The Named-Object Abstraction for Realizing Advanced Mobility Services in the Future Internet*

Francesco Bronzino†
Inria
Paris, France
francesco.bronzino@inria.fr

Shreyasee Mukherjee
WINLAB, Rutgers University
North Brunswick, NJ, USA
shreya@winlab.rutgers.edu

Dipankar Raychaudhuri
WINLAB, Rutgers University
North Brunswick, NJ, USA
ray@winlab.rutgers.edu

ABSTRACT

This paper presents a new abstraction called "Named-Objects" for enabling flexible and advanced mobility services in the future Internet. The concept of named-objects falls under the broad category of "information centric networks (ICN)" and is based on the assignment of a globally unique identifier to all Internet attached objects while separating this "name" from the routable "address" or "locator". This approach is supported by a global name resolution service (GNRS) which dynamically maps names to addresses while also providing supplementary service information where desired. The named-object abstraction is outlined and exemplary mobility related services including device mobility, multihoming and multicast are discussed.

A qualitative and quantitative evaluation of the named-object architecture is given relative to alternative ICN designs such as Content Centric Networking (CCN) as well as name-based protocols evolved from IP, i.e. HIP and LISP, showing superior performance for a wide range of mobility services and the potential to serve as a foundation for future mobile network protocols.

CCS CONCEPTS

• **Networks** → **Naming and addressing**; **Network mobility**; **Naming and addressing**; **Mobile networks**;

KEYWORDS

Future Internet Architecture, Mobility Services, Networking Abstractions, Name Base Networking

ACM Reference format:
Francesco Bronzino, Shreyasee Mukherjee, and Dipankar Raychaudhuri. 2017. The Named-Object Abstraction for Realizing Advanced Mobility Services in the Future Internet. In *Proceedings of MobiArch '17, Los Angeles, CA, USA, August 25, 2017,* 6 pages.
https://doi.org/10.1145/3097620.3097627

*Research supported under NSF Future Internet Architecture - Next Phase (FIA-NP) grant CNS-1345295
†Work done while at Rutgers University

MobiArch '17, August 25, 2017, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5059-4/17/08…$15.00
https://doi.org/10.1145/3097620.3097627

1 INTRODUCTION

The basic abstraction for communicating over the Internet has always been very simple: *send* a message from an interface to a destination address and if an interface is listening for data on that address it will *receive* the message at the other end. From an end-point perspective, this simple concept creates the perception of transmitting data on top of a virtual link connecting two interfaces. This is the base of the Berkeley IP socket layer, the most commonly used network interface. But after many years of reliance on this simple IP service model, the Internet has gradually shifted to a more complex ecosystem, where sophisticated platforms, applications, and services are poised to replace the fixed-host/server model that has dominated the Internet since its inception. While developers have still managed to create advanced services above the IP network layer, the reliance on this core interface forced them to develop ad-hoc and sometimes patched solutions to address even the most common service scenarios. Acknowledging the markedly different population of Internet devices and services, the research community has looked over the years at the possibility of defining new communication abstractions, to address the limitations of the IP/TCP stack as it exists today and to provide solutions to problems such as mobility and multi-point communications in a more integrated and cohesive manner.

As an evolutionary step, the Host Identity Protocol (HIP) [16] has proposed to introduce a naming layer through the use of a shim layer sitting between the classical transport protocols - i.e. UDP/TCP - and the IP network layer. With the goal of exclusively requiring modifications in the end host network stack, HIP provides new tools and functions for future network needs, from supporting seamless host mobility and multi-homing, to the ability to securely identify previously unknown hosts and the ability to securely delegate signaling rights between hosts and from hosts to other nodes. In order to do so, HIP introduces a new namespace made of Host Identifiers, that double as public cryptographic keys. Similar in spirit, SERVAL [20] implements a new Service Access Layer (SAL) that sits above an unmodified network layer, enabling applications to communicate directly on service names aimed at supporting Internet services located at multiple and different locations, while serving clients that are often mobile and multi-homed. For flexibility, uses a large namespace based on 256-bit, while acknowledging that shorter solutions could also be considered.

Using a different namespace to separate names from addresses of the routing layer through the use of a name to address mapping service has also been advocated as a structural change within the network infrastructure itself. For example, the Locator/Identifier Separation Protocol (LISP) [8] proposed this separation through decoupling two types of addresses: EIDs that identify hosts, and RLOCs that identify network attachment points and are used as routing locators, both of which are IP addresses. LISP, with such separation in place and with the use of a name mapping system, is able to offer native mobility through an extension of its base protocol called LISP-MN [26].

A third clean slate approach is that of Content Centric Networks (CCN) [10] or Named Data Networks (NDN) [28], both belonging to the class of Information Centric Networks (ICN). These architectures depart from the conventional point-to-point abstraction by having routers in the network directly operate on content labels making physical network addresses unnecessary. The Data-Oriented Network Architecture (DONA) [12], was perhaps the first comprehensive and detailed proposed architecture that relied on the use of self-certifying names. CCN and its derivative designs [15, 28] evolved the concept and brought it to prominence within the networking community. The CCN architecture through an elegant mechanism shifts the networking paradigm from today's IP locators - *where* - to content descriptors - *what*. This paradigm shift not only enables efficient delivery of content, but also enables advanced services such as mobility and multi-homing which are relatively difficult to support in todays IP networks.

While all these solutions were driven by one or more use cases, none of them ever reached enough maturity to fully replace the original virtual link interface, opening the space for the design of a more compelling abstraction, capable of providing the benefits of ICN techniques while maintaining a higher degree of generality for service creation: the *Named-Object* abstraction. Similar to LISP, the named-object abstraction is created by separating names and addresses allowing for dynamic resolution of destination locations. On top of this, through specification of service intents, named-objects enable the network elements to adapt to the requirements of different services. This distinct feature enables the network participants to easily support different communications modes, spanning from the base virtual link, to seamless mobility and multi-group delivery, to even more advanced scenarios such as content retrieval.

The goal of this paper is to formally present the named-object abstraction (Sec. 2) and provide a qualitative (Sec. 3) and analytical (Sec. 4) evaluation of its merits on fundamental cases, such as mobility, multicast and multihoming, comparing it against three other solution proposals: HIP, LISP and CCN.

2 THE NAMED-OBJECT ABSTRACTION

Named-objects are a new abstraction meant to represent any network entity that could be abstracted as an addressable network element. This should cover any possible abstraction: from the original host based abstraction of a virtual link bridging two interfaces, to recently introduced ones such as contents, to any potential future abstraction - e.g. context. While name based approaches have already been addressed in the past, they were mostly focused on either solving specific issues such as mobility [8] or security [16] or to shift the communication focus to new entities such as contents [10, 12]. Named-objects aim to bring a more comprehensive solution that can enable powerful abstractions and services to underpin the Internet architecture.

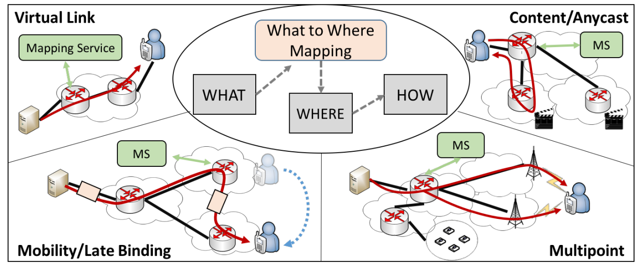


Figure 1: The Named-Object abstraction applied to different use cases.

Fig. 1 outlines the approach in defining the named-object abstraction through separation of names and addresses. Separating names (identities) from addresses has been advocated by the research community [8, 16, 21] for quite some time and has inherent benefits in handling mobility and dynamism for one-to-one communication. If properly employed, names can also provide additional advantages to facilitate the creation of new service abstractions that can be used to support advanced applications. The named-object approach involves three steps: First, *"what"* (or *"who"*) will take part in the communication has to be identified through a unique name that is understandable by all parties involved, e.g. end points, routing elements. When forwarding is required, names are then resolved to *"where"* they are located. While this could be applied at different locations of the network and in the network stack - e.g. having the separation at the end points, previous proposals [8, 24, 25] demonstrated that the use of a globally accessible name resolution service is a suitable approach for this goal, scaling to globally support the size of the namespace while supporting the dynamicity of hybrid routing schemes (i.e. less than 100ms for 95th percentile of lookup operations). Finally, if the semantical value of such element is known, it can be indicated through the use of a service identifier properly located in a packet header, giving an indication of *"how"* such packet should be treated.

2.1 MobilityFirst: A Named-Object Based Architecture

The MobilityFirst (MF) architecture [23] is an example of how the named-object abstraction could be integrated into an Internet network design. At the core of the architecture is a new name-based service layer which serves as the "narrow waist" of the protocol stack. The name-based service layer uses flat Globally Unique Identifiers (GUIDs) of 160 bits to identify all principals or network-attached objects. Names are resolved through a Global Name Resolution Service (GNRS) that provides APIs to insert and query for <*key,value*> mappings and support hybrid schemes that exploit availability of both names and addresses in the network header for dynamic resolution of destination locations [24, 25]. A Service Identifier (SID) flag placed in network header allows network components to be aware of different service types in order to apply different forwarding modes - e.g. multicast [18] and multi network aggregation [11, 17]. Finally, a new name based API [4] designed to offer network primitives for basic messaging (*send, recv*) and content operations (*get,*

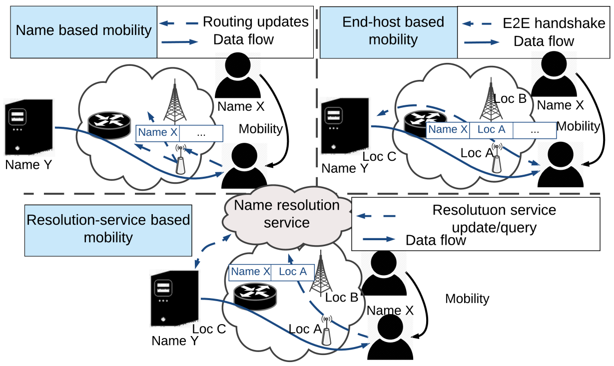


Figure 2: Mobility management techniques: pure name based, end-host based and NRS based

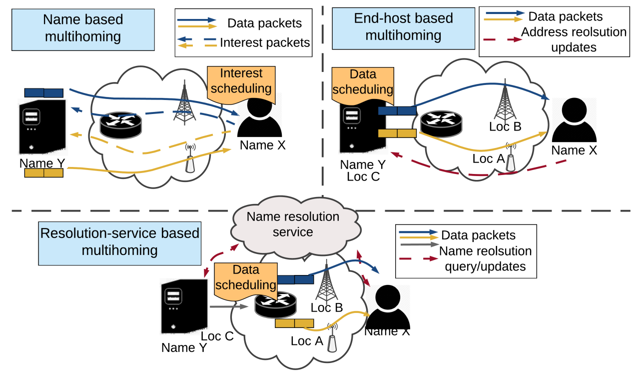


Figure 3: Multihoming techniques: pure name based, end-host based and resolution-service based

post) allow several delivery modes to be innately supported by the network, such as multihoming, multicast and anycast.

3 COMPARISON OF ARCHITECTURES

Two key elements define the named-object abstraction: the name to location separation and the ability to identify the requested service for in transit data. To understand the benefits of the abstraction in supporting advanced mobility services, this section provides a qualitative analysis of the named-object based architecture, MobilityFirst against three classes of alternate approaches: one that employs *no resolution* via pure name based routing (using CCN as an example) and two that provide the name/location separation via *end-host based resolution* (HIP) or using a *Name Resolution Service* (LISP), but that do not incorporate service identification. While for certain services the name/location separation is sufficient and no differentiation will be given between the LISP and MF, more advanced scenarios will show how the second component can provide additional value supporting the case for named-objects. The comparison focuses on three key services: mobility (Fig. 2), multihoming support (Fig. 3) and multicast delivery (Fig. 4).

3.1 Handling Mobility

Pure name based routing architectures require no resolution of names to addresses since routing is also based on names. While this is an interesting paradigm, having an immutable name with no location information whatsoever implies every time a device moves, routing updates need to be flooded, such that the rest of the Internet can build a new route to this name. For architectures with hierarchical names, such as CCN [10], there can be some control on the flood based on name aggregation. However, with networks becoming smaller and denser and entities becoming more mobile, frequent mobility with handoffs is becoming a norm. As shown in [2], with 10 billion mobile devices worldwide, issuing routing updates for every single mobility event is not scalable.

End-to-end mobility management techniques name devices with permanent identifiers but route based on locators, which change at a much slower time scale than host-mobility. Therefore, for every mobility event, there is an end-to-end messaging/handshaking between the mobile end-points to notify their routing locator. This works for scenarios where one of the end-points is relatively static; if *Name X* knows the locator of *Name Y* (in this case *Loc C*) apriori, it can initiate the hand-shaking mechanism. However, if both the end-points are moving simultaneously, there needs to be an additional service or a rendezvous server, which has a fixed known locator. The end-devices need to communicate with this server which then brokers the handshaking between the two [13].

Instead of making end-points responsible for mobility management, the Name Resolution Service (NRS) and named-object based approaches deploy a distributed infrastructure, that maps names of objects to their current locators [23, 26]. As mobile devices move, they update the resolution system with their up-to-date locator and any other device can query the same service to obtain the current locators. This resolution service can be implemented as an overlay system [9] or be a native implementation [25]. The key advantages of having a distributed resolution system are: (i) No explosion of routing updates for individual mobility events, and (ii) higher resilience against failure and fault tolerance compared to a rendezvous server. In addition, distributing name-to-address resolution servers across the Internet enables additional optimizations on replica placements, work-load balance, spatial and temporal locality, etc. thereby enabling faster updates and queries [24]. As shown later in Sec. 4, having a distributed resolution service also incur low control overhead compared to the other techniques and can support low query latencies for highly mobile vehicular scenarios.

3.2 Enabling Device Multihoming

Device multihoming goes one step further wherein, an entity can communicate using a dynamic set of multiple interfaces; a single *who* then maps to multiple *where*(s). This is problematic for architectures with no name to address(es) decoupling, since the name is associated to an entity and not its interfaces. A naive way of multihoming is therefore, to send routing updates across all the interfaces, such that data packets can be received on all of them. For CCN, which works on polling, this means sending out redundant interests across all available interfaces and receiving the same data across all, causing a wastage of network resources. The authors in [5] propose having a scheduler on the end-host that can split

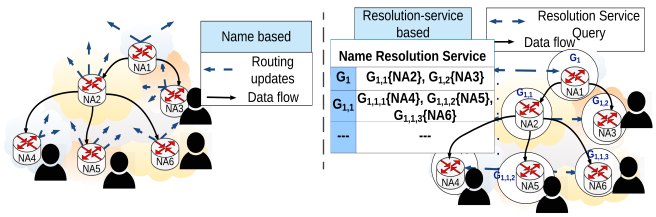
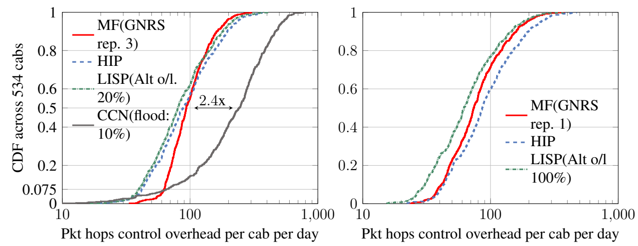


Figure 4: NRS multicast tree overloading compared to name based polling approach.



(a) (b)

Figure 5: Control overhead comparison for (a) maintaining mobility and, (b) using a single replica NRS

interest packets on different interfaces thereby receiving different data packets on the reverse path (Fig. 3). While this helps in improving overall throughput at the client, it still suffers from a control overhead explosion on mobility scenarios, in which case interest messages need to be resent every time one or more of these interfaces change point of attachment. Having a scheduler on the end-device also requires additional sophisticated machinery to implicitly measure or estimate the access link quality and congestion characteristics in order to efficiently schedule interest packets, which may not be viable for a resource and power-constrained end-device.

End-host based multihoming techniques follow similar principles as mobility, namely, end-points communicating directly to exchange name to multiple locator's information. This means a scheduler should run at the other end-point to split the flow for each of the addresses of the interfaces. Similar to the mobility scenario, a rendezvous server is necessary for initiating the flow when both the end-point's locators are changing [13]. Likewise, NRS based architectures update entity names to multiple addresses in the resolution service and query as and when these locators are updated. Moreover, through the service intent specification, the in-network components can be aware of the end-user's desired multihoming mode. Data schedulers can then be located within the network at a suitable bifurcation point, which enables fine-grained control based on accurate link and bandwidth characteristics, leading to significant improvements in overall end-host throughput [11, 17].

3.3 Support for Large Multicast Groups

Pure name based architectures inherently support multicast by, (i) having receivers flood interest packets, and, (ii) routers serving multiple pending interests simultaneously across the reverse path of interest. However, flooding of interest for every multicast request across the internet does not scale. Only the introduction of rendezvous points [6], similar in spirit to IP multicast, and new packet formats can reduce such impact.

While name based architectures can still fall back on IP multicast techniques in order to support multicast [7, 29] and exploit names only to enhance the existing protocols (e.g. HIP security handshake across peers [29]), a more efficient technique is to instead the use of name abstractions (i.e. named-object) to overload multicast trees into the NRS thereby reducing the overall control overhead [18]. In this case, each branching router along the tree is assigned a unique name specific to that tree, which recursively map to their children or downstream branching routers (Fig. 4). This name based tree is stored in the name resolution service for ease of management, allowing branching routers to query and forward packets along the tree. Since name-address decoupling already handles mobility efficiently, this scheme can also handle receiver mobility, ensuring no packet is lost during multicast receiver mobility.

4 EVALUATION

In order to characterize the effectiveness of the named-object abstraction compared to LISP, HIP and CCN, packet-level simulations using both NS-3 and a custom simulator have been performed. These focus on both control overhead and data throughput for the three use cases described earlier, namely: device mobility, multihoming and multicasting.

4.1 Device Mobility Support

Topology Generation: In order to simulate realistic mobility patterns, a dataset of 526 San Francisco cab traces has been used [22]. WiFi access points (AP) are mapped to locations using the WiGLE database [27] and a WiFi association model has been developed for each of these cabs, assuming at every location a cab would connect to the geographically closest AP. However, radio handover does not necessarily mean a change in the network address for the cab, since in many cases internet service providers (ISPs) handle hand-offs transparent to the user, by keeping the routable address the same. Therefore, in order to translate radio handovers to locator changes, geographical locations of point of presence (PoPs) of autonomous systems (ASes) are mapped from Caida [1] onto this topology and each AP is assigned to a PoP based on geographical proximity. While this may not be a perfect representative of the actual network topology, due to the lack of publicly available data for network mobility, this approach can be considered relevant for emulation of realistic mobility scenarios. Finally, this PoP topology is correlated with global inter-domain topology from Caida to obtain a network topology of 21,743 ASes and 735,249 links which is used in our custom simulator to analyze global control overhead to support vehicular mobility.

Update and lookup overhead: Given this topology, an analysis of the global control overhead in maintaining a route to each of these cabs in all the four name based architectures is provided. The data retrieval model follows a web-access scenario with random choice of networks as sources (flow duration and inter-arrival time based on [3]), for the entire duration of each of the traces. Fig. 5(a)

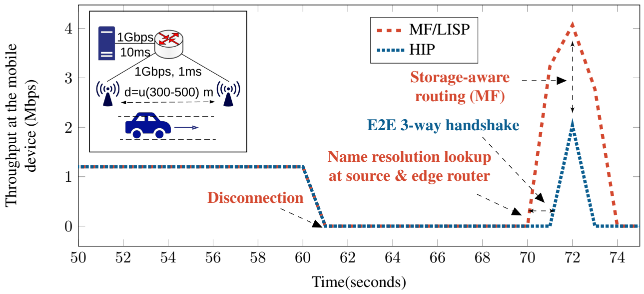


Figure 6: Data delivery to a moving vehicle while it disassociates and re-associates with WiFi APs

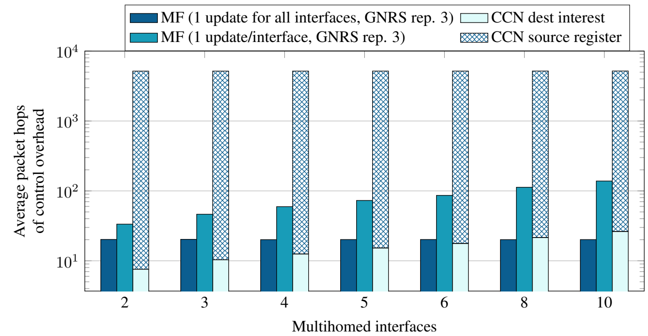


Figure 7: Control overhead for multihoming support with increasing number of interfaces in a 1000 network random graph

plots the cumulative distribution of packet hops of overhead per cab per day. As seen from the plot, control overheads for MF, HIP and LISP are comparable. In comparison median overhead of CCN is about 2.4 times higher than the others. This is because in CCN, every time a device moves, it needs to propagate a routing update to enable other networks to route packets to itself. However due to the hierarchical nature of CCN names, not every mobility event causes an update. For example, if a device named */att/rutgers/alice/* moves from the domain *rutgers* and connects directly to *att*, the latter does not need to propagate an update. For this simulation, it is assumed that every network forwards only 10% of the total routing updates it receives to its neighbors. As seen from the plot less than 8% of the events result in very low routing updates due to hierarchical naming, but on an average, pure name based routing has a much higher overhead.

Although MF, HIP and LISP have comparable overheads, it is worth mentioning that HIP performs an end-to-end handshaking, whereas MF and LISP both update their respective resolution services, for every mobility event. The resolution service for MF (GNRS) is in-network with 3 replicas stored for each name. LISP, on the other hand, uses an overlay based scheme and has an inherent overhead of maintaining overlay topology using BGP. Fig. 5(a) does not take into account BGP and considers 20% of the networks randomly chosen to have an overlay server.

Increasing the number of ALT servers (with corresponding increase in topology maintenance overhead) will further improve LISP control overhead with the minimum being for all networks participating in ALT (similar to the GNRS implementation), as highlighted in Fig. 5(b). On the other hand, reducing the number of GNRS replicas to 1, will reduce the control overhead for MF, but also corresponding reduce its resiliency to failures. As shown in Fig. 5(b), MF overhead would lie somewhere in between LISP and HIP in such cases.

Handover and data throughput: Given that MF, LISP and HIP have much less overhead in managing mobility, a comparison of the three is provided for data throughput during mobility and temporary disconnection with handovers, using packet level simulation in NS-3. In this experiment, a single device moves away from its associated AP along a straight line, and following a period of disconnection connects to another, while downloading a large file form a back-end server. Core link delays are set to 10 ms whereas edge links have 1 ms delay, as shown in the top-left of Fig. 6. This topology although simple, in essence simulates a similar cab mobility trace but with accurate control of AP placements and mobility. MF/LISP evaluation considers a single replica resolution server which all the routers can update and lookup. MF routers are also storage capable [19], so that every router can temporarily store data on disconnection and re-route once an up-to-date mapping is available.

As highlighted in Fig. 6, following disconnection at about 61 seconds, data throughput goes to zero. For MF/LISP both the AP it was last connected to and the source periodically re-queries the server for new address mapping of the device. As soon as the server is updated at about 70 seconds, data delivery resumes for the device. For HIP however, end-to-end 3-way handshaking takes longer to complete and data delivery resumes around 71 seconds. Finally, Fig. 6 also shows the benefit of storage-aware routing in case of MF, wherein the previously associated AP temporarily stores in-flight data and binds them to the new address following resolution server update and re-routes it along the edge.

4.2 Multihoming support

Next control overhead is evaluated for the maintenance of multiple interfaces at a client with data delivery across all interfaces. The graph employed is an *Erdős-Rényi* 1000 node random graph with a randomly chosen source network and variable number of randomly chosen interfaces for the destination. Comparison is done only between resolution-service based approach of MF with pure name based approach of CCN, since comparative studies of in-network multihoming against end-to-end approaches such as that utilized in HIP have already been presented in detail in our earlier works [11, 17]. As shown in Fig. 7, there is no increase in overhead as the number of interfaces increase, since availability of all the interfaces can be populated in the resolution service via a single unicast message per GNRS replica through any one of the attached interfaces. However, even if the interfaces boot up one at a time, a sequential update through each of the interfaces also scales gracefully with increasing number of interfaces, as shown. In comparison, CCN based approaches pay a high cost in register messages flooded from the source to all the available networks with potential receivers. In addition, a receiver also has to send out interest packets through each of its available interfaces which flow in the reverse path of the source register message.

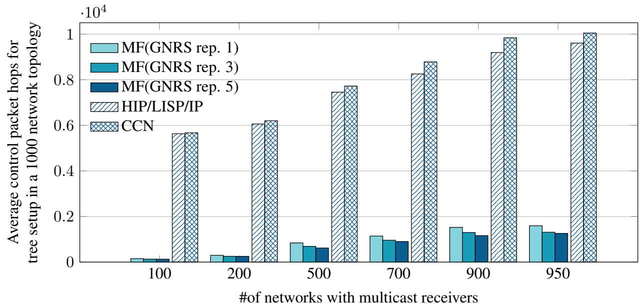


Figure 8: Control overhead for multicast tree maintenance in a 1000 network random graph (single source and multiple receivers)

4.3 Multicast Support

Control overhead for maintaining inter-domain multicast trees has also been analyzed for the different name based architectures, using the same 1000 node graph, a randomly chosen source network and variable number of destination networks. As mentioned earlier, HIP and LISP both utilize IP multicast for tree maintenance, therefore, inter-domain PIM-SM has been evaluated as a representative of these two architectures. For MF, since the tree is maintained in a distributed fashion in the GNRS, control overhead is affected by the number of replicas maintained. Interestingly, increasing the number of replicas leads to a reduction in overall control traffic, as shown in Fig. 8. This is due to two main reasons: (i) Updates are less frequent than lookup queries, and, (ii) during lookup using anycast, there is at least one server which is *close to* the querying router. In comparison, both IP multicast and CCN have a much higher overhead even for a small sized graph, because both of them rely on some form of flooding to build the tree. In CCN, multicast receivers flood interests, whereas in IP multicast, the source floods a source-active message throughout the network [14]. In fact, even with a single replica and 90% of the network having multicast receivers MF improves average control overhead by a factor of 6.4 compared to the alternative schemes and the gain goes up to a factor of 37 when only 10% of the network have multicast receivers.

5 CONCLUSION

This paper presents the Named-Object abstraction as a novel approach that could help support network communications and services in the future mobile Internet architectures. By separating name and addresses, dynamically resolving destination locations through a Name Resolution Service and providing service intents to the network, Named-Objects can support a wide variety of different abstractions, spanning from the original host based virtual link, to recently introduced ones such as contents and context.

Compared with other name based approaches, Named-Objects are capable of reducing control overhead while still improving performance in different common scenarios like mobility, multi-group and multi-homed delivery. As these base services are the underlying foundation of a multitude of different network applications, our future research will explore ways to demonstrate how Named-Objects could be employed in a wider and more advanced set of service scenarios for future mobile Internet architectures.

REFERENCES

[1] 2016. The CAIDA UCSD Internet Topology Data Kit. http://www.caida.org/data/internet-topology-data-kit. (2016).
[2] A. Baid, T. Vu, and D. Raychaudhuri. 2012. Comparing alternative approaches for networking of named objects in the future Internet. In *Proceedings of Computer Communications Workshops*. IEEE.
[3] N. Basher, A. Mahanti, A. Mahanti, C. Williamson, and M. Arlitt. 2008. A comparative analysis of web and peer-to-peer traffic. In *Proceedings of WWW*. ACM.
[4] Francesco Bronzino, Kiran Nagaraja, Ivan Seskar, and Dipankar Raychaudhuri. 2013. Network service abstractions for a mobility-centric future internet architecture. In *Proceedings of the eighth ACM international workshop on Mobility in the evolving internet architecture*.
[5] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. 2013. Optimal multipath congestion control and request forwarding in information-centric networks. In *Proceedings of the eighth ACM international workshop on Mobility in the evolving internet architecture*.
[6] . Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan. 2011. COPSS: An Efficient Content Oriented Pub/Sub System. In *Proceedings of IEEE ANCS*.
[7] D. Farinacci et al. 2013. The Locator/ID Separation Protocol (LISP) for Multicast Environments, RFC 6831. (2013).
[8] D. Farinacci, D. Lewis, D. Meyer, and V. Fuller. 2013. The locator/ID separation protocol (LISP). (2013).
[9] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis. 2011. LISP Alternative Topology (LISP+ALT). *draft-ietf-lisp-alt-01, Internet Engineering Task Force* (2011).
[10] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. 2009. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM.
[11] P. Karimi and D. Raychaudhuri. 2016. Achieving high- performance cellular data Services with multi-network access. In *Proceedings of IEEE Globecom*.
[12] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. 2007. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review*.
[13] J. Laganier and L. Eggert. RFC 5204, 2008. Host identity protocol (HIP) rendezvous extension. (RFC 5204, 2008).
[14] D. Meyer and B. Fenner. 2003. Multicast source discovery protocol (MSDP). (2003).
[15] M. Mosko, I. Solis, E. Uzun, and C. Wood. 2015. *CCNx 1.0 protocol architecture*. Technical Report.
[16] R. Moskowitz et al. 2008. RFC 5201–Host Identity Protocol. (2008).
[17] S. Mukherjee, A. Baid, I. Seskar, and D. Raychaudhuri. 2014. Network-assisted multihoming for emerging heterogeneous wireless access scenarios. In *Proceedings of PIMRC*. IEEE.
[18] Shreyasee Mukherjee, Francesco Bronzino, Suja Srinivasan, Jiachen Chen, and Dipankar Raychaudhuri. 2016. Achieving Scalable Push Multicast Using Global Name Resolution. In *Proceedings of IEEE Globecom*.
[19] S. C. Nelson, G. Bhanage, and D. Raychaudhuri. 2011. GSTAR: generalized storage-aware routing for mobilityfirst in the future mobile internet. In *Proceedings of ACM MobiArch*.
[20] Erik Nordström, David Shue, Prem Gopalan, Robert Kiefer, Matvey Arye, Steven Y Ko, Jennifer Rexford, and Michael J Freedman. 2012. Serval: An end-host stack for service-centric networking. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*.
[21] J. Pan, R. Jain, S. Paul, and C. So-In. 2010. MILSA: A new evolutionary architecture for scalability, mobility, and multihoming in the future Internet. *Selected Areas in Communications, IEEE Journal on* (2010).
[22] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. 2009. CRAWDAD dataset epfl/mobility (v. 2009-02-24). (Feb. 2009).
[23] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani. 2012. Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet. *ACM SIGMOBILE Mobile Computing and Communications Review* (2012).
[24] Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, and Aditya Yadav. 2014. A global name service for a highly mobile internetwork. In *ACM SIGCOMM Computer Communication Review*.
[25] Tam Vu, Akash Baid, Yanyong Zhang, Thu D Nguyen, Junichiro Fukuyama, Richard P Martin, and Dipankar Raychaudhuri. 2012. 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*.
[26] Chris White, Darrel Lewis, David Meyer, and Dino Farinacci. 2016. LISP mobile node. *draft-meyer-lisp-mn-08, Internet Engineering Task Force* (2016).
[27] WiGLE - Wireless Geographic Logging Engine. 2016. http://wigle.net/. (2016).
[28] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* (2014).
[29] X. Zhu and J. W. Atwood. 2007. A secure multicast model for peer-to-peer and access networks using the host identity protocol. In *Proceedings of P2PM*. IEEE.