Issues with Local Naming at a Global Scale (Abstract)

Wes Hardaker

October 26, 2016

Abstract

Today's Internet naming system is founded principally on the notion that a human needs to communicate with a remote, well-known server. The Internet's Domain Name System (DNS) works extremely well in that context; for example: a user Alice can simply type *www.google.com* into their web browser, or bob@comcast.net into their E-Mail software and "magic happens" in her eyes as the web page is displayed or the E-Mail is delivered. But if you ask Alice to share a file with Bob, or print to her printer when she's not at home those tasks also require a centralized server infrastructure when they otherwise shouldn't. The root of this problem is grounded in how Internet naming is constructed today.

The Internet's current naming system (the DNS) falls short of meeting the needs of peer-to-peer and peer-to-device based connections. Alice has no way to simple "send this file to Bob's cell phone", primarily because Bob's phone doesn't have a stable name. Similarly, "send this document to the printer in the den" is equally as impossible, especially when Alice is not within the same network where *mDNS* [2] no longer works. The result is that Alice must fall back to sharing files via cloud-based services like *DropBox [3]*, printing via *Google Cloud Print [5]* and even configuring the thermostat across the room using external middle-ware services like *EcoBee* [4]. (Note that technologies exist to send files to near-by devices, such as over bluetooth, the Wifi-Alliance's *Wifi-Direct [8]* or Apple's proprietary *AirDrop [1]*. However, none of these are cross-platform.) None of these scenarios actually require a round-trip communication through the cloud, but they all do go through the cloud because personal devices and IoT devices have no ability to negotiate long-lived local or global names. Though technologies like dynamic DNS [6] exist, they've had near-zero uptake at the end-user level, don't handle naming conflicts without user intervention and security [7] is especially hard to set up properly without expert-level knowledge. While the cloud often works well today, in the long run it has greater cost, latency and a wider potential attack surface. Furthermore, reliance on a cloud service company means relying on the long-term viability of that company and product line. Now is the ideal time to explore augmenting or reinventing the naming system to allow for the creation of dynamic but long-lived local and global names that can be used both locally and at a distance and without requiring connections to cloud based services. The largest issue with peer-to-peer and peer-to-device based connections is not the communication protocols (with the increasing deployment of IPv6), but rather long term naming conventions. Whether built upon the existing DNS or not, peer-to-peer and peer-to-device naming needs to be addressed for the upcoming IoT era to be truly successful. We would like to research solution spaces for this problem, and how they might make use of DNS or integrate with DNS.

References

[1] Apple. Send content from your mac. https://support.apple.com/en-us/HT203106.

[2] S. Cheshire and M. Krochmal. Multicast DNS. RFC 6762 (Proposed Standard), February 2013.

[3] DropBox. Drobbox. https://www.dropbox.com/.

[4] ecobee. ecobee. https://www.ecobee.com/.

[5] Inc. Google. Google cloud print. https://www.google.com/cloudprint/.

[6] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard), April 1997. Updated by RFCs 3007, 4035, 4033, 4034.

[7] B. Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC 3007 (Proposed Standard), November 2000.

[8] Wifi-Alliance. Wifi direct. http://www.wi-fi.org/discover-wi-fi/wi-fi-direct.