Title

Routing to Multi-Instantiated Destinations: Principles, Practice, and Applications

Authors

Garcia-Luna-Aceves, J.J.
Martinez-Castillo, J.E.
Menchaca-Mendez, R.

Publication Date

2017-08-01

Peer reviewed

Routing to Multi-Instantiated Destinations: Principles, Practice, and Applications

J.J. Garcia-Luna-Aceves, *Fellow, IEEE*, J.E. Martinez-Castillo and Rolando Menchaca-Mendez

Abstract—Prior solutions for routing to multi-instantiated destinations simply adapt existing routing algorithms designed for single-instance destinations, or rely on flooding techniques. In this paper, a new approach for routing to multi-instantiated destinations is introduced, and the MIDR (Multiple Instance Destination Routing) framework is presented as an example of the approach. MIDR uses only distance information to multi-instantiated destinations, without routers having to establish overlays, know the network topology, use complete paths to destination instances, or know about all the instances of destinations. MIDR can be used in name-based content routing, IP unicast routing, multicasting, and anycasting; even in scenarios where the network topology is highly dynamic such as in the case of MANETs. It is shown that MIDR provides multiple loop-free paths to destination instances. Extensive simulation-based experiments performed in the context of MANETs show that MIDR outperforms traditional approaches based on unicast protocols and that it scales to large networks.

Index Terms—Routing, Anycast, Internet, MANETs, Information Centric Network.

1 Introduction

The first routing protocol for packet switching networks can be traced back to Baran's original design of packet switching at the RAND Corporation in the 1960s [5]. The "hot potato heuristic routing doctrine" described by Baran is the first instance of distance-vector routing in which the number of hops traversed by messages originated by destination nodes determine the preferred paths to them over time.

The routing protocols developed for early public data networks and the ARPANET in the 1960s and 70s also assumed that destination nodes advertise either distances to themselves or adjacent links to the rest of the network. The distributed algorithms used as part of these protocols were more efficient than Baran's proposal; they consisted of either a distributed version of the Bellman-Ford algorithm using vectors of distances, or Dijkstra's shortest-path-first algorithm using flooding of link-state updates [48].

The routing protocols developed for the Internet for routing within autonomous systems [38] (e.g., RIP and OSPF) evolved directly from the early routing protocols designed for the ARPANET. The key differences between Internet routing and routing within a packet switching network is that Internet destinations are network address ranges, and the routers attached to a network advertise a link, path, or distance for that network. While Internet routing protocols must build routing tables regarding destinations that are not routers, the distributed algorithms on which they are based were designed assuming that destinations are nodes of packet-switching networks. This choice seems trivial at first glance, because a router to which a destination is attached can report the presence of the destination. Furthermore, as Section 2 summarizes, various forms of routing to multi-instantiated destinations have been proposed or implemented to date based on algorithms designed for routing to single-instance destinations. Examples are multicast routing (e.g., [4], [14]), anycasting (e.g., [3], [30], [51], [56]), routing to multi-homed networks in BGP [38], and routing to replicated content in information centric networking (ICN) architectures (e.g., [1], [6], [53]).

Unfortunately, as discussed in Section 2.1, using algorithms designed for routing to single-instance destinations results in unnecessary signaling overhead and complexity, limited functionality, and in some cases incorrect solutions. This has large implications for the future Internet, given that the nature of Internet destinations is shifting from individual machines to content objects that may be replicated dynamically, services, classes of resources, groups of things or individuals that are distributed over networks, and even networks that may be partitioned.

Other alternatives for routing to multi-instantiated destinations include directory-based (e.g., DNS) solutions which are more flexible with respect to the selection criteria [56] and are easier to deploy because they do not require changes to the underlying protocols. However, DNS-based solutions have the problem of not knowing the location of the clients but the location of the client's name resolvers [51]. Moreover, the overall performance of such directory-based schemes can be improved by having efficient network-layer anycast protocols [17].

We argue that traditional notions of unicast, anycast, and multicast routing need to be revised in favor of an integrated framework for routing in which any destination may have multiple instances. Section 3 presents the main principles of such a framework. In a nutshell, routing to multi-instantiated destinations entails establishing a lexico-

- *J.J. Garcia-Luna-Aceves is with the Department of Computer Engineering, University of California, Santa Cruz, CA 95064, USA, and with the Palo Alto Research Center, Palo Alto, CA 94304, USA.E-mail: jj@soe.ucsc.edu*
- *J.E. Martinez-Castillo and Rolando Menchaca-Mendez are with the Centro de Investigación en Computación of the Instituto Politécnico Nacional, Unidad Profesional ALM 07738, Mexico City, México.E-mail: mcastilloa10@sagitario.cic.ipn.mx, rmen@cic.ipn.mx*

graphic ordering of distances to destinations in which the identifiers of the routers to which destination instances are attached are part of the attribute set used to define the ordering. Many concrete approaches are possible based on this simple observation.

Section 4 presents *Multiple Instance Destination Routing* (MIDR) as a concrete example of a routing framework based on the principles presented here. MIDR is the first routing framework for multi-instantiated destinations and uses only distance information about destinations. MIDR does not require routers to know the network topology, path information, routes to all network sites, or all the instances of any given destination. Instead, for each multi-instantiated destination, MIDR partitions the network into connected components composed of nodes that share a common closest destination instance and establishes an ordering over the nodes that can be used to route data from any node to its closest destination instance through loop-free paths. This way, nodes only need to keep state regarding their closest instance of each destination. This section also presents the *Interest-driven Multiple Instance Destination Routing Protocol* (Id-MIDR) which is an instantiation of MIDR that is well suited to scenarios where the network topology is highly dynamic and the network resources are scarce.

Section 5 shows that MIDR provides multiple paths to destination instances without ever creating a routing-table loop and that it converges to shortest paths to the nearest instances of destinations. Section 6 compares the communication and time complexities of MIDR with that of traditional routing approaches applied to multi-instantiated destinations. MIDR incurs far less signaling overhead and is much faster to converge to correct routing tables than prior approaches, because it does not require routers to know the network topology or all the sites where destination instances are present. This section also presents the results of extensive simulation-based experiments that confirm these notions by showing that Id-MIDR clearly outperforms traditional solutions in the context mobile ad hoc networks. Section 7 discusses how our framework can be applied to improve or redefine IP routing, IP anycasting, and name-based content routing.

2 Related Work

Solutions to routing to multi-instantiated destinations can be classified by the layer of the protocol stack in which they are implemented. In this paper we focus on proposals that work at the network-layer.

2.1 Routing Algorithms

Many routing algorithms have been developed over the years to ensure that the distributed computations incurred in updating all routing tables in a network with correct entries for each destination terminate within a finite time. Examples include: using sequence numbers to identify the most recent update information (e.g., [37]), using path information to detect or block loops (e.g., [7]), or using diffusing computations to avoid loops (e.g., [18]). In addition, many approaches have been proposed that attain routing to single-instance destinations over multiple paths (e.g., [35], [40], [55]). However, attaining loop-free routing does not solve the problem of establishing valid routes to a destination that can be replicated arbitrarily in a network. In fact, none of the algorithms reported to date for routing to single-instance destinations can be used without change to enforce loop-free routing to multi-instantiated destinations. This is because the ordering they establish with respect to each destination inherently assumes that the destination corresponds to a single node of the graph representing a network.

Consider as an example the loop-freedom condition defined for sequence numbered distances. A destination j is the only node that can increase the sequence number used to validate reported distances to j. A router $i \neq j$ stores the most recent sequence number associated with j, and can select a neighbor k as a next hop to destination j if k reports either (a) a larger sequence number from j than i currently stores, or (b) the same sequence number as i stores and a smaller distance to j than i currently attains. Unfortunately, the ordering attempted by the use of sequence numbers originated by a given instance of j is invalidated by the existence of other instantiations of j issuing sequence numbers independently of one another.

Similar problems exist with algorithms based on diffusing computations or path information. Contrary to what some prior work on anycasting has assumed [30], traditional distance-vector routing algorithms *cannot* ensure correct anycasting. Furthermore, the problem exists even when routing is based on link-state algorithms. In this case, the multiple instances of a given destination j appear to be a single node in a graph, which leads to routers having inconsistent topology maps that results in long-term or permanent loops (see Fig. 5 in [30]).

There are only three possible ways to support routing to multi-instantiated destinations using the algorithms designed to date for routing to single-instance destinations. The type of approach used depends on the amount of information known to the sources of data packets.

If sources do not know which nodes constitute instances of the destination, then the sources must flood the entire network with signaling packets; this approach has been called "sender initiated." The opposite approach to the above case consists of each source node knowing the nodes to which instances of the intended destination attach. In this case, signaling from destination instances must reach all potential sources, and sources must have enough information to compute shortest paths to any destination instance.

The alternative to the above two approaches consists of designating a node to serve as the representative (i.e., the address) of the set of destination instances. All nodes maintain routes to the representative node of the multi-instantiated destination and are also capable of learning the mapping from the identifier of the multi-instantiated destination to the identifier of the representative node. Each node with a destination instance establishes a route to the representative node, so that information can flow towards that instance. A source sends its data packets towards the representative node, and either that node or another relay (depending on the specific solution) ensures that the data packets are forwarded to all destination instances based on the routes established previously.

There are three main examples of prior work on routing to multi-instantiated destinations based on the above three approaches: routing to all members of a multicast group; routing to any one member of an anycast group; and name-based routing of content. We summarize the prior work in each of these areas next.

2.2 Multicast Routing Protocols

McQuillan [36] proposed the first link-state routing approach to support multicasting. In essence, each node floods link-state advertisements (LSA) stating the state of adjacent links and the existence of receivers for different multicast groups. Given this information, each node can compute shortest routes to all receivers of each multicast group. Multicast OSPF [38] constitutes a more recent example of this approach. Deering [13] described an approach to multicasting similar to what McQuillan introduced, and also introduced a sender-initiated approach to multicast routing based on distances. Various approaches can be used for flooding signaling packets from each source and for nodes with no receivers of a multicast group to send prune messages towards sources. Examples of this approach are ODMRP [32] and PIM dense mode [14].

The first approach for multicast routing based on representatives was introduced by Ballardie, Francis and Crowcroft [4], and many subsequent examples have been proposed (e.g., [14], [42]). A node serves as the address of a multicast group and is called the core of the group. Nodes maintain routes to all network nodes and hence to all cores, and are able to learn the mapping from the multicast group address to the address of the core. Each receiver of sends join requests towards the core of the group to establish a shared multicast tree spanning all the receivers and the core. Sources simply send data packets towards the core, and data packets are sent to all receivers of the multicast group over the multicast tree.

2.3 Anycast Routing Architectures and Protocols

McQuillan [36] was arguably the first to address routing problems in packet switching networks associated with destinations with multiple instances. He discussed the concepts of logical addressing (which is now called anycasting), broadcast addressing (or intelligent flooding), and group addressing (which is now called multicasting). He also proposed approaches to handle these three addressing methods based on the use of directory nodes supporting indirection services and routing algorithms designed for single-instance destinations.

Interestingly, most of the solutions that have been proposed for anycasting over the years (e.g., [16], [52]) are similar in nature to McQuillan's original proposals, assume the use of routing algorithms designed for single-instance destinations, and none of them is able to truly address the scaling problems in anycast routing associated with the handling of anycast addresses. Patridge et al. [43] defined logical addressing in the context of the Internet Protocol (IP) and called it host anycasting service. C.P. Low et al. [10] proposed a slightly different approach where senders perform breadth first searches to find paths to all the instances of the destination and then select the path with the smallest cost. Unfortunately, this approach does not scale because it requires flooding the network per source per anycast group.

In the context of the MANETs, most of the solutions presented up to date are based on unicast routing protocols such as AODV [45] (e.g., anycast-AODV [31], AODV-Based [50]) and DSR [28] (e.g., anycast-DSR [31], ARDSR [44]). In all these protocols nodes compute routes to all the instances of the destinations using the underlying unicast routing protocol and then select the path of minimum length. In [33], V. Landers et al. proposed the Density-Based Anycast protocol that establishes a potential field per anycast group over the nodes. This potential field is computed as the superposition of the individual potential fields of each group member which are decreasing functions with respect to the hop distance to the group members. Packets are forwarded following the steepest ascent gradient of the field until they reach any group member. This way data packets can be forwarded to the closest group member. This approach is similar to our proposal in the sense that both schemes establish an ordering over the nodes that can be followed to reach the nearest instance of a destination. However, unlike our proposal, Density-Based Anycast requires flooding the whole network per instance per destination which is not scalable.

The Global IP-Anycast (GIA) framework [30] assumes that anycasting within domains can be supported using traditional intra-domain routing protocols and uses an on-demand query-based inter-domain routing protocol based on broadcast signaling. IPv6 [51] provides limited support for anycasting, in that it allows the use of anycast addresses, but there are no known methods to support anycast routing in a scalable manner. The *i*3 approach [49] to anycasting consists of using an overlay of servers that map identifiers to actual IP addresses, and clients contact target destinations through the overlay; however, no new approaches are introduced for anycast routing.

2.4 Name-Based Content Routing Protocols

Name resolution and routing of content are essential in all information centric network (ICN) architectures [1], [6], [53], and several approaches have been proposed to support content routing based on the names of named data objects (NDO) that may be replicated in a network. Interestingly, all these approaches are based on algorithms designed for routing to single-instance destinations.

Like sender-initiated approaches for multicasting, some content routing approaches rely on flooding of content requests to cope with the fact that nodes requesting content by name do not know the locations of copies of content. Directed Diffusion [25] was one of the first proposals for name-based routing of content. Requests for named content (called interests) are diffused throughout a sensor network, and data matching the interests are sent back to the issuers of interests.

A number of approaches are based on maintaining routing information to all replicas of content by means of path-vector algorithms or link-state algorithms. Gritter and Cheriton proposed the Name-Based Routing Protocol (NBRP) [23] as an extension of BGP. The CBCB (combined broadcast and content based) routing scheme for content-based networking [9] is an example of content-routing similar to the receiver-initiated approach to multicasting. CBCB consists of two components. First, a spanning tree of the network or multiple per-source trees spanning the network are established. Then, publish-subscribe requests for content based on predicates are sent between consumers and producers of content over the tree(s) established in the network.

The routing approach in the Mobility First project [39] requires using either network addresses or source routing or partial source routing. Several ICN projects have adopted content routing modalities based on the link-state routing approach (e.g., [11], [12], [15], [26], [41], [47]). NLSR [34] is a recent example of name-based content routing based on complete topology information. Routers flood link-state advertisements (LSA) that describe the state of physical links or the name of prefixes of content for which they have local copies.

Some ICN projects (e.g., [46], [47]) adopt content routing modalities based on distributed hash tables (DHT) running in overlays over the physical infrastructure to accomplish name-based routing. A destination is assigned a home location in the DHT that nodes can determine by using a common hash function from the name space of destinations to the name space of nodes in the DHT. DHT nodes can cache known mappings to improve efficiency, and the DHTs are built using underlying routing protocols that discover the network topology.

3 Routing to Multi-Instantiated Destinations

A multi-instantiated destination is a non-empty set of entities denoted with the same unique identifier consisting of a string of alphanumeric symbols. The identifier can be drawn from a flat or hierarchical naming space and can have a fixed length or variable length depending on the application. An entity can be an information object or collection of objects, a thing or class of things, a process, a service, a network, an end system, or an intermediate system. It may be part of one or more destinations, each denoted with a different identifier.

The objective is to support routing to multi-instantiated destinations in a way that permanent loops are impossible, and without each node having to flood the network, know about all destination instances, or rely on a pre-defined representative node. The services to be provided consist of: (a) reaching the nearest instance of a destination, and (b) reaching a subset of instances of a destination. Due to space limitations, in this paper we focus on the case of reaching the nearest instance of the destination.

Supporting loop-free routing to the nearest instance can be done using distance information in a manner that scales in much the same way as routing to single-instance destinations. Doing so requires two basic functionalities with respect to a given destination. First, using common rules, routers must establish lexicographic orderings with respect to the destination instances that are nearest; different routers may order themselves with respect to different instances. Second, a router must report as its distance to the destination the value of its distance to the nearest instance, and use a common lexicographic ordering to select the nearest instance to use.

Let a router that originates an advertisement for a given destination instance that is locally available be called an *anchor* of the destination. The routing functionality summarized above can be attained by: (a) having each router include the identifier of the anchor corresponding to the preferred nearest destination instance as an attribute of the distance it reports for the destination; and (b) using loop-free routing constraints that make use of this information to establish lexicographic orderings to destination instances.

4 Multiple Instance Destination Routing

We describe MIDR (Multiple Instance Destination Routing) as an example of routing to multi-instantiated destinations. The operation of MIDR assumes that each router is assigned a unique identifier and each destination is assigned a unique identifier. Identifiers may be hierarchical or flat, and they may be user friendly or not.

MIDR relies on sequence numbers created by the anchors of destinations to determine which updates carry the most recent information about a destination, enforce loop-free routing, and maintain a lexicographic ordering among distances reported by different anchors of the same destination.

The lexicographic value of an identifier i is denoted by $|i|$. The set consisting of router i and all its neighbor routers is denoted by N^i. The set of next hops of router i for destination j is denoted by S^i_j. If i is an anchor for destination j, then $S^i_j = \{i\}$. The link from router i to router k is denoted by (i, k) and its cost is denoted by l^i_k. The cost of the link (i, k) is assumed to be a positive number that can be a function of administrative constraints and performance measurements made by router i for the link.

4.1 Information Stored and Exchanged

Router i maintains three tables: (1) a *link cost table* (LT^i) listing the cost of the link from router i to each of its neighbors; (2) a *neighbor table* (NT^i) stating routing information reported by each neighboring router for each destination; and (3) a *routing table* (RT^i) that stores routing information for each destination.

The entry in LT^i for link (i, k) with $k \in N^i - \{i\}$ consists of the identifier of neighbor k and the cost of the link to it (l^i_k). The information stored in NT^i for destination j from each neighbor $k \in N^i$ consists of routing information for the nearest anchor. The routing information consists of: the distance from router k to j (d^i_{jk}); the identifier of an anchor (a^i_{jk}) where j is present, which can be i itself; and the sequence number created by a^i_{jk} for j (sn^i_{jk}). If i is an anchor for j, then $d^i_{ji} = 0$ and $a^i_{ji} = i$.

The row for destination j in RT^i specifies: (1) the identifier of j; (2) the routing update information for j (RUI^i_j); (3) the list of neighbors that are valid next hops (S^i_j), which includes a neighbor s^i_j that offers the shortest distance to j; (4) an interest flag (ι^i_j) that indicates whether router i has interest on destination j; and (5) an anchor list (A^i_j) with a tuple for each different anchor currently reported by neighbors in S^i_j. Each tuple $[m, sn(m)] \in A^i_j$ states the name of an anchor m and the sequence number $sn(m)$ reported by that anchor. The interest flag ι^i_j is set to 1 if router i has recently received a data packet intended for destination j or if it is located one hop away from either an active source or an active anchor and; it is set to 0 otherwise.

The information in RUI^i_j consists of: (1) a flag for each neighbor k denoting whether or not the information needs to be sent in an update to neighbor k (up^i_{jk}); (2) the distance from i to j (d^i_j); (3) the anchor of j that has the smallest name among those that offer the shortest distance to j (a^i_j); and (4) the sequence number created by a^i_j for j (sn^i_j).

An update message sent by router i to neighbor m consists of the identifier of router i; a message sequence number (msn^i) used to identify the message; and a list of updates, one for each destination that needs updating. An update sent by router i for destination j (U^i_j) states: The identifier of the destination (j); the distance to j (ud^i_j); an anchor (ua^i_j), which may be i itself; and the sequence number created by ua^i_j for destination j (usn^i_j).

Router i updates NT^i_{jk} after any input event affecting the information stored for j from neighbor k. It updates NT^i_{jk} with the information reported by k for j only if sn^i_{jk} is the most recent sequence number known from a^i_{jk}.

4.2 Routing to Nearest Instances of Destinations

MIDR establishes a lexicographic ordering of the distances to any given destination reported by routers based on the distance values, anchor identifiers, and sequence numbers created by the anchors. Instead of remembering the most recent sequence number for a given destination, a router maintains the sequence numbers created by *all* the anchors currently reported by its neighbors. The information about a given anchor of a destination is deleted after a finite time that is long enough to ensure that up-to-date information about valid anchors of the destination is received, before anchor information is deleted.

A router can select neighbors as next hops to destinations only if they report up-to-date information and offer shorter distances to the destinations than the router itself, or the same distances but have lexicographically smaller identifiers. To address the case in which routers are unable to find viable next hops to some destinations, anchors send updates about their destinations periodically and increment the sequence numbers they assign to their destinations. Let A^i_j be the set of anchors known to router i for destination j. The following condition is sufficient to ensure that no routing-table loops are formed when routers change their next hops.

Successor-Set Ordering Condition (SOC):

Neighbor $k \in N^i$ can become a member of S^i_j (i.e., be a next hop to destination j) if the following two statements are true:

$$k \in \{ \, v \mid v \in N^i \wedge [\, \forall m \in A^i_j \; (\, a^i_{jv} \neq m \; \vee \; sn^i_{jv} \geq sn(m)) \,] \, \} \tag{1}$$

$$\begin{aligned} (\, d^i_j < \infty \wedge \; [\, d^i_{jk} < d^i_j \vee (\, d^i_{jk} = d^i_j \; \wedge \; |k| < |i| \,) \,] \,) \vee \\ (\, d^i_j = \infty \; \wedge \; d^i_{jk} < d^i_j \; \wedge \\ \forall v \in N^i - \{k\}(\, (\, d^i_{jk} + l^i_k < d^i_{jv} + l^i_v \,) \vee \\ (\, d^i_{jk} + l^i_k = d^i_{jv} + l^i_v \wedge |k| < |v| \,) \,) \end{aligned} \tag{2}$$

With SOC, only those neighbors reporting the most recent sequence numbers from the known anchors of destination j can be considered as next hops (Eq. (1)), and they are ordered lexicographically based on their distances to destination j and their identifiers (Eq. (2)). If router i has a finite distance to destination j, then it can select neighbor k as a next hop to j if either k is closer to the destination than router i or is at the same distance to the destination but $|k| < |i|$. If router i has no finite distance to destination j, then it can have k as a next hop to j only if k reports the smallest *finite* distance to j among all neighbors, or it has the smallest identifier among those neighbors reporting the smallest finite distance to j.

For each destination j, router i determines which routers in N^i report valid sequence numbers created by anchors of the destination, and then determines those routers that can be next hops (i.e., belong to set S^i_j) using SOC. If at least one router in N^i is found that satisfies SOC, router i computes $d^i_j = d_{min} = Min\{d^i_{jm} + l^i_m \mid m \in S^i_j\}$, where S^i_j is the set of routers in N^i that satisfy SOC. Router i then sets $a^i_j = a^i_{jq}$ and $sn^i_j = sn^i_{jq}$, where $q \in S^i_j$, $d^i_{jq} + l^i_q = d_{min}$, and $|q| \leq |m|$ for any $m \in S^i_j$ such that $d^i_{jm} + l^i_m = d_{min}$.

Router i schedules an update $U^i_j[ud^i_j = d^i_j, \, ua^i_j = a^i_j, \, usn^i_j = sn^i_j]$ to neighbor k if $d^i_j < \infty$ and $ud^i_{jk} = \infty$, so that neighbor k can satisfy SOC by making i part of S^k_j. Router i schedules an update U^i_j to all its neighbors if it makes any changes to d^i_j, a^i_j, or sn^i_j after an input event. If SOC is not satisfied at router i after an input event, then router i resets $S^i_j = \emptyset$; sets $ud^i_j = d^i_j = \infty$, $ua^i_j = a^i_j = null$, and $usn^i_j = sn^i_j = 0$; and schedules an update U^i_j to all its neighbors.

Fig. 1 illustrates how MIDR routes to the nearest replica of a single destination when three routers (d, o, and u) serve as the anchors of the destination and each link has unit cost. It is assumed that all routers have received the most-recent sequence numbers from the anchors they know (d, o, or u) for the destination. The first tuple listed next to each node indicates the shortest distance from the router to the destination and the anchor with the smallest identifier at that distance. Updates from each router state only the preferred anchor (e.g., the update from node e states d as the anchor and distance 2 to it). Each additional tuple next to a router, if any, states an alternate anchor for the destination and the distance to it. The arrowheads in the links between nodes indicate the next-hop neighbors of nodes, and the arrow to the lexicographically smallest next hop is shown with the color of the anchor.

As Fig. 1 shows, updates from an anchor propagate only as long as they provide routers with shorter paths to destinations. In the example, no routing update about the destination propagates more than four hops, even though the network diameter is eight. In general, independently of how many anchors exist in a network for given destination, a router only has as many active anchors for the destination as it has neighbors, because each router reports only the best anchor it knows for each destination. Even in this small network of just 23 routers, several routers have multiple paths to the destination; all links can be used to forward data packets; very few routers know about all the anchors of the destination; and traversing any possible directed path to the destination in Fig. 1 necessarily terminates at d, o, or u, without traversing a loop.

4.3 Interest-driven MIDR

The signaling needed to update routing information in MIDR can be based on event-driven, periodic updates or

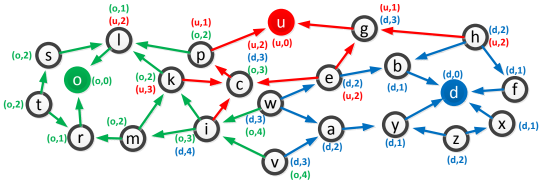


Fig. 1: Routing to the nearest instance of a destination

interest-driven. In this section we describe *Interest-driven-MIDR* (Id-MIDR), an instantiation of the MIDR framework that is based on interest-driven updates. With interest-driven signaling, update control messages are periodically disseminated across *regions of interest*, which are connected components of the network composed of nodes with interest in the destinations. A node i is part of the region of interest of destination j if its interest flag ι_j^i equals 1, namely, if it lays in an active path from a source to an anchor of destination j or if it is located one hop away from either an active source or an active anchor. Regions of interest are activated and deactivated by the presence or absence of data traffic and all routing information is soft-state.

Anchor nodes that have recently received data from interested sources periodically transmit update messages reporting new sequence numbers that are disseminated across the region of interest. A router i sends update messages to report updates made to its routing information regarding destination j, only if it is part of the region of interest of j. The purpose of the regions of interest is to avoid disseminating control information to regions of the network where it is of no use. Algorithm 1 shows the procedure followed by node i to update its routing state with the information $U_j^k = \{j, k, ud_j^k, ua_j^k, usn_j^k\}$ received in an update message from neighbor k, regarding destination j. Updates are accepted if they report either new (line 2) or fresher information (lines 5 and 9).

Algorithm 1: UpdateHandler(i, U_j^k)

```
1  $NT_j^i \leftarrow \begin{cases} NT_j^i \cup \{k, ud_j^k, ua_j^k, usn_j^k\} & \text{if } \{k,*,*,*\} \notin NT_j^i \\ NT_j^i - \{k,*,*,*\} \cup \{k, ud_j^k, ua_j^k, usn_j^k\} & \text{otherwise} \end{cases}$
2  if $\{ua_j^k, usn_j^k\} \notin A_j^i$ then
3      $A_j^i \leftarrow A_j^i \cup \{ua_j^k, usn_j^k\}$ ;
4  else
5      if $usn_j^k > sn(m)$ then
6          $A_j^i \leftarrow A_j^i - \{m, sn(m)\} \cup \{ua_j^k, usn_j^k\}$ ;
7      else
8          return;
9  if $(ud_j^k + l_k^i < d_j^i) \vee (ud_j^k + l_k^i = d_j^i \wedge |k| < |s_j^i|)$ then
10     $up_{j*}^i \leftarrow true$; $d_j^i \leftarrow ud_j^k + l_k^i$;
11     $a_j^i \leftarrow ua_j^k$; $sn_j^i \leftarrow usn_j^k$; $s_j^i \leftarrow k$;
12     if $\{a_j^i, *, *, *\} \in NT_j^i$ then
13         $\iota_j^i \leftarrow true$;
14 Update $S_j^i$ according to SOC;
```

Fig. 2 shows the routing state at the nodes after the region of interest has been established. In the figure, node i is the only active source and node o is its closest anchor. Nodes inside the dotted line belong to the region of interest because they lay in an active path from i to o, or because they are one hop away from either anchor o or source i. Nodes located outside of the region of interest have eliminated their routing state regarding the destination because control packets are not disseminated outside of the region of interest and hence, they have not recently received an update.

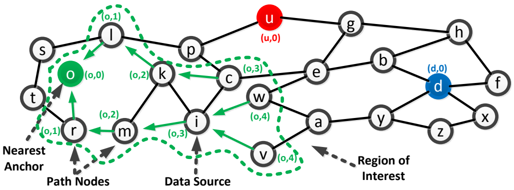


Fig. 2: Routing state at nodes after the region of interest has been established

5 MIDR Correctness

MIDR guarantees that routing-table loops are never formed, even as the instances of destinations and the network topology change. The following theorems prove that this is the case.

Theorem 1. *No routing-table loops can be formed if routers use SOC to select their next hops to destinations.* □

Proof: The proof is by contradiction. Assume that a routing loop L_j for destination j consisting of h hops is created at time t_L when the routers in the vertex set of L_j change successors according to SOC. Let $L_j = (n_1, n_2, ..., n_h)$, with $n_{i+1} \in S_j^{n_i}$ for $1 \leq i \leq h-1$ and $n_1 \in S_j^{n_h}$.

According to SOC, each hop $n_i \in L_j$ $(1 \leq i \leq h)$ can select its next hops (i.e., $S_j^{n_i}$) in only two ways, depending on whether or not $d_j^{n_i} < \infty$ when $n_i \in L_j$ selects its next hops before or at time t_L when L_j is formed.

Assume that there is a subset of hops $I_j \subset L_j$ such that $d_j^{n_m} = \infty$ when n_m joins L_j by adding n_{m+1} to $S_j^{n_m}$ for each $n_m \in I_j$. By assumption, router n_m uses Eq. (2) in SOC; therefore, $d_{jn_{m+1}}^{n_m} < \infty$ and router n_{m+1} must report to n_m either an anchor that router n_m did not know before, or a more recent sequence number created by an anchor known to n_m before the update from n_{m+1}. Furthermore, for all $q \in N^{n_m}$, it must be true that $d_{jq}^{n_m} > d_{jn_{m+1}}^{n_m}$ or $d_{jq}^{n_m} = d_{jn_{m+1}}^{n_m}$ and $|n_m| < |q|$. Therefore, the following relation must hold between $d_{jn_{m+1}}^{n_m}$ and $d_{jn_{m-1}}^{n_m}$ for any $n_m \in I_j$:

$$\begin{aligned} & d_{jn_{m-1}}^{n_m} > d_{jn_{m+1}}^{n_m} \\ \vee\ (& d_{jn_{m-1}}^{n_m} = d_{jn_{m+1}}^{n_m} \wedge |n_{m-1}| > |n_{m+1}|) \end{aligned} \tag{3}$$

Consider a subset of hops $\{n_m, n_{m+1}, ..., n_{m+c}\} \in I_j$ that forms a contiguous chain in L_j, where $c \leq h$. It follows from Eq. (3) that

$$\begin{aligned} & (d_{jn_{m-1}}^{n_m} = d_{jn_{m+c+1}}^{n_{m+c}} \wedge |n_{m-1}| > |n_{m+c}|) \\ & \vee (d_{jn_{m-1}}^{n_m} > d_{jn_{m+c+1}}^{n_{m+c}}) \text{ for } h \geq c \geq 0. \end{aligned} \tag{4}$$

On the other hand, by assumption, every hop $n_i \in L_j - I_j$ must have $d_j^{n_i} < \infty$ when it uses SOC to select its next hops and hence join L_j. Therefore, according to SOC, the following two equations must be satisfied for any $n_i \in L_j - I_j$:

$$d_{jn_i}^{n_i - 1} \geq d_j^{n_i} \tag{5}$$

$$d_j^{n_i} > d_{jn_{i+1}}^{n_i} \geq d_j^{n_{i+1}} \tag{6}$$
$$\vee \, (d_j^{n_i} = d_{jn_{i+1}}^{n_i} \geq d_j^{n_{i+1}} \;\wedge\; |n_i| > |n_{i+1}|).$$

Consider a subset of hops $\{n_l, n_{l+1}, ..., n_{l+k}\} \in L_j - I_j$ that forms a contiguous chain in L_j, where $k \leq h$. It must be the case that either $d_j^{n_{l+i}} > d_{jn_{l+i+1}}^{n_{l+i}} \geq d_j^{n_{l+i}}$ for at least one hop n_{l+i} in the chain, or that $d_j^{n_{l+i}} = d_{jn_{l+i+1}}^{n_{l+i}} \geq d_j^{n_{l+i}}$ and $|n_{l+i}| > |n_{l+i+1}|$ for each hop n_{l+i} in the chain. Accordingly, Eqs. (5) and (6) imply that

$$(\; d_{jn_{l+1}}^{n_l} = d_{jn_{l+k+1}}^{n_{l+k}} \;\wedge\; |n_l| > |n_{l+k}| \;) \tag{7}$$
$$\vee \, (\; d_{jn_{l+1}}^{n_l} > d_{jn_{l+k+1}}^{n_{l+k}} \;) \quad \text{for} \;\; h \geq k \geq 0.$$

It follows from Eqs. (4) and (7) that using SOC enforces the same lexicography ordering among the hops of L_j for any given combination of chains of nodes in L_j that belong to I_j or $L_j - I_j$ and use SOC to select their next hops when they join L_j. Accordingly, it must be true that, if at least one hop in $n_i \in L_j$ is such that $d_{jn_{i+1}}^{n_i} > d_{jn_{k+1}}^{n_k}$, where $n_k \in L_j$ and $k > i$, then $d_{jn_{m+1}}^{n_m} > d_{jn_{m+1}}^{n_m}$ for any given $m \in \{1, 2, ..., h\}$, which is a contradiction. On the other hand, if $d_{jn_{i+1}}^{n_i} = d_{jn_{k+1}}^{n_k}$ for any n_i and n_k in L_j, then $|n_m| > |n_m|$ for any given $m \in \{1, 2, ..., h\}$, which is also a contradiction. Therefore, L_j cannot be formed when routers use SOC to select their next hops to destination j. □

Assume that MIDR is executed in a connected finite network G, that a router is able to detect within a finite time who its neighbor routers are, and that any signaling message sent over a working link between two routers is delivered correctly within a finite time. Further assume that topological changes and destination instance changes stop taking place after a given time t_T. The following theorem proves that MIDR attains shortest paths to the nearest instances of known destinations within a finite time. To simplify the inductive proof, we assume that the cost of any operational link is 1; however, the same basic approach applies to the case of positive link costs [8].

Theorem 2. *If MIDR is used in network G, the routes to destinations converge to the shortest distances to the nearest anchors of the destinations within a finite time after t_T.* □

Proof: Without loss of generality, we focus on a specific destination j. The proof is by simple induction on the number of hops (k) that routers are away from the nearest anchors of destination j. Let the set of anchors in the network for destination j be $A = \{\alpha_1, \alpha_2, ..., \alpha_r\}$, where r is smaller than or equal to the number of routers in the network.

Base case: For $k = 1$, consider an arbitrary neighbor of a given anchor α_i of destination j, with $1 \leq i \leq r$. Given that the signaling between neighbors is reliable and no links fail after time t_T, router n_1 must receive an update $U_j^{\alpha_i}$ from α_i stating $d_j^{\alpha_i} = 0$, $a_j^{\alpha_i} = \alpha_i$, and $sn_j^{\alpha_i} = s(\alpha_i)$ (the most recent sequence number created by α_i) within a finite time after t_T; and it must update $d_{j\alpha_i}^{n_1} = 0$, $a_{j\alpha_i}^{n_1} = \alpha_i$, and $sn_{j\alpha_i}^{n_1} = s(\alpha_i)$.

Because $d_{j\alpha_i}^{n_1} = 0$ and $sn_{j\alpha_i}^{n_1} = s(\alpha_i)$ always satisfy SOC at router n_1 for destination j, it must be the case that $\alpha_i \in S_j^{n_1}$. Furthermore, any other next hop in $S_j^{n_1}$ must also be an anchor, because the smallest link cost between neighbors equals 1 and hence the smallest value of $d_j^{n_1}$ equals 1. Router n_1 must set $d_j^{n_1} = 1$ and send and update stating that distance, together with the identifier of that anchor and the sequence number it created, after a finite time $t_1 > t_T$. Therefore, the theorem is true for the base case.

Inductive step: Assume that the theorem is true for any router n_k that is k hops away from its nearest anchors of destination j. It must be true that $d_j^{n_k} = k$ after a finite time $t_k > t_1$. By assumption, the signaling between neighbors is reliable and no links fail after time $t_T < t_k$; therefore, each neighbor of n_k must receive updates from n_k stating $d_j^{n_k}$, $a_j^{n_k}$, and $sn_j^{n_k}$ a finite time after t_k. Accordingly, each neighbor p of n_k must update $d_{jn_k}^p = k$, $a_{jn_k}^p = a_j^{n_k}$, and $sn_{jn_k}^p = sn_j^{n_k}$ a finite time after t_k.

Let router $q \in N^{n_k}$ be such that it is more than k hops away from any anchor of destination j. Because $d_j^{n_k} = k$ is the shortest distance from n_k to destination j after time t_k, router q cannot have any neighbor reporting a distance to j smaller than k after time t_k. Therefore, $d_{jn_k}^q$ must satisfy SOC within a finite time after t_k and router q must make n_k a next hop to destination j a finite time after t_k. Furthermore, any neighbor of q in S_j^q must have reported a distance of k hops to j. Router q selects the anchor in S_j^q with the smallest identifier, and sends an update within a finite time $t_{k+1} > t_k$ stating $d_j^q = k + 1$, together with the identifier of its chosen nearest anchor and the most recent sequence number created by that anchor. Therefore, router q and hence any router $k + 1$ hops away from the nearest anchors of destination j must attain a shortest distance of $k + 1$ hops to destination j within a finite time, and the theorem is true. □

6 Performance Comparison

In this section we present a theoretical and a simulation-based comparative analysis of the performance of the proposed scheme against that of traditional approaches based on routing to single-instance destinations. In the theoretical performance analysis we show that, even without using interest-driven signaling, MIDR is orders of magnitude more efficient than traditional approaches that require routers to acquire routing information for all instances of each destination. In the simulation based analysis, and in order to show that the principles described here apply not only to semi static networks such as the Internet, but to much more dynamic networks with far less network resources such as the MANETs, we compare the performance of Id-MIDR against that of modified versions of AODV [45] and OLSR [27] that establish routes from sources to the nearest instance of the intended destination. Both analyses confirm that MIDR is more effective, efficient and scalable than traditional approaches.

Other interesting simulation scenarios, including comparisons between solutions based on traditional routing and MIDR on intra-AS and inter-AS topologies are beyond the scope of this paper and deserve further research. This particular type of settings, however, would give MIDR unfair advantage because MIDR limits overhead when destinations are multi-homed and paths to destination instances nearby would rarely change.

6.1 Complexity

Given that our comparison must apply to any protocol based on a given approach, we focus on the communication and time complexities of the approaches. Assuming that all transmissions over any given link are successful, the communication complexity (CC) of a routing algorithm is the number of messages that must be transmitted for each router to have correct routing information about all the destinations. The time complexity (TC) of a routing algorithm is the maximum time needed for all routers to have correct routing information for all destinations.

The number of routers in the network is denoted by N and E denotes the number of network links. The number of different multi-instantiated destinations available in the network is denoted by D, the average number of instances of the same destination is denoted by R, the average number of neighbors per router is l, and the network diameter is d.

We assume a network without hierarchical routing, and that a separate control message is sent for any given link-state advertisement (LSA) or distance update. In practice, multiple LSAs and distance updates can be aggregated to conserve bandwidth. However, given that the maximum size of a control message is a constant value independent of the growth of N or D, this aggregation does not change the order size of the overhead incurred by the routing protocols. We consider three approaches for routing to multi-instantiated destinations based on complete information about destination instances.

6.1.1 Link-State Routing (LSR)

In LSR approach, routers send to each other all the information about the network topology and the location of every instance of each destination. A router must transmit an LSA for each adjacent link and each local destination instance, and each LSA must be sent to all the other routers in the network, which may require transmitting the same LSA over E links. Given that a router can be a maximum of d hops from the source of a given LSA, the time and communication complexities of LSR are: $TC_{LSR} = O(d)$; $CC_{LSR} = O(RDE + lNE)$.

6.1.2 Loop-free Distance-Vector Routing (LDVR)

Routers running LDVR maintain loop-free distances to all network nodes and also the location of every destination instance. The traditional distance-vector routing (DVR) approach subject to looping problems (e.g., RIP) cannot be used for routing to multi-instantiated destinations. DVR signaling can traverse long paths and "counting to infinity" can occur, this approach is known to have $O(N)$ time complexity and $O(N^2)$ communication complexity [29].

By contrast, in a loop-free distance-vector algorithm (LDVR), routing updates regarding a destination would take a time proportional to the network diameter to reach all routers, similar to the LSR case. Each router must send a distance update for each local destination instance and for each node to which destination instances are attached. Hence, the time and communication complexities of LDVR are: $TC_{LDVR} = O(d)$; $CC_{LDVR} = O(RDE + NE)$.

6.1.3 Distributed Hash Table (DHT)

In this approach, all network routers participate in the DHT and routers must maintain the mapping between a destination name or identifier or a destination instance and the node in the DHT representing the destination or an instance of it. The DHT approach that incurs the least amount of overhead is a virtual DHT with one-hop routing [24], such that routers run the DHT locally and maintain routes to all routers in the network. The communication complexity associated with publishing a destination in the DHT and associating R sites with the destination is $O(RdD)$ assuming no loops. The communication complexity of maintaining routes to all routers that form the DHT is the same as in the LSR approach, given that link-state routing is typically used. Hence, assuming the fastest possible propagation of routes to all routers, the time and communication complexities of this approach equal: $TC_{DHT} = O(d)$; $CC_{DHT} = O(RDd + lNE)$.

6.1.4 Multiple Instance Destination Routing (MIDR)

Independently of the number of instances for a given destination, the information a router communicates for a given destination in MIDR is only its distance to the nearest anchor of the destination, plus the anchor identifier and the latest sequence number created by that anchor. Given that MIDR does not incur any routing-table loops, any routing information propagates as fast as the shortest path between its origin and the recipient. However, in contrast to routing schemes that require routing updates for each destination instance, as the number of instances of a destination increases, the distance from a router to the nearest replica of the destination (x) decreases, and $x \leq d$. In addition, the number of messages required for all routers to have a correct distance to the nearest instance of a given destination is always $O(E)$, regardless of the number of instances of the destination, because each router simply communicates to each of its neighbors its distance to the nearest instance of a destination and hence updates about a single instance per destination traverse each network link. Given that there are D destinations in the network, the time and communication complexities of MIDR are: $TC_{MIDR} = O(x)$; $CC_{MIDR} = O(DE)$.

The spatial complexity of Id-MIDR is $O(D)$ because nodes only keep state about the closest instance of a destination if they are part of the region of interest of that destination. On the other hand, traditional approaches to anycasting in MANETs have spatial complexity of either $\Theta(DI)$ for the case of proactive routing or $O(DI)$ for the case of on-demand routing, where I is the maximum number of instances per destination. This is because nodes learn routes to all the instances of the destinations. Similarly, the network complexity measured in terms of the number of sources of control information is $O(D)$ for Id-MIDR, $\Theta(DI)$ for solutions based on proactive routing and $O(DI)$ for solutions based on on-demand routing.

The performance benefits attained with the proposed framework for routing to multi-instantiated destinations are clear from the results we have stated. As the number of destinations and their instantiations become much larger than the number of nodes, any protocol that requires knowledge of all instances of destinations for correct routing requires order $O(RDE)$ (e.g., LSR) or $O(RDd)$ (e.g., DHT) messages and $O(d)$ steps to converge. By contrast, MIDR requires order $O(DE)$ messages and $O(x)$ steps (where $x \leq d$) to converge.

Fig. 3 illustrates the signaling overhead of LSR, DHT and MIDR. LDVR is not shown, given that it is very much the same as LSR for large values of D and R. To focus on the effect of D and R, we set $N = 1000$, and assume that l and d are of order $log(N)$, which makes E order $Nlog(N)$. It is clear that MIDR is always orders of magnitude more efficient than LSR. MIDR is more efficient than all the other approaches when either: (a) the number of destinations is of order $O(N)$, which is the case for IP routing; or (b) the number of destinations is much larger than $O(N)$ and destinations are instantiated opportunistically in the network, which is the intended effect in all ICN architectures. As the number of destinations becomes far larger than the number of network nodes and destination instances proliferate, the signaling overhead of MIDR and an ideal DHT approach become the same.

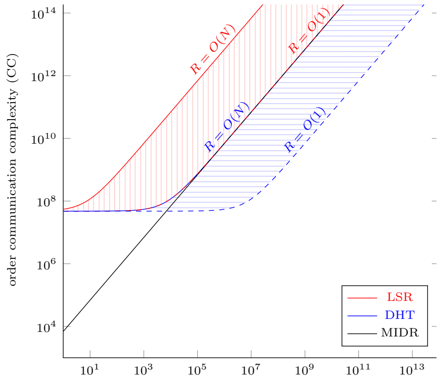


Fig. 3: Signaling overhead as a function of D and R for the forwarding of content requests

6.2 Simulation

We present detailed simulation results comparing the performance of Id-MIDR against that of variations of AODV ("Anycast AODV") and OLSR ("Anycast OLSR") that route data packets to the nearest instance of the intended destinations. In the implementation of Anycast AODV, if a node does not have a valid route to any destination instance, it generates a Route Request that contains the identifier of the group and that can be answered by any destination instance. Route Replay messages travel back to the source, and if more than one of them reaches the source, the latter simply selects the route with the shortest distance. Once a route has been selected, Anycast AODV employs the same route maintenance mechanisms as AODV. In the implementation of Anycast OLSR, nodes explicitly publish if they are an instance of a given destination and sources simply select the nearest destination instance. The source code of the three protocols can be obtained from https://github.com/idmidr/ns-allinone-2.35

We use packet delivery ratio, end-to-end delay, control overhead, and total overhead as our performance metrics. A packet is considered as delivered if it is received by any instance of its intended destination. The control overhead is the average number of control bytes generated by the routing protocols and the total overhead is the average number of bytes that are actually transmitted by the physical layer. The routing protocols are tested with IEEE 802.11 DCF as the underlying MAC protocol with two different data rates, namely, 2Mpbs and 11Mbtps. Nodes move according to the random waypoint mobility model. We used the discrete event simulator ns-2.35, and each simulation was run for twenty different seed values. The confidence level for all the results presented in this section is 95%. Unless stated otherwise, Table 1 lists the details of the simulation environment.

The results are organized in two sets. In the first set (Sections 6.2.1 - 6.2.4) we evaluate the performance of the protocols in a medium-size network of 100 nodes that move around a simulation area of 1400×1400 m^2. In the second set of experiments we evaluate the performance of the protocols in a large network of 1000 nodes that move around a simulation area of 5000×5000 m^2.

TABLE 1: Simulation Environment

Total nodes	100	Node placement	Random
Simulation area	1400×1400m^2	Simulation time	300s
MAC Protocol	IEEE 802.11 DCF	Tx. rate	2Mbps, 11Mbps
Data source	CBR	Pkts. per src.	1000
Mobility model	Random waypoint	Pause time	10s
		Min.-Max. Vel.	1-20m/s

6.2.1 *Number of concurrent data flows*

In these experiments, we evaluate the performance of the protocols as the number of concurrent CBR flows increases from 3 to 24. There are three destinations with three instances per destination. Sources and instances are selected uniformly at random but no source is also an instance of a destination. From Fig. 4(a) we can observe that as the number of concurrent sources increases, the packet delivery ratio (*PDR*) attained by both Anycast AODV and Anycast OLSR decreases sharply, and in particular, for the scenario where nodes have a data rate of 2Mbps. This is mainly due to the extra overhead induced by the traditional protocols (see Fig. 4(c)) while trying to maintain routes to the destinations and to the fact that these protocols tend to use longer routes to the destinations, which is reflected in a larger total overhead that includes data overhead ((see Fig. 4(d))). On the other hand, the *PDR* attained by Id-MIDR remains fairly constant across the different values for the number of concurrent flows. The reasons for this good performance are as follows. (1) By using the Successor-Set Ordering Condition (SOC), nodes running Id-MIDR tend to maintain routes to the closest instance, even if the network topology is constantly changing. (2) The control overhead induced by Id-MIDR grows sub-linearly with respect to the number of concurrent data flows (see Fig. 4(c)). This is because nodes acquire routing state regarding their closest destination instance only. Moreover, as the regions of interest are established, control information is only disseminated among interested nodes. Figure 4(b) shows that the delay attained by Anycast AODV and Anycast OLSR is up to an order of magnitude higher than that of Id-MIDR. This is because Id-MIDR generates less high priority control packets that tend to stale data packets at the data queues, and the fact that the routes computed by Id-MIDR are in general shorter than those computed by Anycast AODV or Anycast OLSR because Id-MIDR always routes packets towards the nearest instance of the destination. Figure 4(d) shows that Id-MIDR is clearly the least expensive protocol.

6.2.2 Number of instances per destination

In this set of experiments, we evaluate the performance of the protocols as the number of instances per destination increases from 1 to 8. There are three destinations with five active sources per destination. As in the previous scenario, sources and instances are selected uniformly at random but no source is also an instance of a destination. The objective of these experiments is to evaluate the ability of the protocols to establish and maintain routes to the closest instance of a destination. From Fig. 5(a) we can notice that Id-MIDR clearly outperforms Anycast AODV and Anycast OLSR for all the values in the number of instances per destination, but in particular in the most challenging scenarios when the destinations have just a few instances. These results show that by using SOC, Id-MIDR is more effective at finding short routes to the closest instances and hence it is able to deliver more packets. The end-to-end delay attained by Id-MIDR is also consistently better than that of Anycast AODV and Anycast OLSR. As in the previous scenario, this is mainly due to the reduced overhead induced by Id-MIDR (see Figs. 5(c) and 5(d)) and to Id-MIDR's ability to compute better routes by means of SOC.

The behavior shown by the control and total overhead induced by Id-MIDR provides more evidence supporting our claim that the use of SOC and regions of interest to confine the dissemination of control traffic tend to yield to efficient protocol implementations. From the figures we can observe that both control and total overhead decrease as the number of instances per destination increases. The reason is as follows. As the number of instances increases, the average path length to the nearest instance also decreases which in term reduces the size of the regions of interest limiting even more the dissemination of control packets. These results are consistent for the two different values of data rate, which indicates that the good performance of Id-MIDR comes not only from the reduced overhead but from the fact that the routing structures established by Id-MIDR are more robust and easier to maintain.

6.2.3 Number of destinations

In these experiments, we increase the number of destinations from 1 to 7. For these experiments, there are five concurrent active sources per destination, and every destination is composed of three instances. Sources and instances are selected uniformly at random but no source is also an instance of the destination. Figures 6(a)-6(d) present the results which are consistent with those reported in the scenario with increasing number of concurrent data flows (Section 6.2.1), namely, that Id-MIDR delivers similar or more data packets with smaller delays while inducing less control and total overhead. The latter is true for the two different values of data rate.

6.2.4 Node speed

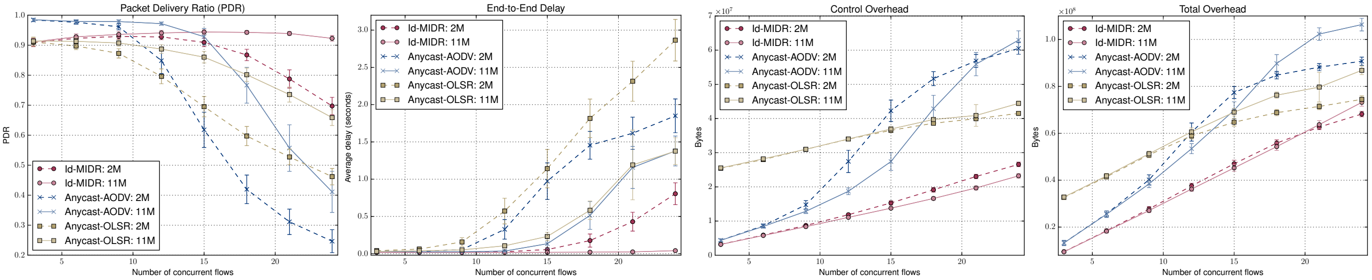
In this experiments we fixed the number of destinations and the number of instances per destination to three, and the number of sources per destination to five; and increase the nodes' speed from 1m/s to 30m/s (or 3.6 to 108 Km/h) to cover pedestrian and vehicular speeds. The pause time is set to 0 seconds to make the network more dynamic. From Figure 7(a) notice that Id-MIDR performs reasonably well by delivering almost all the packets in the static network and close to 85% of the packets when nodes move at relatively high speeds. Unfortunately, nodes running Anycast AODV, but in particular those running Anycast OLSR, are not able to cope with the constant topology changes and their performance falls as the speed of the nodes increases. From Figures 7(c) and 7(d) we can observe that the mechanisms employed to restrict the dissemination of control information, in conjunction with SOC and the proactive signaling used inside of the regions of interest make the control and total overhead induced by Id-MIDR almost insensitive to the network dynamics. Again, this is not the case for neither Anycast AODV nor Anycast OLSR whose control and total overhead increase as the protocols try to repair the routes to the instances of the destinations.

Lastly, Figure 7(b) shows that the end-to-end delay attained by Id-MIDR is considerably better than that of Anycast AODV and Anycast OLSR, and in particular when nodes move at high speeds. The similarity between the results shown in Figures 7(b) and 7(c) allow us to highlight the strong relation between the control overhead and the end-to-end delay which confirms our intuition that well-designed more efficient routing protocols tend to also be more effective.

6.2.5 Large Network

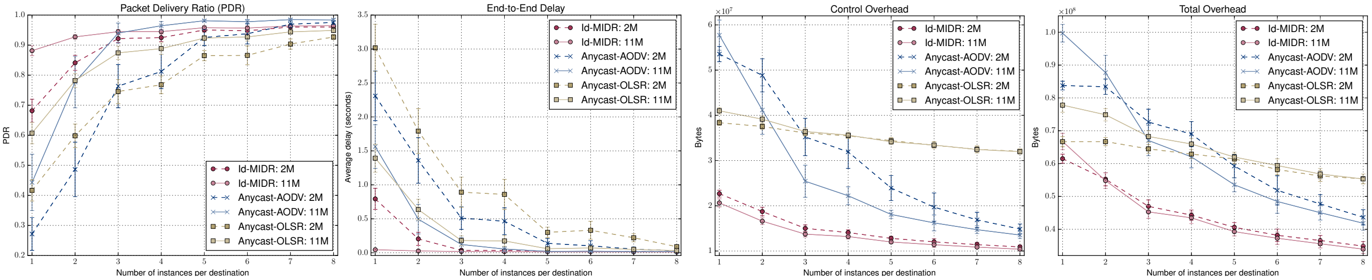
In this set of experiments, we evaluate the performance of the protocols in a large network composed of 1000 nodes that move around a simulation area of $5000 \times 5000\text{m}^2$. There is a single destination with four static instances that were placed at the following set of x, y-coordinates $\{(1250, 1250), (1250, 3750), (3750, 1250), (3750, 3750)\}$. Data flows were generated following an exponential distribution. CBR flows are established from randomly selected nodes with mean interarrival time of $1/\lambda = 2$ seconds and with an increasing mean flow duration time starting from $1/\mu = 100$ seconds and up to $1/\mu = 200$ seconds. This selection of parameters is intended to model a scenario where any MANET node may try to reach a very popular destination such as the Internet.

From Figure 8(a) we can observe that Id-MIDR clearly outperforms the other protocols by delivering up to 20% more data packets than Anycast AODV and up to 40% more data packets than Anycast OLSR. Moreover, Id-MIDR is able to deliver around 60% percent of the packets in the most congested scenario where there are $\lambda/\mu = 100$ concurrent data flows on average, even in the case when nodes transmit at 2Mbps. As in the previous experiments, Id-MIDR consistently attains smaller end-to-end delays than Anycast AODV or Anycast OLSR while inducing far less control and total overhead. Overall, the good performance achieved by Id-MIDR in this set of scenarios shows that by using SOC in conjunction with the use of regions of interest, Id-MIDR is able to scale to large dynamic networks with relatively high number of concurrent data flows. This is particularly true in the most challenging scenario where nodes transmit at 2Mbps.



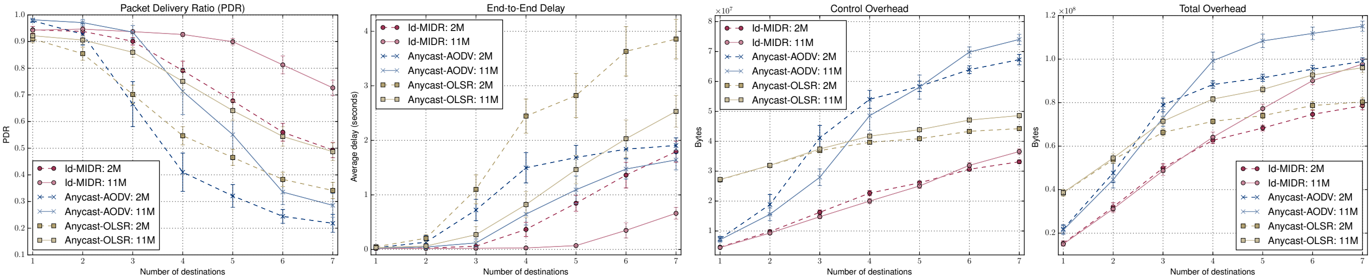
(a) (b) (c) (d)

Fig. 4: Performance with increasing number of concurrent data flows



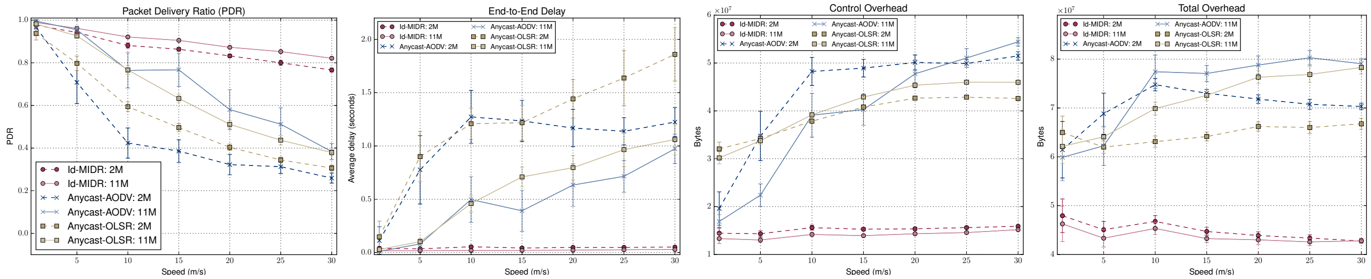
(a) (b) (c) (d)

Fig. 5: Performance with increasing number of instances per destination



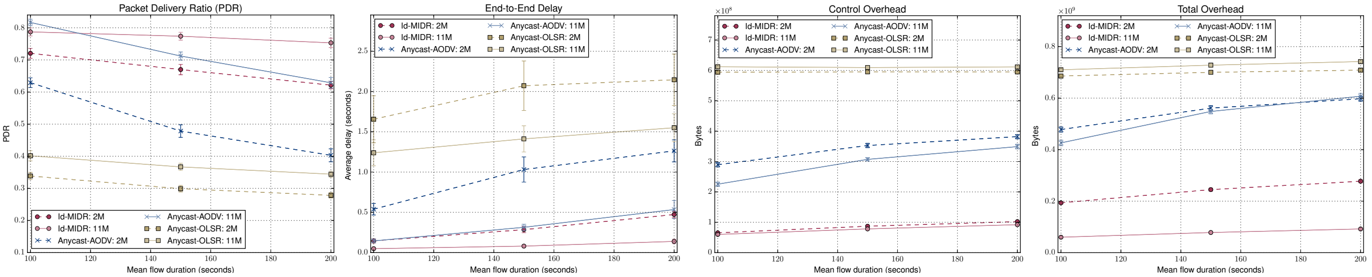
(a) (b) (c) (d)

Fig. 6: Performance with increasing number of destinations



(a) (b) (c) (d)

Fig. 7: Performance with increasing node speed



(a) (b) (c) (d)

Fig. 8: Performance in large network with increasing mean flow duration

7 Applications

Changing the routing protocols for the Internet and ICNs to operate based on algorithms designed for multi-instantiated destinations opens up a vast number of opportunities.

7.1 Efficient and Flexible IP Routing and Anycasting

MIDR constitutes the basis for routing within autonomous systems capable of converging as fast as a link-state routing protocol, but incurring far less signaling overhead. In addition, MIDR supports multi-path Internet routing in which the multi-paths to destinations need not be equal-cost multipaths, which cannot be attained using existing intra-domain Internet routing protocols (RIPv2, EIGRP, and OSPF [38]).

With traditional routing to single-instance destinations, a network assigned a given IP address range cannot be physically partitioned, because permanent routing-table loops can occur. By contrast, MIDR allows the same IP address range to be used by multiple physical networks without causing any routing problems. Allowing an IP address or identifier to be multi-homed (i.e., with physical network components or hosts using the same IP address not being connected directly) opens up many new possibilities for hierarchical routing and anycast routing.

In contrast to all prior algorithms for single-path and multipath routing, MIDR supports loop-free anycasting, because it is inherent in the computation of loop-free multipaths to destinations. Within small networks, MIDR can support routing to "logical addresses" [36] assigned to sites providing well-known services (e.g., directory services, local printers) or nodes connecting to things of a given class.

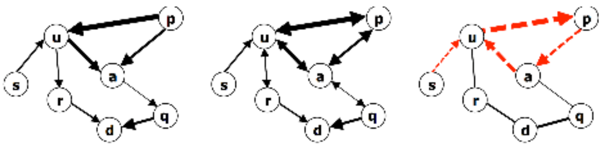
Given that MIDR supports routing to different connected components of a network sharing the same identifier, MIDR can support Internet anycasting within an autonomous system (AS) correctly. Furthermore, if information must be shared among two or more network components sharing the same IP address range, or if a specific physical network in the address range must be contacted, then a MIDST can be established to connect all components with the same IP address range.

The scaling problem of Internet anycasting (i.e., having to advertise too many logical addresses across the Internet) can be addressed with MIDR through hierarchical address assignments to anycast groups and the recursive use of MIDSTs. Anycast groups can be part of IP address ranges associated with Internet anycasting in the same AS, and MIDSTs can be established for them. A request for a specific anycast group whose address is in a given address range can be routed to and over the corresponding MIDST, until it reaches a network that has members in the target anycast group. The request can then be routed within that network either over a MIDST defined for a more specific address range, or directly to the nearest anchor of the target anycast group address. Anycasting across autonomous systems involves policy-based routing and is the subject of future work; however, it can also be addressed using routing to multi-instantiated destinations in the context of BGP or alternative routing protocols.

7.2 Efficient Name-Based Content Routing

ICN architectures (e.g., [12], [15], [26], [41], [47]) rely on the control plane to provide multiple routes to the same named data object (NDO). In this regard, the results in Section 6.1 indicate that MIDR constitutes a promising approach for the design of name-based content routing protocols. MIDR is orders of magnitude more efficient than current approaches being used for name-based content routing (e.g., NLSR [34] in NDN [41]) as the number of destinations D (NDOs) and the destination instances R (replicas of NDOs) increase.

In addition, MIDR has a major positive impact in the data plane of an ICN. For example, the NDN project [34], [54] has argued that loop freedom in the control plane is not needed for Interests (content requests) to be forwarded in an ICN, because Interests that carry a content name and a nonce created by the origin of the Interest cannot loop (e.g., see [54], Section 2). We illustrate below that loop-free multipath routing in the control plane is far more desirable than routing approaches (e.g., NLSR) that do not enforce loop-free paths in an ICN operating with a data plane similar to that of NDN and such that: (a) an Interest is identified simply by the name of the content being requested and a nonce; and (b) forwarding of Interests is done on a hop-by-hop basis using that information and allows for routers to "aggregate" Interests, which means that a router receiving multiple Interests asking for the same NDO forward only the first Interest and remember the interfaces over which subsequent Interests for the same NDO were received. The name and nonce carried in an Interest can be used to *detect* a loop with some high probability *after* the Interest traverses a loop and reaches a router that has recorded the same name and nonce of the Interest in its PIT (Pending Interest Table).



(a) MIDR (b) NLSR (c) Interest looping with NLSR

Fig. 9: Effect of multipaths in FIBs on Interest looping for the forwarding of content requests

Fig. 9 shows the multipath routing induced by MIDR and NLSR in a seven-node network in which NDO j is announced only by router d, and all links have unit cost. The thickness of a link indicates how congested it is, with thinner lines indicating more congestion. The arrowheads in the figure show the direction in which Interests can propagate over links. Fig. 9(a) shows the multipaths built in the FIBs (Forwarding Information Base) when MIDR is used, which are loop-free because routers coordinate the updates they make to their routing tables using SOC, and FIBs are derived from such routing tables. Fig. 9(b) illustrates the multipaths in the FIBs when NLSR is used, which are not loop-free, because routers compute multiple paths to an NDO independently of other routers. Using the information about the topology and the locations of NDOs, each router running NLSR first computes a shortest path to an NDO, deletes the adjacent link belonging to that path and computes a new path; the process continues until the router has considered all its adjacent links.

Let router s issue an Interest for NDO j. With MIDR, the Interest is bound to reach router d, independently of the Interest forwarding strategy used in the data plane. By contrast, the Interest issued by s may traverse a loop once when NLSR is used, depending on the data-plane state, which Fig. 9(c) illustrates with red dashed lines. Even if we assume that Interest loop-detection is certain in NDN, assuming that the content routing protocol does not play an important role has two major limitations.

Waiting-To-Infinity Problem: We have proven [20], [21], [22] that Interest aggregation combined with the Interest-loop detection mechanism used in NDN can lead to Interests being aggregated while traversing forwarding loops without such loops being detected. This results in aggregated Interests "waiting to infinity" for responses that never come. Furthermore, we have shown that no forwarding strategy can be designed that allows Interest aggregation and detects loops by uniquely identifying Interests. In the example of Fig. 9, router s and router a may receive Interests from consumers asking for the same NDO. Router u can forward the Interest received from s and aggregate the Interest received from a. BEcause of the forwarding loop, router a aggregates the Interest received from p that was originated by s. if this happens, the Interests at u and a will have to wait until their time-to-live expires.

Depth-First-Search Problem: A second problem using a routing protocol prone to routing loops in NDN is the response time to loops that are detected. Interest forwarding strategies can be designed to cope with Interest-looping problems resulting from inconsistent loop-prone routes due to network dynamics, or the interaction of forwarding strategies with stable FIBs that need not correspond to loop-free routes (as Fig. 9(c) illustrates). For example, in the example of Fig. 9, router u would inform router a about the forwarding loop with a negative acknowledgment to the Interest. In turn, router a could use router q to forward the Interest. Other strategies similar to what NDN does could be used. However, making the forwarding strategy cope with forwarding loops is not guaranteed to work, and amounts to a depth-first search approach to finding viable paths. This is known to have time complexity $O(N)$ in finding viable paths (i.e.,deliver Interests) because of backtracking due to looping [2]. By contrast, a loop-free routing protocol finds paths with time complexity $O(d)$, because it amounts to a breadth-first search for loop-free paths.

Clearly, a loop-free routing protocol is not absolutely needed to make NDN work correctly. Indeed, the NDN forwarding strategy can be fixed as discussed in [22]. However, a routing protocol like MIDR makes any forwarding strategy more efficient, because it quickly eliminates paths that are not viable and leaves the forwarding strategy with the far simpler task of choosing more efficient paths among the viable paths provided by the routing protocol.

8 Conclusion

Multi-Instantiated Destination Routing (MIDR) was introduced as an example of routing to destinations that can be arbitrarily multi-instantiated in a network. MIDR provides multiple loop-free paths to the nearest instances of destinationsbased solely on distances to destinations. It does not require replicating information about the physical topology of the network or the location of instances of the same destination, or exchanging path information to destination instances. MIDR was shown to be loop-free at every instant and to converge to the shortest paths to the closets instances of destinations. MIDR was shown to have smaller time and communication complexity than traditional routing approaches applied to multi-instantiated destinations, such as routing based on link states, DHTs, or traditional loop-free distance-vector routing. Moreover, the results of a series of detailed simulation experiments illustrate that an instantiation of the proposed framework for MANETs (Id-MIDR) clearly outperforms an AODV-based protocol by consistently attaining higher delivery ratios with lower delays, while inducing far less control and total overhead. Moreover, the results of our experiments show that Id-MIDR is able to scale to relatively large networks even if the traffic load is high.

A couple of applications of MIDR were discussed to further motivate the enormous potential that introducing routing to multi-instantiated destinations has in the future of computer networks in general, and the Internet and ICNs in particular. We argued that much better performance, functionality and simplicity can be attained for IP routing and name-based content routing by adopting multi-instantiated destination routing.

The results we have presented open up many avenues for future research on routing to multi-instantiated destinations for mobile networks, the Internet and ICNs. Protocols are needed to address specific applications (e.g., [19]), and different techniques can be used to establish lexicographic orderings. Lastly, the proposed approach should be extended to address policy-based routing to multi-instantiated destinations.

References

[1] B. Ahlgren et al., "A Survey of Information-centric Networking," *IEEE Commun. Magazine*, July 2012, pp. 26–36.

[2] B. Awerbuch, "A New Distributed Depth-First-Search Algorithm," *Information Processing Letters*, pp. 147-150, 1985.

[3] H. Ballani and P. Francis, "Towards a Global IP Anycast Service," *Proc. ACM SIGCOMM '05*, Aug. 2005.

[4] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT)," *Proc. ACM SIGCOMM '93*, Oct. 1993.

[5] P. Baran, "On Distributed Communications Series," *RAND Corporation*, Reports RM-3420-PR to RM-3767-PR.

[6] M.F. Bari et al., "A Survey of Naming and Routing in Information-Centric Networks," *IEEE Commun. Magazine*, July 2012, pp. 44–53.

[7] J. Behrens and J.J. Garcia-Luna-Aceves, "Hierarchical Routing Using Link Vectors," *Proc. IEEE Infocom '98*, April 1998.

[8] D. Bertsekas and R. Gallager, *Data Networks*, Second Edition, Prentice-Hall, 1992.

[9] A.Carzaniga et al., "A Routing Scheme for Content-Based Networking," *Proc. IEEE Infocom '04*, March 2004.

[10] C. P. Low et al., "On anycast routing with bandwidth constraint," *Computer Communications*, Vol. 26, No. 14, Sep. 2003.

[11] Content Centric Networking Project (CCN) [online]. http://www.ccnx.org/releases/latest/doc/technical/

[12] Content Mediator Architecture for Content-aware Networks (COMET) Project [online]. http://www.comet-project.org/

[13] S. Deering, "Multicast Routing in Internetworks and Extended LANs," *Proc. ACM SIGCOMM '88*, Aug. 1988.

[14] S. Deering et al., "The PIM architecture for wide-area multicast routing," *IEEE/ACM Trans. on Networking*, Vol. 4, No. 2, April 1996.

[15] A. Detti et al., "CONET: A Content-Centric Inter-networking Architecture," *Proc. ACM ICN '12*, Aug. 2012.

[16] S. Doi et al., "IPv6 anycast for simple and effective service-oriented communications," *IEEE Communications Magazine*, Vol. 42, No. 5, 2004.

[17] X. Fan et al.,. "Evaluating Anycast in the Domain Name System," *Proc. INFOCOM 2013*, April 2013.

[18] J.J. Garcia-Luna-Aceves, "A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States," *Proc. ACM SIGCOMM '89*, Sept. 1989.

[19] J.J. Garcia-Luna-Aceves, "Name-Based Content Routing in Information Centric Networks Using Distance Information," *Proc. ACM ICN 2014*, Sept. 2014.

[20] J.J. Garcia-Luna-Aceves, "A Fault-Tolerant Forwarding Strategy for Interest-based Information Centric Networks," *Proc. IFIP Networking '15*, May 2015.

[21] J.J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "Enabling Correct Interest Forwarding and Retransmissions in a Content Centric Network," *Proc. ACM ANCS '15*, May 2015.

[22] J.J. Garcia-Luna-Aceves, "Eliminating Undetected Interest Looping in Content Centric Networks," *Proc. IEEE NOF '15*, Sept. 30-Oct. 2, 2015.

[23] M. Gritter and D. Cheriton, "An Architecture for Content Routing Support in The Internet," *Proc. USENIX Symposium on Internet Technologies and Systems*, Sept. 2001.

[24] A. Gupta, B. Liskov and R. Rodrigues, "Efficient Routing for Peer-to-Peer Overlays," *Proc. ACM NDSI '04*, March 2004.

[25] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom '00*, Aug. 2000.

[26] V. Jacobson et al., "Networking Named Content," *Proc. IEEE CoNEXT '09*, Dec. 2009.

[27] P. Jacquet et al., "Optimized link state routing protocol for ad hoc networks." *Proc. IEEE INMIC 2001*, 2001.

[28] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in Mobile Computing (T. Imielinski and H. Korth, Eds), Chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[29] M.J. Johnson, "Updating Routing Tables after Resource Failure in a Distributed Computer Network," *Networks*, Vol. 14, No. 3, 1984.

[30] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)," *Proc. ACM SIGCOMM '00*, Aug. 2000.

[31] U.C. Kozat, "Network layer support for service discovery in mobile ad hoc networks," *Proc. IEEE INFOCOM*, 2003.

[32] S.J. Lee, M. Gerla, and C. Chiang, "On-demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *Mobile Networks and Applications*, Vol. 7, No. 6, 2002.

[33] V. Lenders et al., "Density-Based Anycast: A Robust Routing Strategy for Wireless Ad Hoc Networks," *IEEE/ACM Transactions on Networking*, Vol. 16, No. 4, 2008. Oct. 1997.

[34] A.K.M. Mahmudul-Hoque et al., "NSLR: Named-Data Link State Routing Protocol," *Proc. ACM ICN '13*, Aug. 2013.

[35] M. Marina and S. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," *Proc. IEEE ICNP '01*, Nov. 2001.

[36] J.M. McQuillan, "Enhanced Message Addressing Capabilities for Computer Networks", *Proceedings of the IEEE* , Nov. 1978.

[37] J.M. McQuillan et al., "The New Routing Algorithm for the ARPANET", *IEEE Trans. Commun.*, May 1980.

[38] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols and Architectures*, Morgan Kaufmann, 2007.

[39] Mobility First project [online]. http://mobilityfirst.winlab.rutgers.edu/

[40] S. Murthy and J.J. Garcia-Luna-Aceves, "Congestion-Oriented Shortest Multipath Routing," *IEEE Infocom '96*, March 1996.

[41] NDN Project [online]. http://www.named-data.net/

[42] M. Parsa and J.J. Garcia-Luna-Aceves, "A Protocol for Scalable Loop-free Multicast Routing," *IEEE JSAC*, April 1997.

[43] C. Patridge et al., "Host Anycasting Service," RFC 1546, 1993.

[44] G. Peng et al., "ARDSR: An Anycast Routing Protocol for Mobile Ad hoc Network," *Proc. IEEE CAS Symp. on Emerging Technologies: Mobile and Wireless Comm.* , pp. 505-508, June 2004.

[45] C. Perkins and E.M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE WMCSA*, pp. 90-100, Feb. 1999.

[46] Publish Subscribe Internet Technology (PURSUIT) Project [online]. http://www.fp7-pursuit.eu/PursuitWeb/

[47] Scalable and Adaptive Internet Solutions (SAIL) Project [online]. http://www.sail-project.eu/

[48] M. Schwartz and T.E. Stern, "Routing Techniques Used in Computer Comm. Networks," *IEEE Trans. Comm.*, April 1980.

[49] I. Stoica et al., "Internet Indirection Infrastructure," *Proc. ACM SIGCOMM '02*, Aug. 2002.

[50] J. Wang et al., "An AODV-Based Anycast Protocol in Mobile AdHoc Network," *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, 2003.

[51] S. Weber and L. Cheng, "A Survey of Anycast in IPv6 Networks," *IEEE Comm. Magazine*, Jan. 2004.

[52] D. Xuan et al., "A Routing Protocol for Anycast Messages," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 6, 2000.

[53] G. Xylomenos et al., "A Survey of Information-centric Networking Research," *IEEE Commun. Surveys and Tutorials*, July 2013.

[54] C. Yi et al., "A Case for Stateful Forwarding Plane," *Computer Communications*, 2013.

[55] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "System for Maintaining Multiple Loop Free Paths between Source Node and Destination Node in Computer Network," US Patent 5881243, 1999.

[56] E. Zegura et al., "Application-Layer Anycasting: A Server Selection Architecture and Use in a Replicated Web Service," *IEEE/ACM Trans. Networking*, Aug. 2000.

J.J. Garcia-Luna-Aceves received the B.S. degree in Electrical Engineering from the Universidad Iberoamericana, Mexico City, Mexico in 1977; and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Hawaii at Manoa, Honolulu, HI in 1980 and 1983, respectively. He is a Distinguished Professor of Computer Engineering at the University of California, Santa Cruz (UCSC), holds the Jack Baskin Endowed Chair of Computer Engineering, and is a Principal Scientist at the Xerox Palo Alto Research Center (PARC). He is a Fellow of the IEEE, ACM, and AAAS; and a corresponding member of the Mexican Academy of Sciences. He received the IEEE Computer Society Technical Achievement Award in 2011 and the IEEE Communications Society Ad Hoc and Sensor Networks Technical Committee (AHSN TC) Technical Recognition Award in 2012.

J.E. Martinez-Castillo received the B.S. degree in Electronic Engineering from the Universidad Autonoma Metropolitana, Mexico City, Mexico in 2006; the M.S. degree in Computer Science from the Instituto Politécnico Nacional, Mexico City, Mexico in 2012. He is a Ph.D. student at the Network and Data Science Laboratory at the Centro de Investigación en Computación (CIC) of the Instituto Politécnico Nacional (IPN).

R. Menchaca-Mendez received the B.S. degree in Electronic Engineering from the Universidad Autonoma Metropolitana, Mexico City, Mexico in 1997; the M.S. degree from the Instituto Politécnico Nacional, Mexico City, Mexico in 1999; and his Ph.D. degree in Computer Engineering from the University of California at Santa Cruz in 2009. He is a professor at the Centro de Investigación en Computación (CIC) of the Instituto Politécnico Nacional (IPN).