An IP-based Manifest Architecture for ICN

Cedric Westphal
Huawei & UCSC
Santa Clara, CA, USA
cedric.westphal@huawei.com

Emrecan Demirors
Northeastern University
Boston, MA, USA
demirors.e@husky.neu.edu

1. INTRODUCTION

Information-Centric Networks [1–4] expose content information to the network layer. CCN, NDN and others want to replace IP as the narrow waist of the Internet. In [5], it is argued that most of the benefits of ICN can be achieved in IP through modification of DNS, and by adding a new *content record (CR)* type to the records returned by a DNS server. This CR type allows the DNS server to respond to request for Named-Data Objects (NDO) or name prefixes. The CR type is a list of IP addresses associated with the NDO.

We argue that this does not go far enough, and that the DNS could return more information about the object, namely that the DNS should return a manifest that describes the object properties. The manifest is an object (say, in xml) describing the properties of the content object that are relevant at the network layer. In particular, the manifest includes the location of the object, some security properties and some information regarding the transport of the object, including for instance its size.

Because the manifest is part of a DNS transaction, it is exposed to the network without the need of Deep Packet Inspection (DPI); because the DNS manifest is a network-layer object, it embeds only information that is exposed to the network layer, and therefore is not encrypted, or is encrypted using material that is shared with network operators.

This allows the network to observe the manifest as it is being requested, and to operate on this manifest. The typical envisioned usage is at the edge network: the client attaches at the edge network, and request a specific file to the DNS server, requesting a manifest. The DNS server (denoted DNS++) returns the manifest; the edge network is therefore informed of the content being requested by the client, and of the properties of the content.

This information can be either used and/or modified. Using the information in the manifest allows the network to accommodate certain properties of the data object. For instance, the manifest includes the size of the data, and therefore the edge network can make a different path selection decision depending on the content being an elephant flow or a mice flow. There is prior work on making resource allocation and traffic engineering decision based upon content properties [6–8].

ICN'15, Sept.30–Oct. 2, 2015, San Francisco, CA, USA.
ACM 978-1-4503-3855-4/15/09.
DOI: http://dx.doi.org/10.1145/2810156.2812614.

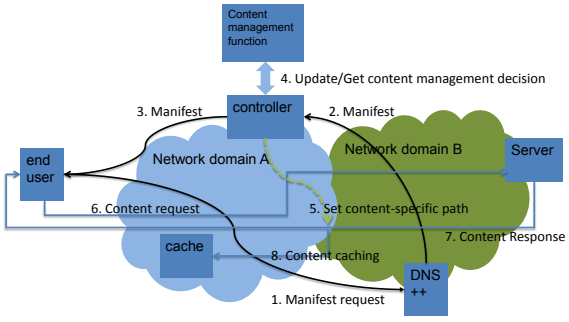

Figure 1: The Demo Architecture -all elements are implemented as VMs on two server blades

Modifying the manifest allows the network to include known copies of the content that reside in a local cache for instance; this of course implies that the manifest can be updated in a trusted manner by the network. Our goal is not to describe such trust mechanism, but rather to highlight the potential benefits of a manifest.

Figure 1 presents the envisioned architecture, where an end-user (or client) will request the manifest from a DNS++ server, receive the manifest in return; the manifest is observed by the edge network on the way back before being returned to the client. The edge network can take some action based upon the manifest.

Note that manifests exist already in many contexts, including DASH MPDs [9]. The idea of notifying the network prior to a transmission is implied in many current network architecture, including SDN/OpenFlow. There, the first packet serves both as a packet within the data exchange, and a notification to the network about the incoming flow. A manifest makes this notification explicit, with the advantage of sharing object/flow properties with the underlying network (unlike an application layer manifest such as in DASH).

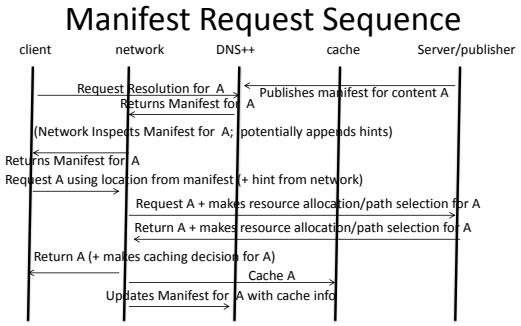

Figure 2: The messaging flow

2. DEMO SET-UP

We have implemented a DNS++ server which returns a manifest. This involves modifying the DNS++ client of the end-user and the DNS++ server. The DNS++ client needs to format the requests differently to include the whole name of the object, or the name prefix. The DNS++ request also includes the source address, as the network needs to know which client is making the request for path optimization. The DNS++ server is modified to return a manifest if it holds one for the request of a manifest record type. If it does not hold a manifest, it returns the A record corresponding to the domain name, as in a typical name resolution.

We have also implemented an edge network that listens for manifests on the DNS port 53 and is able to take a corresponding action based upon the upcoming data transfer. Figure 2 shows the data flow of the demonstration. All these elements have been implemented and will be demonstrated. The demo flow will be as follows: the client will request a file manifest from the DNS and in return, will receive the manifest. The intermediate network will observe the manifest.

- In Scenario I, it will make a routing decision based upon the size of the object mentioned in the manifest. A small object will be returned following one path, while a large object will be returned according to a different path.
- In Scenario II, the edge network will insert the address of a local copy into the manifest, so that the client can download the file from the local cache.

The demo set-up will be hosted in the Huawei data center in Santa Clara, and will be controlled remotely from the demo site. The demo equipment include a server holding virtual machines (VMs) for the server, the DNS and a virtual switch and another server with VM for the cache, the client and the network controller. Both servers will be connected over two distinct links on different ports on the physical switch connecting both servers.

3. DISCUSSION

Defining the proper manifest and what properties to include is an on-going task. There is a tension in making the manifest expressive, but at the same time, in keeping it simple. How to scale the manifest is another issue: small objects do not need a manifest, and the advantage of a native ICN architecture is that they can be fetched directly from any intermediate cache as there is no binding of the session to a destination. There is also a concern of getting a per-object manifest, as certain content may contain many objects. For instance, a web page, or a facebook or twitter-like application, will include many pictures and referral to other objects such as advertisement embeds. While it takes only one DNS resolution step currently, per object resolution would dramatically increase the number of such steps. It could even be a new DDoS avenue to generate objects referring to many more objects.

On the other hand, the manifest of the parent web page could include information regarding the other objects. Furthermore, it could also include information on which manifests to get, and which not to (say, if all embeds are small, none of them would require a manifest resolution). This is also a topic of further study.

4. REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Com Mag, IEEE*, vol. 50, no. 7, pp. 26–36, July 2012.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09, 2009, pp. 1–12.

[3] A. Chanda and C. Westphal, "ContentFlow: Mapping content to flows in Software Defined Networks," in *in Proc. of IEEE Globecom*, Dec. 2013.

[4] B. Azimdoost, C. Westphal, and H. R. Sadjadpour, "On the throughput capacity of Information-Centric Networks," in *Proc. International Teletraffic Congress 25*, Sep. 2013.

[5] S. Sevilla, P. Mahadevan, and J. Garcia-Luna-Aceves, "idns: Enabling information centric networking through the dns," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, April 2014.

[6] A. Chanda, C. Westphal, and D. Raychaudhuri, "Content based traffic engineering in Software Defined Information Centric Networks," in *Proc. IEEE Infocom NOMEN workshop*, Apr. 2013.

[7] K. Su and C. Westphal, "On the benefit of information centric networks for traffic engineering," in *IEEE ICC Conference*, Jun. 2014.

[8] J. Yichao, Y. Wen, and C. Westphal, "Towards joint resource allocation and routing to optimize video distribution over future internet," in *IEEE/IFIP Networking Conference*.

[9] R. Grandl, K. Su, and C. Westphal, "On the interaction of adaptive video streaming with content-centric networking," in *IEEE Packet Video Workshop*, Dec. 2013.