EDITORIAL

Musings

RIK FARROW

Rik is the editor of *;login:*.
rik@usenix.org

This time around I thought I would write about the future of the Internet. Please note the capital "I," as that's what you will find in the Future Internet Design (FIND) final report [1], where the authors suggest strategies to the NSF for funding research into networking.

That initial conference took place in 2009 and looked at 49 projects. One outcome of the NSF NeTS FIND Initiative [2] was to continue funding several of the projects. I was vaguely aware of this work, but I also wondered how in the world anyone could hope to change the Internet, the system of networks we've all grown to rely upon—really, to depend upon—at this point in time.

On the economic side, there is the issue of sunk costs: companies have spent billions creating the network we have today. Then there is conservatism: people have learned (at least enough) to work with TCP/IP, with all its quirks. And, finally, any new protocols will require hardware support, and that's the issue I found worried the people whom I talked to about the NSF project I chose to focus on.

Named Data Networking

I didn't pick Named Data Networking (NDN) out of a hat. kc claffy had just 45 minutes to introduce some of the concepts behind this protocol during LISA15 [3], and I had heard something about NDN earlier. I think it was kc's mention of the importance of security that got me interested. If you read the FIND report [1], you will also see that security often gets mentioned *first* in lists of desirable new features in future protocols. But NDN is about a lot more than just supporting security over network traffic.

NDN comes out of research done by Van Jacobson and others at Palo Alto Research Center (PARC) [4] in 2009. The authors of that paper created a protocol called Content-Centric Networking (CCN), largely because of the realization that then-current Internet traffic was mostly about shared content. Today, streaming video (content) makes up close to two-thirds of all Internet traffic, making the notion of a network focused on content even more relevant.

The NDN researchers started with many of the ideas expressed in the CCN research to create a new protocol with similar goals. The very name, Named Data Networking, hints at the key ideas.

Today's Internet, based on the Internet Protocol (IP), relies on binary addressing for point-to-point communication. We start with DNS names, DNS provides the binary addresses (although we generally think of them as four decimal bytes separated by dots), and communication is between a pair of endpoints. Point-to-point communication made a lot of sense in the 1970s, when computers were rare and just connecting a computer to a shared network required the use of a mini-computer, called Interface Message Processors, IMPs [5], a 16-bit computer the size of a refrigerator, not including its console. The computers that connected to the ARPANET were multi-million-dollar machines themselves. You could say that the world then (just 40 years ago) was very different. Researchers really wanted ways to share data and remotely log in in those days, and those two goals were the focus for designing TCP/IP.

Today, over a quarter of the world's population uses the Internet, and what they want from it is content. *Named data* refers to the requests for data in NDN, called *Interests,* which look a lot like URLs in a RESTful interface. Naming is hierarchical, something that IP addressing has never managed to have, although IPv6 is better in this regard.

The responses to Interests are called *Content*, and the data in Content packets are signed by the source. Having signed data means you can trust that the data came from the source you are interested in, even if that data had been cached by a cooperating router.

Of course, signing relies on there being a secure method for sharing public keys, and secure sharing of certificates is also an important part of NDN. NDN plans on using a Web of Trust, where you have local roots for your own organization, but must trust other certificate signers for trusting certs from the greater Internet. The details of this must still be worked out.

The Hard Part

Well, I jest, because there are lots of hard parts. But one of the things that really caught my attention about this design is how much more involved routers will be in a network where NDN is the underlying protocol. In TCP/IP, IP is what network designers call the "thin waist." What they mean is that one relatively simple protocol, IP, is what is used to get packets delivered across the Internet.

NDN's thin waist are Interests and Content. Routers need to be able to interpret the names in Interests, decide how to forward those Interests, keep track of which port Interests arrived on (so they can return Content via that port), as well as cache Content. Compared to IP routers, that's a huge departure from the way things are currently done.

Since routers replaced gateways (like the IMP, and later Sun and DEC servers), routers started having special hardware that supported the *fastpath*. The fastpath represented the port pair for a particular route and avoided having to use the much, much slower router CPU to make routing decisions for each packet. The fastpath allowed parallel lookups, using Ternary Content Addressable Memory (TCAM [6]) to route packets. TCAM solved what was becoming the problem that would "kill" the Internet in the late '90s, when the number of routes was doubling every several years, requiring four times longer to look up routing information for each packet for each doubling in routing table size.

There aren't any TCAMs for names. In fact, parsing names using current hardware for routing seems like an impossible task today. But then, we faced a similar problem just 20 years ago with IP routing.

There are the other issues that would need to be solved, ones that we have not been able to solve so far, like a trustworthy means for distributing public key certificates. X.509 is itself a terrible protocol—just consider how often libraries for parsing X.509 have resulted in exploits, because X.509 is too ambiguous. We also have certificate authorities, like Symantec, having its root certificates banned by Google [7] because of abuse. And that's not the only case of CAs behaving as paper mills—producers of nice certificates for a fee—instead of identity authorities.

NDN runs over a UDP overlay today, but plans are for NDN to run natively some day. If we ever expect to replace cable with the Internet, we really need a way to stream popular entertainment, like sports events, in an efficient manner. And TCP/IP is not designed for streaming, while NDN would do streaming well, as its design easily and naturally handles multicasting.

The Lineup

We begin the features in this issue with Filebench, a project started within Sun Microsystems many years ago for benchmarking NFS. Vasily Tarasov, Erez Zadok, and Spencer Shepler explain how to use Filebench for benchmarking file systems. Filebench does include templates for several common uses, but the real power in Filebench is your ability to tune the benchmarks to your particular use cases.

Amandeep Khurana and Jayant Shekhar tell us about different systems for processing streaming data. They cover Kafka, Spark, Storm, and Flink, describing the strengths and weakness of each system, all of which add streaming over Hadoop-related architectures. Kafka handles data ingestion, where Spark, Storm, and Flink provide different approaches to analysis.

Jonathan Mace, Rodrigo Fonseca, and Ryan Roelke reprise their SOSP '15 award-winning paper about Pivot Tracing. Pivot Tracing adds metadata to requests in distributed systems on-the-fly, allowing you to monitor and debug these applications, much the way you would use DTrace or Systemtap on local applications.

I interview Doug McIlroy, who was a manager at Bell Labs when the UNIX system was being created. Doug is best known for his work in adding pipes to the UNIX system, but also wrote code from some tools that we still use today.

Arnaud Tomeï takes a comprehensive look at his experiences with creating portable shell scripts. While POSIX was all about creating a standard for UNIX-like features, Tomeï discovered many places where using features found in the most common shells and popular commands will get you in trouble when you try to write one script for multiple *nix systems.

Barclay Osborn, Justin McWilliams, Betsy Beyer, and Max Saltonstall provide another look at BeyondCorp, Google's project

to replace VPNs into sensitive networks with gateways over encrypted connection to services. Rory Ward and Beyer provided a view into this project in a December 2014 *;login:* article [8], and the authors update us on how the project has evolved, and what challenges have been overcome over the intervening year.

Mark Gondree, Zachary N J Peterson, and Portia Pusey share the work being done surrounding the issues of naming in the area of gaming for security education. Terms like Capture the Flag (CTF) have wound up being applied to games that have little to do with the original notion, and not having a standard terminology for styles of games hurts attempts at using gaming for any form of computer education that might take advantage of it.

I interviewed Lixia Zhang and kc claffy about NDN, the subject of this column. I recommend reading this interview and checking out the resources at the end of it so you can learn more about NDN. You might even want to try out some of the sample applications.

Dave Beazley tells us about a problem when using Python 3.5's new asyncio functions: you don't know what other functions will fail when you start using asyncio functions. Dave deftly describes this as the red/blue problem and provides some interesting Python function decorators as possible solutions.

David Blank-Edelman wants us to use Swagger, not an exaggerated way of walking but a Java-based tool that makes writing the code for APIs between client and servers a stroll in the park. Swagger includes code generators for many languages, although only for the client-side of Perl.

Dave Josephsen doesn't want you to be a hero. Dave refers specifically to Brent in the novel *The Phoenix Project,* the one person who can solve any problem, and thus the bottleneck to getting any IT project completed. Dave uses an example to demonstrate how things should work.

Kelsey Hightower introduces his column on Go for sysadmins, where he describes how to use RPCs to build a distributed tool that could be the basis for a monitoring system. Kelsey will be writing Go columns designed to help system administrators, and anyone new to Go, take advantage of one of the best-designed languages.

Dan Geer bets on growth over magnitude. When looking at the problems you will need to solve, do you choose the ones with the most current problems or the ones with the fastest growing list of issues? Dan explains his reasoning behind picking growth.

Robert G. Ferrell, inspired by my look at NDN, considers how he helped with organizing RFCs in the '90s, then ponders NDN, without naming it.

Mark Lamourine has just one book review in this issue. Mark writes about *The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age*. Like the author, Paul Nahin, Mark considers Boole and Shannon unsung heroes (the good kind) in the creation of computers.

In USENIX Notes, Dan Klein tells us why he has worked with USENIX—as education director, paper author, and now Board member—for over 25 years.

It has been said that pornography was the driving force behind the incredible growth of the Internet. During the 1990s, I would meet with UUnet employees at USENIX conferences and hear that since the last time we had seen each other, the size of the Internet had doubled. While I don't really have any idea whether this was because of pornography, attempts at streaming football games might have a similar effect on the introduction of new protocols in the Internet.

Fortuitously, while I was pondering this column, Bloomberg published a magazine article about how, if the NFL were to get serious about live streaming football games [9], they would need a different Internet. TCP/IP was designed for point-to-point transfer, not the one-to-many streaming that huge events require. And entertainment providers like Netflix now dominate Internet traffic. These uses, and more, could really benefit from new protocols like NDN.

Resources

[1] National Science Foundation, "FIND Observer Panel Report," April 9, 2009: http://www.nets-find.net/FIND_report_final.pdf.

[2] NSF NeTS FIND Initiative: http://www.nets-find.net/.

[3] kc claffy, "Named Data Networking": https://www.usenix.org/conference/lisa15/conference-program/presentation/claffy.

[4] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, Rebecca L. Braynard, "Networking Named Content," ACM CoNEXT 2009: http://conferences.sigcomm.org/co-next/2009/papers/Jacobson.pdf.

[5] Interface Message Processor: https://en.wikipedia.org/wiki/Interface_Message_Processor.

[6] Content Addressable Memory: https://en.wikipedia.org/wiki/Content-addressable_memory.

[7] Lucian Armasu, "Google to Remove a Symantec Root Certificate from Chrome and Android," December 11, 2015, Tom's Hardware: http://www.tomshardware.com/news/google-removes-symantec-root-certificate,30742.html.

[8] Rory Ward and Betsy Beyer, "A New Approach to Enterprise Security," *;login:*, vol. 39, no. 6, December 2014: https://www.usenix.org/publications/login/dec14/ward.

[9] Joshua Brustein, "How NFL Thursdays Could Break the Internet," Bloomberg Business, December 15, 2015: http://www.bloomberg.com/news/articles/2015-12-15/how-nfl-thursdays-could-break-the-internet?cmpid=BBD121515.