Achieving Scalable Push Multicast Services Using Global Name Resolution*

Shreyasee Mukherjee, Francesco Bronzino, Suja Srinivasan, Jiachen Chen and Dipankar Raychaudhuri
WINLAB, Rutgers University, North Brunswick, NJ 08902, USA
Email: {shreya, bronzino, sujsrini, jiachen, ray}@winlab.rutgers.edu

***Abstract*—This paper presents a novel approach to achieving scalable push multicast services using the distributed global name resolution service associated with emerging name-based network architectures. The proposed named-object multicast (NOMA) scheme employs unique names to identify multicast groups, while using the global name resolution service (GNRS) to store the tree structure and maintain current mappings to mobile end-user addresses. The NOMA scheme achieves improved scalability and performance over conventional multicast protocols such as PIM-SM and MDSP by taking advantage of the GNRS to simplify tree management and limit control overhead. Performance evaluation results including comparisons with IP multicast are given using a combination of analysis and NS-3 simulation. The results show good scalability properties along with low control overhead for medium to large multicast groups. In addition, NOMA seamlessly handles mobility for end-hosts subscribed to a group, avoiding data losses upon mobility events. Results further demonstrate how separating names from addresses enables NOMA to dynamically forward traffic to mobile users. In conclusion, we describe a proof-of-concept prototype developed for further experimental validation of the proposed NOMA multicast routing scheme.**

I. Introduction

Internet applications like video streaming, online gaming and social networks, e.g. Twitter, often require dissemination of the same piece of information to multiple consumers at the same time. While multicast routing protocols have long been available, most of these applications rely on unicast based solutions that exploit overlay networks aimed at improving the efficiency of pushing the required data without support from the network. Recent increases in network traffic associated with the growth of mobile devices, Internet-of-Things (IoT) devices, smart wearables and connected vehicles, motivate the need for efficient push multicast, a service that is not well-addressed through overlay solutions. Consider for example IoT based messaging scenarios: a typical query involves sending short messages to hundreds or thousands of users or application agents, so that scalability becomes an issue, as multiple unicast messages through an overlay service can easily overload the network. Mobility of end-devices results in additional complexity, especially for dynamic environments such as vehicular communications. For example, if a single warning message needs to be pushed to hundreds of cars and pedestrians in a given area, multicast groups would need to be maintained across a large number of access networks in order to efficiently support such one-to-many communication.

*Research supported under NSF Future Internet Architecture - Next Phase (FIA-NP) grant CNS-1345295

Using appropriate multicast routing solutions would help solve these problems by improving network efficiency, while reducing the complexity and cost of deploying such applications. However, existing network-layer multicast solutions (e.g., PIM-SM [1], MOSPF [2]) have not been widely adopted due to fundamental problems that are a by-product of the original Internet design geared toward static host-centric communication. These solutions implicitly couple the forwarding path (*location*) with the multicast group (*name*). Whenever a receiver moves to a new location, it has to rejoin the multicast tree it was previously a part of and the network has to change the tree structure accordingly. This can cause packet loss during the process and large amount of distributed control traffic is generated to modify the tree structure. The problem becomes particularly acute for applications like Twitter where each receiver might have more than 100 groups to join each time it moves. Secondly, extending these protocols to inter-domain has achieved mixed results, with issues of scalability and coordination across domains [3]. For example IP multicast based on PIM-SM [4] relies on rendezvous points (RPs) as the shared root of a tree. However domains are often unwilling to have RPs for their local groups to be maintained in other domains. This leads to having RPs in every domain connected in a loose mesh, that require periodic flooding of control messages for coordination and management. Multicast group address assignment may require a separate protocol altogether, such as the Multicast Address-Set Claim (MASC) protocol used in conjunction with BGMP [5]. All of these problems have negative consequences for highly dynamic environments and emerging application scenarios. For example, in the vehicular use-case previously described, group membership changes rapidly with vehicular mobility. In addition, the context of data-delivery may change with time as well. An accident or traffic-alert push-notification to a group of cars in NJ Turnpike is such an example. Table I describes a sample set of application scenarios that require efficient multicast primitives and their characteristics.

Application layer solutions for multicast have also been explored in this context; works like SCRIBE [6] and ZIGZAG [7] sought to find scalable and efficient solutions by building an overlay among the receivers in a tree or mesh structure. These solutions do address mobility and inter-domain management issues, but due to the lack of topology awareness, they may incur high levels of network traffic. In addition, forcing the end hosts to replicate packets, instead of dedicated routers results

Application	Multicast Type	Group Size	Group Flux	Group Longevity	Data Flow Size
IoT commands	Push	1000's	Hours	Days	KB-MB
Accident notification	Push	100's	Seconds	Minutes	KB
Twitter	Pull	100's of 1000	Minutes	Months	KB-MB
IPTV	Pull	1000's	Relatively static	Months	GB
Multiplayer games	Push/Pull	100's	Hours	Hours	GB

TABLE I
EMERGING MULTICAST APPLICATION AND THEIR CHARACTERISTICS

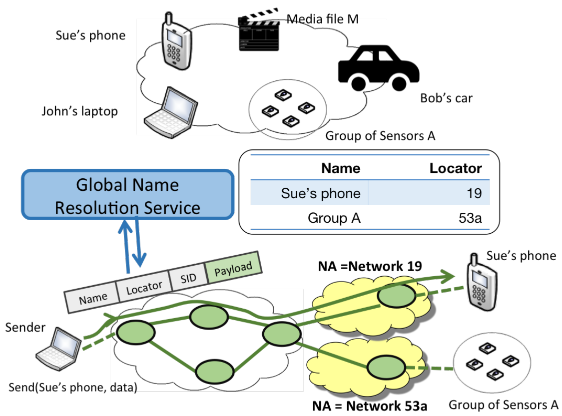

Fig. 1. Named Object abstraction with clean separation of naming and addressing

in heavy workload on the end hosts, which may have intrinsic power and computation constraints.

Based on the above considerations, a native network layer multicasting solution is identified as an important goal for future networks which are increasingly required to support many-to-many communication modes. We propose a solution based on named objects and a dynamic name-resolution service for mapping names to routable network entities. Separating names (identities) from addresses has been advocated by the research community [8]–[10] for quite some time and has inherent benefits in handling mobility and dynamism for one-to-one communication. But they also provide additional advantages by facilitating creation of new service abstractions that can be used to design solutions for multicast services. First, names can be used to represent many different Internet objects; for example, a cell-phone, a person, or a group of devices, as shown in Fig. 1; the latter perfectly applies in the context of multicast to define participation of end-hosts. Moreover, new entities can be integrated within these names, not being constrained to end points; through this, we gain the ability to directly refer to network entities that actively participate in the formation of a multicast tree, such as routers that implement the multicast routing protocols.

We exploit names to design a Named Object Multicast (NOMA) solution which relies on separation of names and addresses using a globally distributed, logically centralized name resolution service, similar in spirit to an evolved DNS. In NOMA each multicast group is identified by a unique name across all domains, thus separating routing logic from group management. NOMA takes advantage of the dynamic name resolution service to manage the tree, using name recursion, to store the tree topology. This is achieved by mapping unique

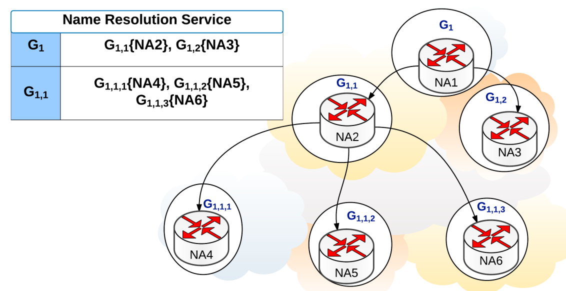

Fig. 2. Hierarchical tree structure maintained in a name resolution service, with names of tree nodes recursively mapping to routable addresses

names assigned to participating routers to their children nodes, as shown in Fig. 2. Data forwarding is then performed using tunnels between participating nodes; end-to-end information is preserved within the packet, while the information globally available in the name resolution service is used to identify next hops in the distribution path allowing for quick branching and replicating decisions. Finally, dynamicity of mobile environments is handled by decoupling the participants name from their location through the resolution service and periodically recomputing the multicast tree; the system first needs to translate the name into a list of host names participating in the multicast group. The routable address (locator) of each host (whether mobile or static) can then be identified by a subsequent query to the name resolution service.

The remainder of the paper provides the details of our design and performance evaluation of the proposed scheme which include:

- The design of NOMA architecture that leverages use of names and global name resolution service to manage multicast routing protocols;
- An efficient centralized tree-construction mechanism that minimizes the network traffic with relatively low computational overhead; and
- Large-scale simulations to demonstrate the reliability, efficiency and scalability of NOMA design even when there is node mobility.

II. NOMA DESIGN

NOMA aims to achieve efficient multicast communications through the employment of a logically centralized, globally distributed name resolution service associated with name based communications. In order to explain NOMA's design, we utilize a Global Name Resolution Service (GNRS) as a network-wide entity that provides an API for inserting and querying

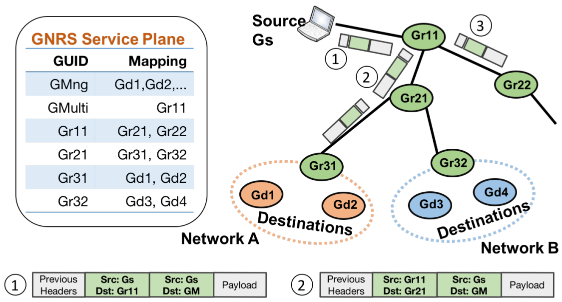

Fig. 3. Multicast architecture overview

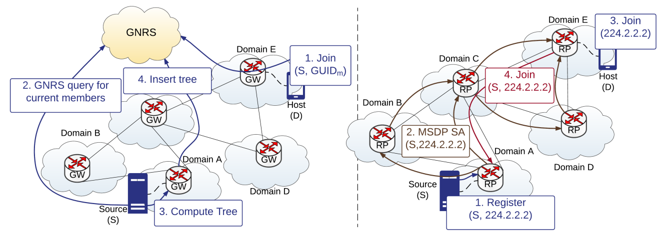

Fig. 4. Tree building steps comparison of NOMA with IP multicast

mappings between unique name identifiers and a set of values which can include network addresses, other name identifiers and related parameters – e.g. node properties, past locations and more. In spirit, this service is very similar to current Internet's DNS, which has already been effectively applied for new service functions such as load balancing and service replication. Even more interesting services can be realized with the next generation of global name resolution services such as DMap [11] and Auspice [12] introduced recently. The key advantage of using a name resolution service is to achieve a clean separation of network names from addresses.

NOMA's design, as proposed here, is based on MobilityFirst (MF [13]), which is a clean-sate network architecture for the next-generation mobile network where DMap [11] is used to provide resolution of names, that are Globally Unique Identifiers (GUIDs), into routable network addresses (NAs). Moreover, MF incorporates a hybrid name-address forwarding scheme, in which routing components use availability of both names and addresses in packet headers to perform forwarding decisions. Note that even though NOMA is based on MF, the same design concept can be applied to IP extensions (such as HIP [8]), overlay protocols (such as SCRIBE [6]), or clean-slate ICN protocols such as NDN [14] and XIA [15] through the use of a similarly designed name resolution service.

A. Multicast Tree Management

Multicast management consists of two core operations: membership of destination nodes and building and management of multicast trees. Both operations can be streamlined by exploiting the logically centralized, globally distributed, name resolution service (GNRS); in particular by using two forms of name indirection. A first unique name ($GMng$ in Fig. 3) is assigned to perform the task of node membership; all entities interested in receiving data from the multicast flow, can request to join by inserting their own unique name into the corresponding mapping in the table. This information is then exploited close to the source by a multicast service manager, which builds an efficient tree based on the available resources and the size of the required tree. Recursive mappings are then used to express the tree graph: by assigning to each branching router a name that exclusively identifies it in the context of the given multicast flow, we recursively follow the tree structure. For example, in Fig. 3, the root of this tree is identified by the multicast flow unique name mapping to the first branching router ($GMulti \rightarrow Gr11$); this router then maps to its children in the tree ($Gr11 \rightarrow \{Gr21, Gr22\}$); this continues until the leaves of the tree are reached, where we identify the leaves as the destination nodes. As time progresses and destinations join or leave the multicast group, the service manager can rebuild the tree information contained in the GNRS to trigger the required update.

One of the novelties of NOMA is that it can support push mode of multicast, where a source can send a single packet of multicast data, without the knowledge of the tree and this can happen even before the tree has been built. On receiving a multicast packet, for a group G_m, the gateway router at the source domain, acting as the multicast service manager, will do a membership query to the GNRS. GNRS supports recursive queries that return the host GUIDs along with the NAs of the domains they are currently connected to. Having the service manager on the gateway enables the tree computation to be topology-aware, as unicast path information of the NAs is available at the gateway, which is then used to build the tree. Once a tree is computed, it is updated in the GNRS such that downstream nodes do not need to recompute the tree again. This is quite different than distributed tree management techniques used in IP multicast since NOMA does not require flooding of multicast control messages (for example, source active (SA) or Join messages in PIM-SM and MSDP [4]) across domains, as shown in Fig. 4. The latter limits the scalability for traditional multicasting techniques to small to medium groups, as shown later in Sec. III. Also, using unique names to represent a group, members of the group as well as the multicast tree eliminates the need of a separate address allocation protocol, similar to MASC required for BGMP [5]. For evaluation purposes, we focused on two categories of multicast tree computation algorithms, i.e. shortest path trees (SPTs) and Steiner trees. A constraint of having centralized computation of trees is complexity and hence we opted for SPT and its modifications, even though our design is not limited to any specific algorithm.

B. Data Forwarding

Once the multicast tree is established, data forwarding can exploit the information contained in the GNRS to efficiently flow through edges between the nodes of the tree. In order to do so, we exploit address encapsulation, where two pieces of information are carried in data packets at the same time: Internally (i.e. second field in the green packets in Fig. 3), the encapsulated information carries the source and destination of the multicast flow, providing valuable information usable by all nodes along the path to easily identify data streams. Externally, routing information to perform hop-by-hop forwarding from one branching node to the next is placed. At each branching node participating in the multicast, forwarding decisions are performed by querying the GNRS to obtain information on how many next hops it has to forward to, generating required duplicates and replacing the external routing information with the new hop source and destination; this process is exemplified in the figure, where node $Gr21$ generates 2 duplicates for its two children, replacing headers accordingly. Intermediate nodes along the path forward encapsulated packets based on normal unicast rules. This reduces complexity of multicast packet processing to only a subset of nodes of the tree. To reduce the need of continuously involving the GNRS in the forwarding procedure, mappings can be cached at each hop, avoiding traffic and computational overhead. The tradeoff for this approach comes at the cost of slower reaction times to tree change events. More details on how to handle tree restructuring and end points mobility is provided in the following section.

C. Handling Mobility:

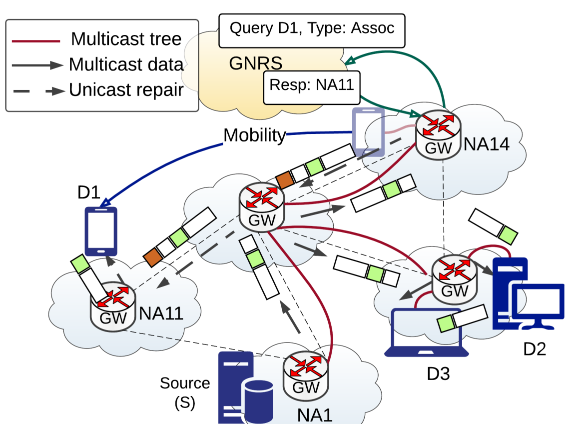

Fig. 5. Device mobility handling through unicast repair messages

End host mobility support has been a challenging problem in both unicast and multicast delivery. For the latter, the situation is further aggravated by the fact that an end-host mobility can significantly alter the multicast tree and hence its efficiency of delivery to other connected end-hosts. Without a clean separation of names and addresses, the onus of *re-booting* an ongoing *session* falls on to the mobile end-host. For an inter-domain multicast delivery, this means that every time an end-host moves and changes its point of association, it needs to send an explicit join at the new point of connectivity. The router at the new domain will then need to join the multicast tree, before the end-host can receive any data. Meanwhile, following a best-effort delivery policy, all the data received at the previous point of association will be lost.

NOMA on the other hand handles mobility by separating names from addresses and maintaining a name-based tree in the GNRS. At any point of the tree, failure in delivery to a downstream node results in temporary storage of data packets (MF routers are storage-capable [16]) and re-querying the GNRS for an up-to-date downstream node name (GUID) to its address (NA) mapping. This is specially relevant for the leaves of the tree which could be mobile end-hosts. As mentioned earlier for a long-lived flow tree, restructuring takes place periodically and any mobility that happens at a faster time-scale than tree re-computation will suffer. In order to ensure that end-hosts do not lose packets while moving, NOMA supports encapsulated 'repair' packets to be sent to the moving client. This again is enabled by the GNRS that maintains the up-to-date location (end-host GUID to NA mapping) as it moves. As shown in Fig. 5, when a end-host $D1$ moves from $NA14$ to $NA11$, which is not part of the multicast distribution tree, the tree does not change immediately. However, failure to deliver at the edge, causes the gateway router at $NA14$ to query the GNRS for up-to-date mapping of $D1$. Following association at $NA11$, the gateway at $NA14$ can encapsulate the pending data and send it as unicast repair to $NA11$ as shown. In contrast to multicasting, the repair procedure is transparent to an end-host or application and does not require explicit re-joining from the client side. However this is only a short-term mechanism to counter moderate mobility of a subset of destinations. With increase in the number of devices and mobility, the frequency of tree updates should increase proportionally.

III. Evaluation

In this section we present detailed performance evaluation based on a combination of large scale analytical modeling and fine-grained packet-level simulation on network simulator (NS3).

A. Tree Generation Algorithms

NOMA provides a framework for managing and deploying multicast communications, independently from the tree generation algorithm employed. While this is a valuable feature of the design, it is necessary to study different algorithms and heuristics in the context of choosing one that can effectively utilize unicast routes, and is lightweight enough to be able to run at a single router. We looked at two main categories of algorithms for building multicast trees, namely shortest path trees (SPTs) and Steiner trees. Although Steiner trees provide an optimal solution in terms of overall network resource utilization, they are NP-hard to compute. Several Steiner heuristics have been proposed over the years to provide near-optimal solutions [17], with relatively high computation cost. However, computational complexity is a key constraint for our design, since the tree computation is *centralized*. We instead opt for the SPT algorithm that uses inter-domain unicast route information and require no further computation, but is less efficient compared with a Steiner tree. In SPT, packets are forwarded along the *longest-common path (LCP)* to all the destinations, as single copy, until the branching point is reached, where the packet is copied and delivered towards multiple destinations. This allows all destinations to receive

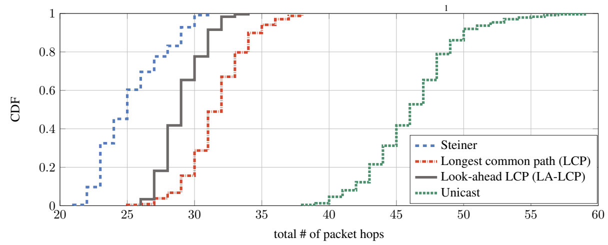

Fig. 6. CDF of performance in terms of packet hops for different multicast tree generation algorithms, for 100 node random graph with 20 randomly chosen destination nodes.

multicast packets across the shortest path from the source. We also analyzed other heuristics that aimed to further minimize the overall network traffic with moderate computation. One of these heuristics is the *look-ahead longest-common path (LA-LCP)* algorithm. Unlike LCP, which branches whenever there a divergence of shortest paths to multiple destinations, LA-LCP, compares the overall network cost of branching from the current node and branching from each of the possible next hops, and decides to branch downstream if the cost is lower for the latter, thereby deviating from the SPT. This reduces the overall packet hops in the network, with slight increase in computation complexity.

Fig. 6 plots the CDF of total packet hops to reach 20 randomly placed destinations from a single source on a 100 node *Erdős-Rényi* random graph for each of these algorithms. As seen from the plot, all the multicast algorithms are much more efficient than unicast. Although Steiner provides the most efficient trees, it is computationally intensive. In comparison, LA-LCP provides *reasonable* performance with lower overall network overhead compared to traditional longest common path.

B. Comparison to IP multicast

In this section we compare pull-based multicast of NOMA with IP based inter-domain multicast, namely, PIM-SM standard coupled with MSDP [4]. Through the results we highlight two key benefits of using NOMA, namely, 1) lower control overhead for maintaining a multicast group, and 2) better handling of mobility for data forwarding. Note that BGMP [5] is another prominent inter-domain IP multicast scheme, however, it is not well-suited for applications that involve dynamism and fast changes in the tree, and hence has not been a focus of our evaluation. BGMP allows multicast route updates to be carried along with inter-domain BGP messages and therefore tree changes occur at a much slower time-scale than PIM-SM/MSDP (typical BGP updates take about 100 seconds to propagate throughout the network [18]).

• **Control overhead:** The advantage of using unicast routes to build the tree is that no multicast specific control overhead needs to be exchanged across networks. This is crucial for inter-domain settings where flooding periodic multicast tree update messages is not tractable. In Fig. 7 we plot the multicast specific messages exchanged for setting up a tree and forwarding packets for increasing graph sizes, with the topology being an *Erdős-Rényi* random graph, and 50% of

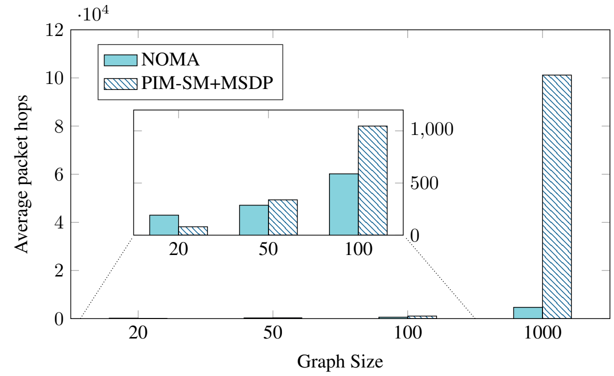

Fig. 7. Control packet overhead for tree setup for varying graph sizes

the nodes being randomly chosen to have destination clients part of the multicast group. For NOMA this includes 1) the GNRS insert messages from each of the destination networks for joining a particular multicast group, 2) the GNRS insert from the gateway at the source domain to insert the generated multicast tree, and, 3) GNRS query and responses during data forwarding at the branching nodes. The GNRS is implemented as a distributed hashmap, following the DMap design [11], with the same mapping stored at multiple locations. For evaluation purposes, 3 GNRS instances were maintained, therefore each insert incurred 3 unicast messages to 3 specific nodes (determined by a hash function), whereas each query was *anycasted* to the nearest of the 3. In comparison, for PIM-SM+MSDP the overhead numbers comprise of, 1) the flooding of Source-Active (SA) messages from the source domain throughout the network, and, 2) the Join messages from the domains which have destinations nodes interested in receiving packet from that particular source. As seen from the plot, maintaining a multicast tree in the GNRS has higher overhead for smaller sized graphs (for example, for a 20 node topology, shown in the zoomed in section of Fig. 7), but it scales elegantly with size. Using PIM-SM+MSDP, on the other hand, becomes intractable as the number of nodes increases. With more than 40 thousand ASes in the Internet today, if every domain was multicast enabled, the cost becomes too high to maintain a distributed tree. Similar trends were observed by varying percentage of destination networks for fixed graph sizes and is not included here for brevity.

• **Handling mobility:** NOMA seamlessly handles client mobility and the dynamism in tree-changes thereof, by periodically recomputing the tree and updating the corresponding GNRS entries. In addition, to counter packet-loss due to mobility, NOMA supports unicast 'repair' packets to be sent from a previous edge node to the current point of attachment of a mobile client, until a tree update restructures the tree. We performed detailed packet level simulations in network-simulator (ns-3) on a 20 domain random topology with randomly placed mobile and static clients, for both NOMA and an IP multicast implementation of PIM-SM + MSDP. Fig. 8 plots the fluctuation in received throughput at a client receiving a multicast stream of 2Mbps on the event of mobility. A mobility event is characterized by disconnection of a client from its

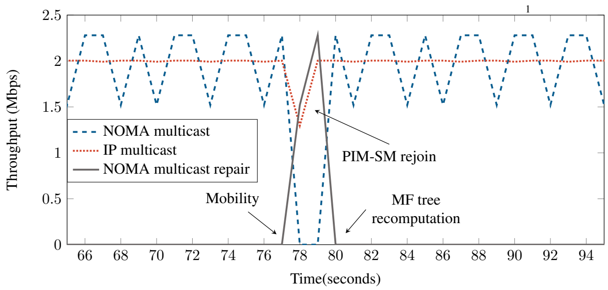

Fig. 8. Comparison of average multicast throughput received at a client with mobility

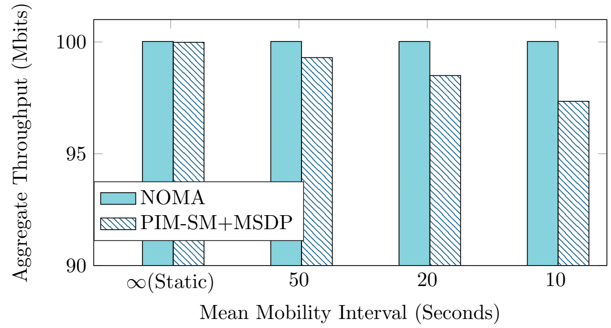

Fig. 9. Aggregate throughput at a mobile client, with increasing mean mobility rates; mobility event is determined by an exponential random variable with the mean

attachment point and re-association to another node, following a period of association (uniform random variable $U(0,1)$ seconds), as highlighted in the figure at $t =\sim 77$ seconds. NOMA periodically restructures the multicast tree every 10 seconds for this scenario, whereas, IP multicast restructures following the client explicitly joining the tree at the new point of association. Therefore, multicast traffic for NOMA falls to 0, until tree is restructured at $t = 80$ seconds. However, repair packets are delivered to counter packet loss and reordering, highlighted by the black trajectory in the figure. Note that NOMA is based on MobilityFirst (MF) transport, that uses reliable hop-by-hop delivery of large chunks, and the throughput received by the client is therefore in steps with the average being 2Mbps. In comparison, for IP multicast, data throughput falls following temporary disconnection and re-connection, as shown by the red dotted trajectory.

Mobility not only affects the instantaneous throughput at a client, it also leads to loss of packets during the interval of disconnection, re-association of the client, re-joining and re-structuring of the multicast tree. Additionally, in a practical setting, for IP multicast, the mobile client will spend a significant amount of time for new IP address allocation through DHCP, which has not been taken into account for this evaluation. This packet loss and reduction in overall throughput is highlighted in Fig. 9 where we plot the aggregate throughput at a mobile client for increasing rates of mobility, that moves randomly with exponential random mean mobility interval of 50, 20 and 10 seconds. As seen from the plot, aggregate throughput for NOMA does not change with mobility, primarily due to native features of MF such as hop-by-hop reliable delivery and storage-capable routers to handle temporary disconnections. In comparison, IP multicast throughput significantly worsens with increasing mobility speeds.

C. Prototype Description

To validate the implementation feasibility of NOMA, we built a Click software router [19] prototype and tested it on a small scale topology on the ORBIT testbed [20]. Fig. 10 highlights the key router and multicast host components that were built for the prototype. In addition, existing DMap based GNRS APIs were modified to allow multicast tree insertion and queries. Ongoing work includes detailed evaluation on larger topologies to validate scalability of NOMA in realistic

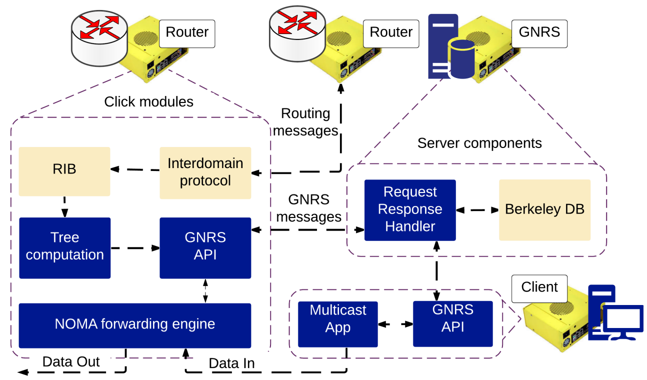

Fig. 10. Components of the NOMA router prototype, GNRS and client implementation, with developed modules shown in blue

scenarios.

IV. Related Work

Work on multicast started in the early 1990's when Deering proposed DVMRP [21]. This was followed by extensions to open shortest path routing (OSPF) called MOSPF [2]. However, most of the early work on multicast concentrated on flood-and-prune based techniques that were difficult to scale. Core-based trees [22] and PIM-SM [1] were thus proposed that introduced shared trees and rendezvous points (RPs). However, even these were mostly deployed within a single domain. Challenges in choosing appropriate RPs for inter-domain settings, led to the multicast source discovery protocol (MSDP) that allowed RP's in different networks to coordinate [4]. MSDP suffered from scaling issues that limited its deployability. BGMP is the multicast extension for BGP for inter-domain routing [23], but requires its own address assignment and address allocation protocols for multicast group management [5]. In this context, overlay solutions such as SCRIBE [6] or ZIGZAG [7] offer better scalability. However, such application layer solutions overload end-hosts with multicast tree computation and active management and also result in higher network traffic load. Clean slate network layer solutions have also been proposed, notably COPSS [24] for content delivery. However, the focus of COPSS is on pub/sub based systems and hence allows pull based multicast. In comparison, NOMA allows both push and pull mechanisms to support a wide variety of applications.

V. Conclusion

In this paper, we have proposed a name based inter-domain multicast approach, leveraging on a distributed name resolution service for membership and tree management. The proposed NOMA framework scales reasonably well to medium to large scale trees and handles client mobility with disconnections. Large scale analytical results for management overhead and fine-grained packet-level simulations for mobility scenarios were provided. In addition, we presented a proof-of-concept prototype with small-scale experiments as feasibility studies. Future work includes further feasibility studies and deploying NOMA on the GENI large scale testbed to evaluate performance in more realistic inter-domain network scenarios.

References

[1] Dino Farinacci, C Liu, S Deering, D Estrin, M Handley, Van Jacobson, L Wei, Puneet Sharma, David Thaler, and A Helmy. Protocol independent multicast-sparse mode (pim-sm): Protocol specification. 1998.

[2] John Moy. Rfc 1584–multicast extensions to ospf. *SRI Network Information Center*, 1994.

[3] Christophe Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, and Doug Balensiefen. Deployment issues for the ip multicast service and architecture. *Network, IEEE*, 14(1):78–88, 2000.

[4] David Meyer and Bill Fenner. Multicast source discovery protocol (msdp). 2003.

[5] P Radoslavov, D Estrin, R Govindan, M Handley, S Kumar, and D Thaler. The multicast address-set claim (masc) protocol, rfc-2909. Technical report, 2000.

[6] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, and Antony I.T. Rowstron. SCRIBE: A Large-Scale and Decentralized Application-Level Multicast Infrastructure. *JSAC*, pages 1489–1499, 2002.

[7] Duc A Tran, Kien Hua, and Thanh Do. ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming. In *INFOCOM*, 2003.

[8] Robert Moskowitz, Pekka Nikander, Petri Jokela, and Thomas Henderson. Rfc 5201–host identity protocol. 2008.

[9] Jianli Pan, Raj Jain, Subharthi Paul, and Chakchai So-In. Milsa: A new evolutionary architecture for scalability, mobility, and multihoming in the future internet. *Selected Areas in Communications, IEEE Journal on*, 28(8):1344–1362, 2010.

[10] Ivan Seskar, Kiran Nagaraja, Sam Nelson, and Dipankar Raychaudhuri. MobilityFirst Future Internet Architecture Project. In *MobilityFirst Project, Proc. ACM AINTec 2011*.

[11] Tam Vu, Akash Baid, Yanyong Zhang, Thu D Nguyen, Junichiro Fukuyama, Richard P Martin, and Dipankar Raychaudhuri. Dmap: A shared hosting scheme for dynamic identifier to locator mappings in the global internet. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 698–707. IEEE, 2012.

[12] Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, and Aditya Yadav. A global name service for a highly mobile internetwork. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 247–258. ACM, 2014.

[13] MobilityFirst Future Internet Architecture Project. http://mobilityfirst.winlab.rutgers.edu/.

[14] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking Named Content. In *CoNEXT*, 2009.

[15] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David G Andersen, John W Byers, et al. XIA: An Architecture for an Evolvable and Trustworthy Internet. In *HotNets*, 2011.

[16] Samuel C Nelson, Gautam Bhanage, and Dipankar Raychaudhuri. Gstar: generalized storage-aware routing for mobilityfirst in the future mobile internet. In *Proceedings of the sixth international workshop on MobiArch*, pages 19–24. ACM, 2011.

[17] Kadaba Bharath-Kumar and Jeffrey M Jaffe. Routing to multiple destinations in computer networks. *Communications, IEEE Transactions on*, 1983.

[18] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. *ACM SIGCOMM Computer Communication Review*, 30(4):175–187, 2000.

[19] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M Frans Kaashoek. The click modular router. *ACM Transactions on Computer Systems (TOCS)*, 18(3):263–297, 2000.

[20] Dipankar Raychaudhuri, Ivan Seskar, Max Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremo, Robert Siracusa, Hang Liu, and Manpreet Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1664–1669. IEEE, 2005.

[21] Stephen E Deering and David R Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems (TOCS)*, 8(2):85–110, 1990.

[22] A Ballaradie, J Crowcroft, and P Francis. Core based tree (cbt)–an architecture for scalable inter-domain routing protocol. In *Proceedings of the 1993 ACM SIGCOM*, pages 85–89, 1993.

[23] Satish Kumar, Pavlin Radoslavov, David Thaler, Cengiz Alaettinoğlu, Deborah Estrin, and Mark Handley. The masc/bgmp architecture for inter-domain multicast routing. *ACM SIGCOMM Computer Communication Review*, 28(4):93–104, 1998.

[24] Jiachen Chen, Mayutan Arumaithurai, Lei Jiao, Xiaoming Fu, and K. K. Ramakrishnan. COPSS: An Efficient Content Oriented Pub/Sub System. In *ANCS*, 2011.