Comparing Alternative Approaches for Networking of Named Objects in the Future Internet*

Akash Baid, Tam Vu, Dipankar Raychaudhuri
WINLAB, Rutgers University, {*baid, tamvu, ray*}*@winlab.rutgers.edu*

***Abstract*—This paper describes and compares alternative architectures for achieving the functional goals of name oriented networking. The CCN (content-centric network) scheme proposed by Van Jacobson is contrasted with hybrid name and address based routing proposed in conjunction with the MobilityFirst (MF) future Internet architecture. In a CCN network, routers forward data directly on content names (such as URNs), achieving elegant and efficient retrieval of content files; the framework can also be extended to other communications services such as VoIP. The MF scheme supports name-based communication services by introducing the concept of a flat 'globally unique identifier (GUID)', which is used as the authoritative header for routing. Further, the GUID is dynamically mapped to one or more topological network addresses using a global name resolution service (GNRS). This leads to a hybrid GUID and network address based routing (HGN) scheme in which routers operate with both flat names (GUIDs) and network addresses, reducing routing table size and overhead at the cost of a fast distributed service for dynamic mapping of names to addresses. Protocol operations for both CCN and HGN are outlined in context of specific services including content retrieval, unicast and mobility. Preliminary evaluations of scalability and performance for both schemes are given using simple analytical models and selected results from Internet-scale simulations for GNRS and BGP.**

I. Introduction

In this paper, we provide a comparative discussion of two different approaches for realizing the functional goals of named data networking. Our goal is to initiate a discussion about alternative design approaches for enabling communication services based on names rather than addresses in the future Internet. We note that the architectures discussed are still in their early stages, so that the concepts and results given here should be viewed as work-in-progress to be updated and refined as further progress is made on analytical and experimental validation. The basic idea of named data networking is to provide communication services based on unique names associated with content, devices, sensors, people, etc. Such a network would operate on abstractions such as get(*'content_name'*), send(*'content_name' , 'receiver_name'*) or send(*'person_name', data*) to give a few examples. These abstractions, which are at a higher level than those of TCP/IP, should simplify application development and avoid centralized processing bottlenecks and control overheads associated with overlay solutions now in use such as Pub/Sub [1] and SIP [2].

In the prominent CCN (content-centric networking) approach proposed by Van Jacobson [3] and now the focus of an NSF FIA project [4], routers in the network directly operate on content labels making physical network addresses unnecessary. In the CCN approach, network-attached objects announce their content descriptor to their access network, and routers in this network further advertise the reachability of this content label to neighboring networks until an aggregated form of the content descriptor is found. End-user applications wishing to access this content simply send out an interest profile announcement to their access network provider, where routers attempt to forward the interest query to the content using reachability information stored in the routing tables. When the interest query reaches the network where the content is located, the file is delivered to the end-user along the reverse path. The CCN architecture is clearly an elegant one that 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.

However, the CCN approach has specific issues with scalability which are still the subject of ongoing research. At a general level the scalability problem stems from the fact that there are ∼10B to ∼100B potential network-attached objects (pieces of content, Internet based phones, sensors, etc.) with unique names, and every CCN router is required to maintain either a unique or an aggregated entry in its routing table for each of these names. Of course, the exact size of the routing table will depend on the achievable degree of aggregation, which is related to the intrinsic structure and locality of content. Although storage and computing technology has progressed considerably since IP was designed, supporting billions of routing table entries may not be practical in the foreseeable future. In comparison, BGP routing tables associated with IP have, at present, ∼400,000 entries, i.e., about 4 orders of magnitude lower than the size associated with content routing. This motivates consideration of alternative architectures which achieve the functional objective of realizing name-based communication services with reasonable storage and network overhead.

In the next section, we outline the architectural differences between CCN and the hybrid name and network address (HGN) based routing alternative central to the MobilityFirst FIA architecture [5], [6]. An approach similar to HGN has also been independently proposed in a recent IRTF contribution [7].

*Research supported by NSF Future Internet Architecture (FIA) grant CNS-1040735

II. Architectural Alternatives

Figure 1 shows the main difference in protocol architecture between CCN (Fig. 1a) and hybrid GUID and network address (HGN) routing (Fig. 1b). In CCN, a naming convention or a logically centralized name assignment service is used to create globally unique names such as *movies/Disney/musicals/Lion_King*. Routers in the CCN network maintain forwarding table entries with these semantically structured content names. The number of entries required may be reduced through content label aggregation, for example *movies/Disney* can be used to represent a large number of content files. The routing layer in the CCN network is thus based entirely on structured names, and hence scalability properties depend on the level of content aggregation and locality of the content.

In contrast, the alternative hybrid approach shown in Fig. 1b uses name assignment services to map content names (or context, device, people names) to 'flat' GUIDs which are randomly selected ∼160 byte public keys without any semantic structure. As shown in the figure, these GUIDs (unique to each network-connected object) are further bound to a set of network addresses (locators) that correspond to the current points of attachment. As an example, a mobile device would be assigned a GUID which is then dynamically bound to a series of network addresses corresponding to the locators of current points of attachment. Similarly, a content file is identified by a unique GUID and when copies of the file are located at multiple networks, the GUID is associated with a set of network addresses. The GUID thus serves as the central 'narrow waist' of the protocol stack, and is used to define communication services provided by the network. We note that the design of the indirection layers has been a central theme in future Internet research in the last decade and our architecture draws insights from many such works including, for example, DONA [8] and NetInf [9].

Routing in the HGN network thus has two types of primitives available the GUID name and the network address (NA), resulting in a hybrid scheme. The GUID is the authoritative routing header which is used to support content caching/retrieval and dynamic services such as mobility and multicast. GUIDs are mapped to NAs using a distributed service called the GNRS (global name resolution service). Any networked object or network element (router, access point, base station) can determine the GUID⇔NA mapping by consulting the GNRS service. Thus for relatively static services such as unicast between fixed end-points, routing can be done with NAs alone, while more advanced services may utilize GUIDs at the cost of higher table lookup latency. The HGN approach outlined above attempts to solve the scalability problem by dividing the problem into two distinct parts, i.e. (a) global name resolution service for mapping ∼10B objects to ∼100K networks, and (b) a routing protocol similar to BGP for distributing a routing table for ∼100K networks. Before moving to a comparison discussion of scalability, we provide a brief review of the MobilityFirst protocol operation and the GNRS functionality it uses in the next section.

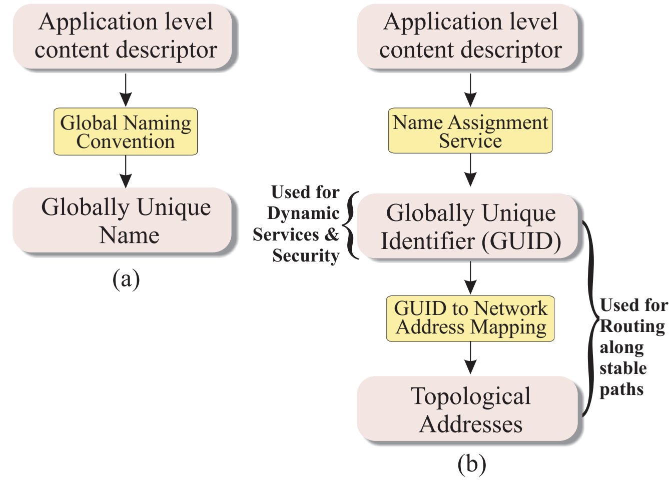

Fig. 1. Indirection layers in (a) CCN (b) HGN

III. Mobility First Protocol Review

The MobilityFirst protocol architecture is based on a clean separation of names (identifying network-attached objects) from network addresses. The design consists of a set of application specific 'name assignment' services which translate human readable names such as *'sensor@xyz'* or *'John's laptop'* to GUIDs. This framework also supports the concept of context-based descriptors such as *'taxis in New Brunswick'* which can be resolved by a context naming service to a particular GUID which serves as a dynamic multicast group for all taxis currently in that area. Once a GUID has been assigned to a network object, MobilityFirst uses the HGN routing scheme described above to route packets through the network.

A key component of the architecture is a fast name resolution service which is implemented as a distributed shared database hosted by network routers. In our ongoing research, we have demonstrated the use of reachability information already present at the routers to enable a single overlay hop DHT shared by the routers [10]. The GNRS supports dynamic mobility simply by providing the current point of attachment of the mobile device, without the need for routing-level indirection associated with current networking protocols such as mobile IP. The network addresses (NAs) are expected to change at a slower time-scale and can use a second distributed network protocol (analogous to BGP in the Internet) for dissemination of routing updates.

A second key design element in the MF architecture is the use of in-network storage at routers along with hop-by-hop transport of large protocol data units (e.g. entire content files). This enables advanced services such as disconnection tolerance, multi-homing, late binding, content- and context-aware delivery. Each router in the MF network has the option of making routing decisions based on the GUID or the NA's in the packet header. Because routing decisions are made on a hop-by-hop basis, there is no concept of an end-to-end connection, and no per-flow state.

The end-to-end packet flow in the MF network is shown

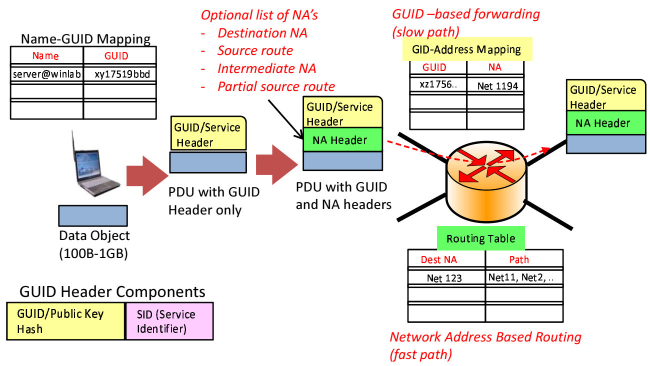

Fig. 2. Hybrid GUID/NA packet headers in MobilityFirst

in Figure 2. Packets entering the network have the destination (and source) GUID attached to the protocol data unit (PDU). There is also a service identifier (SID) in the packet header which indicates the type of service required by the PDU including options such as unicast, multicast, anycast, context delivery, content query, etc. At the first access router, the destination GUID is resolved by accessing the GNRS. The resolved NAs are optionally appended to the packet header thus making it possible for subsequent routers along the path to forward the PDU based on NAs alone - this is referred to as 'fast path' forwarding. Any router along the path has the option of resolving the GUID again by querying the GNRS - this is the so-called 'slow path' which allows for rebinding to a new set of NAs that may have resulted from mobility or temporary disconnection. The GUID routing option makes it possible to implement 'late binding' algorithms where the decision on which network ports to route to is deferred until the PDU is close to the destination.

IV. Comparing CCN and HGN

In this section, we compare the two routing approaches by a rough analysis of their performance on three key metrics: routing table size, update overhead and infrastructure requirement. Further, we highlight the differences between how these two schemes function for three specific use-case scenarios: content retrieval, unicast push/pull and mobile senders/receivers.

A. Routing Table Size

The growing size of the routing table at core routers is one of the major scalability issues in the current Internet [11]. In the HGN approach, since the network address space is decoupled from the content names, it can be designed to contain a network specific prefix unique to each network. Thus each AS only needs to announce its network prefix in the global routing table, making the routing table strictly bounded by the number of networks in the system. With CCN routing, each network can host multiple name prefixes, so the number of entries in the routing tables grows with the total number of named objects. The analysis in [7] and [12] shows that if the current website domain names are used as routable content names, the routing table size could be up to 2-3 orders of magnitude more than those of current BGP routers.

To compare CCN with HGN, Consider a simple content naming model with N levels of hierarchy and each prefix at level i having l_i sub-level prefixes. The routing table size with such a namespace could range from l_1 (if only the very top level names need be announced) to $\prod_{i=1}^{N} l_i$ (if all prefixes up to the last level have to be announced). The extent to which the prefixes can be aggregated depends on the mapping between the naming tree and the topological structure of the network. For example, a possible scenario which leads to the case with just l_1 routing entries is when all content originating from each AS contains the AS name or number as the top level prefix.[1] Of course, such a naming scheme is too restrictive as it requires a one-to-one association between content and the network in which it is located (not usually the case with increasing migration of devices and content that we see in the Internet today).

In order to relax this constraint and understand the dependence of name aggregation on the network topology, we introduce a parameter $n_{top} \in \{1, 2, \ldots, N\}$ which indicates the prefix level below which the naming tree starts being influenced by the network topology. In the case with all names starting with the AS prefix, $n_{top} = 1$ and if the name structure is completely independent of the topology, $n_{top} = N$. In general, the prefix structure $/p_1/p_2/\ldots/p_j/\ldots/p_N/$ with $n_{top} = j$ is such that all contents for each distinct value of p_j are announced by the same AS either by design or by chance. As such, the number of routing table entries that each AS would need to announce would reduce from $\prod_{i=1}^{N} l_i$ to $\prod_{i=1}^{j} l_i$ since more specific prefixes can all be aggregated at the jth level. Figure 3 shows an example of this dependence of prefix aggregation on the network topology and Figure 4 plots the average routing table size as a function of n_{top} for $N = 10$ and different values of L under the simplifying assumption $l_i = L$ for all $i \in \{1, 2, \ldots, N\}$. The marked points on the y-axis in Fig. 4 shows the table size required by this naming model when topology independent names are used. Also shown is the current technology limit on the BGP routing table size estimated to be around 1 million entries and the table size using a hybrid scheme such as HGN (equal to the estimated number of networks). We note that table sizes for name-based routing grow exponentially as we decrease the dependence of topology on names, i.e., increase the n_{top} value.

The key result from this analysis is that hierarchy in name structure reduces the table size only when the name prefixes have some degree of dependence on the physical network topology. If a clean separation between content names and their network location is enforced, architectural enhancements (such as [13]) may be required to overcome this rather basic scalability issue in CCN routing. A two layered approach such as HGN routing decouples the routing table from the content space size and thus can lead to smaller routing tables.

[1] To be more precise, the number of entries would be equal to the number of ASs, but we assume that number is of the same order as l_1 for clarity in exposition.

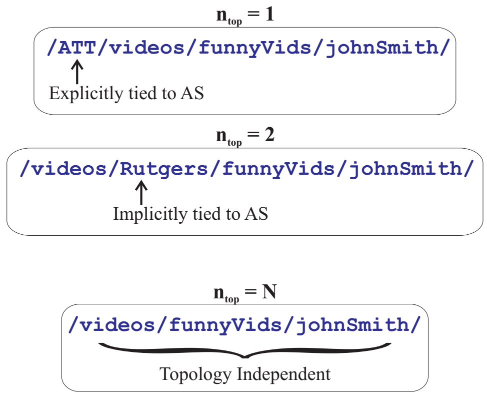

Fig. 3. Simple example of topology (in)dependent prefixes

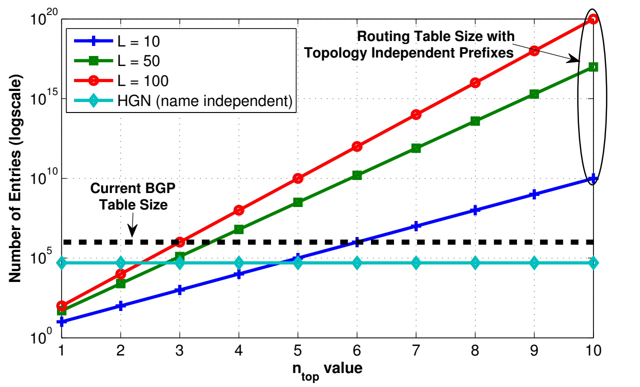

Fig. 4. Variation of routing table size for CCN and HGN as a function of n_{top} for $L = \{10, 50, 100\}$

B. Update Overhead

In HGN routing, network reachability is maintained through the routing protocol and content reachability through the GNRS. As such, unlike CCN, content additions, deletions and changes in its hosting location do not effect the network routing layer. Two factors contribute to the difference in update overhead between the CCN and HGN approaches: (i) Since the number of content names and their volatility may be higher than those for network addresses, CCN routing produces a higher number of routing update messages; (ii) A change in the network topology results in an update of all the name-prefixes affected by the change, thus the size of each update message which is caused by a network route change could be substantially higher.

Next we study the routing overhead for the two approaches, using an AS-level topology generator and BGP simulator as described in [14]. This simulation tool generates realistic AS topologies with three kinds of nodes: tier-1 nodes (T) which form a clique and do not have any providers; mid-level nodes (M) which have one or more providers (other M nodes or T nodes) and can peer with other mid-level nodes; and customer nodes (C) which have one or more mid-level nodes as providers. In our simulation, we generate three topologies corresponding to different values of the total number of AS nodes $A = \{1000, 5000, 10000\}$. The number of T nodes

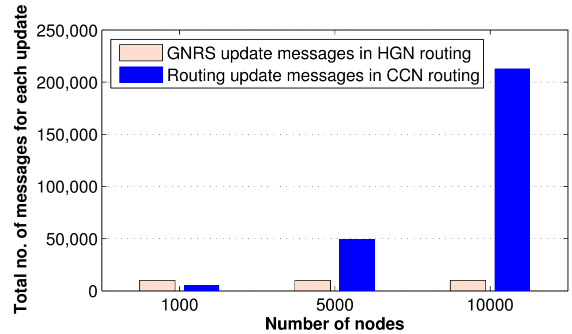

Fig. 5. Total update messages generated due to a single update event

is kept constant at 5 while the rest is divided into 15% M nodes and 85% C nodes. All other parameters are same as in [14]. Since our focus here is the update overhead and not the table size, we consider one distinct name prefix per C node containing 1,000 content objects. Further, we capture the effect of routing updates by reusing the BGP policy described in [14]: all ASs announce routing updates from customers to its providers and peers and not to other customers, while updates received from providers and peers are announced to all customers.

Using this setup, we simulate the effect of dynamism in name prefix announcement by withdrawing the name prefix from an AS and re-announcing it after the network routing table has converged. The total number of name update messages passed between nodes due to this event is recorded and the value is averaged over similar events at each AS. Note that such an event results in a routing update message in CCN routing which is then propagated to other networks while in HGN routing, the update for each individual content object is updated in the GNRS database. Using the same GNRS implementation parameters described in [10], we assume that each GUID⇔NA mapping entry is replicated at $K = 5$ different places, thus requiring a total of $2*K$ one-hop overlay messages for each update: K each for name withdrawal and re-announcement. Figure 5 shows the total number of routing update messages generated in the system to propagate each name-prefix change event in both CCN and HGN. The plots show that the protocol overhead for name updates in CCN grows exponentially with increasing name space size, whereas the cost of a GNRS update remains constant.

C. Infrastructure Requirements

The scalability properties of the HGN approach in terms of routing table size and protocol overhead comes at the cost of a global name resolution infrastructure. Some common concerns related to Internet-scale services (such as DNS) are: storage space requirement and its distribution, amount of maintenance overhead, concerns about single points of failure and the lookup latency. The MF approach [10] to overcoming these problems is to apply distributed hash table (DHT) technology directly to the network infrastructure to create a virtual GUID⇔NA table at every router without requiring any additional servers or centralized resources. Specifically,

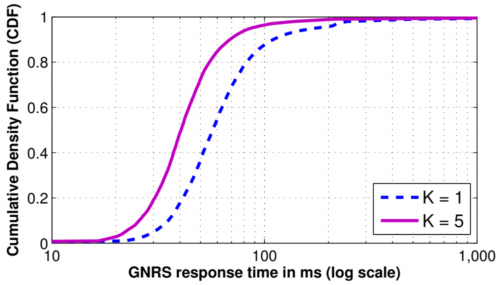

Fig. 6. CDF of GNRS lookup response times for 1 million lookups originated from uniformly random end-hosts in the Internet

storage at routers is scaled by distributing the GUID⇔NA mappings across the network (by hashing the GUID to the AS address space) and maintenance of routing table is minimized by piggy-backing on the underlying network protocol (such as the BGP). Replication (DHT-based storage at $K > 1$ networks) is used to prevent aggregated points of failure which also helps in keeping the lookup latency under check.

In order to evaluate the GNRS response time, we use a measurement-driven simulator described in [10]. This AS level simulator uses the inter-AS and intra-AS latency measured in the Internet through the DIMES project [15] to estimate the end-to-end latency required given a pair of end-points. We evaluate the GNRS response time by querying the GNRS from 1 million randomly selected end-hosts distributed uniformly across all ASs. Figure 6 shows the cumulative density plot of the round trip response time of GNRS for two different replication factors: $K = 1$ storage location per GUID⇔NA mapping and $K = 5$ storage locations per mapping. These results show that the extra latency caused by the GNRS lookup can be bounded to ∼100 ms for an Internet-scale DHT deployment. These numbers are sufficiently low to support dynamic mobility and represent an acceptable latency for most services (note that caching of NAs can be used where faster response times are needed).

D. Use-case Comparison

In this section, we compare how CCN and HGN approaches can be applied to a set of basic service scenarios (use cases).

1) Content Retrieval: Since CCN routing treats content retrieval as the basic networking primitive, caching and retrieval of content is performed very naturally. The process of interest packet routing through name prefix announcement for content retrieval in CCN is outlined in Sec. I. HGN achieves the same functionality by enabling caching/retrieval based on GUIDs. In this regime, content providers create GUIDs for their content and insert an entry into the GNRS denoting its network address and the content GUID. A consumer retrieving this content first obtains it's GUID through a well-known name assignment service and sends a *get('GUID')* primitive to the network along with its own network address. The first router queries the GNRS to resolve the GUID to a network address and relays the query to the provider. The content on its path from the provider to the consumer can optionally be cached at every router by its GUID just like CCN. Future queries for the same GUID received by one of these routers triggers the router to send back a cached copy of the content and the packet is terminated without being sent to the original content provider.

Service Model	CCN		HGN	
	Mean	Median	Mean	Median
Unicast Pull	440	382	488	430
Unicast Push	659	573	269	238

TABLE I
END-TO-END LATENCIES (IN MS) FOR UNICAST PUSH AND PULL SERVICES

2) Unicast Push/Pull: Standard unicast message delivery between two network-attached objects remains important for basic services like email, instant messaging and voice calls. When the two end-points are fixed and stable, both name and address prefixes can be assumed to have sufficiently propagated to enable policy-compliant shortest path routing of packets between the end-points. We consider two cases: push services where a message is sent by the originator and pull services where the message is retrieved from a server. Since pull services require a flow of data from one end-point to the other and back again, a minimum of one round trip time is required for both CCN and HGN schemes whereas push services requires only the one-way transit latency. HGN routing incurs a varying initial latency due to GNRS lookup in both push and pull models. CCN routing, on the other hand slightly differs between the push and pull cases: For push services the sender has to first send out an interest packet soliciting the receiver's interest in the data which causes an extra initial delay of one round trip time, while for data pull services there is no additional latency.

Table I compares the two approaches in terms of the mean and median latencies for 100K unicast push and pull messages. The numbers are obtained by using current Internet latency values measured through [15] for data path latency between random pairs of end-points and adding the overheads corresponding to each scheme as mentioned above. Thus if x denotes the one way end-to-end message transit time and y denotes the lookup latency, pull messages incur a $2x$ latency for CCN and $2x + y$ for HGN. Whereas for push messages, the latencies are $3x$ for CCN and $x + y$ for HGN.

3) Mobile Receivers/Senders: CCN handles receiver mobility through in-network caching and sender mobility through direct signalling or routing updates [3]. When a receiver while waiting for data in response to an interest packet moves to a new network location, all routers along the path of the data store the content but the data packet is ultimately dropped. Upon joining the new network, the receiver re-issues an interest packet which is propagated upstream and depending on specifics of network topology and content prefix aggregation, fetches the data from one of the cache stores. However when a content *source* moves from a network to another, it needs to announce the reachability of the content through the routing protocol. This leads to a routing update message and depending on the prefixes already being announced by the second network, could require further propagation in the global routing tables. In HGN routing, the receiver (which retains the same GUID during mobility) upon joining a new network updates the GNRS with its current NA. This enables the last-hop router which had to drop and subsequently cache the packet to proactively push the stored data instead of waiting for a fetch by the receiver. More importantly, sender mobility in HGN is supported in an equally seamless manner. The content source after joining a new network updates the GNRS with its current NA and future lookups for the content GUIDs are automatically directed to the new location of the content provider.

To compare CCN with HGN, we focus on a specific mobility use case: voice over Internet calls through mobile phones. Reference [16] describes the functioning of VoIP-like services in a CCN routing framework; albeit for fixed end-points. Using the same example as in [16], we assume that Alice, with a registered routing name */ccnx.org/sip/alice/* wants to make a call to Bob who has the name */parc.com/sip/bob*. If Bob wishes to receive calls through the same name as he moves, he must announce his intent in receiving interest packets for his name through whichever network he visits, thus requiring routing announcements within and possibly outside the network.

Extending this example to the estimated 5 billion mobile phones in current use, we analyze the total number of update messages generated per day in the entire network assuming that each mobile switches an average 10 (considered typical today) or 100 (may be more realistic for future mobile Internet scenarios) networks in a day. Since the propagation of CCN updates depend on the aggregation of the names, we reuse the BGP simulator described in Sec. IV-B with 10,000 AS nodes and assume a variable $x\%$ of the updates received by an AS needs to be propagated to the global routing tables (i.e., the visiting mobile has a name which is not covered by any of the prefixes already announced by the AS). Figure 7 shows the mobility related overhead in the entire network in terms of number of update messages for a varying value of x between 0.01 and 10%. The results shown indicate a significantly higher rate of updates (typically 2-3 orders of magnitude) for CCN over HGN, increasing with the frequency of prefix updates propagated outside the network.

Similar comparisons can be made for other important services such as multicast push/pull, anycast, context-aware delivery, and so on. Discussion of these use cases is omitted here due to space constraints, and will be presented in future work.

V. CONCLUSION

In this paper, we have presented a general comparison between CCN and an alternative hybrid name and network address based routing scheme currently under consideration for the MobilityFirst future Internet architecture. Key differences in the architectures (direct support of content labels vs. GUID indirection layer) have been explained and protocol operations were outlined for some sample uses cases. An informal evaluation of scalability and protocol overhead properties of the CCN and HGN schemes was presented with the objective of initiating further discussion and motivating future improvements to both schemes. The results presented show that in certain scenarios, the hybrid GUID and network address approach may offer scalability and performance improvements over baseline CCN. It is recognized that future Internet architectures discussed are still at an early stage, and this contribution is thus intended as work-in-progress discussion rather than as a definitive assessment. Future work on this topic includes detailed simulation and large-scale GENI experimentation to further validate the scalability and performance of the MF HGN scheme.

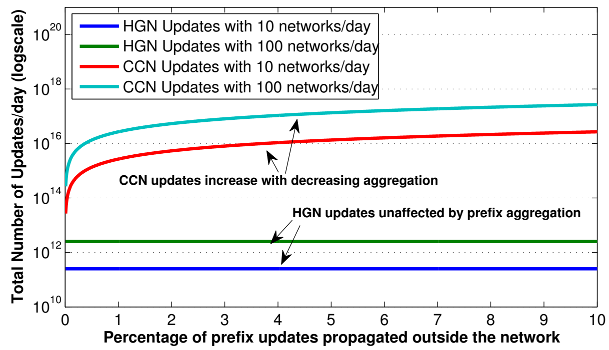

Fig. 7. Mobility overhead in the entire network for CCN and HGN

REFERENCES

[1] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg, "Content-based publish-subscribe over structured overlay networks," in *Proceedings of ICDCS 2005*, June 2005, pp. 437 –446.
[2] *SIP: Session Initiation Protocol*, IETF Internet Standard, RFC 3261, June 2002.
[3] V. Jacobson et al., "Networking named content," in *Proceedings of CoNEXT '09*, pp. 1–12.
[4] "Named Data Networking (NDN) Project," 2010. [Online]. Available: http://www.named-data.net/ndn-proj.pdf
[5] MobilityFirst Future Internet Architecture Project, http://mobilityfirst.winlab.rutgers.edu/.
[6] D. Raychaudhuri, *MobilityFirst: A Robust and Trustworth Mobility-Centric Architecture for the Future Internet*, PIMRC 2011 Keynote Speech. [Online]. Available: http://bit.ly/v3HrMD
[7] A. Narayanan and D. Oran, *Content Routing using Internet Routing Protocols: Can it scale?* [Online]. Available: http://trac.tools.ietf.org/group/irtf/trac/wiki/icnrg
[8] T. Koponen et al., "A data-oriented (and beyond) network architecture," in *Proceedings of SIGCOMM '07*, 2007, pp. 181–192.
[9] B. Ahlgren et al., "Design considerations for a network of information," in *Proceedings of CoNEXT '08*, 2008, pp. 66:1–66:6.
[10] T. Vu et al., Technical Report WINLAB-TR-397 - DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet, Fall 2011.
[11] T. Narten, *On the Scalability of Internet Routing*, IETF Internet Draft, draft-narten-radir-problem-statement-05.txt, Aug. 2010.
[12] D. Perino and M. Varvello, "A reality check for content centric networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ser. ICN '11, 2011, pp. 44–49.
[13] F. Papadopoulos, D. Krioukov, M. Boguñá, and A. Vahdat, "Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces," in *Proceedings of INFOCOM'10*, pp. 2973–2981.
[14] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "On the Scalability of BGP: The Role of Topology Growth," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1250 – 1261, 2010.
[15] Y. Shavitt and E. Shir, DIMES - Letting the Internet Measure Itself, http://www.netdimes.org/.
[16] V. Jacobson et al., "VoCCN: voice-over content-centric networks," in *Proceedings of ReArch 2009*, pp. 1–6.