An Overview Of Information-Centric Networking

Spencer Sevilla
CMPE 257
This Talk

• Intro/Motivations/History
• CCN
• ICN
• The Forest For The Trees
• A New Proposal
A Quick Note...

- CCN is an individual project
- ICN is the overarching abstract concept
- A good analogy: SDN and OpenFlow
The Current State

• Today’s network architecture revolves primarily around connecting **hosts**

![Diagram](image)
Problems

- End-to-end model made sense in the ’60s when we only had a few mainframes
- The Web now dominates Internet traffic
- What about CDNs, caches, and replication?
- Web users don’t care about hosts or IPs, they care about content!
What Is ICN?

- A new way of looking at networking
- Focus on connecting users with **content**
- No longer concerned with who, where, how, etc...

---

give me object “X”

here is object “X”

Network
History

- Lots of statistical work on HTTP cooperative caching throughout the 90s
- TRIAD (Cheriton et. al.) proposed in 2000
- DONA (Koponen et. al.) in 2007
- The golden rule of systems: everything has been done before!
CCN: End Hosts

- Content Centric Networking (CCN) proposed in 2009 by Van Jacobson at PARC
- End-hosts publish and request Named Data Objects (NDOs) with Data and Interest packets
- Routers forward Interests appropriately and respond with Data packets
CCN: End Hosts

CCN: Routers

- Okay, that’s cool… but how on earth does it work? (We’re networking geeks here)

- Routers are comprised of three parts: the Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB)

- CCN also uses the concept of a “Face,” which is just an abstraction of a link between two CCN routers
CCN: Routers

CCN: Routing

- Okay, cool… but how does THAT work?
- When a router receives an Interest, it stores it in the PIT and forwards it onwards.
- When a router receives a Data packet, it caches it in the CS and uses the PIT to forward it backwards.
In CCN, NDOs are published at nodes, and routing protocols are employed to distribute information about NDO location. Routing in CCN can leverage aggregation through a hierarchical naming scheme. NDO security is achieved through Public Key cryptography. Trust in keys can be established via different means, such as a PKI-like certificate chain based on the naming hierarchy, or information provided by a friend. Requests (interest packets) for an NDO are forwarded toward a publisher location, as illustrated in Fig. 3, steps 1–3. A CCN router maintains a pending interest table (PIT) for outstanding forwarded requests, which enables request aggregation; that is, a CCN router would normally not forward a second request for a specific NDO when it has recently sent a request for that particular NDO. The PIT maintains state for all interests and maps them to network interface where corresponding requests have been received from. Data is then routed back on the reverse request path using this state (steps 4–6). CCN supports on-path caching: NDOs a CCN router receives (in responses to requests) can be cached so that subsequent received requests for the same object can be answered from that cache (as depicted in steps 7–8, Fig. 3). From a CCN node's perspective, there is balance of requests and responses; that is, every single sent request is answered by one response (or no response). CCN nodes can employ different strategies for requests (re-) transmission pace and interface selection depending on local configuration, observed network performance, and other factors. The NDN project advances the CCN approach. It provides a topology-independent naming scheme and is exploring greedy routing for better router routing scalability.
CCN: Routing

• How does a router or host know where to send Interests?  
  Use the FIB!

• But how do we populate the FIB?  
  Stop asking questions!
  Just kidding... kind-of...
Towards ICN

• Since CCN, there has been a LOT of work done on many different aspects of ICN

• Some works adapt/fork CCN, others propose entirely separate architectures

• These works generally focus on one or more key aspects/drawbacks of ICN
Popular ICN Topics

• Naming: How do users name content?
• Directory: How do they learn those names?
• Routing: How do we locate content?
• Caching: Where should we store content?
• Security: Without e2e, how does it work?
• Two main styles: Flat or Hierarchical

• Hierarchical names are legible and easy to prefix-match, but hard to verify

• Flat names can self-certify using PKI, but are harder (impossible?) to aggregate or read
Hierarchical: CCN names use / to indicate structure. ex: /ucsc/spencer/icn_slides.ppt

Flat: Object names are just a cryptographic hash of the content with its public key!
Naming

Figure 1: A depiction of the three entities and the different types of names. The entities are Name, Public Key, and RWI (Required World Identity). The arrows indicate the relationships:
1. Human-readable names map to RWI.
2. Self-certifying names map to public key.

Naming Problems

• Hierarchical: If the name doesn’t secure the content, can the security info be spoofed?

• Flat: If the name doesn’t reflect the RWI, how do you ensure that binding securely?

• These problems are essentially isomorphic!
Routing

- ICN routing generally takes two forms, for outgoing Interests and Data responses

- Approaches are architecture-specific and closely tied to the naming form employed
Routing Challenges

- Flat names: how to aggregate names under a single routing prefix?
- Hierarchical names: how to find the closest cached copy of an NDO?
- Both: How do we deal with routing table size when $O >> N$? (by ~$10^5$)
When forwarding REGISTERs, the RH signs it so that the receiving machine authorized to serve a datum or service with name P:L when processing REGISTERs and FINDs. RHs use which peer with those of their neighbors and friends. RHs use so, if a REGISTER comes from a copy closer than the previous copy. If does not forward it onward unless no such record exists or the new FIND(P:L) packet to locate the object named P:L, and RHs route the provider/customer/peer (or, alternatively, parent/child/peer) of appending their distance/cost to the previous-hop RH before sending. In a similar manner, the RHs accumulate the distances; they are hop-by-hop and accumulated in a REGISTER along the path. RHs know that the data came from a trusted RH. These signatures from P empowering this other key to register this piece of data. administrative policy (such as a corporate firewall). of a REGISTER can be terminated at any point if dictated by some interdomain policies as reflected in BGP. In addition, the forwarding driven by the local policies, DONA can faithfully respect the basic content). By letting the forwarding of FINDs and REGISTERs be for example, on whether the AS is willing to serve as a transit AS for updating its registration table). If the REGISTER comes from a peer, the RH has neither P:* or P:L entries do we say that P:L does not have an entry in the registration table. The RH has neither P:* or P:L entries do we say that P:L does not have an entry in the registration table. Only when for both P:* and P:L exist, the RH uses the one for P:L. Only when entry for P:* but not P:L, the RH uses the entry for P:*; when entries P:*, in addition to individual entries for the various P:L. RHs use of RH hops, or some other metric). There is a separate entry for a next-hop RH and the distance to the copy (in terms of the number of RH hops, or some other metric). There is a separate entry for a next-hop RH and the distance to the copy (in terms of the number of RH hops, or some other metric). There is a separate entry for a next-hop RH and the distance to the copy (in terms of the number of RH hops, or some other metric). There is a separate entry for a next-hop RH and the distance to the copy (in terms of the number of RH hops, or some other metric). There is a separate entry for a next-hop RH and the distance to the copy (in terms of the number of RH hops, or some other metric).
Routing

• DONA makes it easy to find local copies, but depends on a Tier-1 entry for every single piece of content: no aggregation!

• CCN easily supports name-aggregation, but makes it harder to find local copies

• Tradeoff between flat and hierarchical?

• How many entries can a router handle?
What Is “Content?”

- Should it be a whole file or a small chunk?
- A whole site (CNN.com) or just an article?
- Should we de-duplicate content published multiple times (Kazaa, HBO)?
- What about content aggregators (Reddit) or content-search (Google)?
- And what to do about content metadata?
Security

- Content can come from anywhere in the network... for better or worse!
- Major shift: previous security model was designed for end-to-end communication
- ICN security revolves around verifying/authenticating the content, not the source
- This makes some attacks (spoofing) easier, others (MITM) harder/impossible
Security: Practices

• Most proposals rely heavily on PKI for content signing and verification

• Approaches vary in their ability to protect end-to-end communication (VoIP etc)

• The network itself has to be able to verify and reject malicious Data and Interests
Security: New Attacks

- Content poisoning can disseminate throughout the entire network if not caught
- Content requests and responses are visible to the entire network - what about privacy?
- Many social engineering attacks still apply
- What about other threats? (DDOS, etc)
The Forest For The Trees

• Let’s get back to the basics: why are we even bothering with ICN?

• Efficient content distribution
• Persistent and unique content naming
• Secure content provenance
• Support network mobility and disruption
The Forest For The Trees

• Not to be confused with the common properties of all ICN proposals:

  • Routing based on content-names
  • Location-independent content naming
  • Ubiquitous content-caching
  • Nearest-replica-routing
The Forest For The Trees

The Question: Are these benefits really even that beneficial, do they really matter?

The Punchline: If we can achieve the same benefits of ICN, who cares how we do it?
Zipf Distributions

• Small number of very popular objects (ie CNN) dominates the observed traffic

• Other unpopular objects (ie my bad poetry website) constitute the “long tail”

• This same distribution can be seen in books, files, and on the Web!
Zipf Distributions

![Zipf Distribution Graph](image)

Zipf Distributions
Zipf Distributions

The Punchline: A small cache size (in MB) gets us almost all the bandwidth savings!!!
In DONA, NDOs are published into the network by the sources. Nodes that are authorized to serve data, register to the resolution infrastructure consisting of resolution handlers (RHs). Requests (FIND packets) are routed by name toward the appropriate RH, as illustrated in Fig. 2, steps 1–4. Data is sent back in response, either through the reverse RH path (steps 5–8), enabling caching, or over a more direct route (step 9). Content providers can perform a wildcard registration of their principal in the RH, so that queries can be directed to them without needing to register specific objects. It is also possible to register NDO names before the NDO content is created and made available. Register commands have expiry times. When the expiry time is reached, the registration needs to be renewed. The RH resolution infrastructure routes requests by name in a hierarchical fashion and tries to find a copy of the content closest to the client. DONA's anycast name resolution process allows clean support for network-imposed middleboxes (e.g., firewalls, proxies).

In CCN, NDOs are published at nodes, and routing protocols are employed to distribute information about NDO location. Routing in CCN can leverage aggregation through a hierarchical naming scheme. NDO security is achieved through public key cryptography. Trust in keys can be established via different means, such as a PKI-like certificate chain based on the naming hierarchy, or information provided by a friend. Requests (interest packets) for an NDO are forwarded toward a publisher location, as illustrated in Fig. 3, steps 1–3. A CCN router maintains a pending interest table (PIT) for outstanding forwarded requests, which enables request aggregation; that is, a CCN router would normally not forward a second request for a specific NDO when it has recently sent a request for that particular NDO. The PIT maintains state for all interests and maps them to network interface where corresponding requests have been received from. Data is then routed back on the reverse request path using this state (steps 4–6). CCN supports on-path caching: NDOs a CCN router receives (in responses to requests) can be cached so that subsequent received requests for the same object can be answered from that cache (as depicted in steps 7–8, Fig. 3). From a CCN node's perspective, there is balance of requests and responses; that is, every single sent request is answered by one response (or no response). CCN nodes can employ different strategies for requests (re-)transmission pace and interface selection depending on local configuration, observed network performance, and other factors. The NDN project advances the CCN approach. It provides a topology-independent naming scheme and is exploring greedy routing for better router routing scalability.

In PSIRP, NDOs are also published into the network by the NDO sources as illustrated in step 1, Fig. 4. The publication belongs to a part...

---

Cache Distributions

- This scenario occurs when a content object is cached at C3 but not at C2 or C1

- How often do we see that happen?

- Is the content object popular or not???
Cache Distributions

- Turns out, this scenario is exceedingly rare!

- If the content object is popular, it probably was already at C1

- If not, it probably was evicted from all caches before the next request for it
Cache Distributions

Punchline #1: Small edge-caches are sufficient, and actually get us the vast majority of benefits

Punchline #2: Today’s HTTP edge caches and proxies are already quite close to this goal!!!
idICN

- “Incrementally Deployable ICN” combines HTTP servers with CDNs and Proxies

- We just have to adapt these protocols to be more location-independent and ICN-ish

- Slightly easier than replacing TCP/IP!!!
Conclusions

• ICN has been around for a while, and got kicked off this time around by CCN

• Many different ways to do ICN, each with their own benefits and drawbacks

• There’s a lot of heated debate around these topics, and the dust hasn’t quite settled

• Recent works question the overall benefits and if we should even be doing ICN at all!
Thanks!