance. Recognizing this limitation, this paper proposes a novel knowledge dimension. In the knowledge dimension, two knowledge-driven mobility support approaches, Topology-driven Intermediate Placement (TIP) and Trajectory-driven Reachability Update (TRU), are proposed. These approaches exploit knowledge such as network topology and movement trajectory to tweak the network and network services for better overall mobility support performance. A cross-architectural quantitative evaluation framework covering two communication scenarios and 5 quantifiable metrics is proposed to evaluate mobility support performance. Experiment results show that the knowledge-driven approaches significantly improve mobility support performance, demonstrating the potential of the knowledge-driven vision for providing better mobility support.

Keywords: Mobility · Mobility support approach · Knowledge-driven networking · Internet architecture

1 Introduction

With the proliferation of the computer networking technology, applications that thrive on the cyberspace are booming in every aspect of human activities. Along comes the ever-increasing demand on innovations that can satisfy novel needs such as supporting mobility [28], security [12,26], and autonomy [24]. Specifically, mobility refers to the ability to conduct “seamless” communication with network entities (e.g., subnets, hosts, applications or data) whose network location constantly changes. This paper examines the mobility support problem in IP and NDN, and identifies two dimensions in the mobility support solution

space: host dimension and data dimension. Host dimension solutions employ a certain intermediate to track the location of mobile entities; such solutions have exhausted the available design choices in the host dimension, and have not been able to achieve new breakthroughs in performance. Recognizing this limitation, this paper proposes a new design dimension, namely the knowledge dimension. In the knowledge dimension, knowledge such as network topology and movement trajectory can be used to tweak the network to both improve user experience and reduce network overhead. Under this vision, two novel knowledge-driven mobility support approaches, Topology-driven Intermediate Placement (TIP) and Trajectory-driven Reachability Update (TRU), are proposed. TIP places the intermediate at a location to shorten most forwarding paths, while TRU optimizes the way reachability information is updated to reduce signaling overhead.

To evaluate mobility support performance across network architectures, a cross-architectural quantitative evaluation framework is proposed. Under two communication scenarios, 5 quantifiable metrics reflecting user experience and network overhead are defined. The calculation formulae for these metrics are given by analyzing the key differences among mobility support approaches and architectures. By feeding network topology and movement track to the formulae, mobility support performance can be quantified via numerical simulations. Experiment results show that the proposed knowledge-driven approaches are superior according to all 5 metrics, demonstrating the potential of the knowledge-driven vision for providing better mobility support.

2 Background

2.1 IP and NDN Architecture

IP and NDN transmit application-layer information across a network by specifying a network-wide consensus including network identifier namespace, packet format, and per-hop behaviors (PHB). In IP, an IP address identifies a network interface, and each IP packet carries two IP addresses, namely source address and destination address. Each IP node maintains a Forwarding Information Base (FIB), which maps IP address (prefix) to network interface. Upon receiving an IP packet, the carried destination address is matched against the FIB using the longest prefix match (LPM) algorithm to determine the output interface. In NDN, a data name identifies a piece of named data. An Interest packet carries a data name, and is sent by data consumers as a request for data; a Data packet carries a data name and represents a piece of named data. Each NDN node maintains a FIB, a Pending Interest Table (PIT), and a Content Store (CS). The FIB maps data name (prefix) to network interface. The PIT keeps track of incoming Interests, including storing the input network interfaces. The CS caches received Data. Upon receiving an Interest, the Interest is forwarded according to the FIB if no matching Data is found in the CS and the Interest is not in the PIT, the incoming interface is then stored in the PIT. Upon receiving

a Data, matching Interest is looked up in PIT, and if a match is found, Data is sent through all recorded interfaces towards consumers.

2.2 The Mobility Support Problem and Solution Space

Under the computer networking paradigm, mobility often refers to the ability to conduct “seamless” communication with network entities whose network location constantly changes. In IP, the mobility support problem concerns how to deliver IP packets to mobile entities using the same IP address. In NDN, there are two types of mobility, namely consumer mobility and producer mobility. Consumer mobility is natively supported in NDN since Data is forwarded along the reverse path of Interest. To retrieve lost Data after relocation, a mobile consumer simply retransmits the Interest. The NDN producer mobility problem is rather similar to the IP mobility problem, Interests need to be forwarded to/toward the mobile producer according to FIBs.

This paper considers IP and NDN producer mobility support problem, which share a common design space. Solutions under this common design space take a host-centric approach by maintaining the persistence and reachability of the network identifiers associated with mobile network entities. In NDN, the design space is expanded by data-centric solutions that attempt to make data constantly available by moving or disseminating data [27]. Thus, we depict the current design space as a two-dimensional space consisting of a host and a data dimension. Along the host dimension, host-centric solutions employ an intermediate to track the network location of mobile entities. Along the data dimension, solutions employ an intermediate to rendezvous requests for data with requested data, which may rely on host dimension approaches for reachability.

3 Cross-architectural Host Dimension Approaches

Host dimension solutions can be broken down into two processes: reachability information update and packet forwarding. The former updates the reachability information of a mobile entity’s identifier, while the latter uses the updated information to deliver packets to the mobile entity. Considering the two processes, existing host dimension solutions may be categorized as 4 types of cross-architectural approaches, namely proxy-based, resolution-based, routing-based and trace-based approach. In the rest of this section, M represents a mobile device, S represents an immobile server that provides mobility support service, and C represents the entity that communicates with M . The term relocation refers to the event that M changes its Point-of-Attachment (PoA) to the network.

3.1 Proxy-Based Approach (Proxy)

Proxy maintains the mapping between M ’s network identifier and a globally reachable network identifier associated with M ’s current PoA, namely M ’s locator. In the reachability information update process, M sends a signaling packet

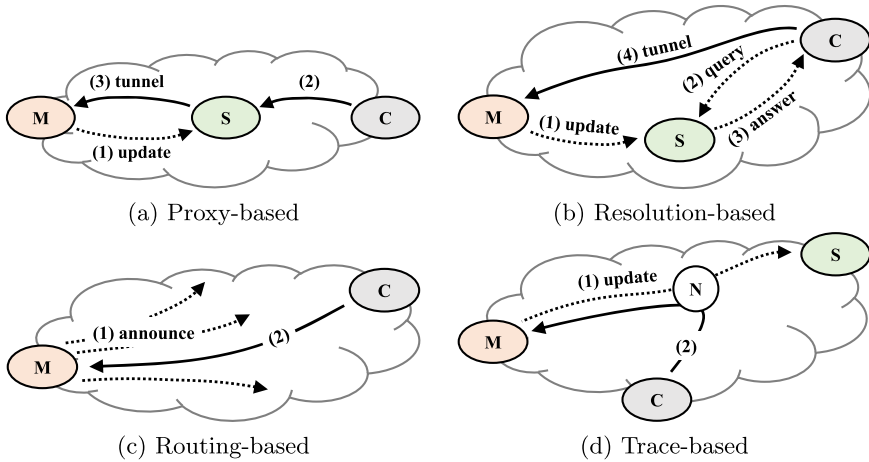


Fig. 1. Host dimension approaches

to S after a relocation (Fig. 1a: step 1); the signaling packet identifies M and contains M 's current locator; S then updates the mapping accordingly. In the packet forwarding process, M 's network identifier is made constantly reachable at S , so that packets destined to M always reach S first; upon receiving such packets, S will tunnel the packet to M using M 's locator (Fig. 1a: step 3).

Compared with the described approach, existing work in IP [17, 22, 25] and NDN [4, 10, 13, 15] differs in three major aspects: (1) S may be deployed in a distributed manner as multiple cooperative instances; (2) tunnelling may be implemented in various ways; and (3) the protocol may or may not be transparent to endpoints. Considering mobility support performance evaluation, (2) is irrelevant because tunnelling by definition produces a direct path to M ; (3) is irrelevant because it is mostly an engineering choice; (1) is relevant, and this research does not consider distributed deployment.

3.2 Resolution-Based Approach (Resolution)

Like Proxy (Sect. 3.1), Resolution also maintains the mapping between M 's network identifier and locator, thus shares the same reachability information update process. In the packet forwarding process, C will query S about M 's locator, producing an exchange of packets between C and M (Fig. 1b: step 2 and 3). After learning M 's locator, C will directly tunnel packets to M .

Resolution shares with Proxy the same differences between existing work in IP [2, 9, 20, 29] and NDN [1, 10, 13], and the same considerations regarding these differences, thus not repeated here.

3.3 Routing-Based Approach (Routing)

Routing updates the FIB on each node to produce optimal forwarding paths to M 's current PoA. In the reachability information update process, M broadcasts signaling packets into the network after a relocation (Fig. 1c: step 1). As a result, FIBs are updated to establish shortest forwarding paths for M 's network identifier from anywhere in the network. In the packet forwarding process, packets sent by C will follow FIBs to reach M through a shortest path.

Compared with the described approach, existing solutions in IP [11, 19] and NDN [3] differ in two major aspects: (1) propagation range of signaling messages; (2) route calculation. Both are relevant to evaluation, and Routing behaves as follows: for (1), signaling messages will be flooded to reach every node; for (2), shortest forwarding paths are always produced.

3.4 Trace-Based Approach (Trace)

Trace establishes forwarding paths to M on top of existing routing information. In the reachability information update process, M sends a signaling packet to S after a relocation (Fig. 1d: step 1), making each node on the shortest path between M and S record a FIB entry for M 's network identifier pointing to M , namely a trace, constructing a forwarding path from S to M . In the packet forwarding process, packets destined to M will be first forwarded toward S , and eventually meet traces either at S or before (e.g., at node N as marked in Fig. 1d). After meeting a trace, the packet will follow traces to reach M .

Compared with the described approach, existing work in IP [7, 18] and NDN [8, 13, 23, 28] differs in two major aspects: (1) how traces are created, and (2) how traces are maintained. Both are relevant, and Trace behaves as follows: for (1), signaling packets from M will directly establish the complete forwarding path from S to M ; for (2), traces will disappear upon relocation.

4 The Knowledge Dimension of Mobility Support

Existing solutions have exhausted the design choices in available design space, thus can only make tradeoff decisions to balance between user experience and network overhead. This paper proposes the knowledge dimension in the mobility support solution space. This new dimension offers a new set of design choices that may nurture solutions that comprehensively improve mobility support performance. In the knowledge dimension, various types of knowledge including current states (ground truth knowledge, e.g., network topology, placement of immobile entities), predictable states (inferred knowledge, e.g., movement track of mobile entities), and task-specific strategies (driver knowledge, i.e., how various information should be used to meet application needs) are exploited as the knowledge to drive the network to satisfy stringent performance requirements.

The expanded mobility support solution space is illustrated in Fig. 2. In the knowledge dimension, knowledge-driven approaches (or KD approaches for

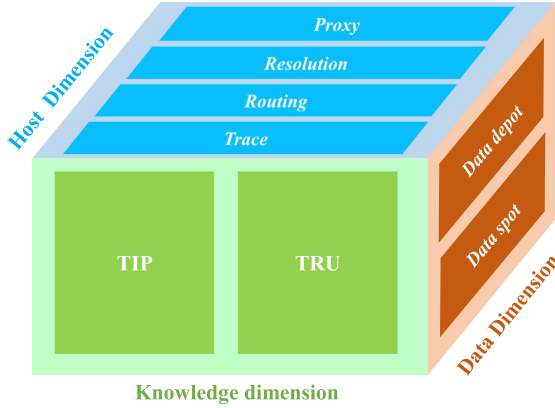


Fig. 2. The expanded mobility support solution design space

short) take knowledge as input, and output operational decisions that tweak the network and network services for better mobility support performance. This paper proposes two novel KD approaches: (1) Topology-driven Intermediate Placement (TIP), and (2) Trajectory-driven Reachability Update (TRU). Two knowledge-driven approaches can be combined with host dimension approaches to improve user experience AND reduce network overhead. In the rest of this section, we first describe the two knowledge-driven approaches, then briefly sketch out the architectural requirements to support them.

4.1 Topology-Driven Intermediate Placement (TIP)

TIP places the intermediate at an optimal network location in order to optimize forwarding paths, and may be combined with host dimension approaches that rely on an intermediate without affecting the internal mechanisms. Given a network topology and network locations (i.e., nodes in the topology) of relevant entities, an optimal network location of the intermediate is calculated using the betweenness centrality algorithm. The intermediate is then migrated to this optimal location. TIP employs the following ground truth knowledge:

- network topology graph (G): a graph representing the actual topology;
- network location set of mobile endpoints ($MLoc$): a set consisting of all potential network locations of the mobile endpoints that use the mobility support service provided by the intermediate in question;
- network location set of correspondent endpoints ($CLoc$): a set consisting of the network locations of the endpoints that communicate with the mobile endpoints in question.

The driver knowledge employed by TIP is an algorithm that outputs an optimal location. Specifically, the betweenness centrality of each node in a given

topology G is calculated for a set of source nodes $CLoc$ and a set of destination nodes $MLoc$, the node with the highest value is selected as the output.

TIP then drives the network to change the location of the intermediate. Let the optimal location be B , and the intermediate's current network location be A , the migration of the intermediate from A to B may involve the following steps:

1. A collection of executables and runtime states of the service provided by the intermediate is encapsulated, and transferred from A to B ;
2. routing announcements are made from B if necessary, to lead packets to B .

4.2 Trajectory-Driven Reachability Update (TRU)

TRU optimizes the reachability information update process of each host dimension approach when fed with a network topology and predicted movement track of a mobile endpoint. TRU employs the following ground truth knowledge:

- network topology graph (G): a graph representing the actual topology;
- underlying host dimension approach (App): which host dimension approach is already deployed and used.

TRU also employs an inferred knowledge:

- mobile endpoint's network location sequence ($Track$): a temporal sequence of network locations that may be inferred from movement pattern or trajectory that may further be based on auxiliary information about the mobile endpoint; for a mobile endpoint that relocates n times, the sequence is $Track = (t_i, P_i) (1 \leq i \leq n)$, where t_i is time when the i th relocation finishes, and P_i is the mobile endpoint's PoA after the i th relocation.

The driver knowledge employed by TRU is a centralized reachability information update mechanism that exploits the knowledge above to reduce signaling traffic. According to App , the mechanism exploits other knowledge as follows:

- Proxy/Resolution-based: schedule the intermediate to automatically change the stored locator to P_i at t_i ;
- routing-based: calculate FIB changes at each t_i , and schedule each node to automatically update its FIB upon t_i as needed;
- trace-based approach: upon t_i , schedule each node on the shortest path between P_i and the intermediate to update its FIB to create the trace pointing to P_i , and schedule all other nodes to purge traces.

4.3 Architectural Requirements of Supporting the Knowledge Dimension

KD approaches involves the collection and processing of knowledge, and operating the network and network services. Similar functions can be provided by control-oriented network architectures/frameworks such as Software-Defined Networking (SDN) [14] and Network Function Virtualization (NFV) [16], where

controller applications or orchestrators process information about the network to dynamically adjust the network behavior. Such architectures have the potential to bridge the gap between the knowledge-driven vision and actual practice, but the details are yet unclear. Here we sketch out several critical requirements to an architectural design that supports the knowledge-driven vision:

- **Abstraction:** the architecture should provide a way to abstract underlying network resources and application needs;
- **Compatibility:** the architecture should support existing network architectures that provide packet delivery service like IP and NDN;
- **Reactivity:** the overall operational process should form a feedback loop: states and observations of the underlying network generate actions, which in turn impact the network.

5 A Quantitative Cross-architectural Mobility Support Performance Evaluation Framework

The proposed framework takes metric definition, network architecture (architecture for short) description, and mobility support approach (approach for short, including KD approaches) description as input, and outputs the formulae for each metric. The formulae can then be used to calculate metric values from given network topologies and movement tracks.

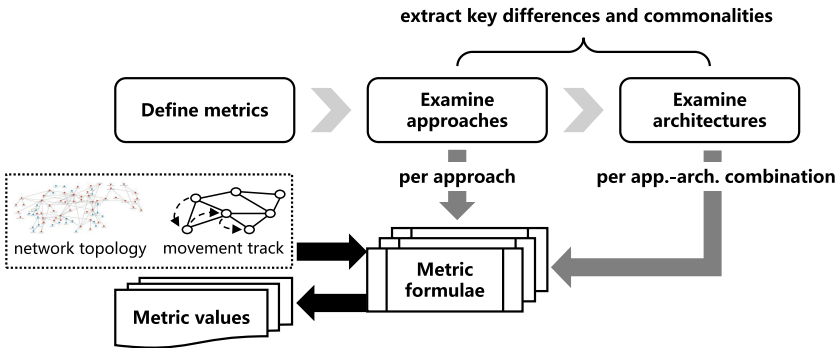


Fig. 3. The thought process of determining metric formulae

The thought process of determining metric formulae is illustrated in Fig. 3. At the core is the definition of metrics, which is independent from architectures and approaches. Based on the definition, how various design choices of each approach and architecture affect metric calculation is carefully examined. Among architectures and approaches, irrelevant differences are ignored, while key differences are reflected in the formulae. Eventually, a list of formulae is given for a metric, respecting each approach, or architecture/approach combination.

5.1 Basic Models

Network Topology: This paper considers an infrastructure network environment consisting of inter-connected nodes. The network topology can be represented by an undirected, weightless and fully-connected graph $T = (V, E)$, where V is the set of vertices in the graph, corresponding to each node in the network, and E is the edges, corresponding to each link between connected nodes.

Network endpoints such as user devices and servers gain access to the network by connecting to nodes. The node that an endpoint connects to is the PoA of the endpoint. Only nodes are considered in shortest path calculation. Thus, the shortest path between endpoints is the shortest path between their PoAs.

Movement Track: A movement track of a mobile endpoint is a temporally ordered sequence of PoAs, that the mobile endpoint attaches to after each relocation. In a topology $T = (V, E)$, the movement track of a mobile endpoint M covering n relocations is identified by $MT(T, M) = (M_0, M_1, M_2, \dots, M_n)$, where $M_i \in V (0 \leq i \leq n)$, and $M_i \neq M_{i+1} (0 \leq i \leq n - 1)$.

5.2 Metric Definition and Calculation

A total of 5 metrics are defined under two mobile communication scenarios to reflect the two critical aspects of mobility support performance: user experience and network overhead. Due to page limit, the full deduction process of each formula is not given. The complete list of formula and important comments are given instead, which should suffice to demonstrate how the formulae differentiate architectures and approaches in terms of mobility support performance.

This section inherits the definitions of M , C , and S in Sect. 3 to represent three types of endpoints. In the formulae, endpoints stand for the starting, ending, or breaking point of shortest paths. $l()$ produces the length of a path, e.g., $l(MS)$ produces the length of MS . $j()$ stands for the junctional node of two paths, e.g., $j(MS, CS)$ stands for the junctional node of MS and CS .

Basic Scenario. In this scenario, C communicates with M by continuously sending packets to M , and those packets lost due to M 's movement will eventually reach M after being retransmitted. Four metrics are defined in the basic scenario: path stretch, handover delay, signaling traffic, and maintained states.

Path Stretch (Stretch): Stretch is defined as the ratio of the actual forwarding path length to the shortest path length between C and M . This metric reflects how well an approach optimizes the packet forwarding path: the smaller the value, the smaller the average communication delay experienced by users, indicating better performance. The formulae for each approach is given in Table 1. Proxy and Trace suffer from triangular path, the actual forwarding path may be longer than the optimal path. But because Trace updates FIBs, packets may bypass S if trace is met before reaching S . Neither KD approaches affect the formulae when combined with any approach. TIP only changes the PoA of S ,

the forwarding path should be calculated in the same way as when TIP is not enabled. TRU does not affect the packet forwarding process at all.

Table 1. Metric calculation formulae - Stretch

Approach/Arch. IP	NDN
Proxy	$l(CSM)/l(CM)$
Resolution	1
Routing	
Trace	$l(Cj(CS, MS)M)/l(CM)$

Handover Delay (Delay): Delay is defined as the time it takes for the first packet to reach M after relocation. Where the first packet is always the retransmission of the last packet sent to M , which is lost due to relocation. This metric reflects the “seamless” communication capability of an approach under an architecture: the smaller the value, the less disruptions experienced by users, meaning better communication quality. The formulae for each approach and architecture combination is given in Table 2. M' stands for the last PoA of M . The formulae produce the smallest possible value in an ideal situation, so that retransmission takes place as early as possible. Note that we assume that relocation happens instantaneously, so the earliest transmission time is the exact moment M connects to a PoA. For Routing and Trace, the difference between IP and NDN comes from NDN’s in-network retransmission capability: a node may retransmit Interest itself when a FIB update points to a new forwarding direction. Considering KD approaches, TIP still does not change the formulae. TRU, however, removes the time required to propagate signaling packets, thus changes the formulae when combined with each combination.

Signaling Traffic (Traffic): Traffic is defined as the total amount of traffic produced by signaling packets. This metric reflects the extra traffic generated by

Table 2. Metric calculation formulae - Delay ($X = j(MS, CS)$, $Y = j(M'S, MS)$)

Approach/Arch.	IP	NDN
Proxy		$max(l(MS), l(CS)) + l(MS)$
Proxy-TRU		$l(MS)$
Resolution		$max(l(MS), l(CS)) + l(CS) + l(CM)$
Resolution-TRU		$l(CS) + l(CM)$
Routing	$l(CM) \cdot 2$	$l(j(CM, CM')M) \cdot 2$
Routing-TRU	$l(CM)$	$l(j(CM, CM')M)$
Trace	$max(l(XC), l(XM)) + l(XM)$	$min(l(YM) \cdot 2, max(l(XC), l(XM))) + l(XM)$
Trace-TRU		$l(YM)$

an approach: the smaller the value, the smaller the extra traffic, the higher the scalability. The formulae for each approach is given in Table 3. For Routing, the formulae consider a star topology and maximize aggregation ratio, and $\#N$ represents the total number of nodes, i.e., $|V|$. Considering KD approaches, TIP still does not affect the formulae, while TRU completely eliminates signaling packet traffic (Traffic is zero) except for Resolution, which still generates signaling packets in the query process.

Table 3. Metric calculation formulae - Traffic

Approach/Arch.	IP	NDN
Proxy	$l(MS)$	
Proxy-TRU	0	
Resolution	$l(MS) + l(CS) \cdot 2$	
Resolution-TRU	$l(CS) \cdot 2$	
Routing	$\#N - 1$	
Routing-TRU	0	
Trace	$l(MS)$	
Trace-TRU	0	

Maintained States (State): State is defined as the total number of states that need to be maintained to keep the reachability of M up-to-date. This metric reflects the storage cost of each approach, like traffic, smaller value reflects higher scalability. The formulae for each approach is given in Table 4. For Routing, each node needs to maintain one entry. Neither KD approaches affect the formulae, because they do not affect what the reachability information is or how such information is stored.

Table 4. Metric calculation formulae - State

Approach/Arch.	IP	NDN
Proxy	1	
Resolution		
Routing	$\#N$	
Trace	$l(MS)$	

5.3 Compound Scenario - Upload

In this scenario, M uploads a certain amount of data to C . A compound metric upload time (Time) is defined for this scenario. Time is defined as the total amount of time needed to finish the uploading job. The calculation of Time reuses Stretch and Delay defined in the basic scenario.

The following parameters are defined for this scenario ($i \geq 0$):

- UT : total amount of data to upload, in unit size;
- r : upload rate ratio, reflects network bandwidth;
- T_i : stay period, i.e., the period M stays before i th relocation, $T_0 = 0$;
- D_i : relocation period, i.e., the time taken to finish the i th relocation, $D_0 = 0$;

The formula is deducted as follows: the actual upload time during each stay is $T_i - Delay_i$; let U_i be the actual upload amount at the i th stay, then:

$$U_i = \frac{(T_i - Delay_i)r}{L(CM_i)Stretch_i} \quad (\text{if } T_i < Delay_i, \text{ then } U_i = 0) \quad (1)$$

Assume that after n relocations, $\sum_{i=1}^n U_i > UT$, then upload is finished during the last stay period, elapsed time:

$$GTime = \sum_{i=1}^{n-1} (D_i + T_i) \quad (2)$$

The time used to upload data during the last stay is thus:

$$UTime = T_n \frac{UT - \sum_{i=1}^n U_i}{U_n} \quad (3)$$

And further:

$$\begin{aligned} Time &= GTime + UTime \\ &= \sum_{i=1}^{n-1} (D_i + T_i) + T_n \frac{UT - \sum_{i=1}^n (T_i - Delay_i)r / (L(CM_i)Stretch_i)}{(T_n - Delay_n)r / (L(CM_n)Stretch_n)} \quad (4) \end{aligned}$$

6 Evaluation

6.1 Network Topology Settings

A total of three topologies of three types are used, including:

- Rocketfuel topology (RT): topologies in the Rocketfuel [21] dataset, which are generated from the measurement results on the real Internet; the topology for AS 1775 is used;
- balanced tree topology (BTT): topologies with the balanced tree shape, that can be generated by setting the branching factor and height; the used topology is generated with branching factor of 2, and height of 6;
- random graph topology (RGT): topologies with a specific number of nodes and randomly generated links; the used topology is generated with 256 nodes, and a link growth probability of 0.6.

6.2 Movement Pattern Settings

A movement pattern describes the probability distribution of an endpoint’s next PoA, based on its movement track so far. A movement track can be randomly generated according to a movement pattern. The movement patterns used in experiments include:

- Completely random movement (CRM): regardless of the history movement track, the next PoA is chosen uniformly from all the nodes;
- local movement (LM): there is a very high probability that the next PoA is a neighbor of the current PoA, this probability is set to 0.7;
- powerlaw movement (PM): the mobile endpoint will often attach to a few “frequent” nodes, mimicking the real life scenario where a person regularly visits a set of locations; the α parameter is set to 2.

6.3 Results

Experiment results are given for each metric, regarding each evaluated approach and architecture combination. A combination is named by concatenating the corresponding host dimension approach, any combined KD approach or KD approaches, and finally architecture with a “-”, e.g., “Trace-TIP-NDN”. If the result is the same for both IP and NDN, architecture is ignored. If multiple host dimension approaches share the same value, the approaches are concatenated with a “/”, e.g., “Proxy/Resolution-IP”.

Table 5. Results for Stretch

Combination	Mo. pattern	RT	BTT	RGT	Avg.
Proxy	CRM	2.738	2.481	2.219	2.513
	LM	2.458	3.18	2.22	
	PM	2.38	2.725	2.216	
Proxy-TIP	CRM	1.156	1.07	1.809	1.288
	LM	1.032	1.032	1.79	
	PM	1.088	1.053	1.562	
Resolution/Routing	CRM	1			1
	LM				
	PM				
Trace	CRM	1.612	1	2.13	1.563
	LM	1.541	1	2.121	
	PM	1.53	1	2.131	
Trace-TIP	CRM	1.101	1	1.805	1.258
	LM	1.023	1	1.79	
	PM	1.05	1	1.55	

Stretch: Results for Stretch are shown in Table 5. Proxy has the highest value across all topologies and mobility patterns, reaching the overall peak value of 3.18 under balanced tree topology (BTT) and local movement (LM). The reason is that Proxy suffers the most from triangular path, especially under tree-like topology. Also suffering from triangular path, the average value of Trace is 62% of Proxy. Because Trace updates FIBs, traffic between C and M does not necessarily traverse S , generating shorter forwarding path than Proxy when C and M are near each other. Regarding KD approaches, TIP results in 48% and 19% reduction respectively when combined with Proxy and Trace, almost leveling the performance of the two. The reason is that TIP migrates S near most of the shortest paths between each pair of C and M 's network locations, thus relieves the triangular path issue of Proxy, and weakens the advantage of Trace.

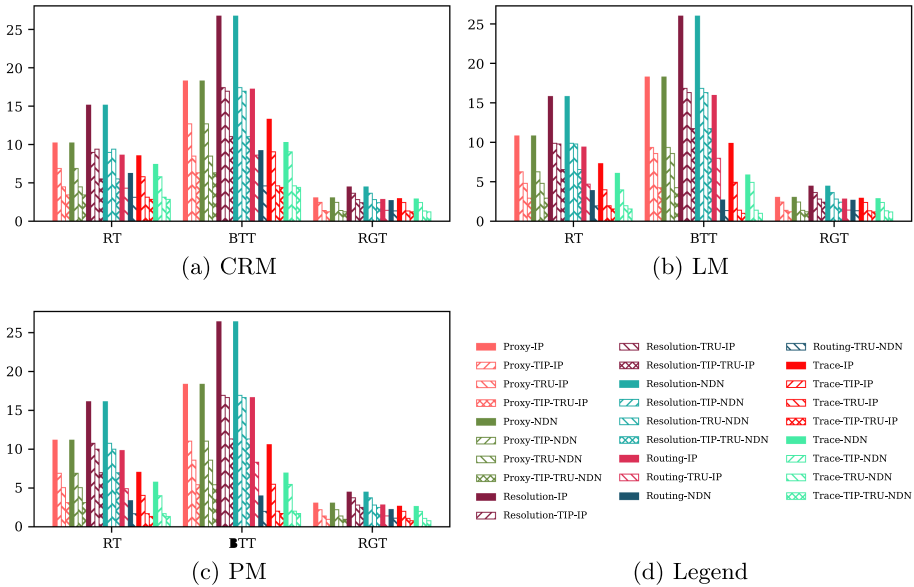


Fig. 4. Results for Delay (with legend). The X axis is topology, the Y axis is the value of Delay, bars represent approach and architecture combinations.

Delay: Results for Delay are shown in Fig. 4. In both IP and NDN, Resolution has the highest delay, averaging 15.517 time units, which is 3.8 times of the lowest average (Routing in NDN). The reason is that Resolution requires a packet exchange between C and S prior to sending a packet to M . Routing and Trace performs considerably better in NDN than in IP, the reason is that NDN's in-network Interest retransmission feature is effective when FIBs are updated.

For KD approaches, TIP reduces the length of the path taken by signaling packets, while TRU completely eliminates the delay caused by the propagation of signaling packets. With both TIP and TRU enabled, the value further drops.

Table 6. Results for Traffic

Combination	Mo. pattern	RT	BTT	RGT	Avg.
Proxy/Trace	CRM	8.484	1.395	4.469	4.903
	LM	8.575	1.38	4.794	
	PM	8.59	1.395	5.048	
Proxy/Routing/Trace-TIP	CRM	6.311	1.225	3.417	3.202
	LM	4.245	1.196	2.922	
	PM	5.425	0.964	3.115	
Routing	CRM	126	255	171	184
	LM				
	PM				
Proxy/Trace-TRU	CRM	0			0
	LM				
	PM				
Resolution	CRM	25.144	4.195	14.649	14.783
	LM	25.235	4.18	14.974	
	PM	25.25	4.195	15.228	
Resolution-TIP	CRM	11.151	3.225	5.837	7.066
	LM	11.765	3.196	6.622	
	PM	11.425	3.144	7.235	
Resolution-TRU	CRM	16.66	2.8	10.18	9.88
	LM	16.66	2.8	10.18	
	PM	16.66	2.8	10.18	
Resolution-TIP-TRU	CRM	4.84	2	2.42	3.864
	LM	7.52	2	3.7	
	PM	6	2.18	4.12	

Traffic: In IP and NDN, Routing produces the most traffic, because the signaling packet needs to be flooded across the network. For KD approaches, TIP reduces the length of the propagation path, reducing traffic for Proxy, Resolution and Trace; and TRU completely eliminates propagation of signaling packets, reducing traffic to 0 except for Resolution (Table 6).

State: Routing has the highest value thus performs the worst, because all nodes need to maintain a FIB entry, while for other approaches, only the intermediate or certain nodes need to maintain a piece of reachability information. When TIP is combined with Trace, the average distance between the mobile entity and the intermediate is shortened, and the value is slightly reduced.

Time: For the Time metric, Upload scenario parameters are set as follows: total upload size is set to 1024; upload rate factor is set to 2; stay periods are chosen

uniformly between 16 and 48; and relocation periods are set to the length of the shortest path between the current and previous PoA.

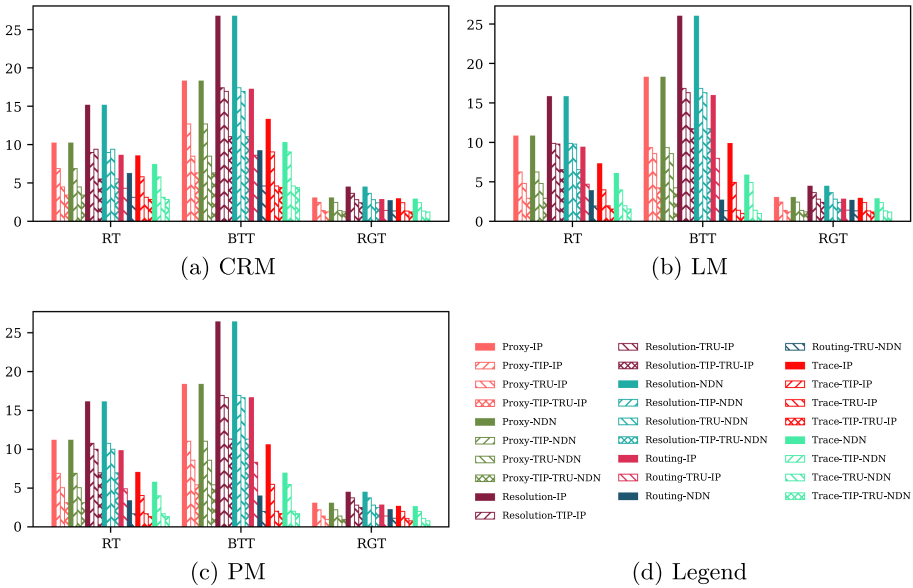


Fig. 5. Results for Time (with legend). The X axis is topology, the Y axis is the value of Delay, bars represent approach and architecture combinations.

Results for Time are shown in Fig. 5. Mobility pattern has a greater impact on this metric, and the value is generally higher (worse performance) for CRM. The main reason is that the mobility pattern determines the average distance covered in each relocation, which will greatly affect the ratio of the stay period to the total time. The larger the average travel distance, the longer the relocation period, thus the stay period is smaller, yielding higher total upload time. Across combinations, the Routing in NDN has an advantage in each topology and mobility pattern. The reason is that NDN produces smaller Delay due to the Interest retransmission mechanism, and Routing generates now path stretch, thus making the effective upload time longer in each stay.

6.4 The Effectiveness of KD Approaches

The way KD approaches affect mobility support performance is analyzed for each approach and architecture combination. For each metric, an Optimization Ratio (OR) is by comparing the value before and after KD approach(es) is enabled on top of each host dimension approach and network architecture combination. A higher positive value of OR indicates a more significant improvement in performance when combining with the corresponding KD approach(es), while a

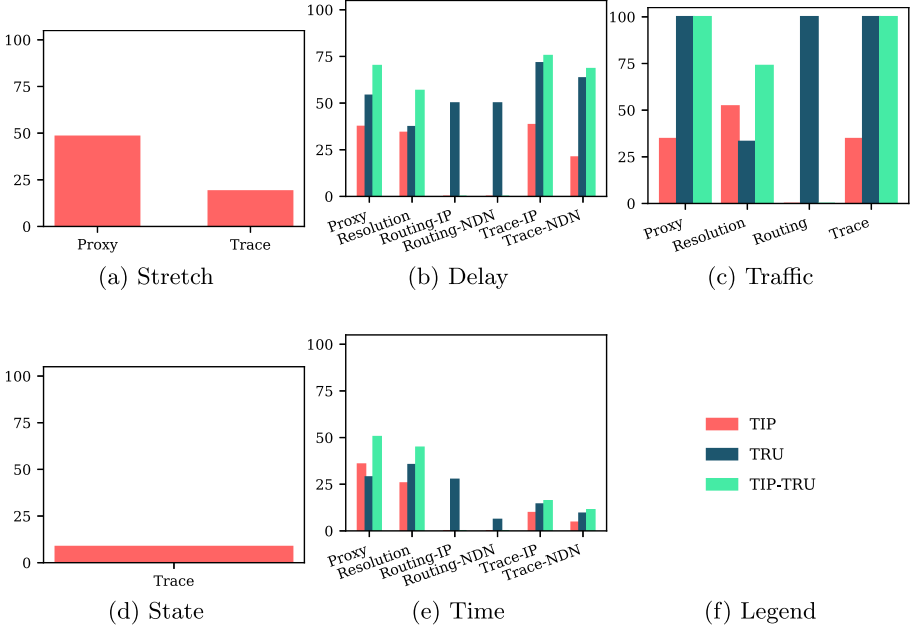


Fig. 6. Results for Optimization Ratio (with legend). The X axis is approach and architecture combinations, the Y axis is the value of OR as percentage, bars represent KD approaches.

negative value indicates worse performance. Results (see Fig. 6) show that KD approaches improve all 5 metrics. The OR is generally over or near 50%, and some even reach 100%. Such a result preliminarily demonstrates the effectiveness of the knowledge-driven paradigm, and is also expected because KD approaches operate in a new dimension to take advantage of a new set of design choices. However, this also suggests the existence of knowledge-related costs not evaluated in this paper. Evaluating such costs is crucial to this line of research and will be covered in future work. Also note that when both KD approaches are enabled, the performance boost is always more significant. This shows that the two KD approaches do not conflict with each other, and should always be enabled together if possible.

7 Related Work

Gao et al. [6] performed empirical evaluation on the mobility support performance of three “puristic” mobility support approaches, aiming at determining the most suitable architectural approach to realizing location-independent communication, i.e., communicating without caring about changing network locations. The paper presents a quantitative methodology that supports three quantifiable metrics: update cost, path stretch, and FIB size. By modeling the route

update process of name-based routing, and feeding realistic topology and mobility datasets to the model, the paper analyses the trade-off between path stretch and update cost. Results regarding device mobility show that when producing optimal routes, name-based routing causes as many as 14% of all the routers in the Internet to update their FIBs, rendering the approach infeasible.

Chaganti et al. [5] extends the work of Gao et al. [6], by proposing metrics more closely related to practical concerns, and using parameterized mobility distributions. Proposed metrics come in two types: time-to-connect (TTC) and traffic belong to the data plane, while update cost and update propagation cost belong to the control plane. By using a discrete-event simulator, the paper analyzes how each approach trades-off TTC, which is experienced by users, against a combined cost of other three metrics, which has an effect on the network. Results show that by adjusting the number and distribution of distributed name resolution servers, the name resolution approach is able to achieve the best balance between user experience and scalability, suggesting that name resolution is the most suitable approach for supporting device mobility across the Internet.

This paper aims to investigate how exploiting knowledge may further optimize mobility support performance, on top of various combinations of network architectures and mobility support approaches. Thus, compared with [5,6], the proposed evaluation framework considers existing network architectures and how their design choices affect mobility support performance, and how communication scenarios shape the way mobility support performance should be evaluated.

In terms of quantitative evaluation method, both [6] and our work relies on modeling and numerical simulation. However, [6] only models how FIBs are updated in the routing process, while the proposed one is built upon the careful examination of major design choices of network architectures and mobility support approaches, thus has richer expressive power covering more types of metrics. Chaganti et al. [5] uses discrete-event simulator which are rather straight-forward to develop and has the desired versatility for collecting a multitude of metrics, they are prone to hard-to-detect implementation faults, and the development and adaptation of such an implementation is generally more time consuming.

8 Concluding Remarks

A future Internet must support the rapidly increasing traffic from mobile endpoints in a scalable and low-latency manner. This paper proposes a knowledge-driven dimension for designing mobility support solutions and two novel knowledge-driven approaches. Experiment results show that knowledge driven approaches improves all 5 proposed metrics, demonstrating the advantage of exploiting knowledge. This research makes a preliminary attempt to apply knowledge to the mobility support problem, future research will focus on three directions: expand the framework to support quantitative analysis of knowledge-related costs; propose an architectural design to support the knowledge dimension; and apply knowledge to new research topics, such as in-network mobile computing.

References

1. Afanasyev, A., Yi, C., Wang, L., Zhang, B., Zhang, L.: SNAMP: secure namespace mapping to scale NDN forwarding. In: 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 281–286. <https://doi.org/10.1109/INFOCOMW.2015.7179398>
2. Atkinson, R., Bhatti, S., Hailes, S.: A proposal for unifying mobility with multi-homing, NAT, & security. In: Proceedings of the 5th ACM International Workshop on Mobility Management and Wireless Access, MobiWac 2007, pp. 74–83. Association for Computing Machinery. <https://doi.org/10.1145/1298091.1298105>
3. Augé, J., Carofiglio, G., Grassi, G., Muscariello, L., Pau, G., Zeng, X.: MAP-me: managing anchor-less producer mobility in content-centric networks **15**(2), 596–610. <https://doi.org/10.1109/TNSM.2018.2796720>
4. Azgin, A., Ravindran, R., Wang, G.: A scalable mobility-centric architecture for named data networking
5. Chaganti, V., Kurose, J., Venkataramani, A.: A cross-architectural quantitative evaluation of mobility approaches. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, pp. 639–647. <https://doi.org/10.1109/INFOCOM.2018.8485893>
6. Gao, Z., Venkataramani, A., Kurose, J.F., Heimlicher, S.: Towards a quantitative comparison of location-independent network architectures. In: Proceedings of the 2014 ACM Conference on SIGCOMM, SIGCOMM 2014, pp. 259–270. Association for Computing Machinery. <https://doi.org/10.1145/2619239.2626333>
7. Grilo, A., Estrela, P., Nunes, M.: Terminal independent mobility for IP (TIMIP) **39**(12), 34–41. <https://doi.org/10.1109/35.968810>
8. Han, D., Lee, M., Cho, K., Kwon, T.T., Choi, Y.: Publisher mobility support in content centric networks. In: The International Conference on Information Networking 2014 (ICOIN 2014), pp. 214–219. <https://doi.org/10.1109/ICOIN.2014.6799694>
9. Henderson, T.R., Jokela, P., Nikander, P., Moskowitz, R.: Host identity protocol. <https://tools.ietf.org/html/rfc5201>
10. Hermans, F., Ngai, E., Gunningberg, P.: Global source mobility in the content-centric networking architecture. In: Proceedings of the 1st ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications, NoM 2012, pp. 13–18. Association for Computing Machinery. <https://doi.org/10.1145/2248361.2248366>
11. Hu, X., Li, L., Mao, Z.M., Yang, Y.R.: Wide-area IP network mobility. In: IEEE INFOCOM 2008 - The 27th Conference on Computer Communications, pp. 951–959. <https://doi.org/10.1109/INFOCOM.2008.148>
12. Huang, K., Zhang, Q., Zhou, C., Xiong, N., Qin, Y.: An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning. *IEEE Trans. Syst. Man Cybern. Syst.* **47**(10), 2704–2713 (2017). <https://doi.org/10.1109/TSMC.2017.2698457>
13. Kim, D.h., Kim, J.h., Kim, Y.s., Yoon, H.s., Yeom, I.: Mobility support in content centric networks. In: Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking, ICN 2012, pp. 13–18. Association for Computing Machinery. <https://doi.org/10.1145/2342488.2342492>
14. Kim, H., Feamster, N.: Improving network management with software defined networking **51**(2), 114–119. <https://doi.org/10.1109/MCOM.2013.6461195>

15. Li, D., CHuah, M.C.: SCOM: a scalable content centric network architecture with mobility support. In: 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks, pp. 25–32. <https://doi.org/10.1109/MSN.2013.44>
16. Mijumbi, R., Serrat, J., Gorricho, J.L., Bouten, N., De Turck, F., Boutaba, R.: Network function virtualization: state-of-the-art and research challenges **18**(1), 236–262. <https://doi.org/10.1109/COMST.2015.2477041>
17. Perkins, C., Bhagwat, P.: A mobile networking system based on internet protocol **1**(1), 32–41. <https://doi.org/10.1109/98.911984>
18. Ramjee, R., Varadhan, K., Salgarelli, L., Thuel, S., Wang, S.Y., La Porta, T.: HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks **10**(3), 396–410. <https://doi.org/10.1109/TNET.2002.1012370>
19. Rekhter, Y., Li, T.: A border gateway protocol 4 (BGP-4). <https://tools.ietf.org/html/rfc1771>
20. Rodríguez Natal, A., Jakab, L., Portolés, M., Ermagan, V., Natarajan, P., Maino, F., Meyer, D., Cabellos Aparicio, A.: LISP-MN: mobile networking through LISP **70**(1), 253–266. <https://doi.org/10.1007/s11277-012-0692-5>
21. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with rocketfuel. *ACM SIGCOMM Comput. Commun. Rev.* **32**(4), 133–145 (2002)
22. Teraoka, F., Uehara, K., Sunahara, H., Murai, J.: VIP: a protocol providing host mobility **37**(8), 67–ff. <https://doi.org/10.1145/179606.179657>
23. Wang, L., Waltari, O., Kangasharju, J.: MobiCCN: mobility support with greedy routing in content-centric networks. In: 2013 IEEE Global Communications Conference (GLOBECOM), pp. 2069–2075. <https://doi.org/10.1109/GLOCOM.2013.6831380>
24. Yang, Y., Xiong, N., Chong, N.Y., Défago, X.: A decentralized and adaptive flocking algorithm for autonomous mobile robots. In: 2008 The 3rd International Conference on Grid and Pervasive Computing - Workshops, pp. 262–268 (2008). <https://doi.org/10.1109/GPC.WORKSHOPS.2008.18>
25. Zhang, L., Wakikawa, R., Zhu, Z.: Support mobility in the global internet. In: Proceedings of the 1st ACM Workshop on Mobile Internet Through Cellular Networks, MICNET 2009, pp. 1–6. Association for Computing Machinery. <https://doi.org/10.1145/1614255.1614257>
26. Zhang, Q., Zhou, C., Xiong, N., Qin, Y., Li, X., Huang, S.: Multimodel-based incident prediction and risk assessment in dynamic cybersecurity protection for industrial control systems. *IEEE Trans. Syst. Man Cybern. Syst.* **46**(10), 1429–1444 (2016). <https://doi.org/10.1109/TSMC.2015.2503399>
27. Zhang, Y., Afanasyev, A., Burke, J., Zhang, L.: A survey of mobility support in named data networking. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 83–88. <https://doi.org/10.1109/INFOCOMW.2016.7562050>
28. Zhang, Y., Xia, Z., Mastorakis, S., Zhang, L.: KITE: producer mobility support in named data networking. In: Proceedings of the 5th ACM Conference on Information-Centric Networking, ICN 2018, pp. 125–136. Association for Computing Machinery. <https://doi.org/10.1145/3267955.3267959>
29. Zhu, Z., Zhang, L., Wakikawa, R.: Understanding apple’s back to my mac (BTMM) service