the PARC CCN Team

- Jim Thornton
- Diana Smetters
- Russ Atkinson
- Simon Barber
- Rebecca Braynard
- Nick Briggs
- Tim Diebert
- Priya Mahadevan
- Mark Mosko
- Michael Plass
- Paul Rasmussen
- Elaine Shi
- Ignacio Solis
- Ersin Uzun
Named Data Networking (NDN) NSF FIA 3 year project

UCLA
UCIrvine
University of California, Irvine
UCSD
ARIZONA
Colorado State University
Palo Alto Research Center
University of Illinois at Urbana-Champaign
The University of Memphis
Yale University
University in St. Louis
Lixia Zhang (lead)
Jim Thornton (manager)
Deborah Estrin (advisor)
Van Jacobson (architect)
Tarek Abdelzaher
Jeff Burke
K Claffy
Patrick Crowley

Dmitri Krioukov
Dan Massey
Christos Papadopoulos
Gene Tsudik
Ersin Uzun
Lan Wang
Edmund Yeh
Beichuan Zhang
CCN offers ...

- painless mobility, wireless, virtualization, ...
- (provably) optimal content distribution
- simple, secure, robust configuration
- much better security
- same scalability & efficiency as TCP/IP
- an easy, incremental, evolutionary path
Design Criteria

1) A system that’s complete & self-contained.
   
   IP runs over anything
   
   Anything runs over IP
Design Criteria

1) A system that’s complete & self-contained.

   IP runs over anything

   Anything runs over IP

   CCN runs over anything

   Anything runs over CCN
Design Criteria

1) A system that’s complete & self-contained.

IP runs over anything
(that moves bits to some endpoint)

Anything runs over IP

CCN runs over anything
(that moves bits in space or time)

Anything runs over CCN
Design Criteria

2) An evolutionary path:
   - incremental deployment gives immediate value
   - painless, viral, bottom-up growth.
   - use existing infrastructure, including its lessons.
For 150 years ‘communication’ has meant a conversation over a wire connecting two devices:
For 150 years ‘communication’ has meant a conversation over a wire connecting two devices:

For consumers, the Web forever changed that. The information matters, not how or where you get it.
Diffusion / Percolation / Viral Propagation
Full-mesh Pt-Pt Connectivity abstraction

Topology creation
• Neighbor discovery
• Map building
• Path building (spanning tree)

Diffusion / Percolation / Viral Propagation
Distribution Networks

Move in Space

Physical

- Airplane
- Ship
- Train
- Truck
# Distribution Networks

<table>
<thead>
<tr>
<th>Move in Space</th>
<th>Move in Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="airplane.png" alt="Airplane" /></td>
<td><img src="forklift.png" alt="Forklift" /></td>
</tr>
<tr>
<td><img src="ship.png" alt="Ship" /></td>
<td><img src="warehouse.png" alt="Warehouse" /></td>
</tr>
<tr>
<td><img src="train.png" alt="Train" /></td>
<td><img src="cart.png" alt="Shopping Cart" /></td>
</tr>
<tr>
<td><img src="truck.png" alt="Truck" /></td>
<td></td>
</tr>
</tbody>
</table>

**Physical**
## Distribution Networks

<table>
<thead>
<tr>
<th>Physical Move in Space</th>
<th>Digital Move in Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Airplane" /></td>
<td><img src="image2" alt="Cable" /></td>
</tr>
<tr>
<td><img src="image3" alt="Ship" /></td>
<td><img src="image4" alt="Router" /></td>
</tr>
<tr>
<td><img src="image5" alt="Train" /></td>
<td><img src="image6" alt="Antenna" /></td>
</tr>
<tr>
<td><img src="image7" alt="Truck" /></td>
<td><img src="image8" alt="Antenna" /></td>
</tr>
<tr>
<td><img src="image9" alt="Cart" /></td>
<td><img src="image10" alt="X" /></td>
</tr>
</tbody>
</table>

**Physical**
- Movement in space
  - Airplane
  - Ship
  - Train
  - Truck
  - Cart

**Digital**
- Movement in time
  - Cable
  - Router
  - Antenna

**X**
- No direct correspondence

---

14
<table>
<thead>
<tr>
<th>Naming</th>
<th>Communication</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Endpoints</td>
<td>Stuff</td>
</tr>
</tbody>
</table>

Invisible

Explicit – storage and wires equivalent

Secure the process

Secure the stuff
<table>
<thead>
<tr>
<th>Communication</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naming</td>
<td>Endpoints</td>
</tr>
<tr>
<td>Memory</td>
<td>Invisible</td>
</tr>
<tr>
<td></td>
<td>Communication</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Naming</td>
<td>Endpoints</td>
</tr>
<tr>
<td>Memory</td>
<td>Invisible</td>
</tr>
<tr>
<td>Security</td>
<td>Secure the process</td>
</tr>
<tr>
<td></td>
<td>Communication</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Naming</td>
<td>Endpoints</td>
</tr>
<tr>
<td>Memory</td>
<td>Invisible</td>
</tr>
<tr>
<td>Security</td>
<td>Secure the process</td>
</tr>
</tbody>
</table>

Distribution networks solve a larger class of problems than communication networks (a strict superset).
van’s calendar?
pointless mtg 08:30
• Users specify objective, not how to accomplish it.

• Data appears wherever it needs to be.

• Model loves wireless and broadcast.

• There’s no distinction between bits in a memory and bits in a wire.

• Data security and integrity are the architectural foundation, not an add-on.
CCN packets

There are two packet types: *Interest* (a question) and *Data* (an answer). Both are encoded in an efficient binary XML.

<table>
<thead>
<tr>
<th>“interest”</th>
<th>“data”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Name</td>
<td>Content Name</td>
</tr>
<tr>
<td>Selector</td>
<td>Signature</td>
</tr>
<tr>
<td>(order preference, publisher filter, scope, ...)</td>
<td>(digest algorithm, witness, ...)</td>
</tr>
<tr>
<td>Nonce</td>
<td>Signed Info</td>
</tr>
<tr>
<td></td>
<td>(publisher ID, key locator, stale time, ...)</td>
</tr>
<tr>
<td></td>
<td>Data</td>
</tr>
</tbody>
</table>
CCN names are opaque, structured byte strings

/parc.com/van/cal/417.vcf/v3/s0/0x3fdc96a4...

is represented as a sequence of components followed by an end marker. Each component is a byte count followed by that many bytes.

8: parc.com 3: van 3: cal ... 32: 3FDC96... 0

The only assumption CCN makes about names is hierarchical structure. E.g., names or components can be encrypted or contain arbitrary binary data.
Basic CCN forwarding

• Consumer ‘broadcasts’ an interest over any available communications media:

  want ‘/parc.com/van/slides.pdf’

• Interest identifies a *collection* of data - all data items whose name has the interest as a prefix.

• Anything that hears the interest and has an element of the collection can respond with it:

  HereIs ‘/parc.com/van/slides.pdf/v6/p1’ <data>
Basic CCN transport

• Data that matches an interest ‘consumes’ it.

• Interest must be re-expressed to get new data. (Controlling the re-expression allows for traffic management and environmental adaptation.)

• Multiple (distinct) interests in same collection may be expressed (similar to TCP window).
Forwarding
IP node model

Transport

FIB lookup → Check & decr. TTL → N → Dst is me?

Y

FIB
Prefix | Interface
---|---
10.* | 2

Interface 0

Interface 1

Interface 2
IP node model

FIB lookup → Check & decr. TTL → Dst is me?

FIB
 Prefix | Interface
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10.*</td>
<td>2</td>
</tr>
</tbody>
</table>

Transport

Interface 0
Interface 1
Interface 2
CCN node model

**Content Store**

<table>
<thead>
<tr>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s0</td>
<td>…</td>
</tr>
</tbody>
</table>

**Pending Interest Table (PIT)**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Requesting Face(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s1</td>
<td>0</td>
</tr>
</tbody>
</table>

**FIB**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Face list</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com</td>
<td>0, 1</td>
</tr>
</tbody>
</table>

**Index**

<table>
<thead>
<tr>
<th>ptr</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>F</td>
</tr>
</tbody>
</table>

C = Content store
P = PIT
F = FIB

Face 0

Face 1

Face 2

Application
Comparison

Content Store & PIT are same as buffer memory - same contents, different replacement policy.
Comparison

FIBs are almost identical except CCN has list of output faces.
Comparison

Half the transport state becomes the (multi-point) Pending Interest table

Transport

FIB lookup → Check & decr. TTL → Dst is me?

Y

Interface 0

Interface 1

Interface 2

FIB
Prefix Interface
10.* 2

Content Store
Name Data
/parc.com/videos/WidgetA.mpg/v3/s0 …
/parc.com/videos/WidgetA.mpg/v3/s1 0

Pending Interest Table (PIT)
Prefix Requesting Face(s)
/parc.com/videos/WidgetA.mpg/v3/s1 0

Index
ptr type
C = Content store
P = PIT
F = FIB

FIB
Prefix Face list
/parc.com 0, 1

Application

Face 0

Face 1

Face 2
Comparison

There’s no “me” test since CCN demuxing & forwarding are the same.
Comparison

There's no TTL decrement since nothing can loop. (CCN packets are never modified in transit.)
CCN forwarding

**Content Store**

<table>
<thead>
<tr>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s0</td>
<td>...</td>
</tr>
</tbody>
</table>

**Pending Interest Table (PIT)**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Requesting Face(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s1</td>
<td>0</td>
</tr>
</tbody>
</table>

**FIB**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Face list</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com</td>
<td>0, 1</td>
</tr>
</tbody>
</table>

**Index**

<table>
<thead>
<tr>
<th>ptr type</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

C = Content store  
P = PIT  
F = FIB

get /parc.com/videos/WidgetA.mpg/v3/s2

Application
CCN forwarding

Content Store

<table>
<thead>
<tr>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s0</td>
<td>...</td>
</tr>
</tbody>
</table>

Pending Interest Table (PIT)

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Requesting Face(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s1</td>
<td>0</td>
</tr>
</tbody>
</table>

FIB

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Face list</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com</td>
<td>0, 1</td>
</tr>
</tbody>
</table>

Index

<table>
<thead>
<tr>
<th>ptr</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

C = Content store
P = PIT
F = FIB

get /parc.com/videos/WidgetA.mpg/v3/s2
CCN forwarding

<table>
<thead>
<tr>
<th>Name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s0</td>
<td>…</td>
</tr>
</tbody>
</table>

Pending Interest Table (PIT)

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Requesting Face(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s1</td>
<td>0</td>
</tr>
<tr>
<td>/parc.com/videos/WidgetA.mpg/v3/s2</td>
<td>0</td>
</tr>
</tbody>
</table>

FIB

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Face list</th>
</tr>
</thead>
<tbody>
<tr>
<td>/parc.com</td>
<td>0,1</td>
</tr>
</tbody>
</table>

Index

<table>
<thead>
<tr>
<th>ptr</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

C = Content store
P = PIT
F = FIB

get /parc.com/videos/WidgetA.mpg/v3/s2
generic forwarding ASIC
Routing
IGP routing

B sends a ‘hello’ out all its links
IGP routing

C floods that it’s adjacent to B
IGP routing

Same thing happens in C→B direction. When ‘B adj C’ announcement flooded, everyone adds B-C link to map.
IGP routing
Some ‘external’ (non-IGP) agent injects a ‘prefix announcement’ which B floods to all other IGP nodes
Existing link-state routing protocols can be used, unmodified, to construct a CCN FIB
Information on CCN is available at

www.ccnx.org

including a GPL’d open-source release of our current research prototype.

Information on NDN is available at

named-data.net