

CacheMule: a Mobile-Carried Content Location Mechanism

Cedric Westphal

Innovation Center, Huawei & Dept. of Computer Engineering, University of California, Santa Cruz
cwestphal@huawei.com & cedric@soe.ucsc.edu

Abstract—We consider the abstractions that are required in an information centric network. The usual publish and subscribe primitives need to be extended with some content location additions. Indeed, one of the main challenges of information centric networks is to keep the content routing consistent, and this would be eased by offering content location support in the protocol to get and push data. The two primitives we suggest are a content location message that requests not the content, but its availability, and a content hint message that lets the consumer specifies where to find the content to the network. Both are needed to cope with the uncertainty of the content location that comes from the huge scale and the highly dynamic nature of content routing tables.

I. INTRODUCTION

Content-centric networks [1], [2], [3], [4] allow intermediate routers to store data; as a consequence, a file might be stored in multiple locations. Taking advantage of the content cached in the network involves locating the copies of the content. This can be done in an *opportunistic* manner, by "running" into a copy of the content on the way to the publisher's servers; but it might also require updating the (content) routing table in the routers so that they are able to direct requests for content to the nearest copy.

The complexity and scale of updating routing tables for a large amount of content might be prohibitive in practice. The current Internet is said to be holding 10^{15} pieces of content[5], and growing. Aggregating content by names allows to route on (name) prefixes, but routing data to its nearest copy would run counter to aggregation as it would detach specific items from their default routing prefix.

It might be possible for a domain to keep track of which content is stored within its administrative boundaries. However, exchanging this information across multiple domains would require some synchronization protocols which can provide a *strong* consistency only at the cost of a high overhead.

It might also be possible to place content over a restricted subset of the network topology, to reduce the scope of searching and make the content resolution scalable. However, this goes against the principles of a content-centric architecture where content could be located anywhere.

Two consequences thus arise: 1) The knowledge of the location of a piece of content (other than its default location at the publisher's server) is thus relatively precious for the consumer of such content: finding cached content in the network is difficult, unless the routing mechanism is able to dynamically update in real time a list of 10^{15} items. 2) The end user needs a mechanism to obtain this location information, in particular to be able to request the location of a piece of content rather than the content itself.

A node's mobility would often disconnect the consumer of the content from the copy it is downloading, streaming, or using. Even in the case of a static node, the user might be connected through several interfaces over different network who have different views of where the copies of the content is located.

We propose here a method we call CacheMule to expand the primitives exposed by an Information-Centric Network to: 1) assist with identifying the location, using a location probing mechanism; and 2) attach the known location of the content as an attribute of the content request as an indication for the network to route to the content. The former will assist with selecting proper interfaces for getting the content, will help with network debugging and can be viewed as a content-centric form of ping. The latter will assist session continuity in mobile environment.

We base our description on the CCN [2] architecture, but we hope that the basic primitives are applicable to other ICN architectures. Within this context, our proposed solutions are extension of the *interest* semantics and require no new control plane and are forwarded in-band.

The paper is organized as follows: in Section II we briefly go over some relevant related works and describe some CCN background as well as our architecture model, before introducing our CacheMule proposal in Section III. In Section IV we discuss the implication of our proposal as well as provide a cursory evaluation, before finalizing the paper in Section V.

II. RELATED WORK & MODEL

Related Work: Content-centric network proposals [1], [2], [3], [4] allow the network to store data, and to perform routing by the content's name. This creates a content routing structure at the router which contains either the path to the content in the case of a content FIB (Forwarding Information Base), or a copy of the data, in the case of a cache or content store.

Exchanging the information at the routers can be traced back to web caches [6] up to current effort to share information in between CDNs [7] or to synchronize content routers (see [8] for such a efficient sync protocol for content routers located near each other).

[5] consider the issue of scale in content routing, and begin to address it by proposing scale-appropriate data structures and routing protocols. Other works [9], [10], [11] present analytical models to predict the scaling behavior of information dissemination in large networks. However, the issue of consistency in the routing tables [12] needs to be addressed, and there is a trade-off between the routing performance and the rate of synchronization in between the different systems.

Our approach in this paper takes a different tack, namely that content routing cannot be left exclusively to the network. While the network must have final decision on where to get the content from, the end user might be able to assist the network, by carrying some information with her. In particular, since the content usage patterns of a single user are self-correlated, a user of content is more likely to have accessed this content in the past, and can speed up the resolution of the content location by sharing this history with the network. Our work attempts to add new primitive to existing architectures, and is not unlike [13] in this regard.

Model: We assume a content-routed network architecture, where a node request data by its name. Examples of such networks include NetInf [4], PURSUIT [3], etc. However, unlike in CCN/NDN, we assume that the content is mapped to a location. This can be achieved in CCN/NDN by assigning a prefix to a specific location, say assigning a globally unique identifier to each potential cache with the proper NDN/CCN naming semantics, and appending the content name to the cache identifier¹.

We assume a file (or an exchange) is composed of multiple chunks (or segments) and that several round trips are required to complete the transaction (namely, downloading a file or completing some service).

A wireless (mobile) node attaches to the network using an access point. In an ICN, it does not (necessarily) have to configure a network address, and it can use multiple interfaces, so we cannot assume that the node knows which AP it connects to, nor which corresponding node or nodes it is getting data from. We contend that the cache location information is valuable, and should be included in the requests for the subsequent segments of a file, in order to assist the routing layer in resolving the content location.

III. CACHEMULE

The consumer will request data using a GET(name) primitive, for instance, by sending an interest packet. In return, it will receive the object, using a SEND (or PUT) primitive (for instance, a data packet in CCN). We propose to extend the GET primitive by adding a location hint. The primitive thus becomes GET(name; location hints) where name describes the requested object, and location hints is a list of known network locations that hold this object.

In return for a GET request (say, an interest, or a subscribe message in a pub/sub infrastructure), the network responds with a SEND(name, location) message which is composed of the data (or some part of the data, if the data is segmented over several chunks) and the location. This allows the receiver of the data to compile a list of potential locations where the data could be located. This is illustrated on Figure 1.

¹For instance, a file such as *www.foo.com/videos/cat.mp4* would be copied at the cache with name (address) *xyz* as: *xyz/www.foo.com/videos/cat.mp4*. Note that in this case, the only routing that is necessary is that to *xyz*, as a prefix for the full name is sufficient. As such, the prefix *xyz* becomes the location of the content, and allows us to make the distinction between the content name and its location. The prefix *www.foo.com* is of course the location of the publisher of the content.

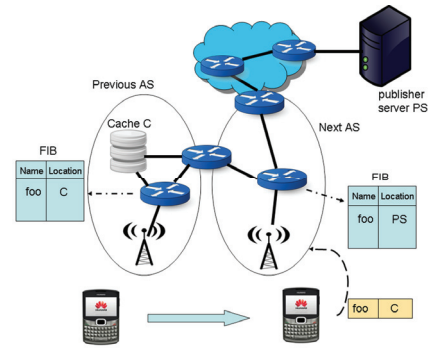


Fig. 1. A mobile node can carry location information into the next domain it connects to.

In addition to the attributes described in other ICN proposals, we also append some attributes to the GET request, such as: a "listed location only" (LLO) attributes, to force the routing of the request to one of the specified locations in the location list; and "probe" (P) attribute to fetch only the locations that could answer to the request for a specific name, but does not return the actual data.

The LLO attribute is similar to the "origin only" attribute in CCN, with the caveat that it could be any location specified by the requester, and not only the original publisher's location. We believe those primitives are necessary in order to offer a smooth performance in many scenarios.

IV. SCENARIOS AND EVALUATION

Location hint mode: As motivated earlier, we cannot expect the network to know the location of every copy of an object being accessed by a mobile node. Absent the location hint, the network could behave as follows: a mobile node MN accesses an object from a local cache LC. The MN changes access points, leaves the previous access router (PAR) and connects to a new access router (NAR). In order to retrieve the next segment of the object, it issues a GET(name) for the next segment. The new AR will need to: 1) resolve the name into a location; 2) fetch the object from this location. The only location that we can expect the new AR to know is the object's original location OL at its publisher's server. The routing path was: MN-PAR-LC and thus becomes MN-NAR-OL. The performance degradation can be substantial if the OL is further away from the MN.

If the GET carries the location LC, the MN would request GET(name;LC) and the NAR could then make a decision based on: the reachability of LC from NAR, the cost of doing so, the processing time to resolve OL from the name, and the comparative cost of routing to OL and NAR.

The NAR could even send a GET(name;location;P) probe where location is any potential content location, including OL and LC to in essence ping the content. Based upon the response to a probe packet, the NAR can then make a routing decision to retrieve the data.

The NAR could also decide to bicast the traffic to both LC and OL to assess the liveliness of each path and empirically

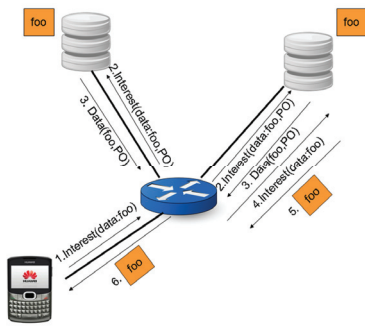


Fig. 2. A probe only interest avoids duplicate request and unnecessary data transfer from the cache to the router: only one interest will be satisfied, rather than both.

compare. This would achieve the fastest handoff recovery, since it would place the known copies of the content (at LC and OL) in competition with each other.

Note that knowing explicitly where the content is located is a big advantage in this scenarios, as this allows the MN to inform the NAR of where to look-up. NDN for instance does not consider content location, and thus cannot avail itself natively of taking advantage of copies that are known in some part of the networks, but unknown in others (the custodian-based information sharing [14] offers to some extent a mechanism to locate content).

Probe-only Mode: We hinted at the benefit of the probe-only mode in the previous section. Content routing requires different network debugging tool compared with traditional IP networks. One can assess the liveness of links on a link-by-link basis. Each router/switch can ensure that the interfaces it is connected by are up and running. For multi-link paths, there are no systematic tools to debug.

In particular, since the location of the content might evolve in time, there is no predictability of the path taken by a packet. Consider the following sequence. A node downloads a piece of content from server A. One of the routers might not function properly. However, an ICN protocol would find another copy of the content. The network administrator would thus not immediately notice the failure of the router. There is no ability to test a specific *path* in such an information centric network.

We suggest the use of a *Probe-only* mode to test the liveness of a path. The probe-only is a specific type of packet which intends to test the network capability without adding undue overhead by having the network send some traffic and consume the resource.

We thus propose to expand our GET primitive to support a PO mode. This PO mode is a flag in the request for the content (for instance, an option in the parameters of the GET(.) function). Upon receiving a request for content with the PO flag, an intermediate router (namely, which cannot answer with the data) forwards the packet according to the route resolution performed upon the request content's name. An endpoint (namely, a router which can answer with the data)

responds with an empty file, and with its own location, using the SEND(.) primitive we described in the previous section. To respond, the content router needs to hold the data, but does not need to send it.

The SEND(.) primitive is extended with its PO mode as well, which means that the node which responds to the GET will set this flag to PO, and include the content name, the information included in the GET (depending on the ICN architecture, this could include a nonce to identify the GET request, and other parameters), its own location, but no data.

The PO mode may request a signature of the data to be forwarded to ensure that the responder actually holds the data. This is described in Figure 2

Routers forward back the stream created by the function SEND(name, location, PO, emptyfile) back to the routers or hosts which generated the GET request in the PO mode. If multiple responses are received from different nodes (but with different locations), then these copies are all forwarded back. However, only one response from a given location is forwarded back to the requester. The router can also use these responses to populate its FIB.

There is a significant difference with basic CCN, as the processing of the data does not "consume" the interest, and the interest is kept until it times out even though it has returned some data, in order to forward different potential locations back to the sender of the probe.

The PO mode tests the path between a node and various copies of the content and returns a list of known location of the content. Note that this list is not exhaustive: there might be other contents that are not reached by the GET(.) message. However, if there is a copy of the content in the network and the network is able to properly route, then the GET(.,PO) should return at least one response.

Further, the PO mode can also be used to test a specific location. This is achieved by specifying this location in the GET(name,location,PO) parameters. The network then routes the packet to the location, and returns a yes/no answer as to whether the requested content is indeed at this location. This achieves a PING feature which is currently missing in CCN for instance.

LLO Mode: The "Listed Location Only" LLO mode is used to request content from a specific location. If the network is unable to reach the listed location, then the request is dropped. This can be used again for debugging purpose, by testing the specific route to the specified location. However, in this case, the response is the actual data, and the request/response exchange consumes network resource.

A combination of PO and LLO mode allows to first gather the potential content locations, and then decide which location to use.

Upon specifying several location, the network operator can optimize based upon its own incentive which location to select to deliver the content.

Evaluation: The benefit of the location hint is hard to assess: it requires making assumption regarding the temporal self-correlation of the traffic from a user, the spatial correlation

of the requests within a given network, the hand-off patterns between two networks, and the likelihood of caching in each network and the round-trip time to a cache within the previous network and the new access point vs the original server and the new access points. We feel that in many scenarios, the ability to locate the content from the mobile node will be useful. In particular, it should be obvious that a user leaving a corporate network or a home network to join a wide area network might lose connection to a local copy, and that the WAN might not be aware of all the files contained within such home or campus network. In such scenarios, assisting the network with the file location will bring a significant benefit.

For the probing scenario, we go back to Figure 2 and observe that, if a regular interest is sent at the intermediate router, and both storage possess the requested content, then both will respond with the content; per CCN specification, the intermediate router will forward one data object and drop the other. The transfer of one of the copies from one of the storages to the intermediate router thus end up wasted.

We can then quantify the gain of having a PO mode, which will return the location of the copies in the cache. If both caches have the data, then one cache will be selected and only one copy will be fetched (note that an alternative strategy would be to fetch multiple segments from the multiple copies, potentially using network coding, as in [15]). Thus, in the case of Figure 2, only two data transfers will occur with the probe mode, vs. three with regular CCN.

Files are requested by the user according to a Zipf distribution out of 100,000 files. We generate 100,000 requests. Under a vanilla CCN strategy the router forwards the interest to both caches. If both copies have the content (which happens with probability p), both respond with the content. At least one cache has the content and responds. When the intermediate router receives a piece of content, it keeps it in its cache according to a LRU replacement policy.

Figure 3 depicts the consumed bandwidth for various cache sizes at the router (LRU policy, $p = 1$). The larger the cache, the less bandwidth is consumed. The probing mode keeps the bandwidth consumed constant, while as the duplicated interests find copies of the requested pieces of content with higher probability, the wasted bandwidth increases.

Figure 3 also displays the bandwidth saving of probing vs. the probability that the content is duplicated in both caches (with cache size at the middle router set to 100). As the duplicated interests find copies of the content with higher probability, the wasted bandwidth increases.

V. CONCLUSION

We have presented some network primitives that are necessary in an ICN: once a (potentially mobile) node resolves the location of a piece of content, it should share this location with its next networks of attachment; an in-band mechanism to locate content without triggering a data transfer is required and useful for, among other uses, network debugging. We have demonstrated that locating content before initiating a data transfer could significantly reduce the bandwidth usage.

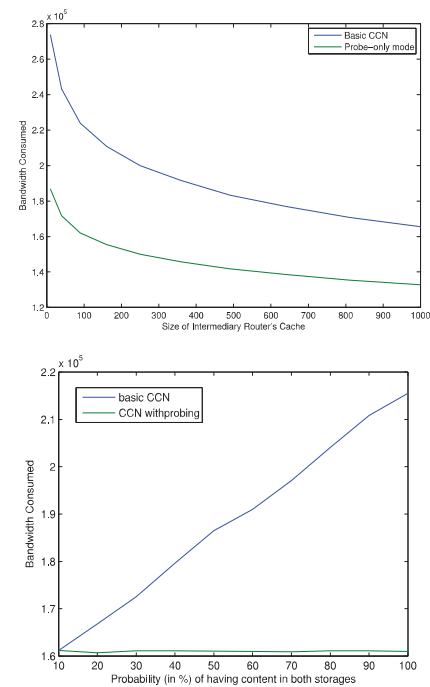


Fig. 3. Top: Bandwidth consumed vs Size of the Cache of the Intermediary Router; Bottom: Bandwidth Consumed vs Probability p of Finding the Content in both Caches

REFERENCES

- [1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Comm. Mag.*, vol. 50, July 2012.
- [2] "Named data networking." <http://named-data.org/>, Aug. 2010.
- [3] "PURSUIT: Pursuing a pub/sub internet." <http://www.fp7-pursuit.eu/>, Sept. 2010.
- [4] C. Dannewitz, "NetInf: An Information-Centric design for the future internet." Proc. of 3rd GI/ITG KuVS Workshop, May 2009.
- [5] M. Djalalo, S. Fdida, and P. Spathis, "Towards extreme scale Content-Based networking for the next generation internet," *IEICE Transactions on Communications*, vol. E94-B, no. 10, pp. 2706–2714, 2011.
- [6] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," *IEEE/ACM Trans. Neww.*, vol. 8, pp. 281–293, June 2000.
- [7] L. Peterson and B. Davie, "Framework for CDN interconnection." IETF CDNi Working Group, draft-ietf-cdni-framework-02.txt, Dec. 2012.
- [8] C. Westphal, "Synchronizing state with strong similarity between local and remote systems," in *Proc. ACM MobiSys MCS'12*, 2012.
- [9] C. Westphal, "On maximizing the lifetime of distributed information in ad-hoc networks with individual constraints," in *ACM MobiHoc'05*, 2005.
- [10] C. Westphal, K. Seada, C. Perkins, and R. Wakikawa, "An epidemiological study of information dissemination in mobile networks.," in *Proceedings of IEEE SECON '09*, pp. 1–9, June 2009.
- [11] E. J. Rosensweig and J. Kurose, "Breadcrumbs: Efficient, Best-Effort content location in cache networks," in *INFOCOM 2009, IEEE*, pp. 2631–2635, IEEE, Apr. 2009.
- [12] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized?: state distribution trade-offs in software defined networks," in *ACM Sigcomm HotSDN'12*, 2012.
- [13] L. Popa, A. Ghodsi, and I. Stoica, "HTTP as the narrow waist of the future internet," in *ACM SIGCOMM HotNets'10*, 2010.
- [14] V. Jacobson et al., "Custodian-based information sharing," *Communications Magazine, IEEE*, vol. 50, pp. 38–43, July 2012.
- [15] M.-J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets Information-Centric networking," in *NOM Workshop, ACM MobiHoc'12*, June 2012.