Custodian-Based Information Sharing

Van Jacobson, Rebecca L. Braynard, Tim Diebert, Priya Mahadevan, Marc Mosko, Nicholas H. Briggs, Simon Barber, Michael F. Plass, Ignacio Solis, and Ersin Uzun, Palo Alto Research Center
Byoung-Joon (BJ) Lee, Myeong-Wuk Jang, and Dojun Byun, Samsung Electronics
Diana K. Smetters and James D. Thornton, Google Inc.

ABSTRACT

Information sharing systems such as iCloud, Dropbox, Facebook, and Twitter are ubiquitous today, but all of them depend on massive server infrastructure and always-on Internet connectivity. We have designed and implemented a sharing system that does not require infrastructure yet supports robust, distributed, secure sharing by opportunistically using any and all connectivity, local or global, permanent or transient, to communicate. One key element of this system is a new information routing model that so far has proven to be as scalable and efficient as the best of the current Internet routing protocols, while operating in an environment more complex and dynamic than they can tolerate. The new routing model is made possible by new affordances offered by information-centric networking, in particular, the open source CCN [1] release. This article describes the new system and its routing model, and provides some performance measurements.

INTRODUCTION

Routing is the glue that converts a collection of wires and switches into a network. Thanks to decades of experience, routing protocols such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) are highly efficient, scalable, and robust. But this efficiency is derived, in part, from constraints on the network. In particular, both OSPF and IS-IS assume that the network adjacency graph (which nodes have direct links between them) is relatively static even though links go up and down dynamically. In other words, link availability can change rapidly, but not link existence. This model works well for desktop PCs talking to servers over wires but not for our increasingly wireless mobile world. There is ongoing research on adapting routing to function when adjacencies change rapidly, but to date these adaptations have significant efficiency and scalability issues [4].

Information-centric networking (ICN) offers new ways to solve these issues:

- In host-centric networking, applications specify where their communication terminates (the destination host); thus, the only role for routing is to determine how to get there. With ICN, applications specify only *what* information they want, leaving both *where* and *how* to the routing/forwarding subsystem. As described below, this extra degree of freedom can be used to increase both the energy and bandwidth efficiency of the system.
- Host-centric routing is responsible for creating loop-free paths between each pair of nodes. The time required for all involved nodes to reach agreement on which paths to use is called the *convergence time*. As mobile networks scale up, the rate of topological change can become greater than the convergence time, making data delivery impossible. The CCN ICN architecture slightly changes the semantics of router buffer memory so that it functions as a short-term cache. This small change has many profound effects, one of which is to prevent forwarding loops, making routing behave as if it converged instantaneously [1, sec. 4.1].
- Host-centric transport is conversational and point-to-point. To find if some piece of information is available in the local environment, it is necessary to first discover the local hosts and then query each of them for the desired information. Since ICN architectures do not presume a destination, they can efficiently use inherently broadcast media such as wireless. Thus, asking all hosts in radio range for a piece of information has the same cost in energy and time as asking one, and there is no need for a "discovery" phase.

We have recently developed Custodian-Based Information Sharing (CBIS), which allows simple, secure, and distributed information sharing between a user's devices, family, and friends. Unlike the cloud-based sharing models in use today, which require both always-on Internet connectivity and substantial centralized server capacity, CBIS is designed to function with any available connectivity, local or global, permanent or transient, and to keep a user's information under complete control of the user. It uses CCN for its communication layer, together with a novel routing architecture specifically designed to take advantage of the ICN affordances mentioned above. In the next section we describe CBIS and its routing model, followed by measurement studies comparing our prototype implementation's delivery efficiency and routing scalability to that of conventional approaches.

Diana K. Smetters and James D. Thornton were at PARC when the work described in this article was performed.

0163-6804/12/$25.00 © 2012 IEEE

CBIS is built on top of the open source CCN implementation available at www.ccnx.org. This implementation provides application programming interfaces (APIs) to create, manage, and securely access content. *Content Objects* are retrieved by sending *Interest* packets containing a name prefix identifying the desired content. Interest packets are routed similar to the way IP packets are routed toward their destination: a longest-match lookup is done in the router's forwarding information base (FIB). If the router recently retrieved the content and still has it in its buffer memory, the FIB lookup resolves to a pointer to that content, which can then be immediately returned. Otherwise, the Interest is remembered and forwarded on to one or more of the interfaces listed in the FIB entry. Received Content Objects are matched to outstanding Interests and sent back out onto the link over which the Interest arrived.

CBIS Design

Thanks to an exponential decrease in the cost of storage, modern mobile phones and tablets hold tens of thousands of content items, and typical family media centers and backup servers hold millions. At these scales, people cannot manage individual items of information. Hierarchical, ontological naming structures have been successfully used for information management since the 1980s. They are easy to comprehend, simplifying the information production process, and easy to navigate, simplifying the discovery/foraging process. Hierarchies also make it easy to specify policies that should be applied to collections of information, allowing a user to delegate the detailed management work to machines. For example, John Smith might want all the pictures taken with his phone or camera to automatically go to the family media server and for all his calendar items to be on his phone, laptop, and the family backup server. These policies could be specified as:

```
/JohnSmith/photos -> SmithFamily_Media,
JohnSmith_phone, JohnSmith_camera
/JohnSmith/calendar -> JohnSmith_phone,
JohnSmith_laptop, SmithFamily_backups
```

The first item on each line specifies a collection of information: the set of items all of whose names start with the given prefix. For example, if a picture taken on John's phone were named `/JohnSmith/photos/phone/img0123.jpg` it would automatically become a member of this collection (item naming need not be done explicitly since the phone has sufficient context to synthesize all the components of the name).[1]

Most ICN architectures allow information to be obtained from anywhere there is a valid copy [see survey articles in this special issue]. Social sharing results in copies of each item of information on many different devices. If, say, every device advertised each piece of information it held, control traffic would scale as the number of information items times the number of devices. To avoid this explosion of control traffic, CBIS distinguishes *custodians*, entities that have an explicit relationship with some collection of data, from those who happen to have a copy of it. In the example policy specification, items after the "->" name custodians. For example, `SmithFamily_Media` is the name of the Smith family's media center.

The custodian name identifies an entity that may have the desired information but says nothing about how to communicate with it. Each custodian publishes an *Endpoint Table* listing all the communication endpoints that can currently be used to reach it. Since CCN works over a large variety of media, endpoint information is quite general. It can be a globally routable IP address, an Ethernet, WiFi or Bluetooth MAC address, a DNS, DynDNS or ZeroConf mDNS name, a local-use address with disambiguator such as a gateway or AP MAC address, the target identifier needed to make a SIP call and set up a DTLS tunnel, etc.

These two collections of information, the *Prefix-to-Custodian* table (PCT) and the *Custodian-to-Endpoint* table (CET), are the routing information exchanged by CBIS. Since prefix-to-custodian bindings are independent of each other, each custodian publishes a PCT item for each prefix it is responsible for. Each item is a single, versioned, Content Object signed by the custodian's key (attesting that it agrees to host the content associated with the prefix). Since a custodian's endpoints are typically not independent (e.g., when a mobile moves, its old IP endpoint goes away and a new one takes its place), the complete endpoint list is published as one Content Object and a new version of the object is published whenever one or more endpoints change.

The routing model explicitly includes the custodian entity as an intermediary between the prefix and communication endpoints, even though the CCN FIB entries constructed by CBIS from the routing information map directly from the prefix to a set of endpoints (Fig. 1). There are two reasons for doing this. First, the prefix-to-custodian bindings are relatively long lived but custodian endpoints are not, particularly for mobiles. If the routing data were published in a form closer to the FIB representation, all the prefix bindings would have to be republished each time an endpoint changed. Internet routing has long had problems with the traffic churn caused by this kind of representation. For example, the Border Gateway Protocol (BGP) provides no way to identify the set of prefixes that transit a particular next-hop router other than enumeration. Thus, the loss of a single next-hop can cause BGP to issue tens of thousands of withdraws for the prefixes that were going through it followed by tens of thousands of announcements to associate those prefixes with their new next hop [7, p. 26].

The more important reason is that the different custodians of an information collection are not interchangeable. In the second example above, John's calendar items could be obtained from his phone, his laptop, or the family backup server. But the phone and laptop both operate on batteries, and their usability suffers if they drain those batteries servicing information requests. Thus, CBIS custodian announcements contain a priority field that helps an informa-

Thanks to an exponential decrease in the cost of storage, modern mobile phones and tablets hold tens of thousands of content items and typical family media centers or back-up servers hold millions. At these scales, people cannot manage individual items of information.

[1] These names are for illustration purposes. Clearly "JohnSmith" is not unique and thus a poor choice for an information prefix. Most of the top-level identifiers in CBIS are the fingerprint of an entity's public key, and user-friendly names like "John Smith" are attached to that identifier via metadata conventions. CBIS naming details were primarily driven by authentication, privacy, and trust issues that are outside the scope of an article on routing. Some information is given in [5], and we plan to write a future article on this topic.

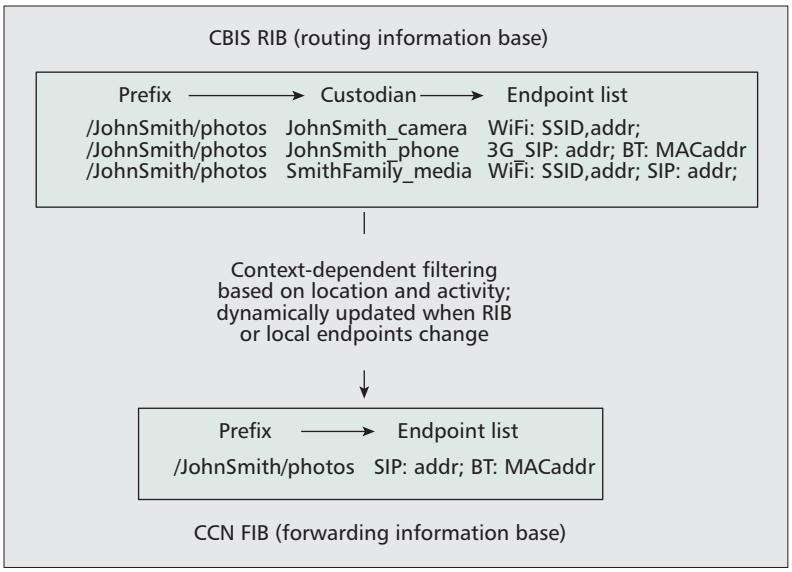

Figure 1. *Relationship of CBIS to CCN forwarding.*

tion consumer pick which custodian(s) should be queried. High priority is typically given to a server with continuous power and connectivity, lower priority to powered non-servers such as a desktop, on down to the lowest priority, which is given to mobiles that will only accept requests from a server-level custodian for a prefix they have in common. For example, when John Smith tweets the name of a picture he just took on his phone, the default custodian priorities cause Interests for the picture to go to the Smith family media center. If the media center does not already have a copy of the picture, it may ask the phone for it. This policy both offloads work from the phone (it only has to service one request for the picture no matter how popular it becomes) and causes the media server to back up the new content at the earliest possible time.

Like custodians, endpoints are also not interchangeable. A mobile might be unwilling to receive anything but infrequent high-priority notifications on its cell interface but happy to accept anything on its WiFi interface. This kind of policy is easily implemented with a priority similar to the custodian priority. But even if the mobile has a functioning WiFi connection, firewalls or network address translation (NAT) boxes may make it only useful to communicate with other hosts on the same access point (AP). This kind of restriction requires that the information consumer use its current communication context to ignore unreachable custodian endpoints. For example, WiFi endpoints are ignored unless the consumer has a WiFi connection with the same SSID and AP medium access control (MAC) address, and Bluetooth addresses are ignored unless they have been detected in proximity. Given the ubiquity of NAT and firewalls, the only custodians that can be reached easily are those on the same net or those with globally reachable IP addresses (which are expected to be rare for the CBIS target user community). Anything else requires setting up a tunnel to punch through the firewalls at both ends. Unfortunately, this operation is expensive since it requires the assistance of external infrastructure like STUN and TURN servers. Because of this, CBIS always prioritizes currently active endpoints to reach a custodian, even if the static priority of that custodian or endpoint is low. Only if there are no endpoints active or the active endpoints fail to retrieve the content will alternatives be tried. The custodians are tried in priority order, and all feasible endpoints of that custodian are tried in endpoint priority order until one succeeds or none are left.

Custodian priorities offload the burden of mobile information production but do not benefit information consumers. To improve consumer efficiency, local servers such as a home gateway can announce that they will serve requests for information they are not custodians of (typically for "/," the root prefix which matches everything, but they could announce a more restricted set of prefixes). This announcement, combined with the opportunistic endpoint usage described above, will cause the local server to be queried first. If it does not already have the information, its normal CBIS routing forwards the Interest to the correct custodian. This essentially turns the local gateway into a proxy cache for external content. This offloads traffic from the remote custodian, improves latency and bandwidth if multiple local clients want the same information, and reduces mobile power consumption since the mobile can consume data from the local gateway based on the mobile's most efficient sleep/wake schedule. Studies reported in [2] demonstrated that this can result in a 4x power reduction compared to conventional network transport.

A similar idea can be used to implement the broadcast transfer mentioned in the introduction. CBIS can (optionally) send low-rate probe packets on broadcast-capable interfaces to learn if there are any other devices on that interface that might source or sink information which may be of interest. If so, 224.0.23.170 (the IANA-assigned multicast address for CCN) will be registered on the interface and added to the FIB entry of each interesting prefix discovered.

CBIS helps to illustrate that both the nature of routing and the division of labor between the routing (control) plane and forwarding (data) plane are fundamentally different with CCN. In IP, routing does almost everything and forwarding almost nothing: Routing is responsible for picking the single "best" next hop to some destination. This decision is normally based on a shortest path computation using statically assigned link weights. Since IP is a unidirectional datagram protocol, the forwarding plane has no idea if the downstream hop is actually functioning. Thus, the burden of determining which adjacent nodes are feasible (have not failed) is left to the routing protocol, which does it by exchanging periodic hello messages with its peers. Since reroute (repair) time is dominated by failure detection time, these messages need to be frequent, which results in both the high level of control traffic and heavy weight peerings that characterize IP routing.

CCN forwarding is bidirectional—the forwarding plane is guaranteed to see both packet types: questions (Interests) and their answers (Content Objects). Thus, it knows not only which downstream peers are functioning but how well they are functioning (e.g., which ones return answers the quickest or most reliably). Since the forwarding plane, as part of its normal functioning, does both fine grain fault detection and relative peer performance evaluation, routing should do neither of these. This means that CCN routing needs no notion of 'peer' (in the IP sense) and has none of the associated control traffic. It also means that the relationship between routing and forwarding is more balanced: routing provides choices and forwarding chooses.

Figure 2 shows a CBIS use case illustrating the routing described above. It starts with John running into his father on the train and sharing a new baby picture via Bluetooth. This works because John's phone is one custodian of his photos, and its Bluetooth MAC address is listed as one of its active communication endpoints. When the photo browser app on John's father's phone looks for new photos from John, normal Bluetooth proximity detection tells it that John's phone is nearby, so it establishes a Bluetooth transport channel, and then uses Interests to browse for and retrieve John's new picture. When he gets off the train, John's father has no Internet connectivity and thus sends the name of the photo (encoded as a "cbis://" URL) as an SMS to his wife. When she taps the URL, her tablet sends an Interest for the picture to the Parent's House media center, which is advertising a route to "/" (it is willing to retrieve any content) on the local WiFi network, and the tablet is connected to that network. The Parent's House media center does not have active connections with any of John's photo custodians, so it starts with the highest-priority custodian, the Smith Family media center. The highest-priority endpoint, the Smith Family house WiFi, is not accessible at the Parent's House, so the next-highest-priority endpoint, a SIP contact address for setting up a DTLS tunnel, is tried and succeeds. The Smith Family media center then gets the Interest for John's new picture, but since John has not been home since he took the picture, it does not have it yet. So it looks for the highest-priority custodian for John's photos that it is allowed to contact and tries John's phone. The phone is outside the house and is not advertising an endpoint on the local WiFi, and the media center cannot talk to its Bluetooth address, so the last choice, a SIP call/DTLS tunnel to the phone's third generation (3G) IP address, is tried. Since the credentials of the SIP call identify the caller as the Smith Family media center, the only entity John has allowed to call his phone for data, the tunnel is set up, the Smith Family media center retrieves the new picture, keeps a copy (since it is a custodian of that name space), and sends it via the DTLS tunnel to Parent's media center (which caches a copy since other devices in the house often request newly arriving content) which then sends it to John's Mother's tablet.

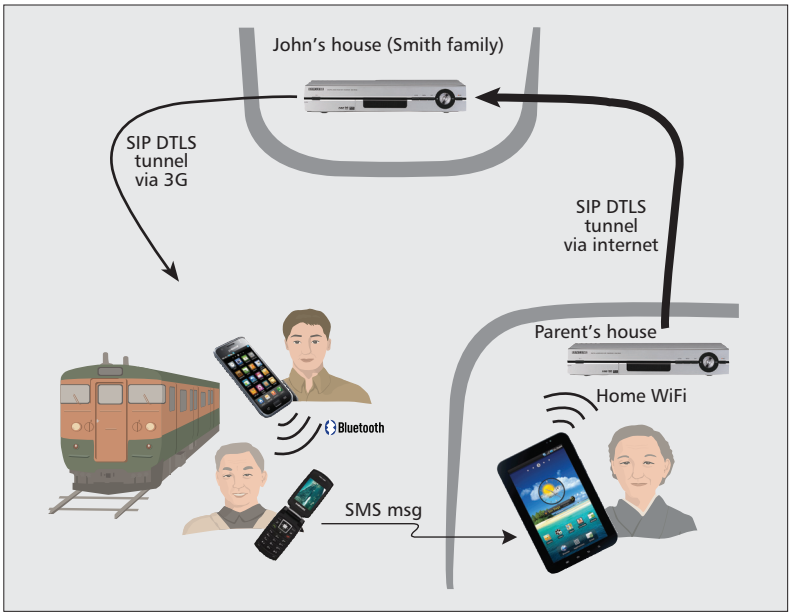

Figure 2. *CBIS use case.*

CBIS Routing Data Exchange

In Internet routing, peers identify each other via Hello message exchanges, compare databases to ensure they each have the same view of the network, then go to a steady-state mode where they send keep-alives to each other, listen for state change packets, and flood new ones they receive. As long as the topology is stable (no new nodes or links), the traffic during this phase is proportional to the rate of link state changes, which is the minimum possible for this problem. The high overhead portions of peer-based routing are initial database synchronization and keep-alives. The previous section pointed out that CCN's forwarding plane makes the second unnecessary, so one wonders about the first. Since the goal of a routing protocol is to ensure that all routers have the same view of the network (the same set of routing data), CBIS takes the information-centric approach that this should not be done indirectly by comparing notes with peers but directly in terms of the collection of routing data. It does this by means of a novel transport protocol we call Sync.

Sync was inspired by problems like having a user's calendar or contacts data automatically be the same on all their devices. Solutions to these problems generally have three common elements:

- There is a name for the collection (e.g., /JohnSmith/calendar).
- The collection is monotone (item x is deleted by publishing a new item saying "x is gone" rather than by creating a new collection without x; this makes it possible to distinguish a collection that never contained x from one where x was deliberately removed).
- When parties holding the same collection communicate, their goal is to each end up with the union of their collections (in computer science, this is the well-known set reconciliation problem). CCN data is named

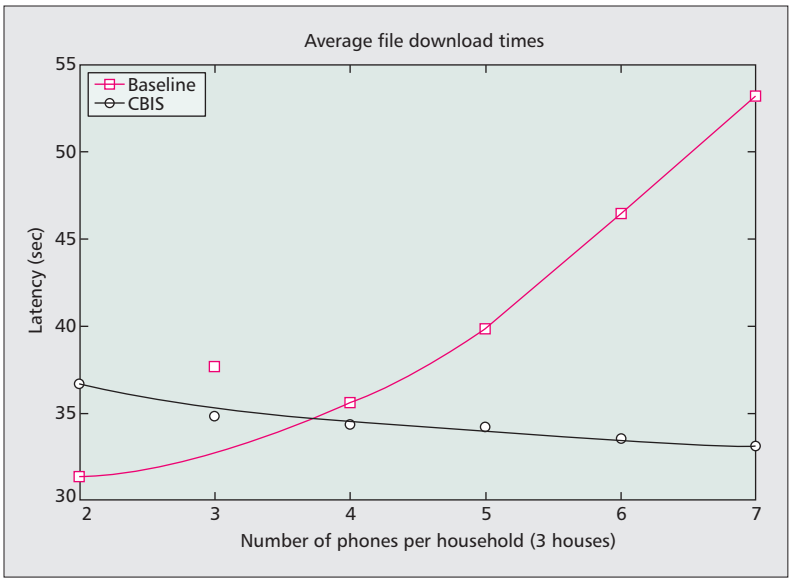

Figure 3. *Average file download times.*

and persistent, so the first two elements are almost automatic. The third element has two parts:

–The parties detecting that they hold different instantiations of the same collection

–Reconciling the differences

Sync solves part 1 by having nodes announce an Interest containing the name of the collection concatenated with a checksum of all the items in that node's collection. Part 2 is solved by having the semantics of this Interest be that any node whose collection has a different checksum should initiate a "set reconciliation" by replying with a Content Object containing the initial data for a fast, bandwidth efficient reconciliation algorithm such as the one described in [3]. These algorithms have communication and computational complexity proportional to the difference of the two sets.

Thus, Sync traffic for the collection "/local/CBISrouting" is proportional to the state differences, just as is best-case steady-state peer-routing, but Sync communication involves only information about sets of routing data and nothing about the peers holding those sets. Thus, Content sent in response to the initial Interest updates the routing data for all nodes with the same data as the node that sent the Interest. Since neither the Interest nor the associated Content are peer-specific, they can be sent to anyone at any time and can synchronize routing (via data muling) even in networks such as DTNs where there is never a connected path between some pairs of nodes. For CBIS this means that the routing reconciliation need only be run opportunistically (i.e., when a node connects or is connected to for some other reason), so there is no "background" control traffic as in peer-based routing or mobile ad hoc networks (MANETs). Even though most of the nodes do not talk most of the time, the information propagates virally to all members of the transitive closure of the node contact graph, and they will be synchronized as quickly as physically possible.

While CBIS Sync-based routing gets rid of almost all the overhead associated with conventional mobile routing approaches, a scaling problem remains. Internet routing has scaled up to planetary size because IP prefixes have been assigned in a way that allows hierarchical aggregation — even though there are four billion addresses in use, the default-free transit core can connect them all with routes for only 250,000 prefixes. Since CBIS routing entries ultimately tie to cryptographic identities for individuals, it is not possible to impose an IP-like hierarchical structure. However, to have well founded privacy, identity, and trust, CBIS requires that entities who want to share information first participate in a lightweight "enrollment" procedure that results in each certifying information about the other. This enrollment information is part of the CBIS routing collection and essentially defines a "social graph" containing all the peers with which some entity is capable of sharing data. Since the purpose of CBIS routing is data sharing, this graph defines exactly the set of useful routing information. The routing name space was designed such that each peer group (social graph) forms a subtree of the CBIS route data. Sync was designed to work on any subtree of the collection name space, not just the root; thus, reconciling differences consists of computing the intersection of the two sets of peer groups, then syncing the subtrees in the intersection. This technique guarantees that any node's routing information is bounded by its enrollments. Since the enrollments represent the social sharing network of the entity, and for humans many studies have found that such networks contain at most 100–300 entities (Dunbar's Number [6]), the CBIS routing state burden is modest.

Measured Data Sharing Performance

We have implemented CBIS and run it on Linux machines and Android phones. A number of test and demonstration programs have been written using it. Figure 3 shows a test comparing a CBIS photo-sharing application to the equivalent implemented using a conventional client-server model. We used three Ethernet-connected Linux desktops to emulate the custodians/gateways of three distinct households, each with its own WiFi. The number of mobiles in each "house" was varied between 1 and 7 Samsung Galaxy-S phones for a total of up to 21 mobiles. One phone acted as the content source, sending a 4MB photo. The other phones all simultaneously requested that content. We plot the number of phones per house vs. the average time it took the phones to download the content. To minimize the differences between the two tests, the "baseline" case uses CCN transport but not CBIS — all data comes from the phone sourcing the picture. Each test was run twice. Each result is the average over all the phones involved in the experiment in both runs of the experiment (phones_per_house × houses × runs). The variation between runs was negligible.[2] It is clear that by 5 phones/house (15 total), the source is completely saturated and the download time increases linearly with each added phone. In the CBIS

[2] *The anomalous data point at three phones per house (nine phones total) is due to the wireless AP used to simulate "house 2." It gave consistently lower throughput (higher transfer times) than the other two, but scaled the same way as them (and so did not effect the trend line) except at this one point. The different behavior was not noticed in time to be investigated.*

case, the phones request the content from their local gateway, which then requests it from the phone sourcing the picture. At the time this test was run, the CBIS custodian priorities were not implemented so the three gateways went straight to the phone rather than all going to the source phone's gateway, which would then pull only one copy from the phone. Even with incompletely implemented routing, the source never saturated and latency decreased monotonically as late starters found that earlier starters had essentially caused their gateway to prefetch the content.

Peer-based Routing vs. CBIS Sync-based Routing

Figure 4 shows the results of an experiment where six phones are incrementally enrolled in a household, one roughly every five minutes. We plot the observed traffic captured by a wireshark packet trace. The red line shows the total bits sent (Interests plus Content Objects) vs. time for a routing protocol similar to normal Internet peer routing – each phone discovers every other phone and enumerates all its control state. Almost all the traffic is Interest packets that each node sends to each of its peers to probe whether that peer's state has changed. With n phones, each probing $n - 1$ peers, the traffic scales as $O(n^2)$. The black line shows the total bits sent for CBIS sync-based routing. The line makes a step up at each new enrollment, reflecting the control data from the new phone, but rapidly flattens (a horizontal line indicates no data on the wire) as the control data at all the nodes gets in sync. Thanks to viral propagation, each node syncs with a small, fixed number of other nodes, so the total traffic scales at worst linearly with the number of nodes, and the per-node traffic is small and constant.

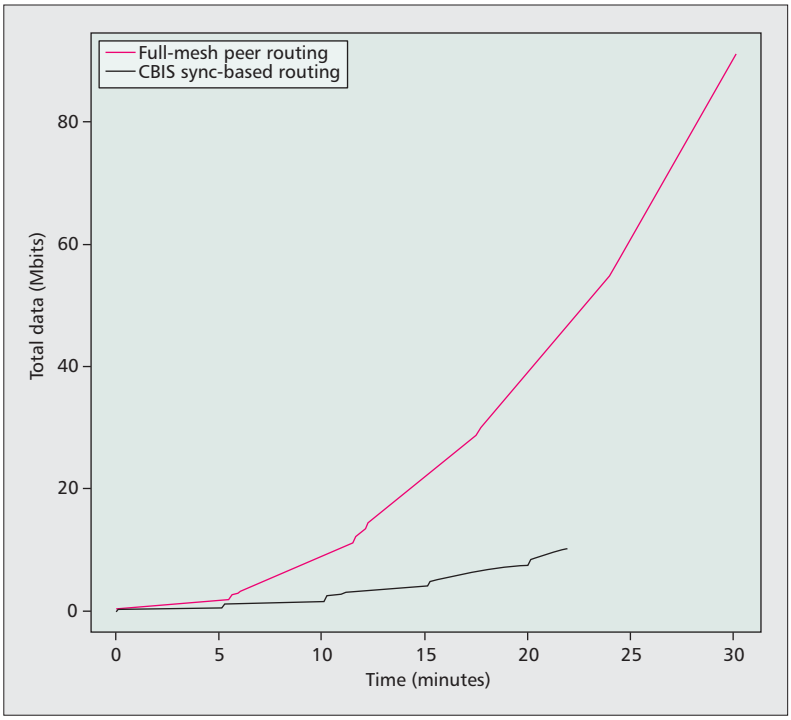

Figure 4. *Routing traffic scaling as number of nodes increases.*

Conclusion

Since information-centric networks interface with applications in terms of what information is wanted, where and how to obtain it become new degrees of freedom for the networking subsystem. CBIS is one example of how exercising this freedom can produce new routing protocols that perform better than the best possible with today's Internet routing. At the application and user level, it becomes possible to have a network that tolerates high levels of mobility by seamlessly managing multiple ways of communicating, each with context dependent, intermittent or partial connectivity. At the implementation level, the better division of labor between routing and forwarding allows for routing protocols that have much lower overhead yet detect problems faster and converge instantly. Transport innovations like Sync replace heavy weight, high-overhead peering establishment with light weight set reconciliation, allowing routing to function efficiently over mobile-friendly broadcast media and/or opportunistic viral propagation. Viewing transport as Sync also allows special structure in the routing information, such as the CBIS social graph, to easily be used to dramatically improve the scaling and communications overhead. Although not discussed in this article, while working on CBIS we also observed that an ICN approach was equally successful when applied to problems in security and application information transport. We have found that ICN is a new, exciting, and effective way to deal with today's communication problems.

References

[1] V. Jacobson *et al.*, "Networking Named Content," *Proc. CoNEXT '09*, Dec. 2009.
[2] E. Mutafungwa *et al.*, "Strategies for Energy-Efficient Mobile Web Access: An East African Case Study," *Africomm Conf. 2011*.
[3] D. Eppstein *et al.*, "What's the Difference? Efficient Set Difference without Prior Context," *Proc. SIGCOMM 2011*.
[4] N. Javaid *et al.*, "Evaluating Impact of Mobility on Wireless Routing Protocols," *IEEE Symp. Wireless Technology and Applications*, Sept. 2011.
[5] J. Kim *et al.*, "Content Centric Network-based Virtual Private Community," *IEEE Int'l. Conf. Consumer Electronics,* 2011.
[6] R. I. M. Dunbar, "Neocortex Size as A Constraint on Group Size in Primates," *J. Human Evolution*, vol. 20, 1992.
[7] C. Alaettinoglu, "Analysis of RIPE/RIS Project's BGP Data," *NANOG 23*, Oakland, CA, Oct. 2001;t http://www.nanog.org/meetings/nanog23/presentations/cengiz.pdf