Networking Named Content

Van Jacobson, Diana K. Smetters, James D. Thornton, Michael Plass, Nick Briggs, and Rebecca Braynard

Presented by Ross Lagerwall

University of Cambridge

January 27, 2013
Networking is done host-to-host in a conversation fashion.
But our internet usage doesn’t fit this pattern.
We care about content, not location of the data.
Research has tried to use content as a first class citizen rather than location.

Previous work has used flat opaque names to data. These names are then mapped to user friendly ones.

TRIAD uses structured, user friendly names with a modified DNS server.
CCN tries to solve the problem by using user friendly names which describe content.

Routing is done on content rather than host location.

CCN defines Interest and Data packets.

To retrieve data, a node broadcasts an Interest.
**Interest packet**

- Content Name
- Selector (order preference, publisher filter, scope, ...)
- Nonce

**Data packet**

- Content Name
- Signature (digest algorithm, witness, ...)
- Signed Info (publisher ID, key locator, stale time, ...)
- Data
On receiving an Interest:

- reply with data
- reply with cached data
- store a breadcrumb in the pending interest table and forward it
Data follows the breadcrumbs back.
Breadcrumbs are replaced with a cached copy of the packet.
Interests can be pipelined similar to TCP’s window.
Sequencing is achieved by altering the structured name.
A limited form of querying is available with primitives such as RightMostChild.
The strategy layer is responsible for reliability.
It uses timeouts to resend interests.
The sender of data is stateless.
The strategy layer can also use multiple faces with a policy such as *UsageBasedCharging* or *PeakUseLimited*.
Sender will only deliver data once.
CCN can work with IP’s routing scheme.
Protection and trust are part of the content, not the connection.

Content packets are signed which binds the name, key and content giving integrity, pertinence and provenance.

Keys are retrieved as content which allows arbitrary trust models to be developed.

Example: hierarchical or peer-to-peer.
Implemented as userspace daemon and client library. For testing, CCN packets are encapsulated in UDP.
Evaluation

[Graph showing the comparison between CCN and TCP in terms of total download time vs. number of clients. The graph indicates that CCN performs significantly better than TCP as the number of clients increases.]
The paper is overly ambitious.

It tries to replace:

- IP
- TCP and UDP
- HTTPS and other application layer protocols
- PKI
Opinion

Provides a good solution to the original problem.
It seems to match up with HTTP GET requests for static data.
It does not seem like a complete replacement for the TCP/IP network stack.

What about SSH, VPNs, pushing data with online backup, gaming, etc?
What about unwanted caching?
What happens to all the application layer protocols?
What about broadcasting in a large busy LAN?
How does application demultiplexing work?
How does caching work with dynamically generated content?