Information Centric Networking

MSc in Computer Networking
January 09, 2015

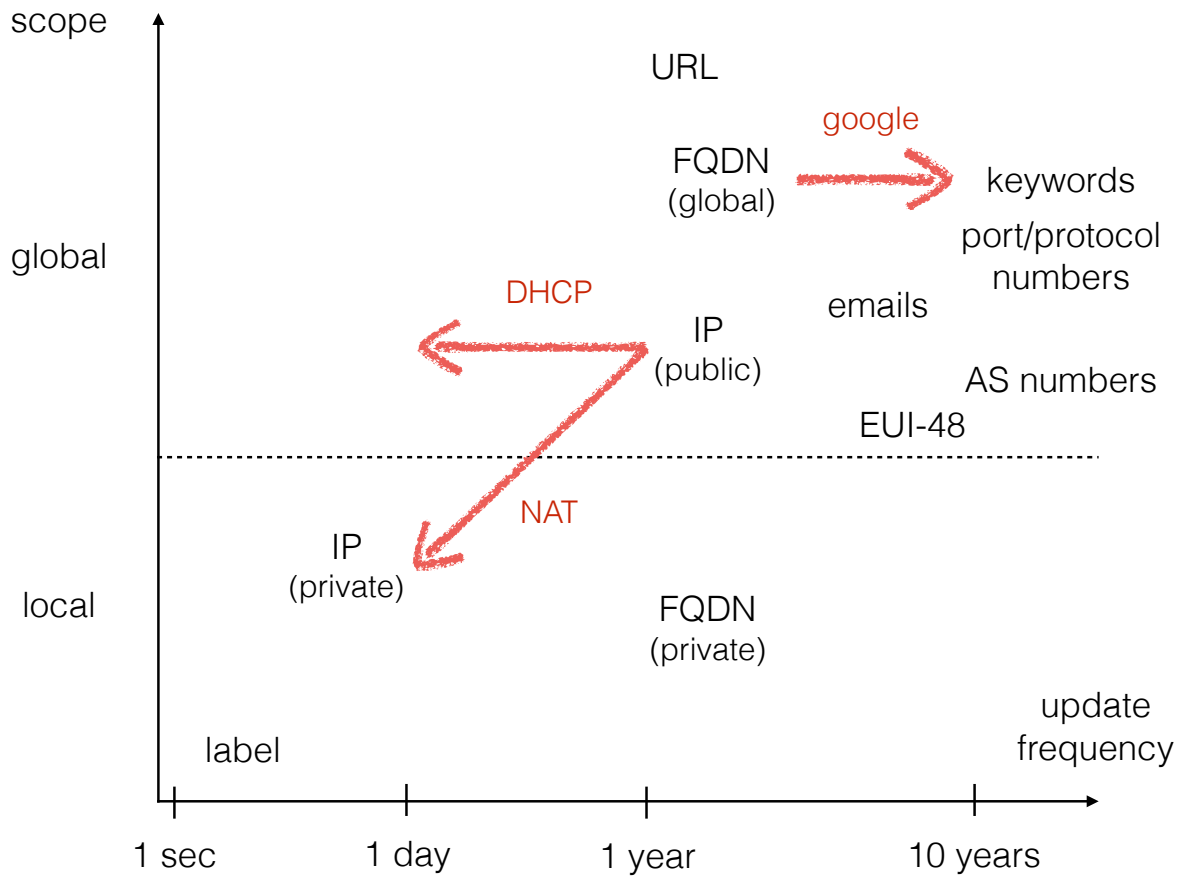
Prométhée Spathis
promethee.spathis@upmc.fr

Once, IP Adresses used to rule the Internet

- RFC 597 Host Status (1973)

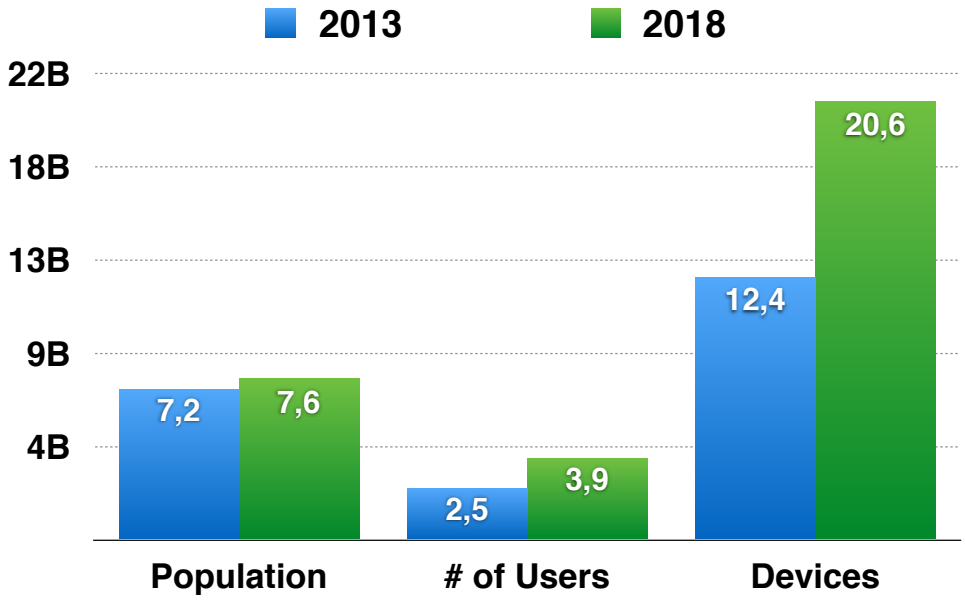
Host (8)	Address (10)	Hostname	(Interface)-> Computer	Status/ System
1	001	UCLA-NMC	Sigma 7	Server till 12/31/73 SEX
			PDP-11/45	User 1/1/74 ANTS
101	65	UCLA-CCn	IBM 360/91	Server
201	129	UCLA-CCBS	(PDP-15)-> PDP-10	limited Server
002	2	SRI-ARC	PDP-10	dedicated Server TENEX, NLS

- RFC 990 (1986)
  - ~10K assigned network numbers



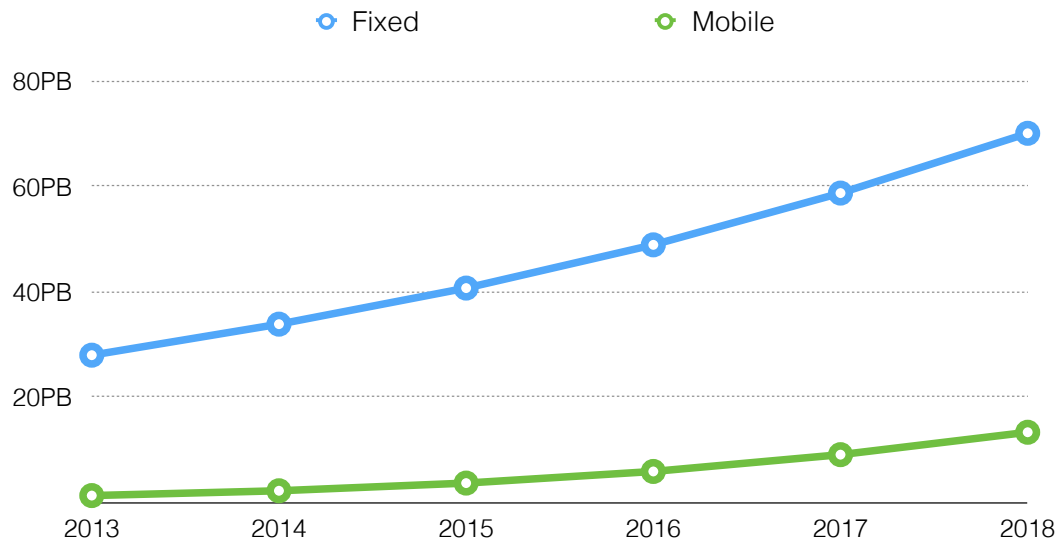
Man vs Human

(Billions per Year)



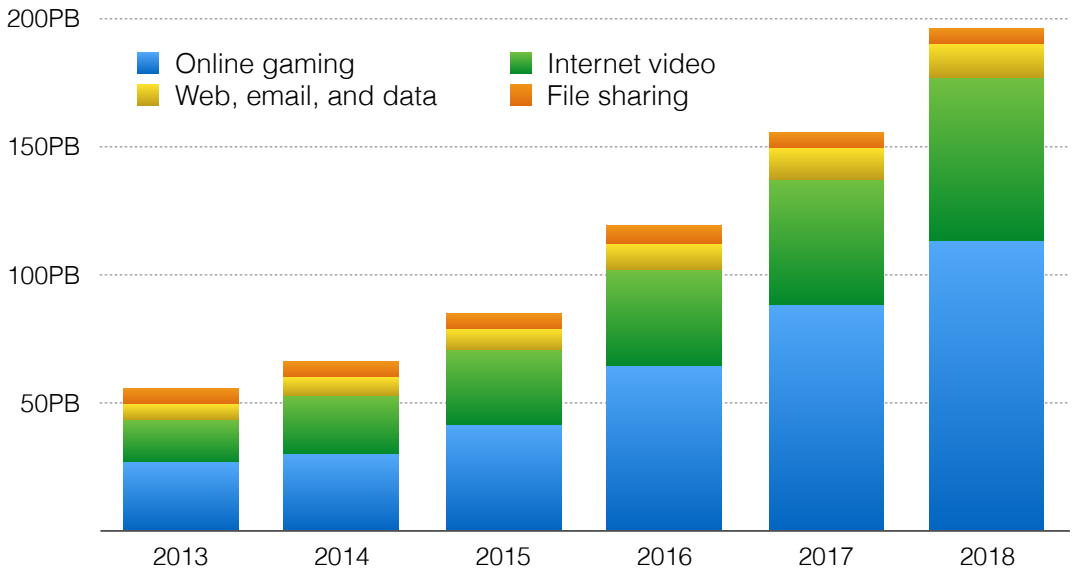
Internet Traffic by Network

(Petabytes per Month)



Internet Traffic By Type

(Petabytes per Month)



IP Addresses Bashing

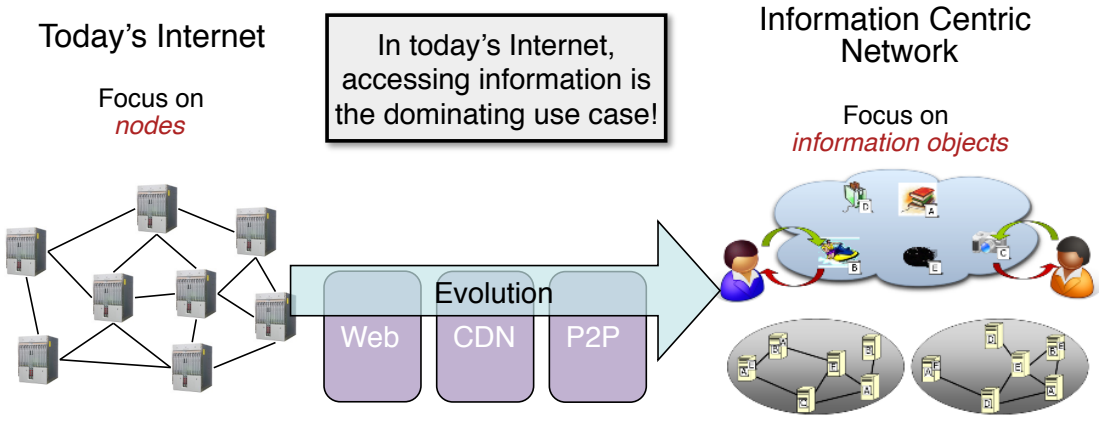
- Mobility/Migration: IP adds change over time
- Multihoming/Replication: Multiple IP adds at same time
- Routing and name resolution: Scaling and convergence time limitations
- Mobility and multihoming: ID/Loc split architectures
- Migration and replication: Data-intensive applications?

Design Usage Mismatch

- Internet was designed for host-to-host communication
  - "contact this host…"
- Internet is mainly used for data access
  - "get me this data....."

- Mismatch between usage and design:
  - data migration and replication unnecessarily hard
  - requires Akamai- and BitTorrent-like designs to scale
  - mobility and multi-homing pose problems

Information Centric Networking



- Considering important requirements
  - **Accessing named resources – not hosts**
  - **Scalable distribution through replication and caching**
  - **Good control of resolution/routing and access**
- With ubiquitous caching
  - But for all applications
  - And for all users and content/service providers

« Named Data Network (NDN) is a general-purpose protocol built on requests for named data. »

— Patrick Crowley (University of Washington in St. Louis)
ACM Conference on Information-Centric Networking (ICN-2014) Paris, France

« Content-centric Networking (CCN) is a communication architecture based on transferring named data. »

— Glenn Scott (Senior Research Scientist and Principal Engineer at Parc)
ACM Conference on Information-Centric Networking (ICN-2014) Paris, France

CCN Timeline

- 2005: PARC starts work on ICN
- 2006: Van Jacobson's talk at Google
- 2007: CCN 0.x begins
- 2009: Seminal CCN Paper at CoNext 2009
- 2010: NSF-funded NDN project
- 2013: CCN 1.x begins
- 2014: NDN 2nd phase (w/o Parc)

1st Dagstuhl Seminar (2010)
1st SIGCOMM workshop (2011)
1st CCNxCon (2011)
1st INFOCOM workshop (2012)
1st ICNRG meeting (2012)
1st NDNCom meeting (2014)
1st ICN Conference (2014)

CCN Key Tenets

- Content centric paradigm
  - access data objects in a location-independent manner
  - secure the content rather then the connection
  - add general-purpose memory into the network
- CCN design is not a clean-state approach …
  - not a replacement for IP
  - … but more conventional

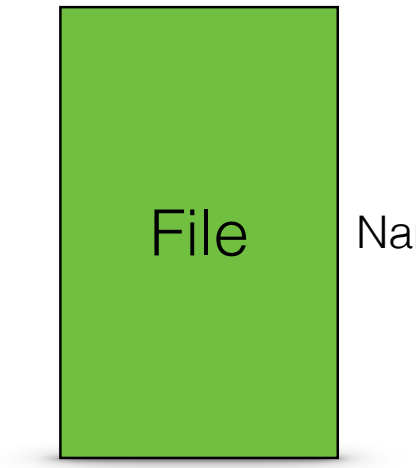
CCN Design Choices

- Which IP engineering principles can remain still valid after removing addresses from the Internet?
- Maximize the reuse of well-tried mechanisms and techniques directly borrowed from IP
  - DNS-like naming scheme
  - CIDR-like prefix aggregation
  - Longest prefix match forwarding
  - Link-state protocols (i.e. OSPF) opaque capacities

Agenda

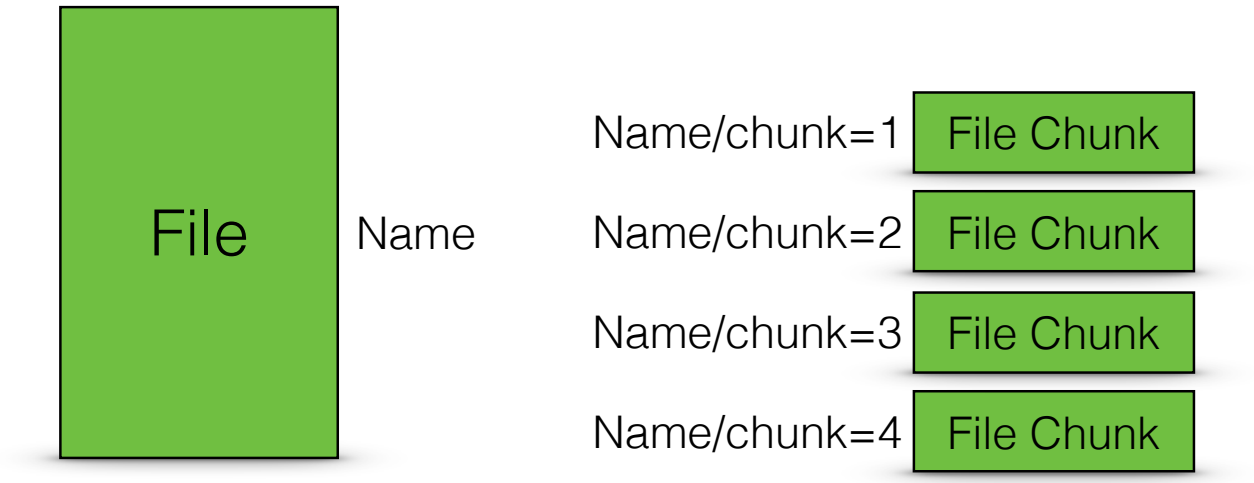
- CCN object and chunks
- CCN names format
- CCN signature
- CCN packet and messages format
- Content-based security
- CCN node model and forwarder
- CCN routing
- Application examples
- Conclusions

Named Object



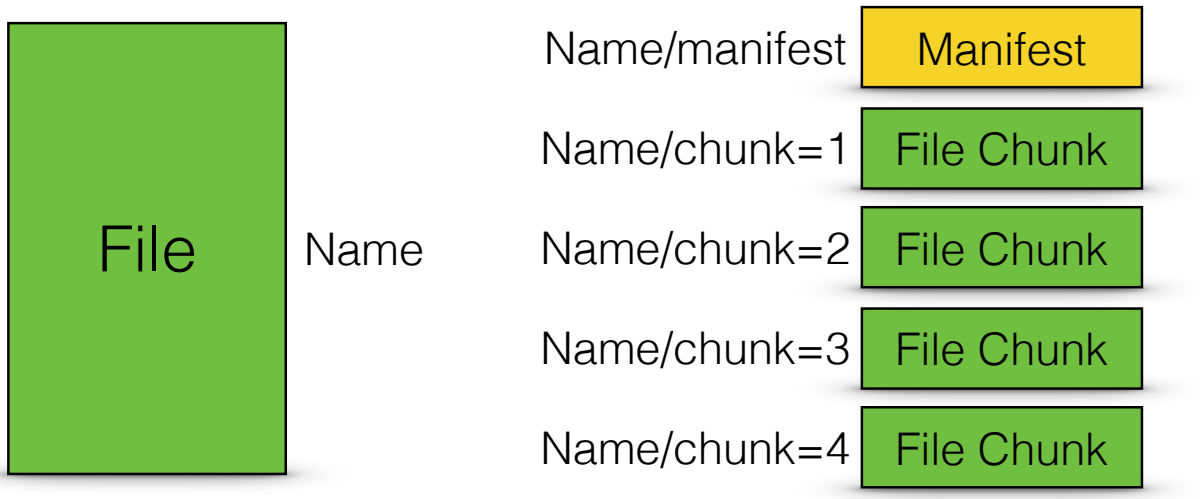
A large object has a CCN publisher-given name

Named Chunks



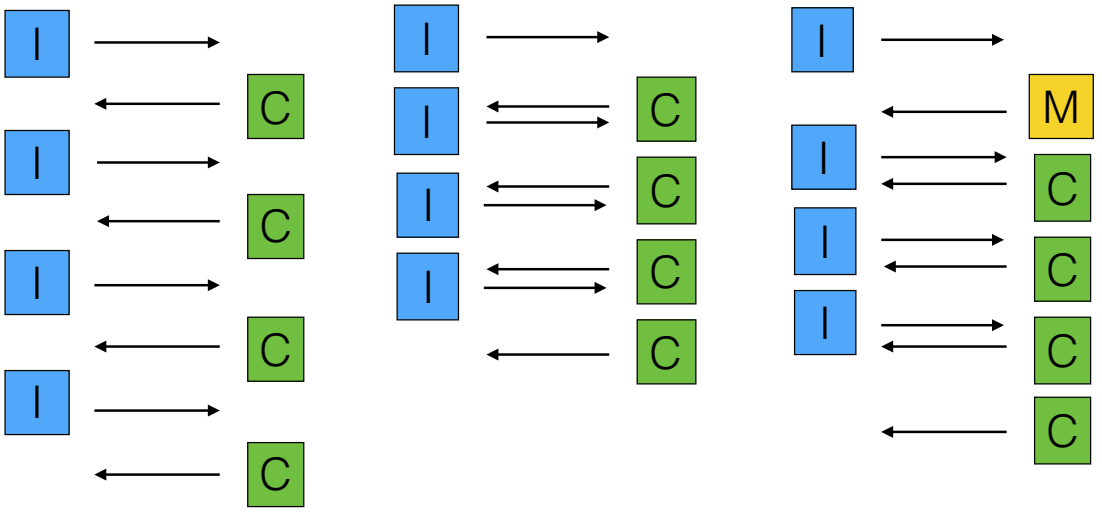
Each network-sized chunk also has a CCN name

Named Chunks



CCN creates a manifest describing the file

Transport Protocols



Core exchange parallel requests use of manifests

CCN Names Format

/upmc/spathis/ccn/tutorial/slide13/v=2/c=0

Ordered labeled sequence of binary segments

CCN Names Format

/upmc/spathis/ccn/tutorial/slide13/v=2/c=0

↓

global routable
name segments

Ordered labeled sequence of binary segments

CCN Names Format

/upmc/spathis/ccn/tutorial/slide13/v=2/c=0

↓ global routable name segments

↓ application dependent name segments

Ordered labeled sequence of binary segments

CCN Names Format

/upmc/spathis/ccn/tutorial/slide13/v=2/c=0

↓ global routable name segments

↓ application dependent name segments

↓ protocol dependent name segments

Ordered labeled sequence of binary segments

CCN Names Qualifier

Interests can identify content hash or publisher's key

/upmc/spathis/ccn/tutorial/slide13/v=2/c=0

ContentObjectHash

↓

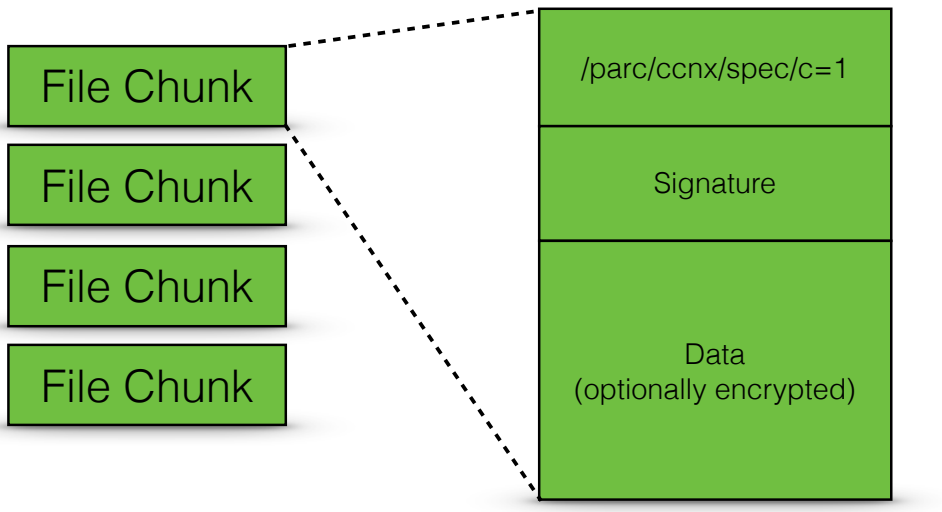
secure cryptographic hash of the Content Object message

KeyId

↓

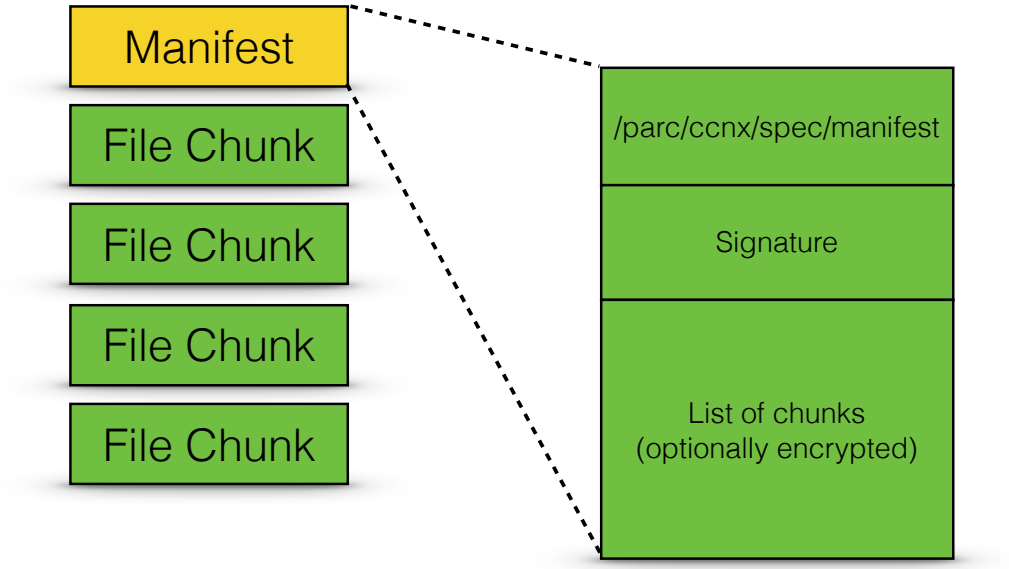
Identifier of the Content Objet publisher

Secure Single Chunk



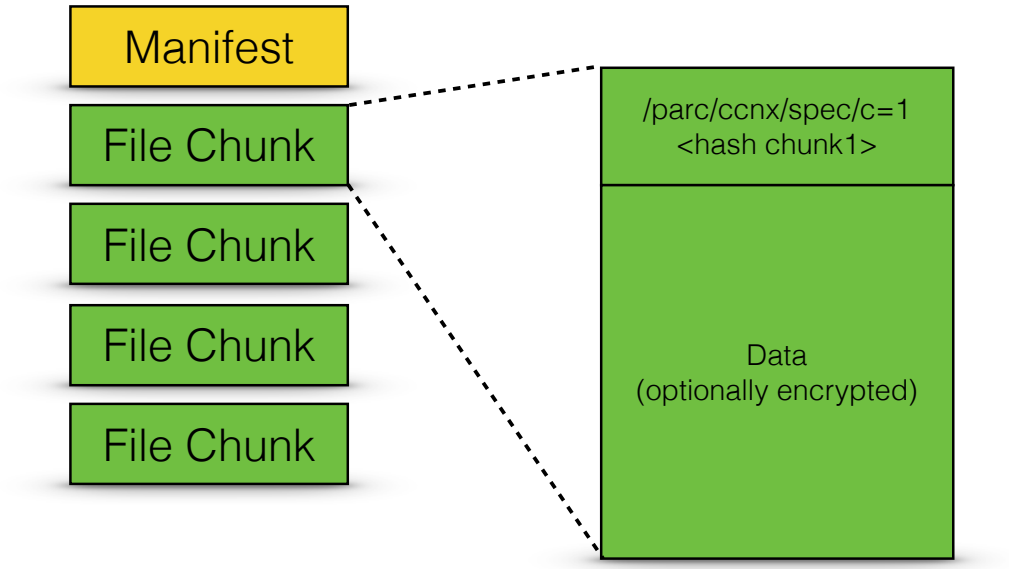
CCN names and signs every chunk

Secure Whole Object via Manifest



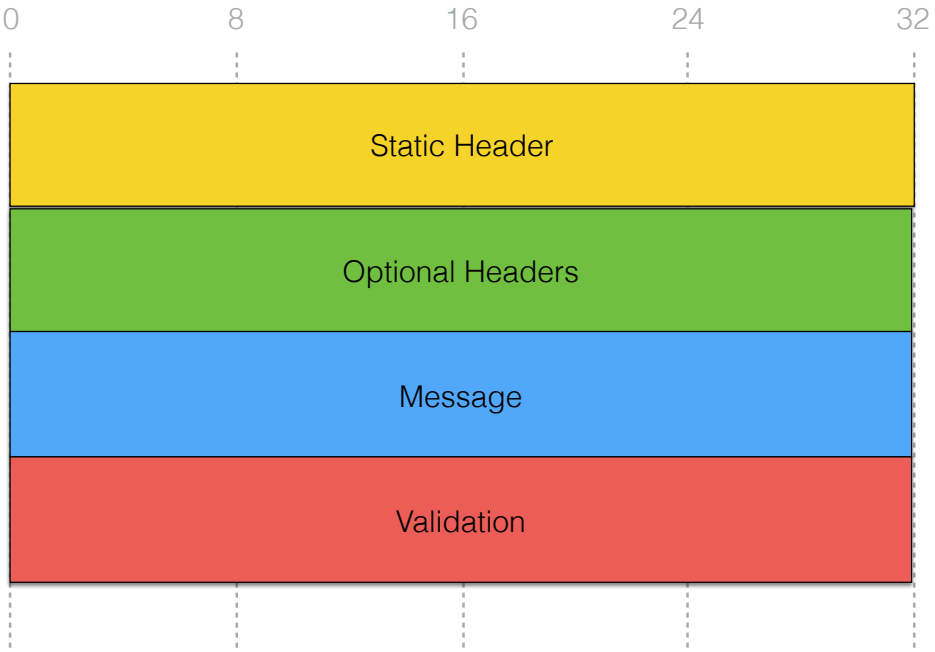
CCN names and signs the file via a manifest

Secure Whole Object via Manifest

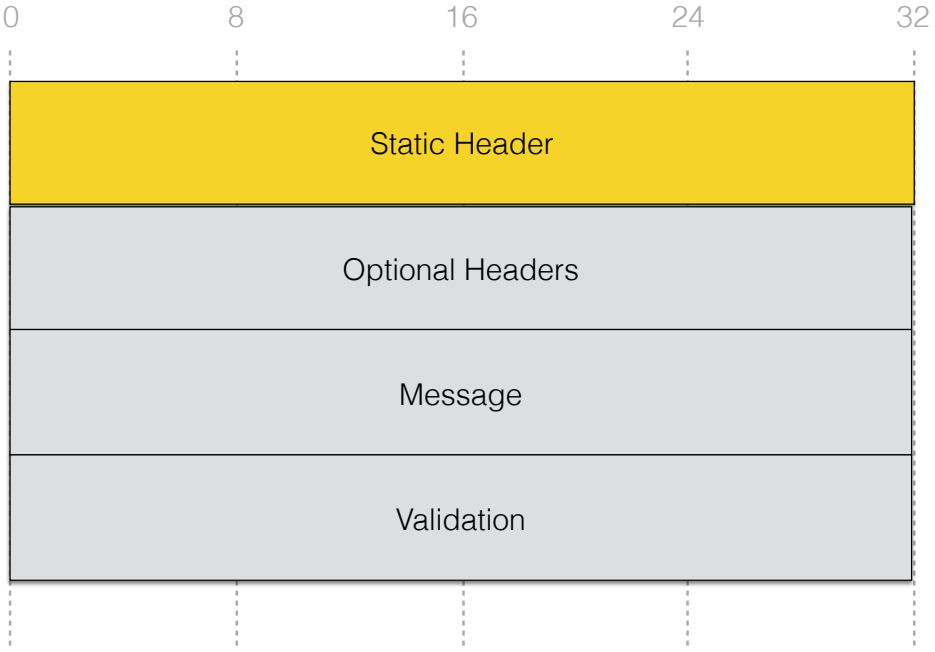


Indirectly sign every chunk through the manifest

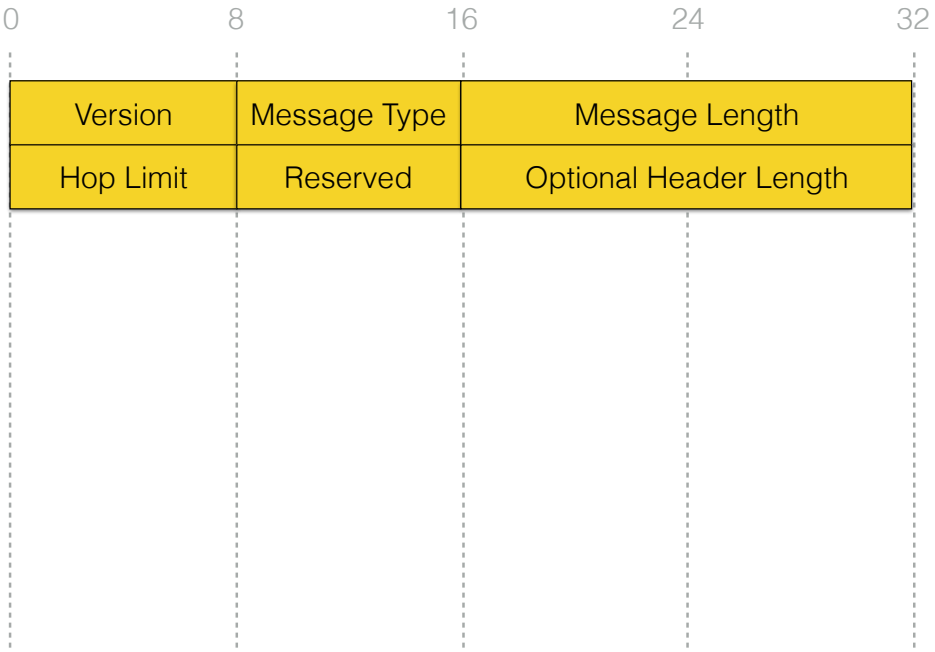
Packet Format



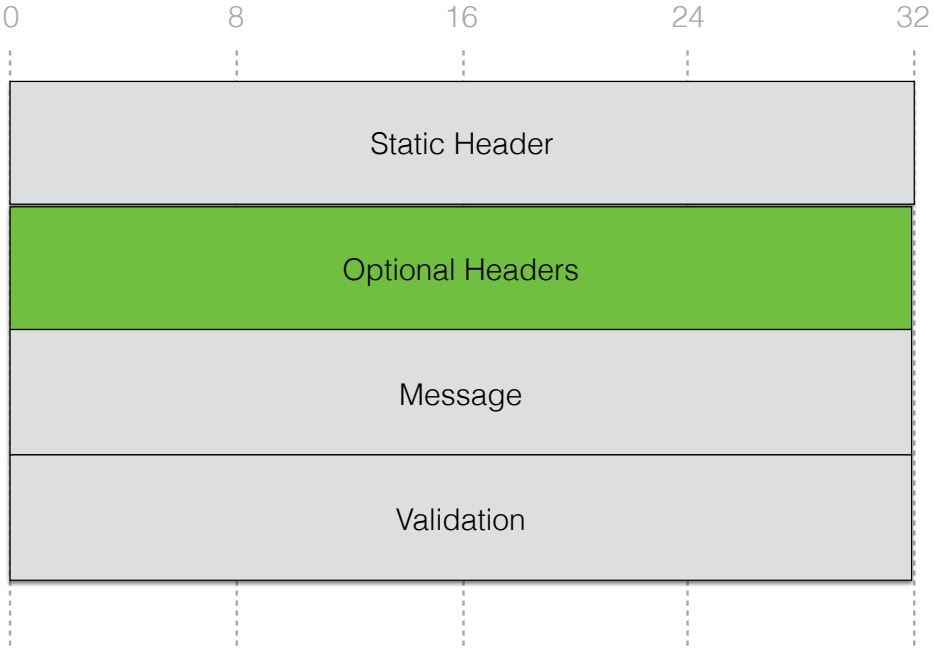
Static Header



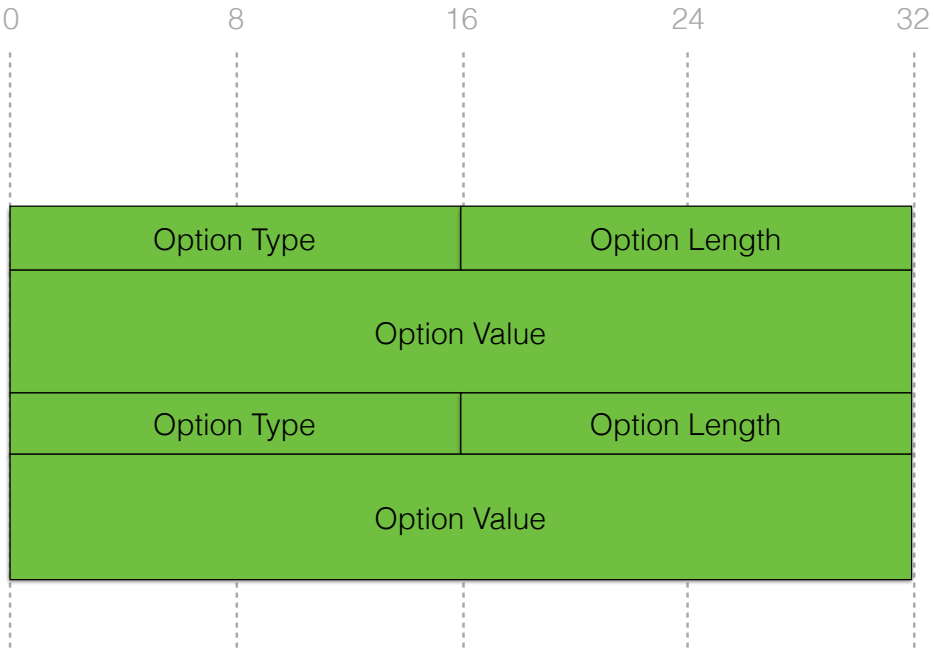
Static Header



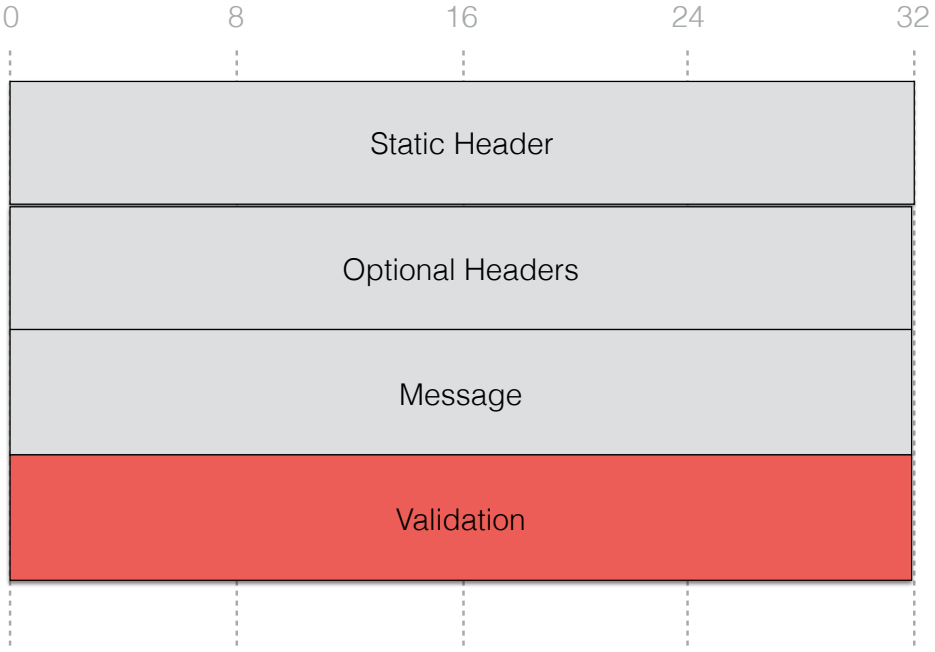
Packet Format



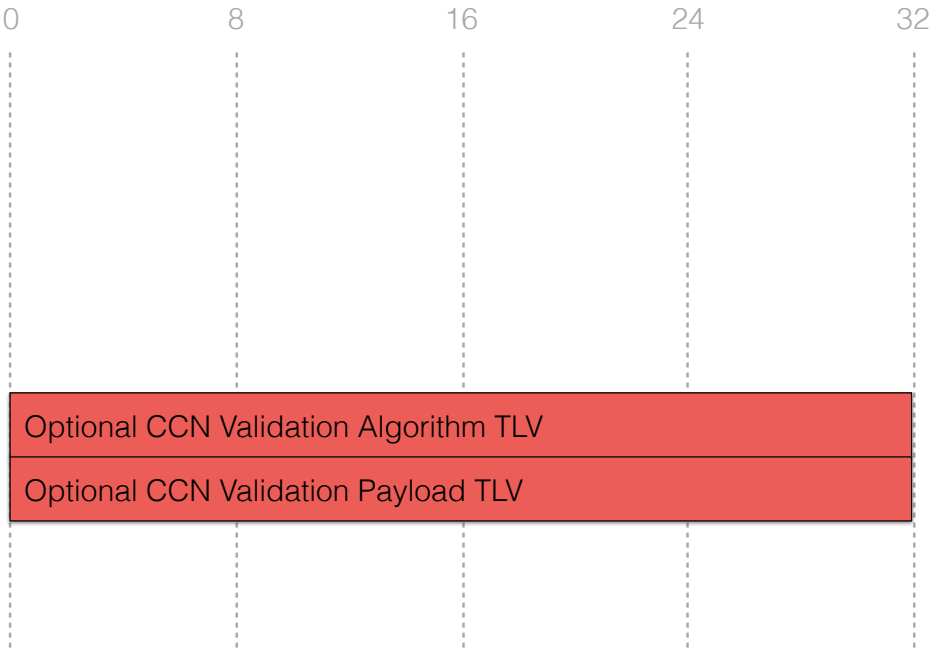
Optional Header



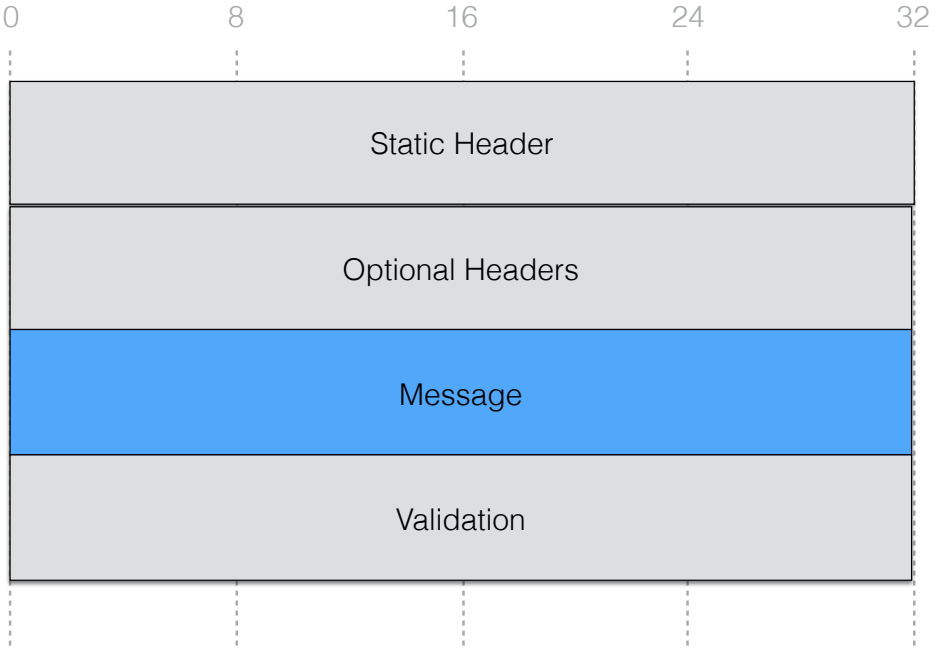
Packet Format



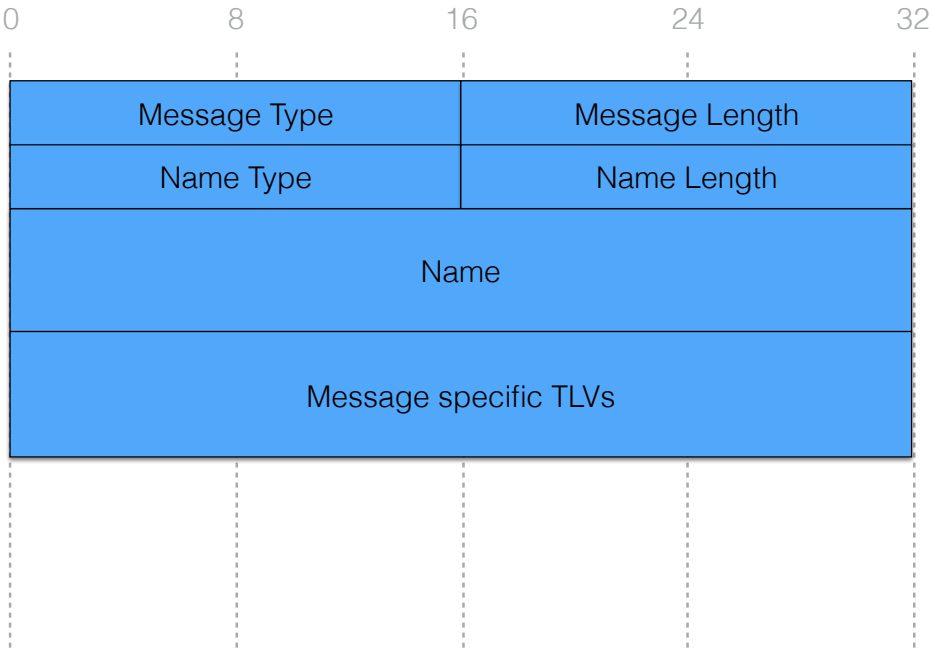
Validation



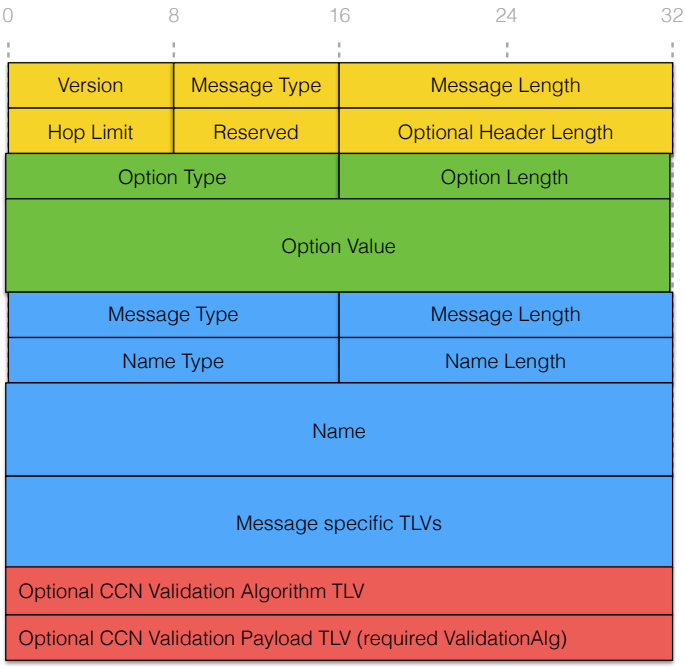
Packet Format



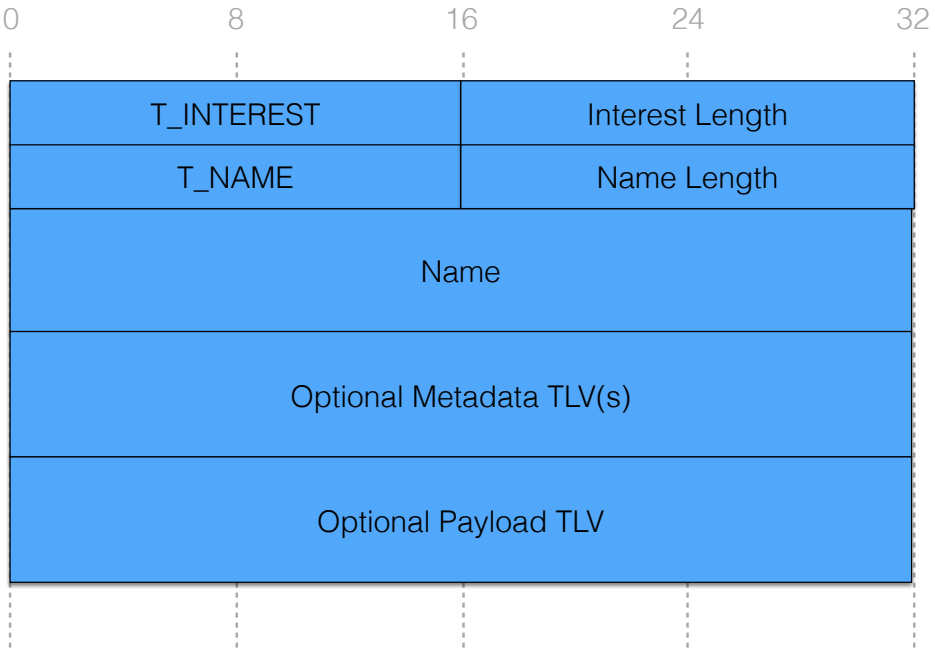
CCN Message



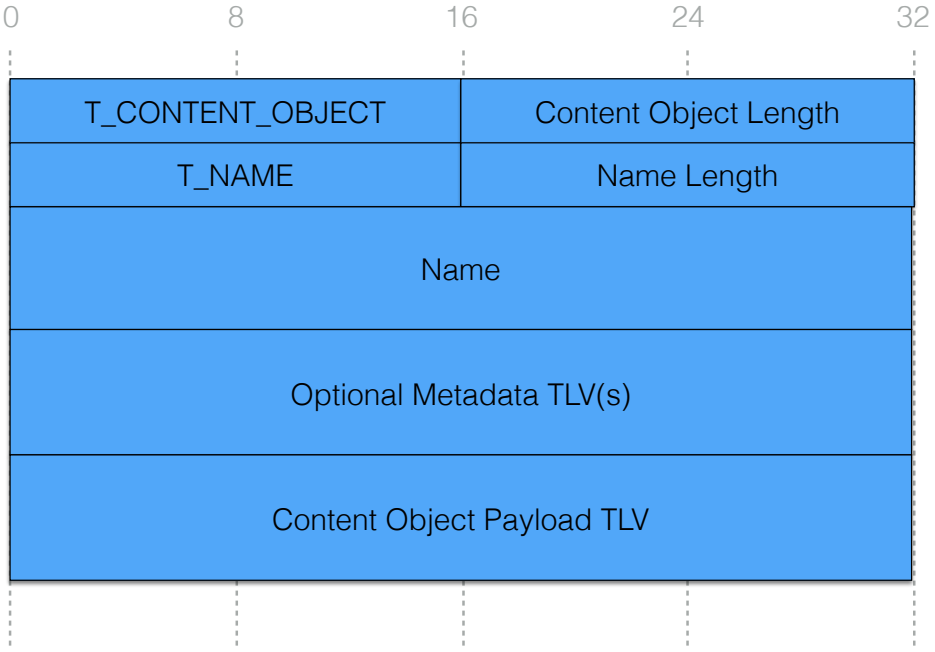
CCN Message



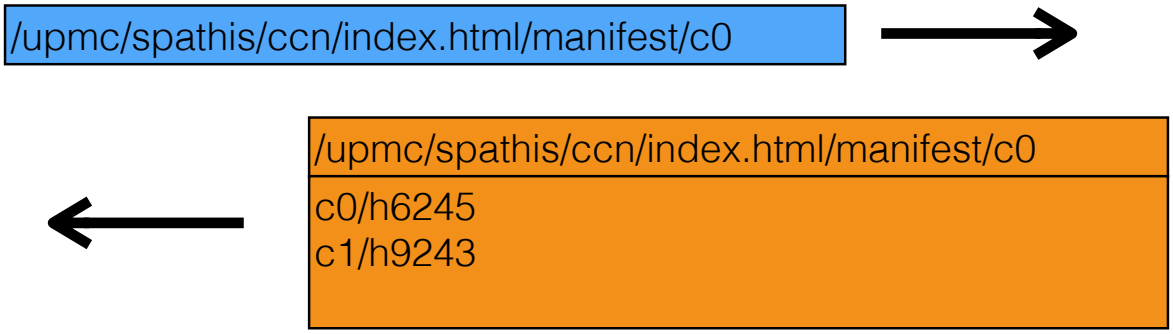
Interest Message



Content Object Message

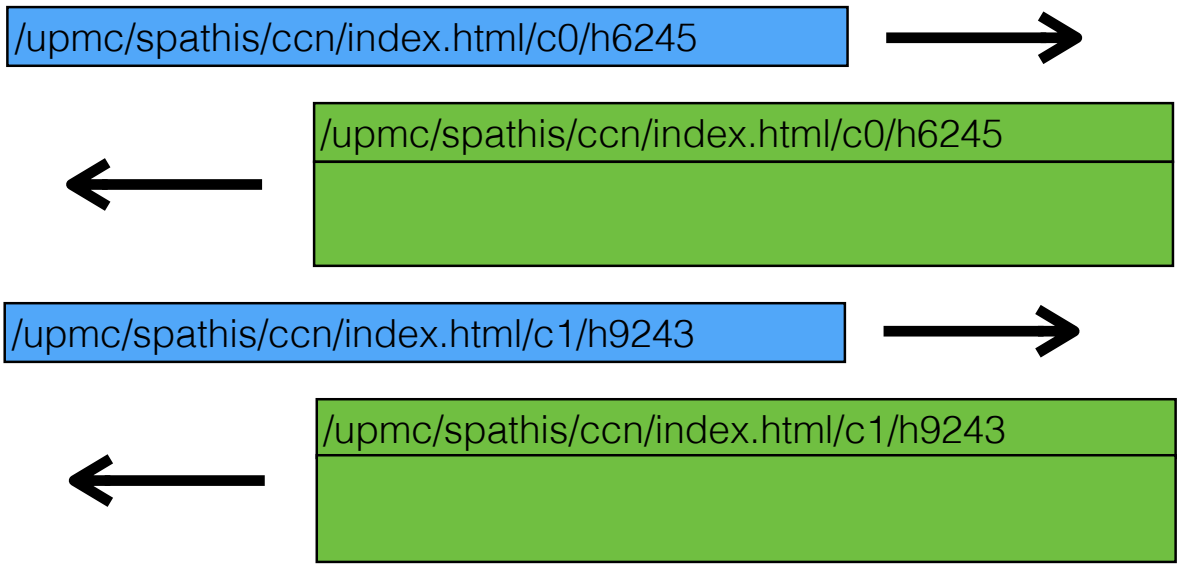


Quick Example Break



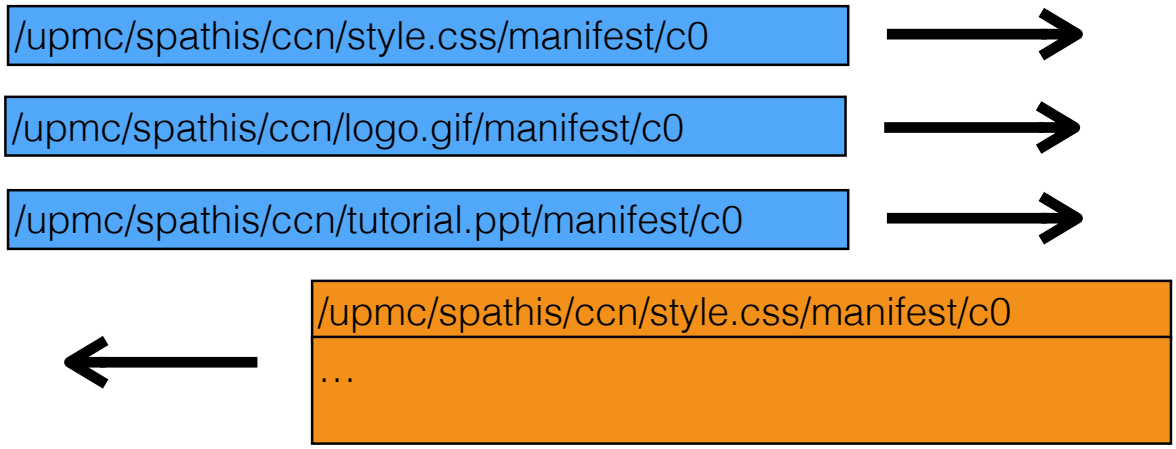
Request the list of chunks for the main webpage
Each chunk identified with a hash

Quick Example Break



Request each chunk using name and hash

Quick Example Break



Retrieve references

Securing Content

Content Packet = 〈 name, data, signature 〉

- Integrity & authenticity
  - Is it a complete, uncorrupted copy of what the publisher sent?
- Origin Authentication
  - Is the publisher one the receiver is willing to trust to supply this content?
- Correctness
  - Is this content an answer to the question the receiver asked.

Publishing and Verifying Content

$$M_{(N,C,P)} = \langle\ N,\ C,\ Sign_P(N,\ C)\ \rangle$$

- A content publisher
  - determines the name of its content (how it will be found)
  - generates a digital signature over that name and the content
- A content consumer, given N, must be able to retrieve
  - the content C, the authenticator $Sign_P(N, C)$
  - sufficient supporting information to determine what public key to use and where to find a copy

➡ User-friendly mechanisms to manage public and private keys

➡ Easy to deploy mechanisms to determine trust in keys and content

Content-based Security

- Security travels with the content
  - Content can be authenticated by any node (public key signatures)
  - Secure caching: can get content from anyone with a copy, and still authenticate it
  - Confidentiality: encrypt content for access control
- Move the security perimeter from the host to the application
  - Content decrypted only inside the target application
  - Use of encryption tailored to application needs
- Host protection
  - Harder to mount an attack against a host if you can't address packets to it
- Access control by policy routing

CCN Forwarder

- Routing

  Finding the path alternatives

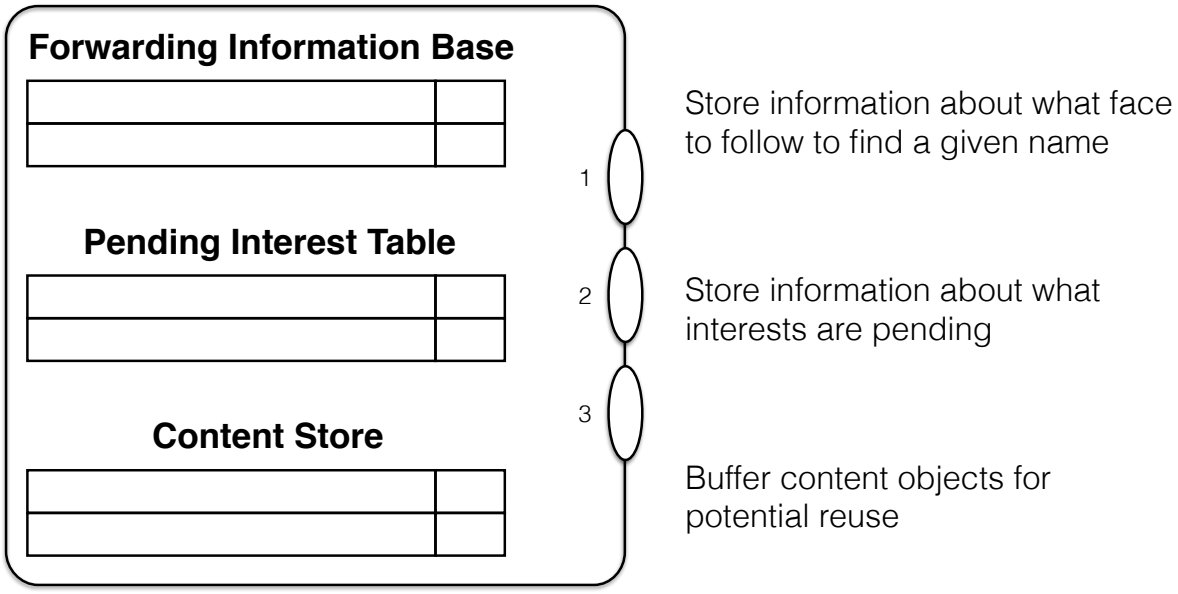
- Strategy

  How to use the alternatives

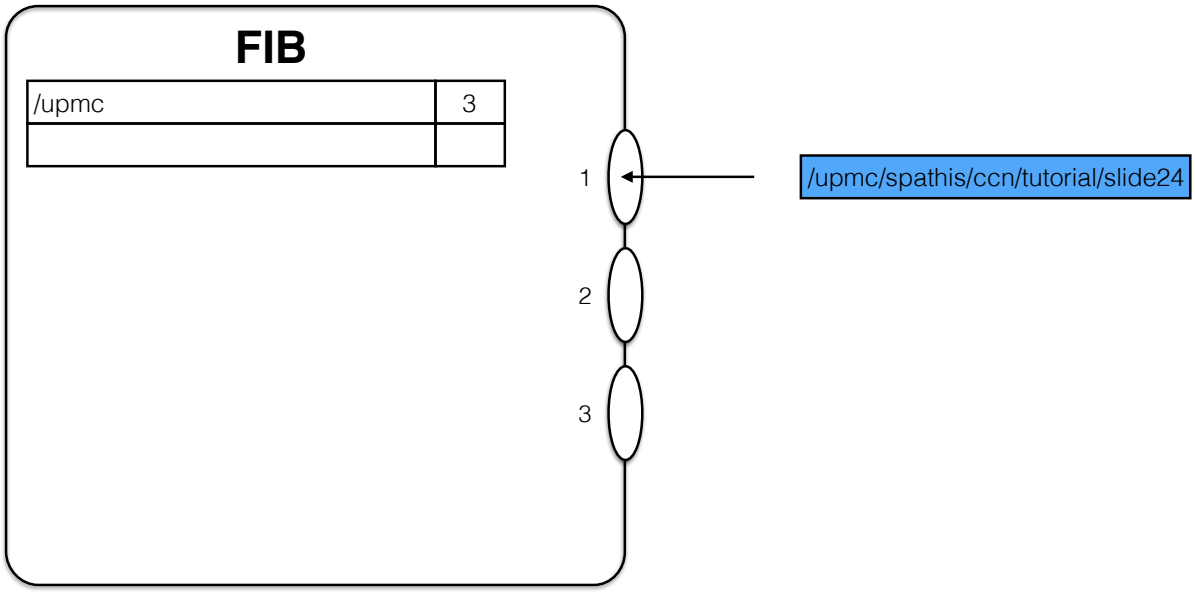
- Forwarding

  Processing a packet based on a strategy

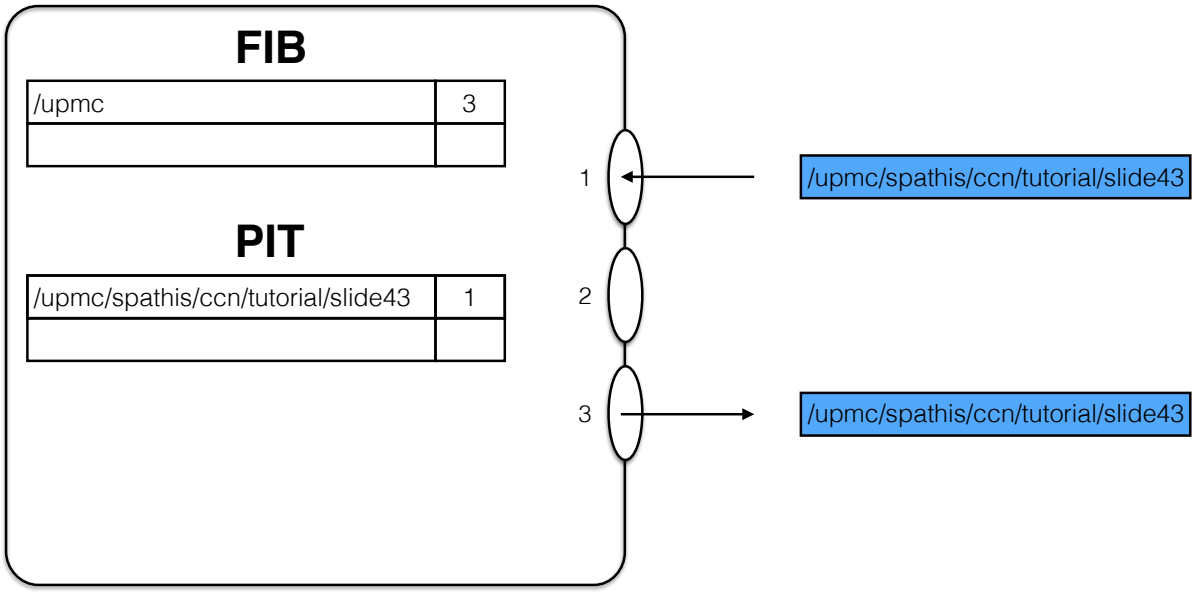
CCN Forwarder



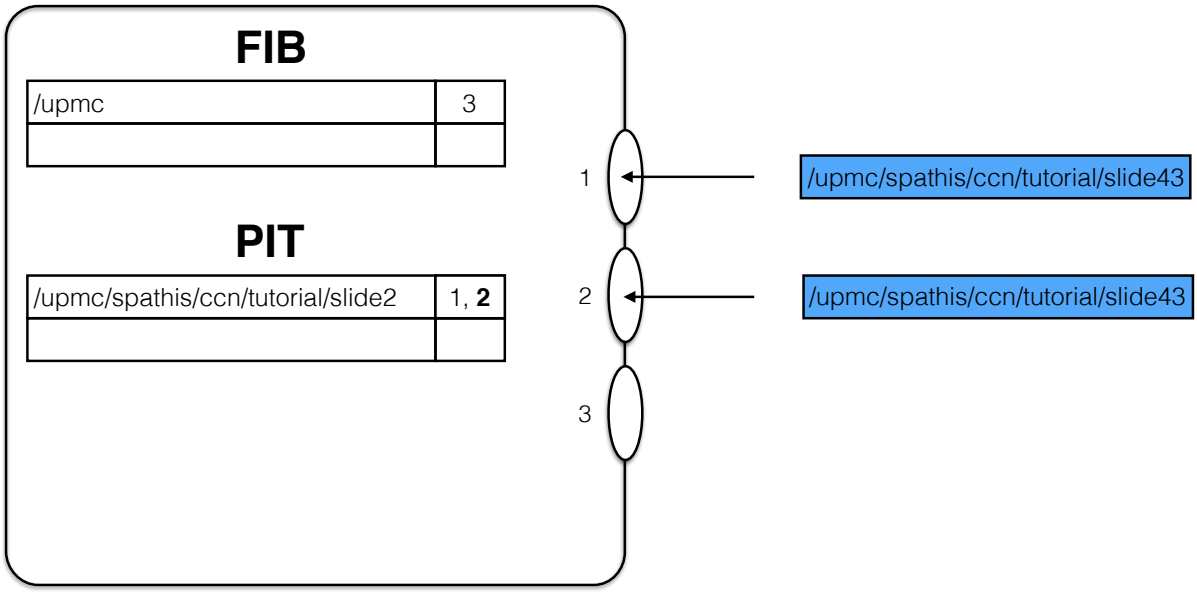
CCN Forwarder



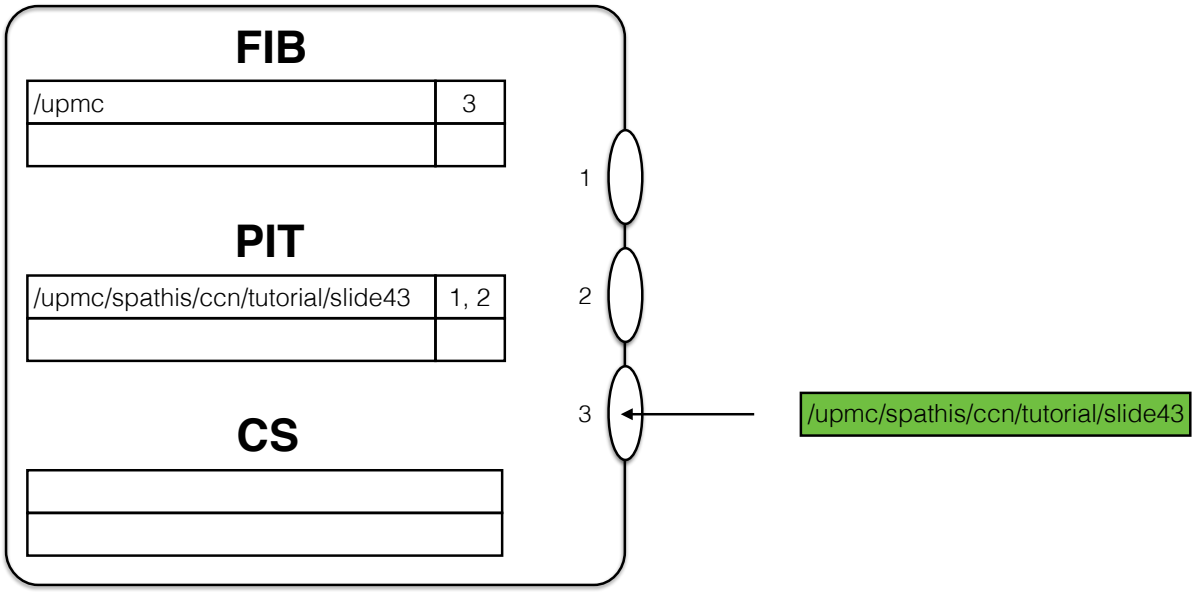
CCN Forwarder



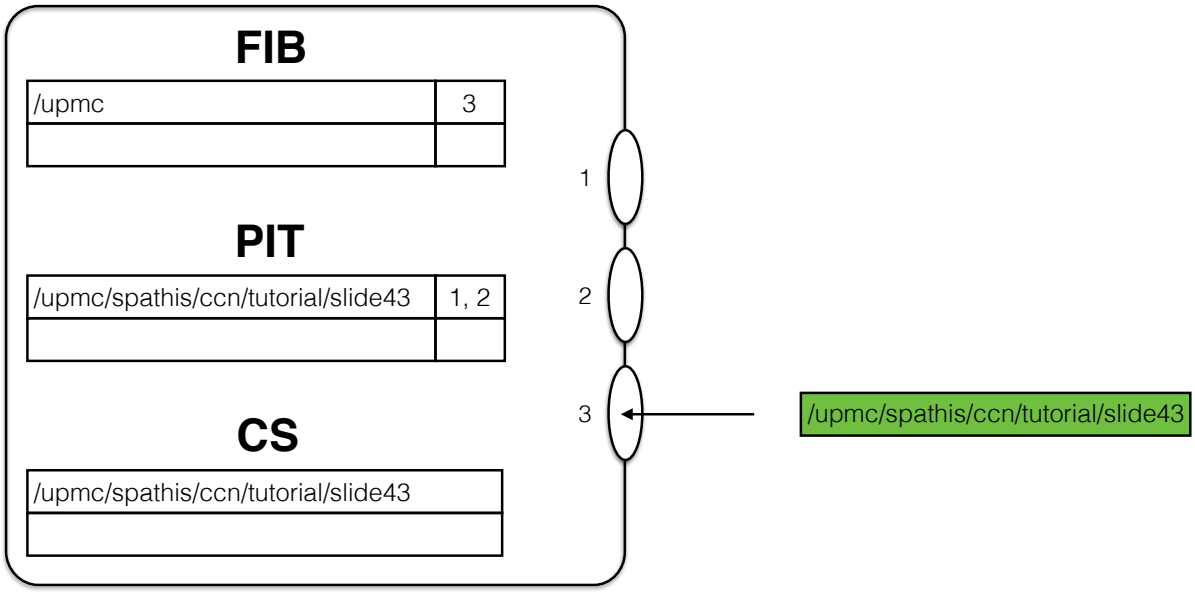
CCN Forwarder



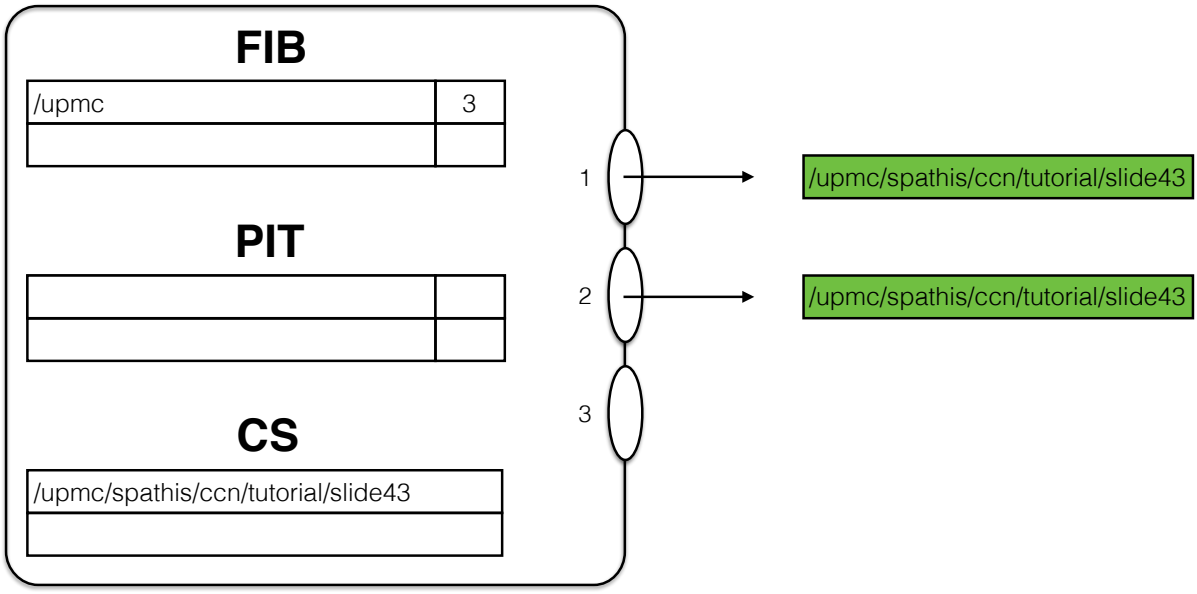
CCN Forwarder



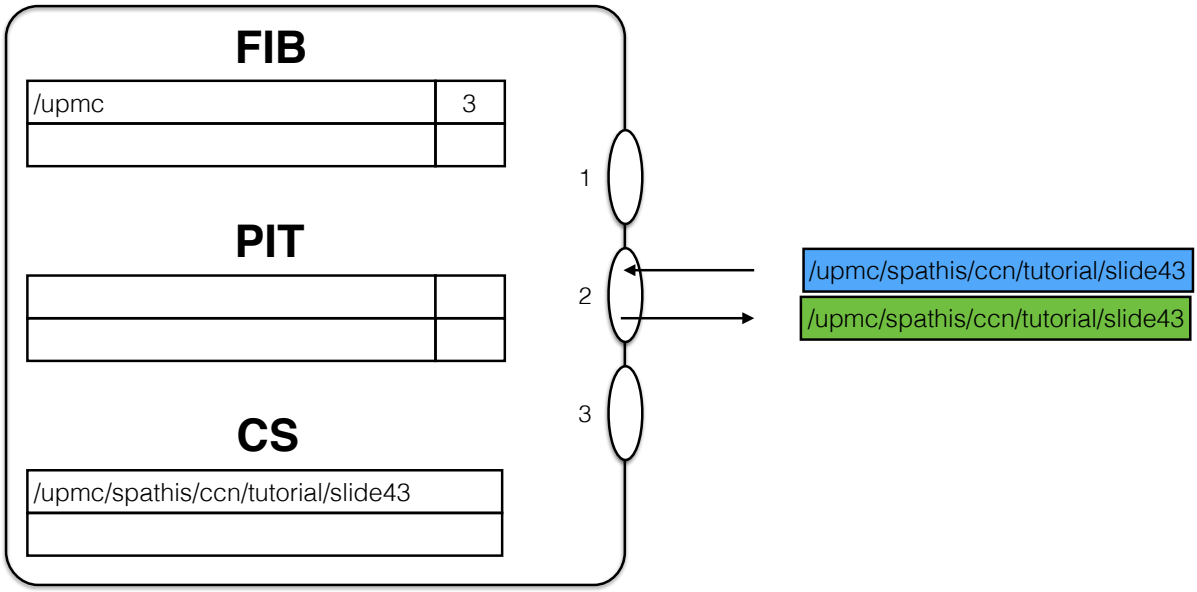
CCN Forwarder



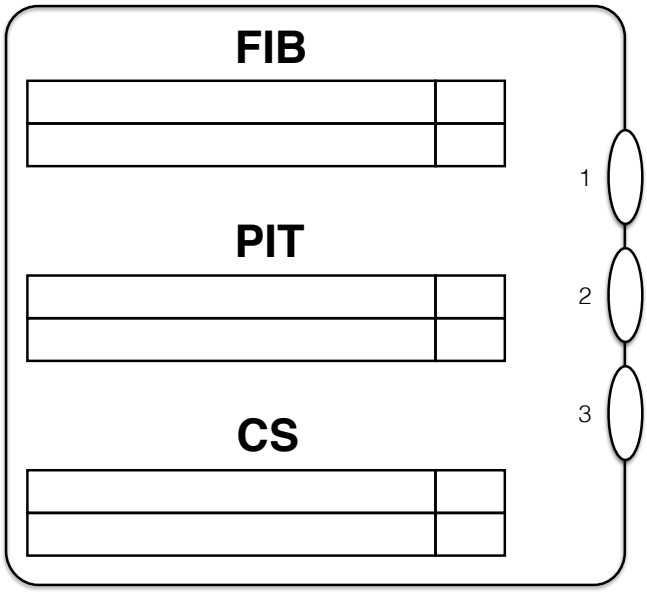
CCN Forwarder



CCN Forwarder



CCN Node Model

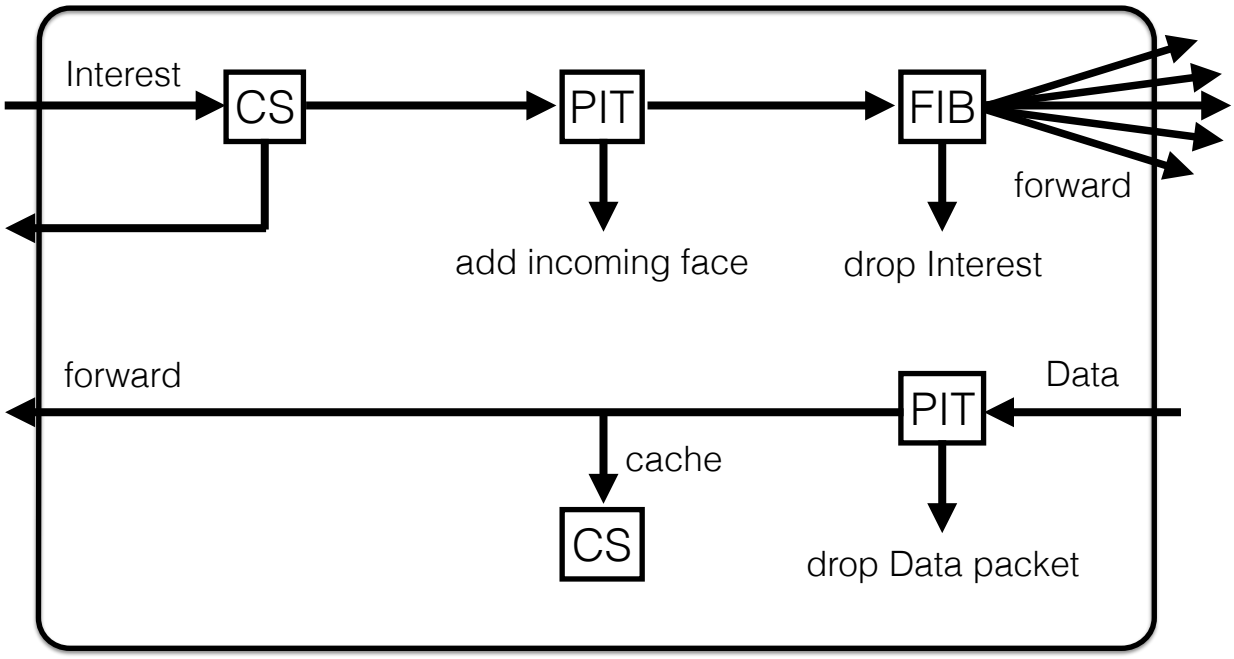


Longest matching prefix on names
Managed by external routing protocols

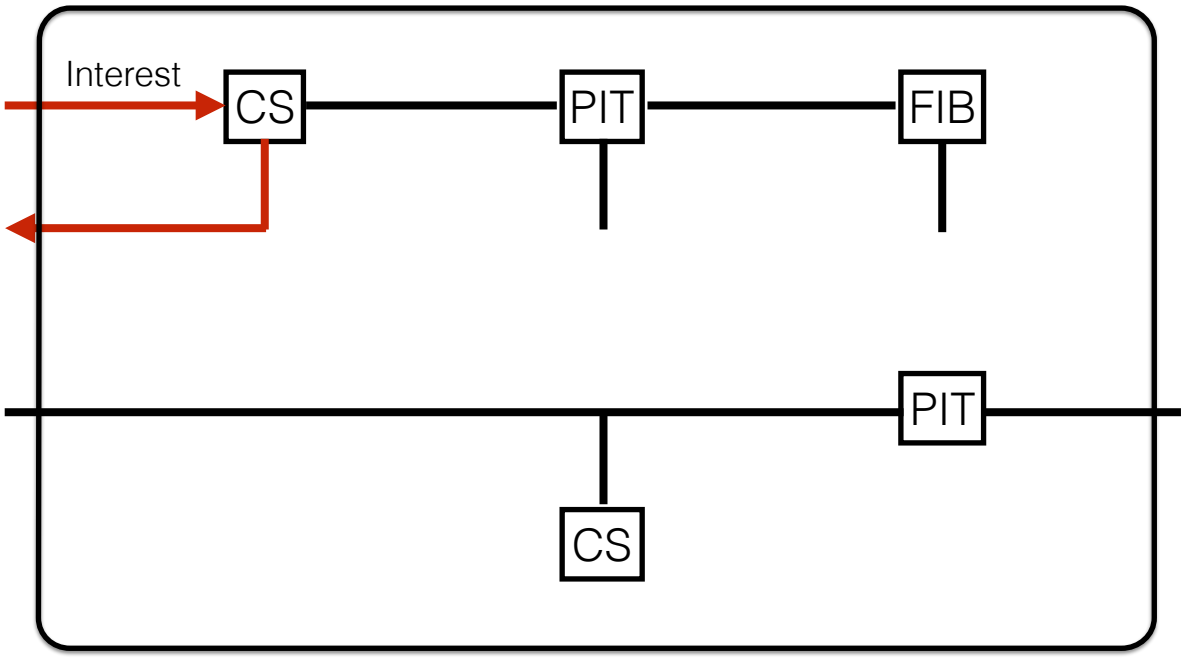
State for reverse-path forwarding

Long-term packet memory

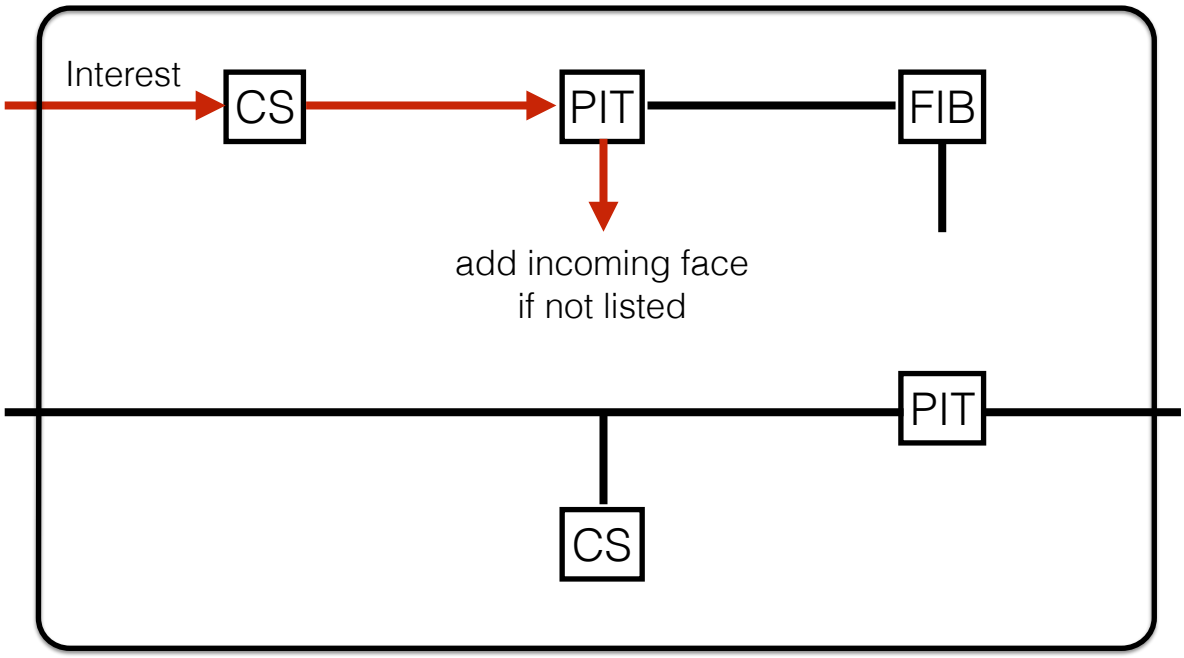
CCN Node Model



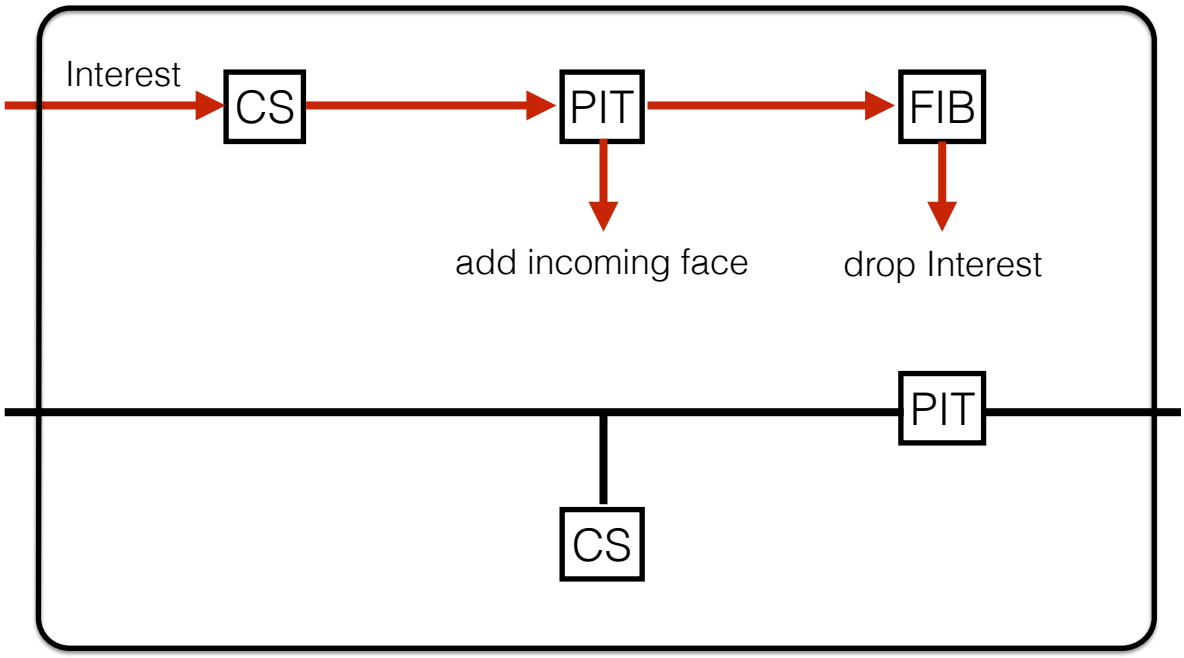
Interest satisfied by CS



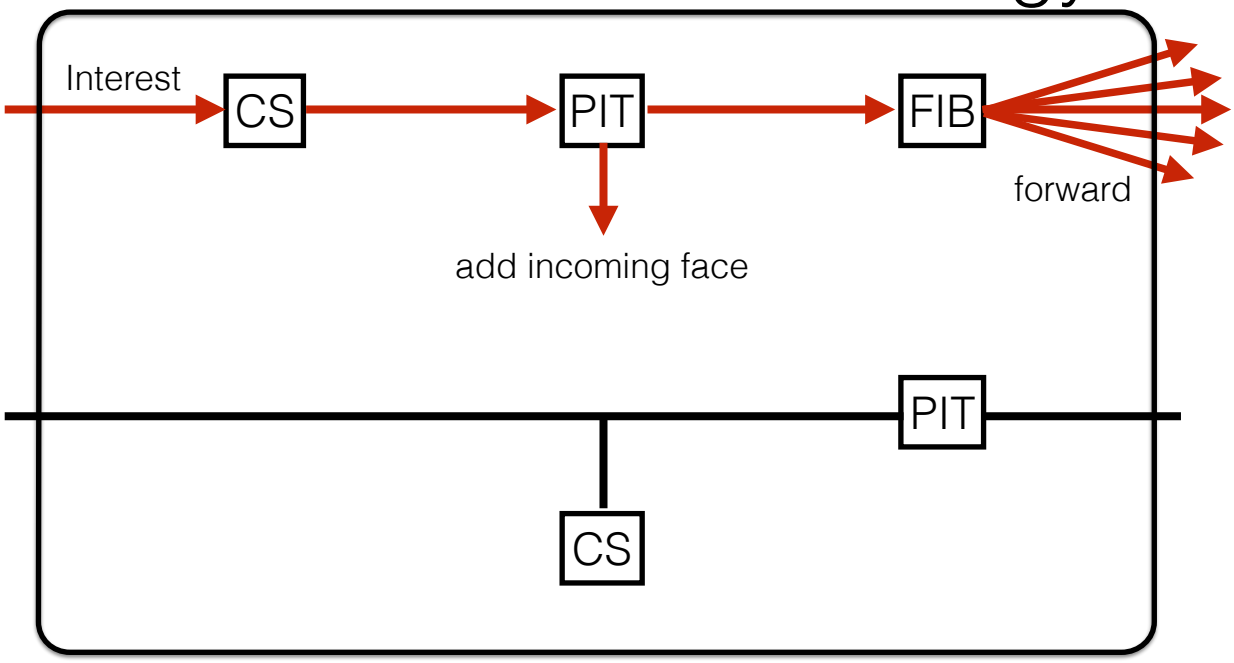
CS miss - Interest already in PIT



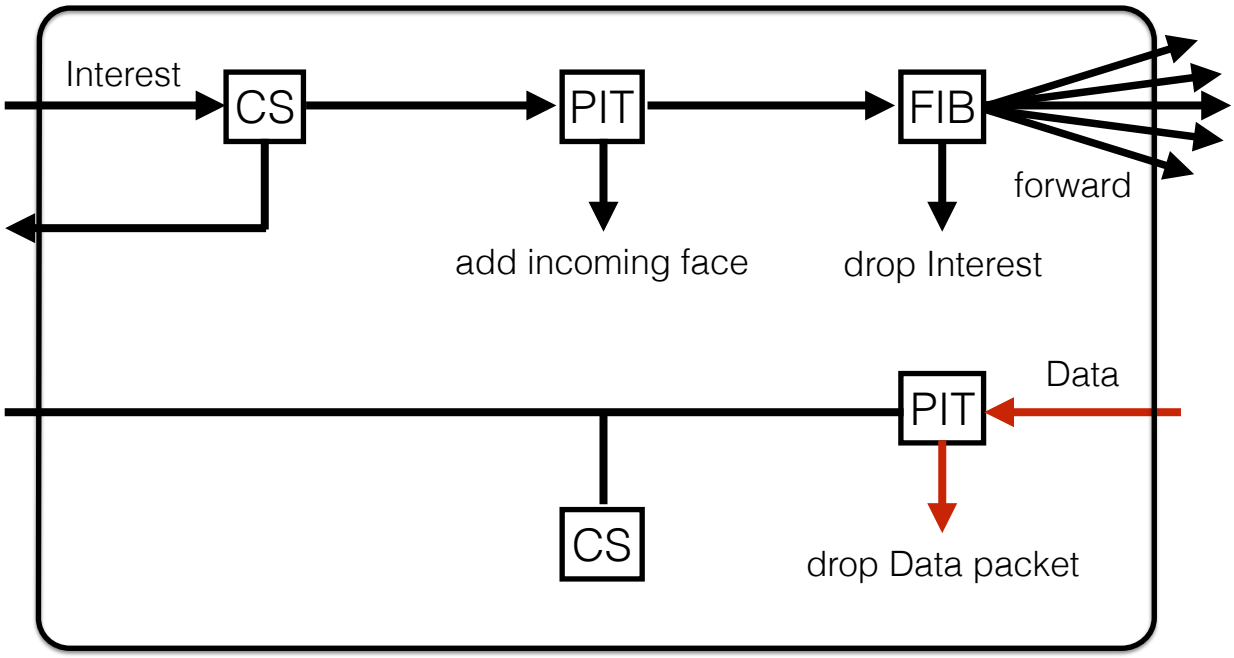
CS miss, PIT miss, no route in FIB for Interest



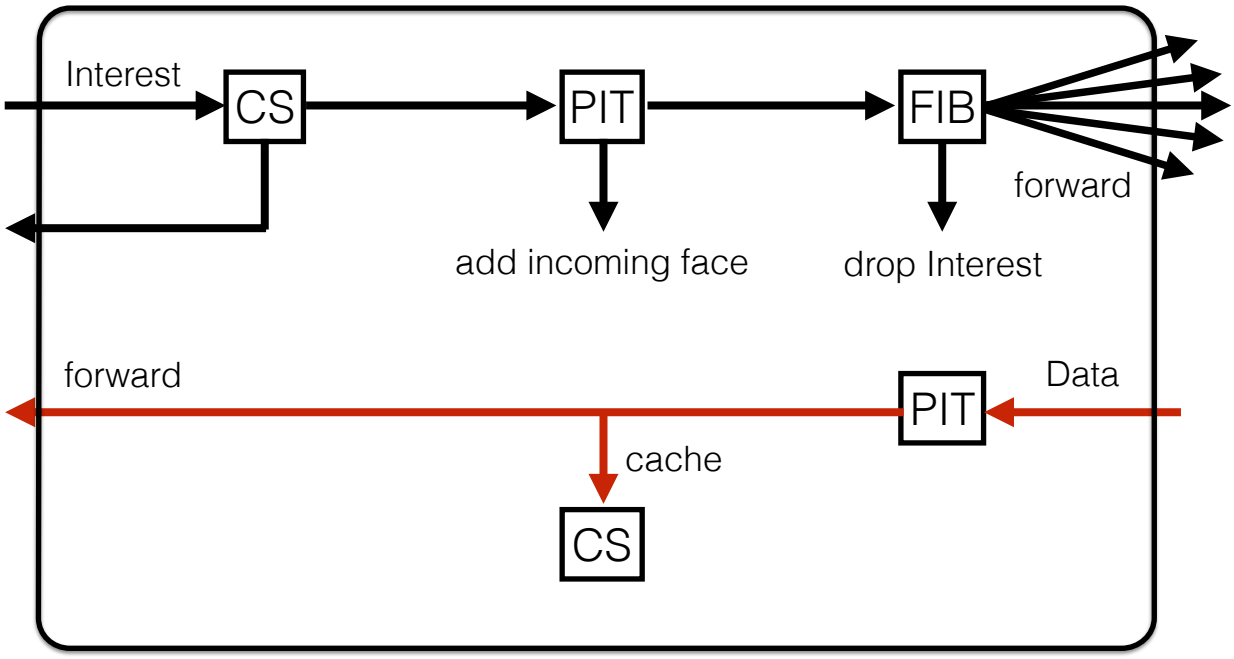
CS miss, PIT miss, route match in FIB - Forward via Strategy



PIT miss for Content



PIT match - Remove PIT entry, store copy in CS, reverse-path Forward



CCN Stateful Data Plane

- Named-based anycast and multicast delivery
  - Reverse-Path Forwarding
  - Scalable content distribution
  - Multipath forwarding
- Content Store
  - Offload sources for popular content
  - Retransmission buffer

CCN Data Plane Resilience

- CCN content delivery is a 2-step process:
  - Interest forwarding to set up state
  - Content traversal of interest path in reverse
- Content not forwarded w/out interests (i.e., request) for it
  - Multiple interests for same content are collapsed and one
  - copy of content per “interested” interface is returned
- Interest forwarding state eliminates looping, allows exploitation of topological redundancies and multipath forwarding
- Content packets measure quality of selected (interest) paths
  - Forwarding plane can incorporate congestion and fault mitigation into path decisions
  - Content caching increases availability & mitigates DoS attacks

CCN Routing

- Multiple copies of Content Objects
  - Opportunist replication, caching, or migration
- How to route towards one or all copies of the same object?
  - Source does not know any instances
  - Source knows all instances
  - In-between: Designated representative

Remember Multicast

- Flood and Prune (PIM-DM)
  - Reach all possible destinations, prune all except instances
- Link-state Multicast (MOSPF)
  - Topology is known so the location of the instances
  - Compute the source-tree to all instances
- Core-based Trees (aka PIM-SM)
  - Rendez-vous point in-between the source and the instances
  - Compute the tree rooted at the RV point

CCN Current Routing Proposals

- Network topology and location of all instances known
  - Named-data Link State Routing Protocol (NLSR)
- w/o knowing all, flooding, or predefined core

Named Link-State Routing NLSR Protocol (ICN'13)

- Naming routers and routing process

```
/<network>/<site>/<router>/NLSR
```

- Link-state advertisements
  - Adjacency LSAs (Neighbor 1 Name, Link 1 Cost, ...)

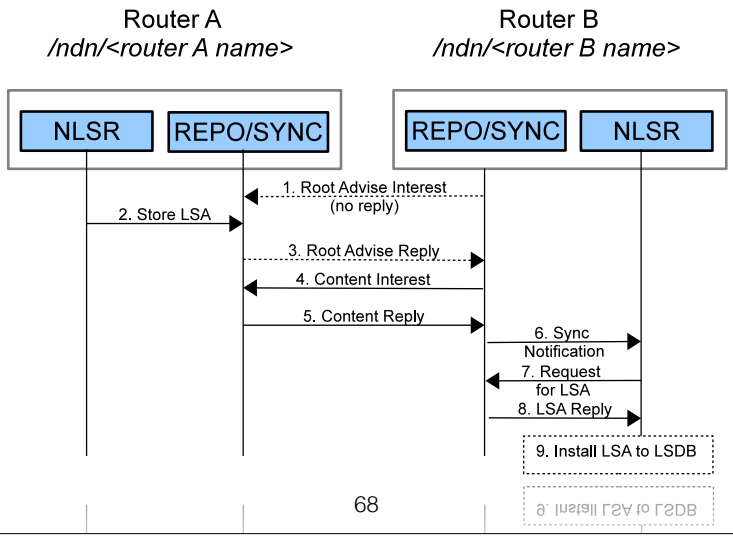
```
/<LSA-prefix> /<site>/<router>/LsType.1/<version>
```

  - Prefix LSAs (isValid, name prefix)

```
/<LSA-prefix>/<site> /<router>/LsType.2/LsId.<ID>/
                     <version>
```

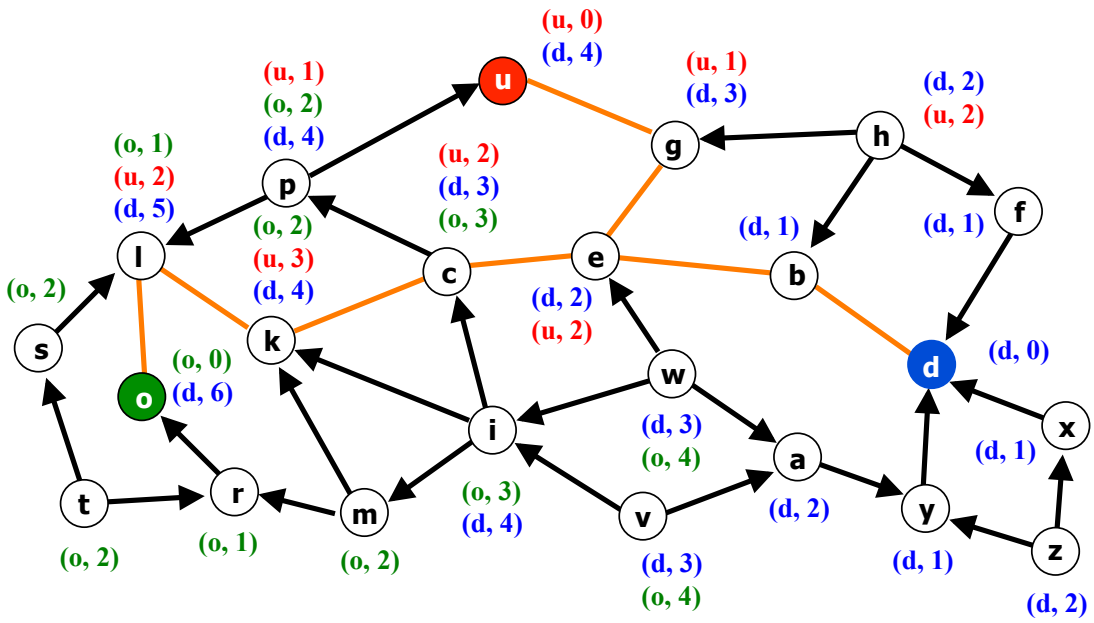

LSDB Synchronization

- Use of CCNx synchronization protocol
  - LSDB seen as a CCNx repository, collections of named data
  - Detection of inconsistencies through periodic exchanges of LSDB hashes



Distance-based Content Routing DCR (ICN'14)

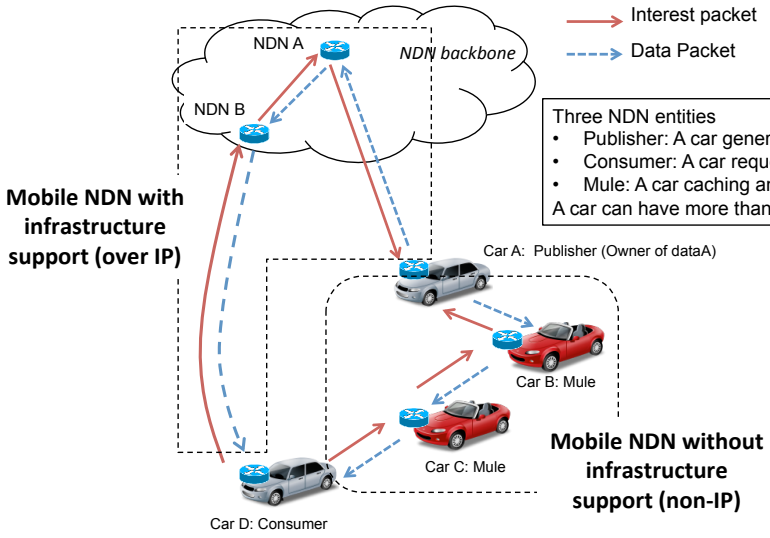
- Establish a lexicographic ordering of distances to multi-instantiated destination
- The name of a router "speaking for" a destination instance (called anchor) is an attribute used in the ordering
- Routers choose what to share with their peers (e.g., "the best distance according to the lexicographic ordering")
- Lexicographic ordering among instances defines an instance where a DAG spanning all instances is rooted



- Lexicographic ordering based on hop count to instance, ID of instance's anchor , sequence number from anchor
- Route to some *nearest* instances w/o knowing all, flooding, or predefined core

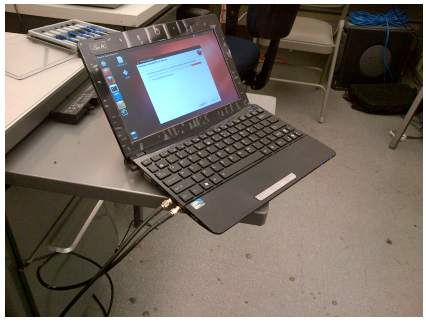
Vehicular Named Data Networking

Implemented a Linux-based NDN daemon, with enhancement to WiFi broadcast support



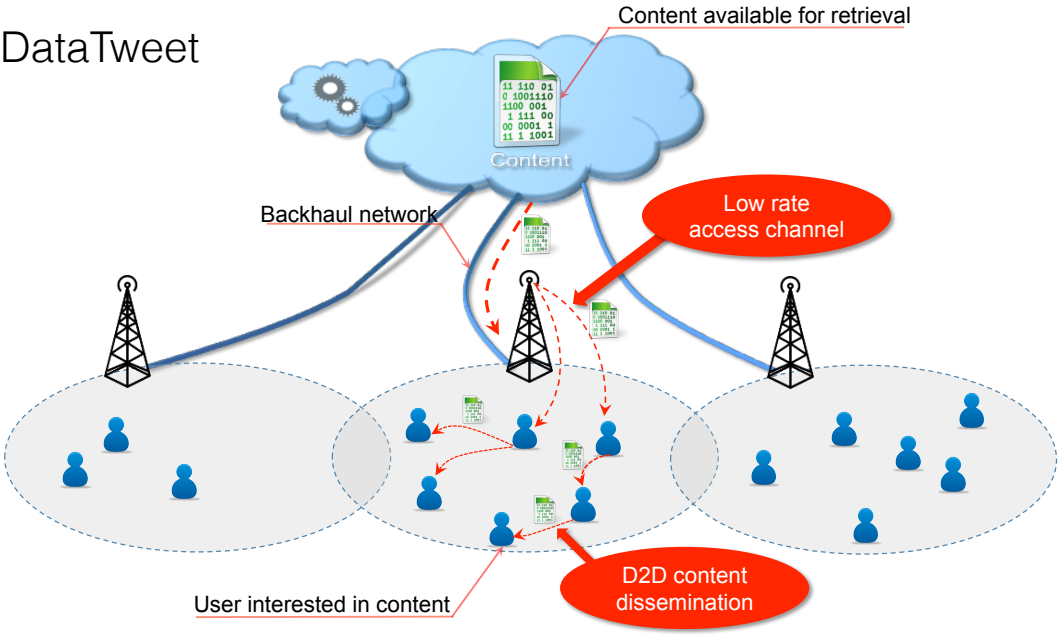
Three NDN entities
- Publisher: A car generating data
- Consumer: A car requesting data
- Mule: A car caching and forwarding data

A car can have more than one role at a time.



Content Centricity in Constrained Cellular-Assisted D2D Communications

ANR DataTweet



Key Issues

- Define Autonomous Systems for CCN, incorporate routing policies and an inter-domain trust model
- Applying CCN to solve real networking problems
  - Internet is already information-centric
    - youtube, netflix, amazon, facebook
    - new generations of applications
  - Solving info distribution problems via IP point-to-point communication, as we do today, is complex & error-prone

Analogy with TCP/IP in the 80s

- Promising new technology
- Largely unknown outside its small community
- Federal funding led TCP/IP to its success
  - BSD development, NSFnet
  - Various research projects over Internet
- A number of problems exposed and resolved through larger scale experimentation
  - DNS development
  - Congestion control
  - Evolution of the routing system
  - …

Takeaways

- Recognizing the right communication abstraction
- CCN focuses on retrieving data rather than conceptualizing communication between hosts
- Security, delivery efficiency, mobility and disruption tolerance than TCP/IP
- CCNx, NDN: Tools for experimenting new apps in emerging environments