Information Centric Networking

MSc in Computer Networking
January 20, 2017

Prométhée Spathis
promethee.spathis@upmc.fr

Salah-Eddine Belouanas
salah-eddine.belouanas@lip6.fr

Agenda

- A Brief History of Networking (+Motivation)
- Node Model
- Routing
- Transport

A Brief History of Networking

- Generation 1:

The phone system
(focus on the wires)

- Generation 2:

The Internet
(focus on the endpoints)

- Generation 3?

dissemination
(focus on the data)

The Phone System is not about phones, it's about connecting wires to other wires!

- The utility of the system depends on running wires to every home & office
- The wires are the dominant cost
- Revenue comes from dynamically constructing a path from caller to callee

A business model built on side effects

- For a telco, a call is *not* the conversation you have with your gf in Marseille, it's a *path* between two end-office line cards

- A phone number is not the name of your gf's phone, it's a program for the end-office switch fabric to build a path to the destination line card

Some ways to build paths

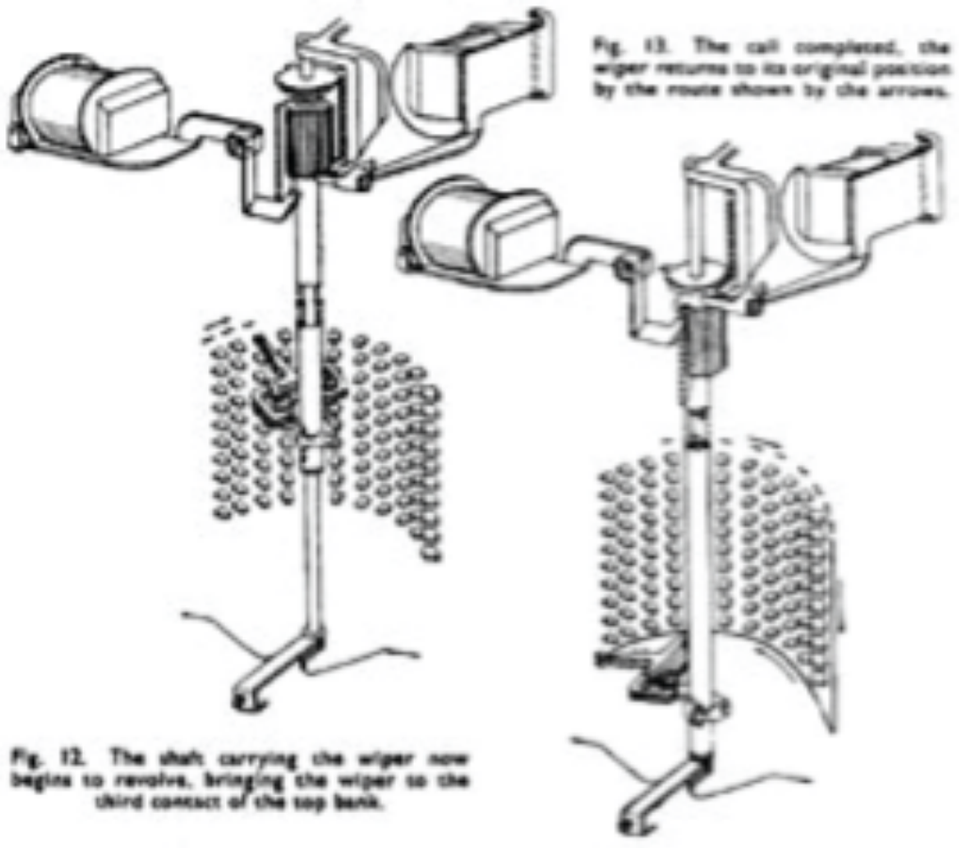


Fig. 13. The call completed, the wiper returns to its original position by the route shown by the arrows.

Fig. 12. The shaft carrying the wiper now begins to revolve, bringing the wiper to the third contact of the top bank.

Structural problems with phone systems

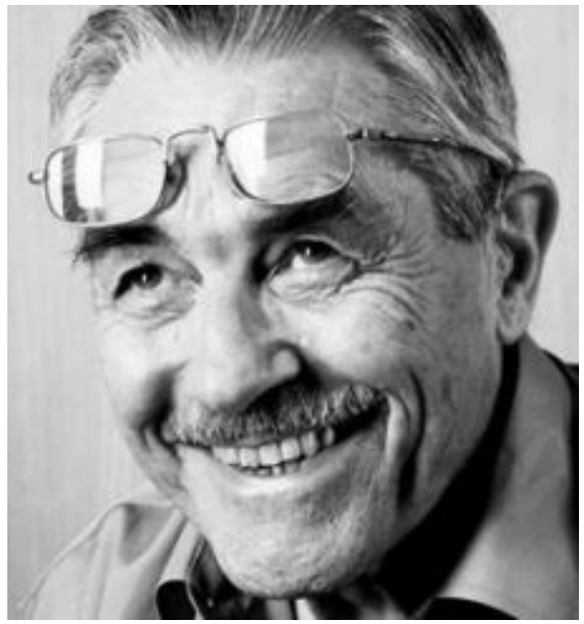
- Path building is non-local and encourages centralization and monopoly

- Calls fail if any element in path fails so reliability goes down exponentially as the system scales up

- Data can’t flow until a path is set up so efficiency decreases when setup time or bandwidth increases or holding time decreases

Gen 2: Packet switching

- Change point view to focus on endpoints rather than paths
- Data sent in independent chunks and each chunk contains the name of the final destination
- (Transitivity) If node gets a chunk for a different destination, tries to forward it using static configuration or distributed routing computation



Packet switching used the existing wires, it just used them differently

On Distributed Communications Networks

PAUL BARAN, SENIOR MEMBER, IEEE

Summary—This paper[1] briefly reviews the distributed communication network concept in which each station is connected to all adjacent stations rather than to a few switching points, as in a centralized system. The payoff for a distributed configuration in terms of survivability in the cases of enemy attack directed against nodes, links or combinations of nodes and links is demonstrated.

A comparison is made between diversity of assignment and perfect switching in distributed networks, and the feasibility of using low-cost unreliable communication links, even links so unreliable as to be unusable in present type networks, to form highly reliable networks is discussed.

The requirements for a future all-digital data distributed network which provides common user service for a wide range of users having different requirements is considered. The use of a standard format message block permits building relatively simple switching mechanisms using an adaptive store-and-forward routing policy to handle all forms of digital data including digital voice. This network rapidly responds to changes in the network status. Recent history of measured network traffic is used to modify path selection. Simulation results are shown to indicate that highly efficient routing can be performed by local control without the necessity for any central, and therefore vulnerable, control point.

Introduction

LET US CONSIDER the synthesis of a communication network which will allow several hundred major communications stations to talk with one another after an enemy attack. As a criterion of survi-

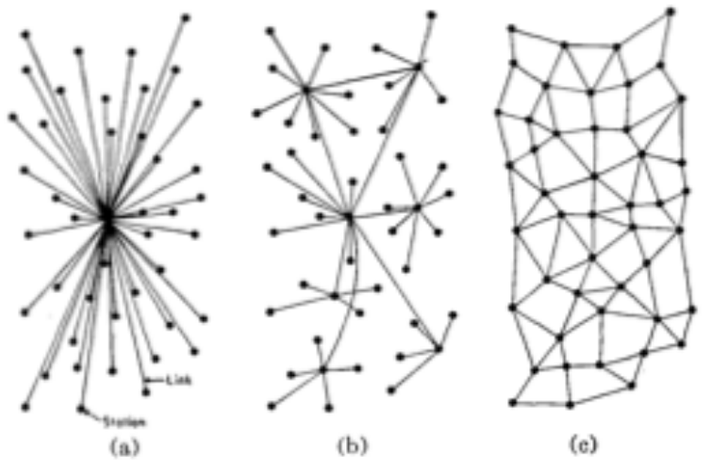


Fig. 1—(a) Centralized. (b) Decentralized. (c) Distributed networks.

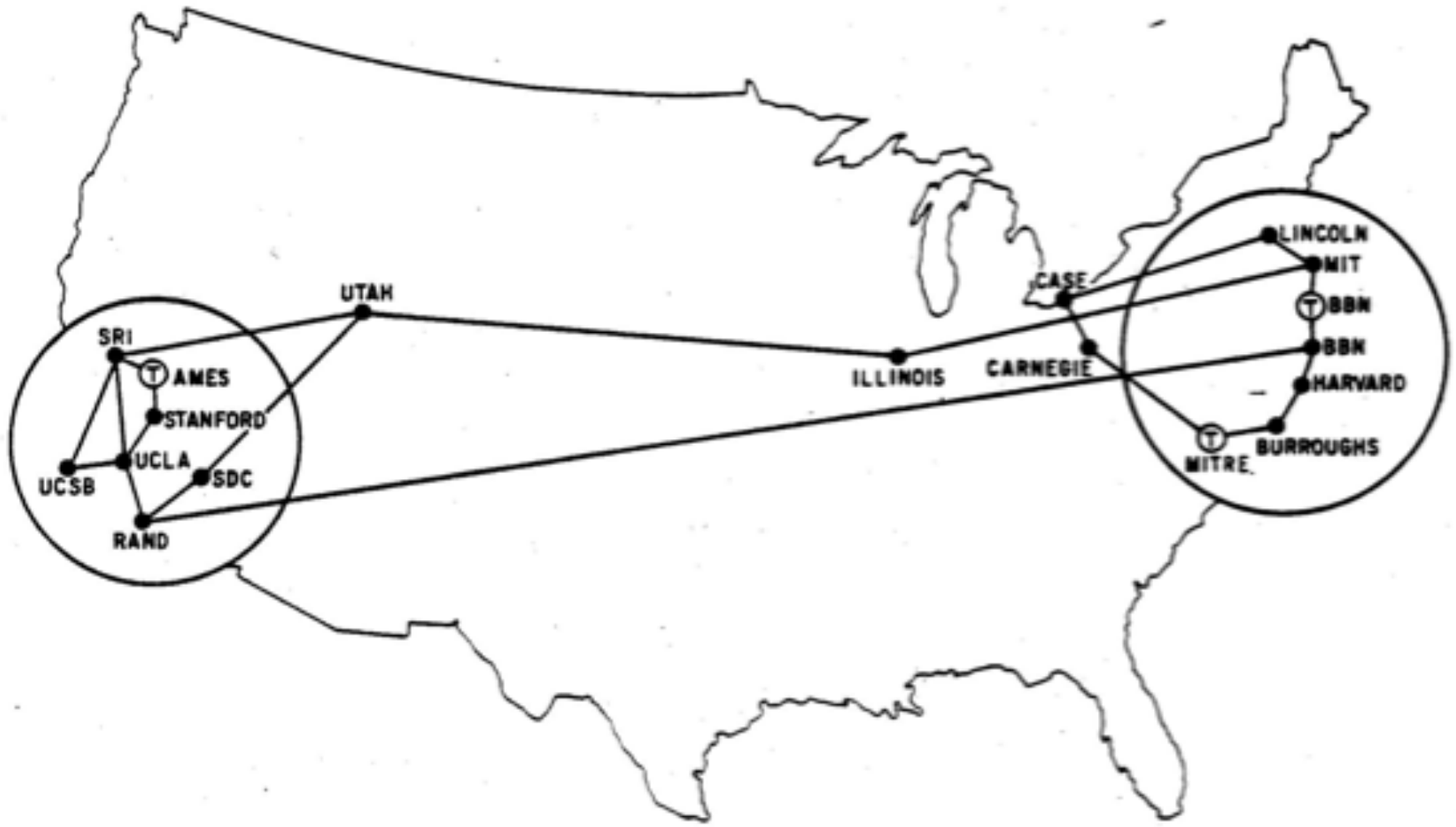
loop. Such a network is sometimes called a "decentralized" network, because complete reliance upon a single point is not always required.

Examination of a Distributed Network

Since destruction of a small number of nodes in a decentralized network can destroy communications, the properties, problems, and hopes of building "distributed"

In 1964 these ideas were 'lunatic fringe' — anyone who knew anything about communications knew this could never work.

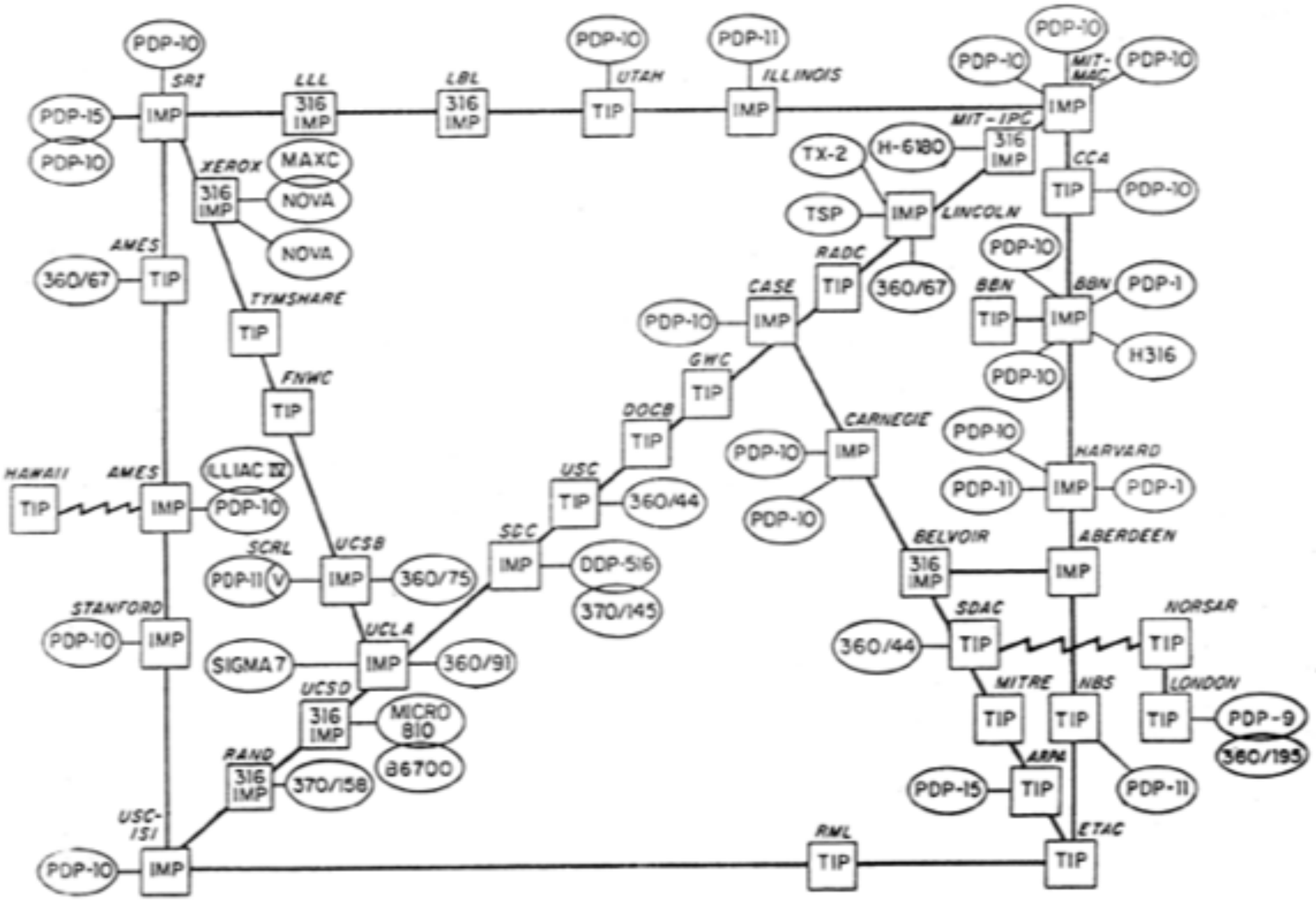
ARPAnet
September, 1971



ARPAnet

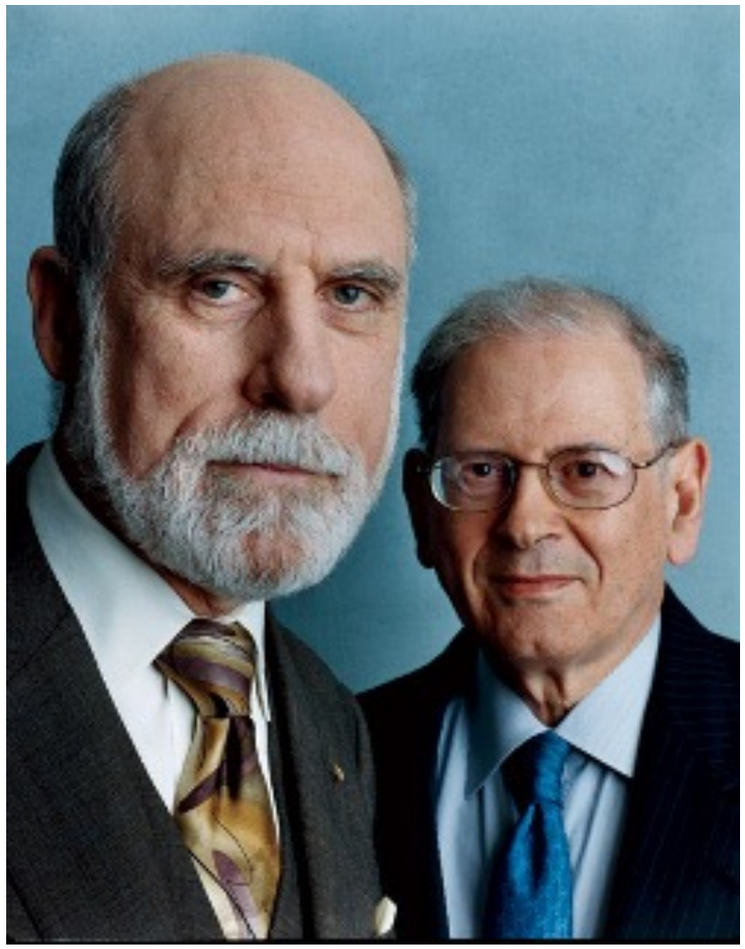
- The ARPAnet was built on top of the existing phone system
- It needed cheap, ubiquitous wires
- It needed a digital signaling technology (but not anything like the state of the art)
- At the outset, the new network looked like an inefficient way to use the old network
- The rest of the research community put enormous effort into the details of circuit switched data. In the end it didn't matter

ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973

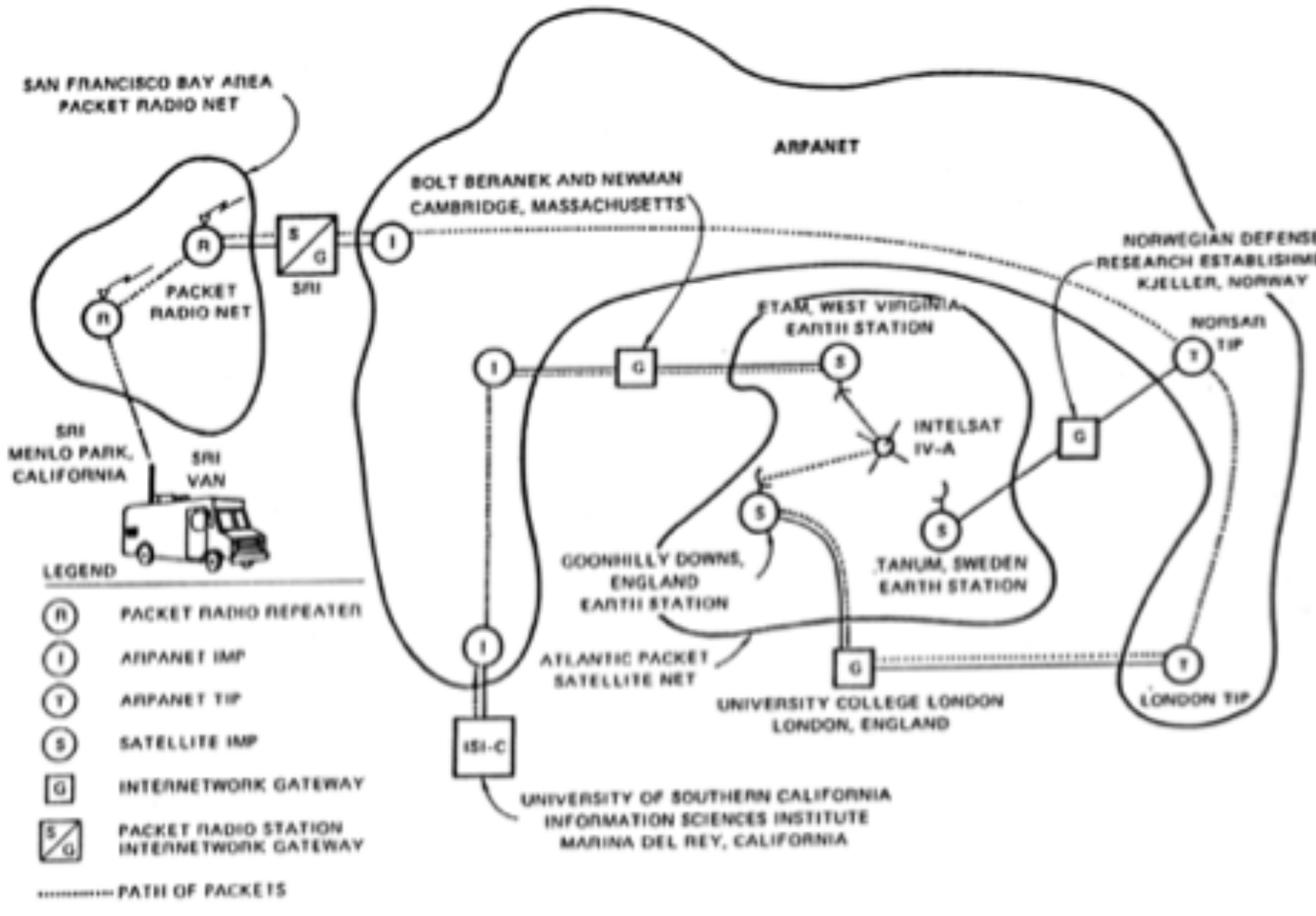


The CATENET and TCP/IP

- Packet switching worked so well that by 1973 everyone wanted a network
- Each was done as a clean slate so they didn't interoperate
- Since Paul Baran had already abstracted out all the topological details Vint Cerf realized that a common encapsulation & addressing structure could glue together arbitrary networks

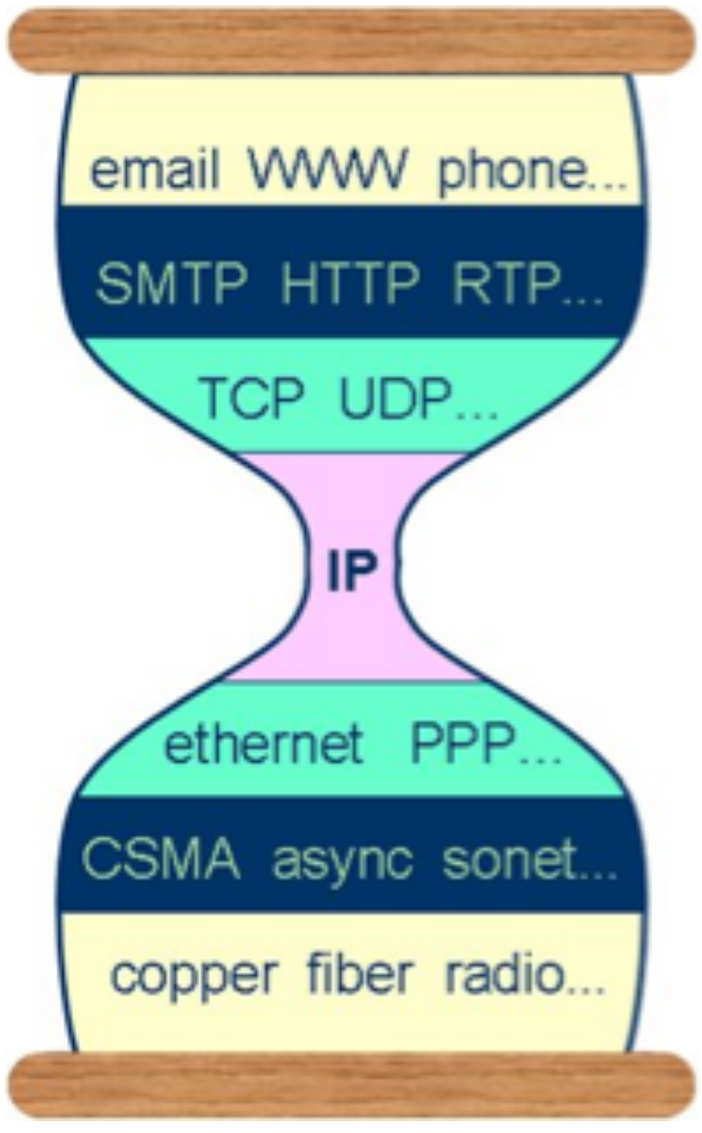


Multinetwork Demonstration 1977



TCP/IP wins

- Adaptive routing lets system repair failures and hook itself up initially.
- Reliability increases exponentially with system size.
- No call setup means high efficiency at any bandwidth, holding time or scale.
- Distributed routing supports any topology and tends to spread load and avoid a hierarchy's hot spots.



TCP/IP issues

- "Connected" is a binary attribute: you're either part of the internet and can talk to everything or you're isolated.

- Becoming part of the internet requires a globally unique, globally known IP address that's topologically stable on routing time scales (minutes to hours).

  - connecting is a heavy weight operation

  - the net doesn't like things that move

- Like the phone system before it, TCP/IP solved the problems it set out to solve so well that even today it's hard to conceive of an alternative.

- TCP/IP's issues don't reflect an architectural failing but rather its massive success in creating a world rich in information & communication.

- When TCP/IP was invented there were few machines and many users per machine. Today there are many machines and many machines per user, all with vast amounts of data to be synchronized & shared.

- And that creates an entirely new class of problem . . .

Once IP Adresses used to rule the Internet

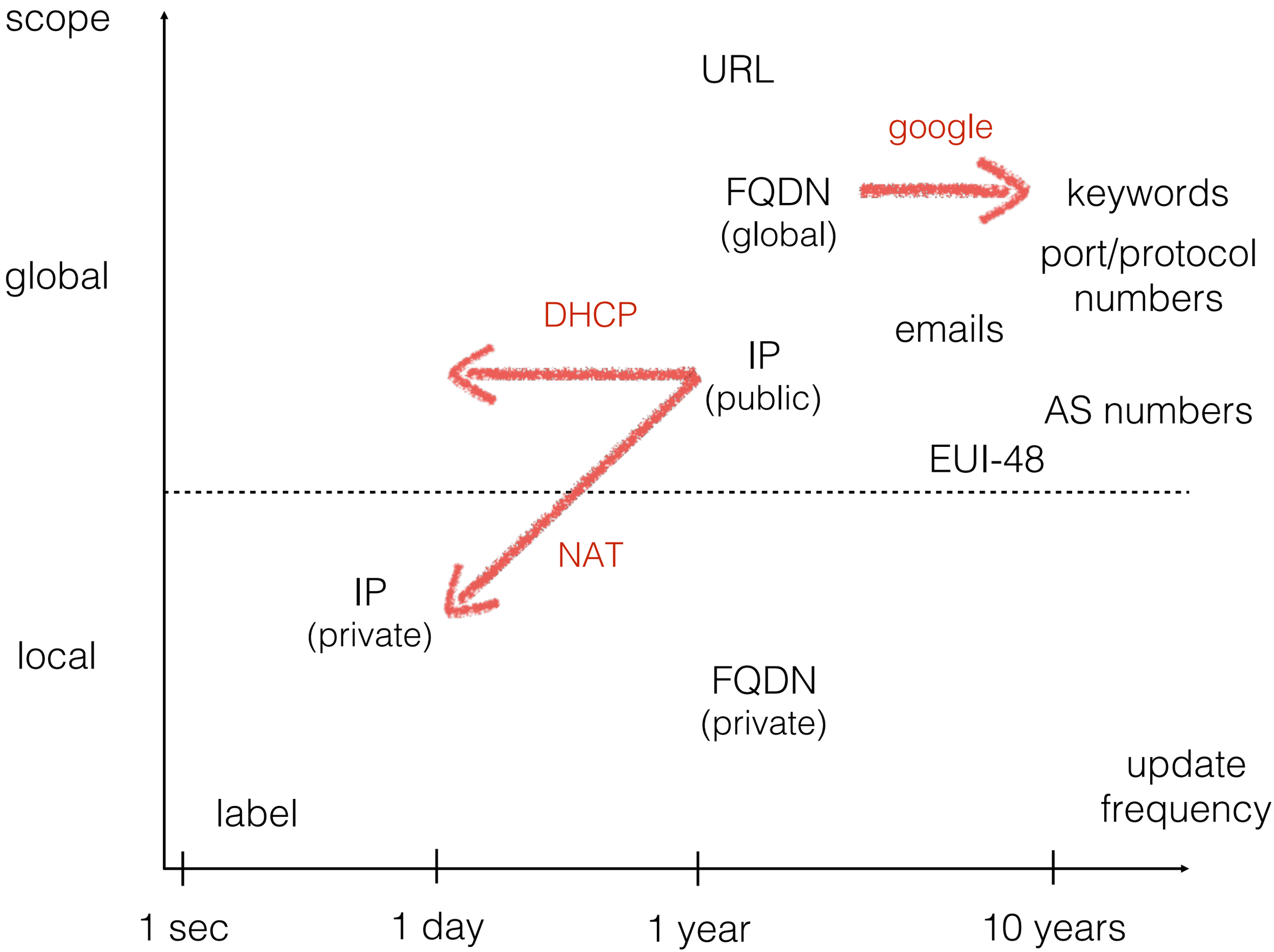
- In the ARPANET, addresses were fixed and could actually be used as "long-lived names"
  - Address 1 is UCLA-NMC [RFC-597, 1973]
  - HOST.TXT file is mapping a static symbol into another static symbol
- Same with the early Internet
  - 1.0.0.0/8 is BBN-PR [RFC-820, 1982]
- After the introduction of DNS in 1984 addresses are still very long-lived: RFCs still contain the static list of assigned addresses
  - estimated number of hosts ≈ 1,000
- RFC-990 (1986) is the last one to contain a list of assigned addresses
  - estimated number of hosts <≈ 10,000

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


scope
global
local
URL
google
FQDN
(global)
keywords
port/protocol
numbers
DHCP
emails
IP
(public)
AS numbers
EUI-48
NAT
IP
(private)
FQDN
(private)
update
frequency
label
1 sec
1 day
1 year
10 years

- The raison d'être of today's networking, both circuit switched and TCP/IP is to allow two machines to have a conversation

- The overwhelming use (>99% by most measurements) of today's networks is for a machine to acquire named chunks of data (like web pages, or email messages)

Acquiring named chunks of data is not a conversation, it's a "dissemination" (the computer equivalent of "Does anybody have the time?")

IP Addresses Bashing

- Mobility/Migration: IP adds change over time
- NAT/DHCP: IP scope is shrinking
- 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?

In a dissemination the data matters, not the supplier

- It’s possible to disseminate via conversation and get the data as a side effect, But:

- Security is an afterthought. Channels are secured, not data, so there’s no way to know if what you got is complete, consistent or even what you asked for.

- It’s inefficient (hotspots, poor reliability, poor utilization).

- Users have to do the translation between their goal & its realization and manually set up the plumbing to make things happen.

Dissemination networking

- Data is request by name, using any and all means available (IP, VPN tunnels, multicast, proxies, etc).

- Anything that hears the request and has a valid copy of the data can respond.

- The returned data is signed, and optionally secured, so its integrity & association with name can be validated (data centric security)

Design Usage Mismatch

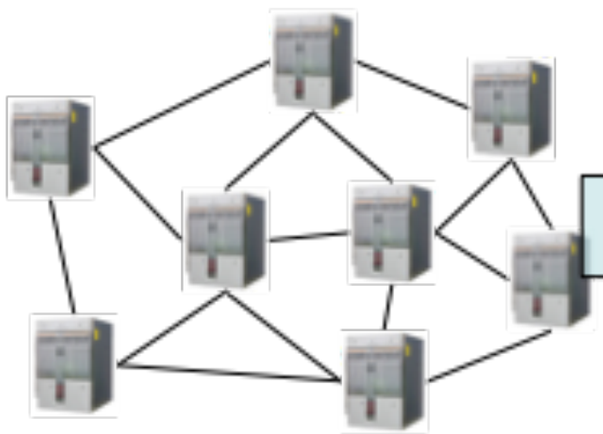
- Internet was designed for host-to-host communication
  - "contact this host…"
- Internet is mainly used for data access
  - "get me this data....."
- Mismatch between usage and design:
  - data migration and replication unnecessarily hard
  - requires Akamai- and BitTorrent-like designs to scale
  - mobility and multi-homing pose problems

what would the Internet look like if we designed it around data access?

Information Centric Networking

Today's Internet

Focus on
nodes



In today's Internet, accessing information is the dominating use case!

Information Centric Network

Focus on
information objects

- 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

Content Centric Networking (CCN): Goals

- Create a simple, universal, flexible communication architecture that:

  - Matches today's communication problems

  - Matches today's application design patterns

  - Is at least as scalable & efficient as TCP/IP

  - Is much more secure

  - Requires far less configuration

Universal?

- Any architecture that runs over anything is an overlay (IP is an overlay).

- IP started as a phone system overlay; today much of the phone system is an IP overlay. System theorists would say ‘IP is universal’.

- CCN has the same character: it can run over anything, including IP, and anything can run over CCN, including IP.

- And CCN has a simpler, more general relationship with lower layers than IP.

There are two ways to view CCN

- a ‘universal’ middleware
- an IP for content

- The difference between these is deployment time horizon

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

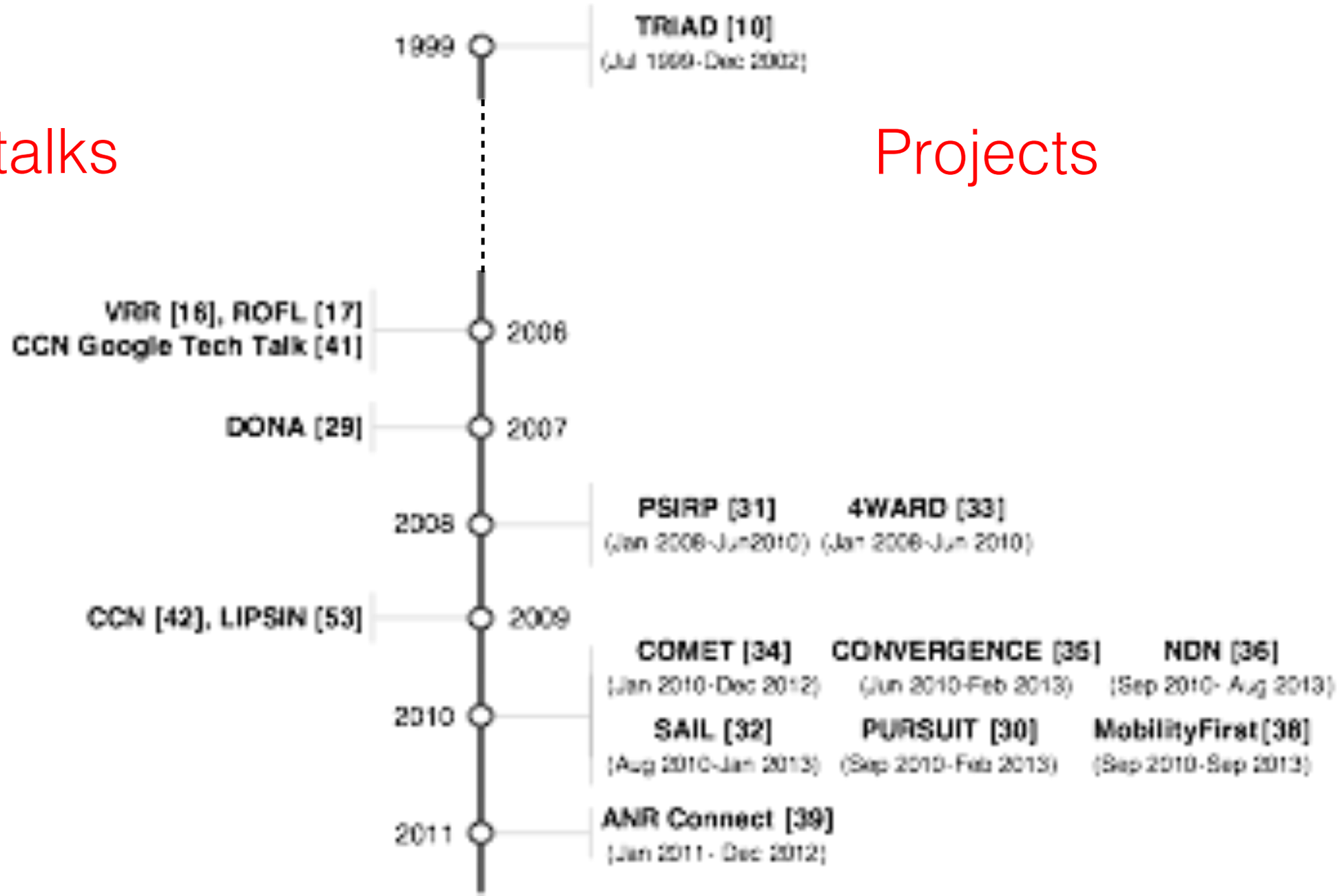
  - Longest prefix match forwarding

  - Link-state protocols (i.e. OSPF) opaque capacities

Timeline of key ICN milestones

Papers/talks

Projects

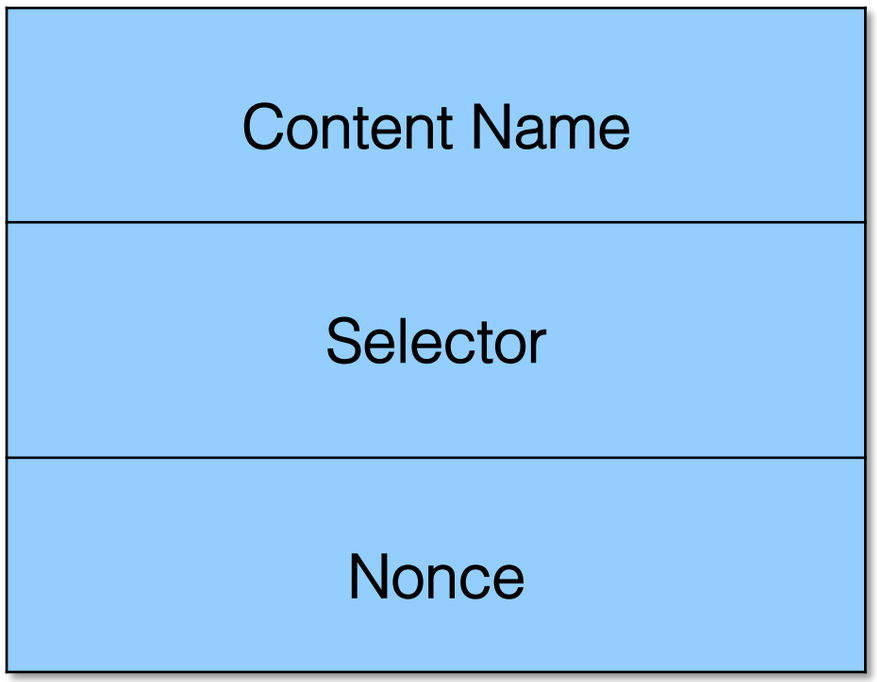


CCN Timeline

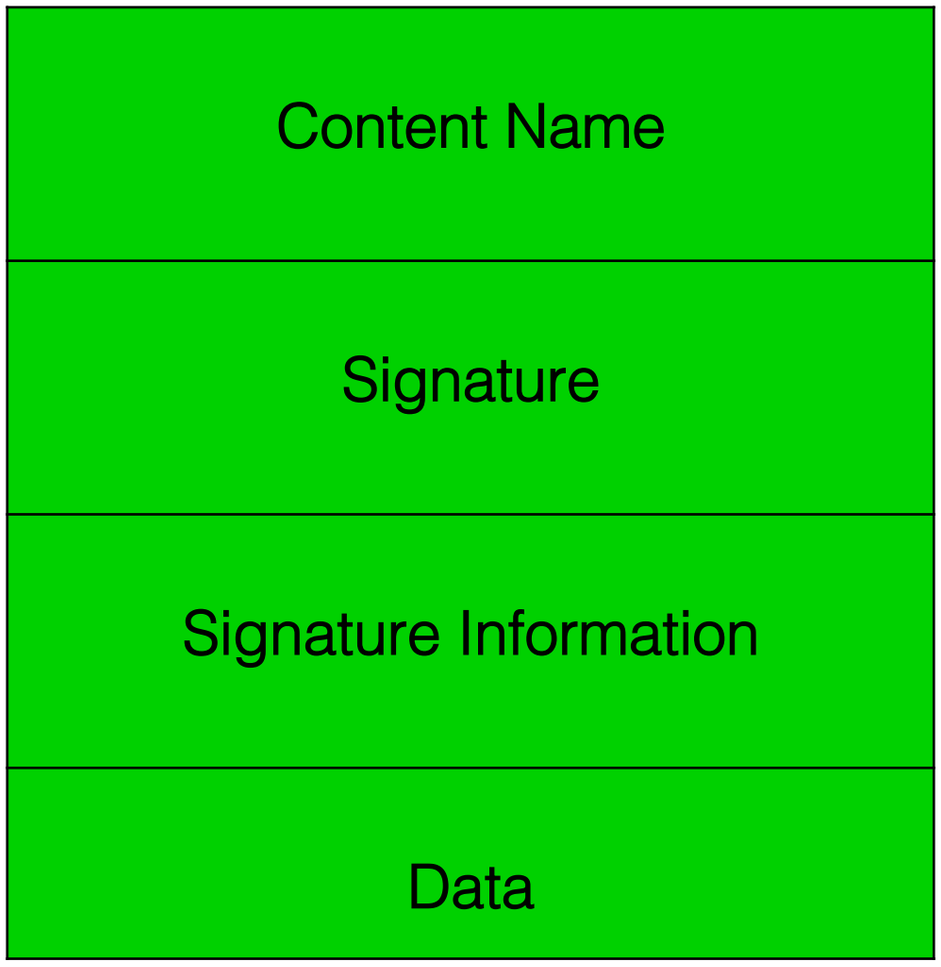
- 2005: PARC starts work on ICN
- 2006: Van Jacobson's talk at Google
- 2007: CCN 0.x begins
- 2009: Seminal CCN Paper at CoNext 2009
- 2010: NSF-funded NDN project
- 2013: CCN 1.x begins
- 2014: NDN 2nd phase (w/o Parc)

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)

Interest/Data packets



Interest packet



Data packet

Interest/Data Exchange

Interest

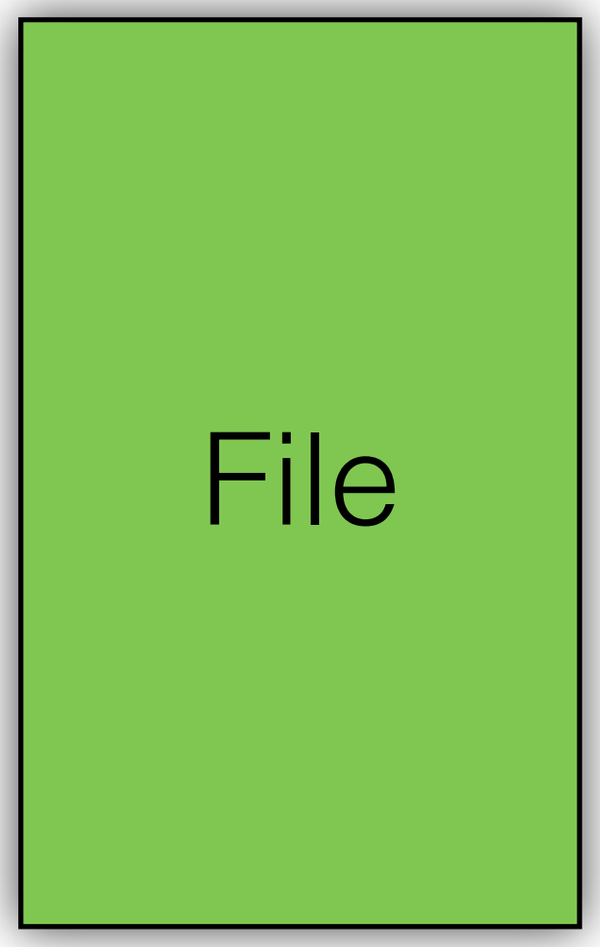
/upmc/spathis/ccn/index.html/manifest/c0

/upmc/spathis/ccn/index.html/manifest/c0
c0/h6245 c1/h9243

Data

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

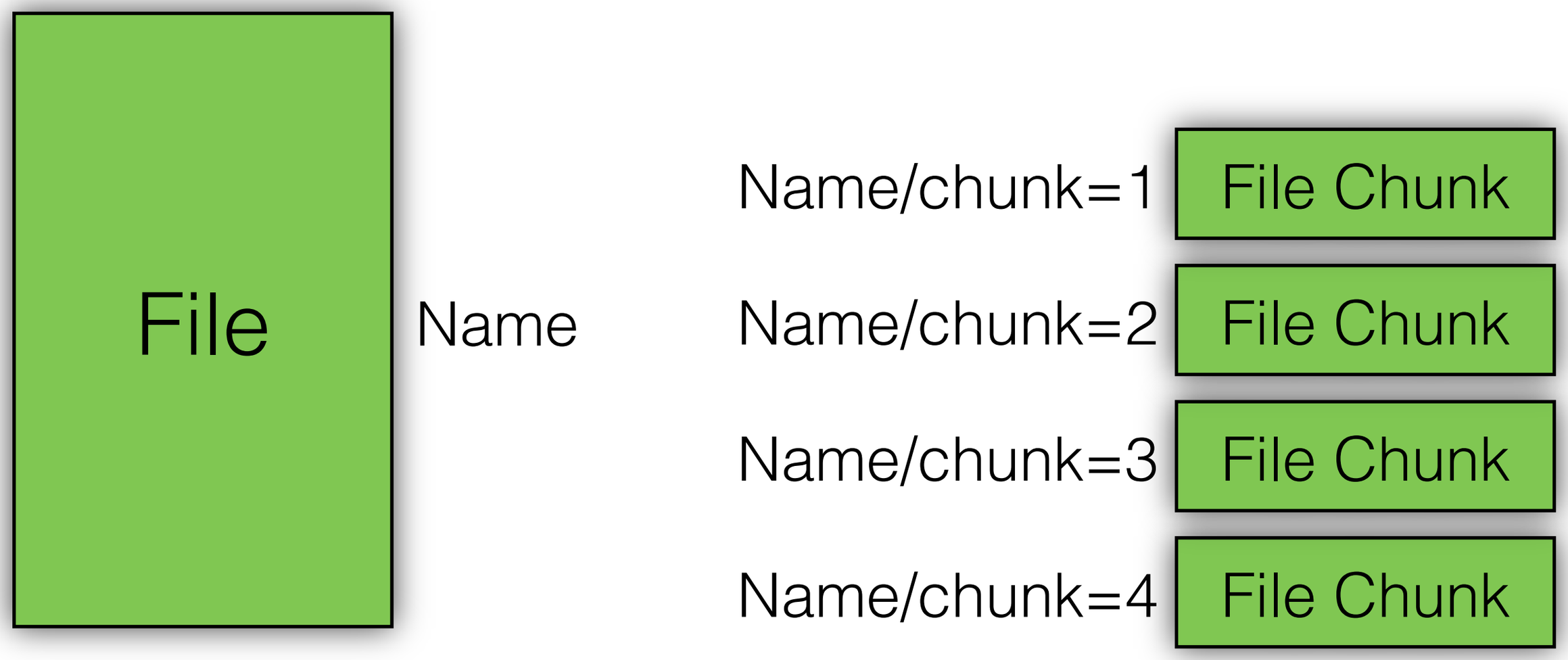
Named Object



Name

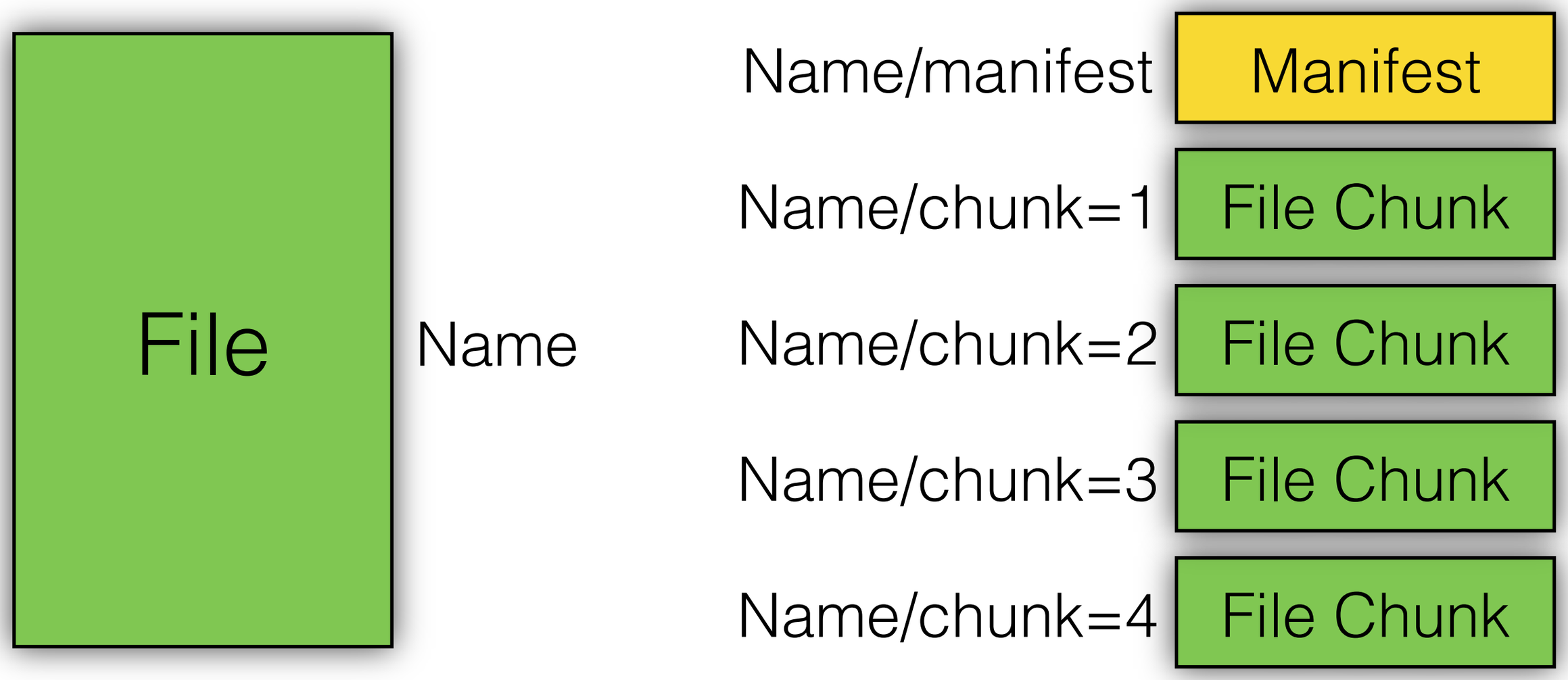
A large object has a CCN publisher-given name

Named Chunks



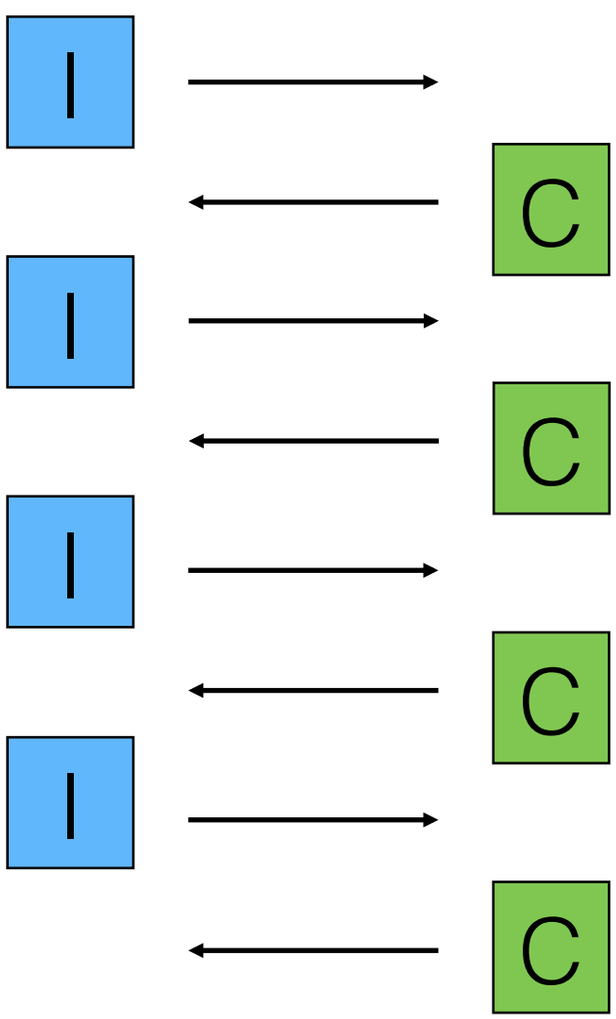
Each network-sized chunk also has a CCN name

Named Chunks

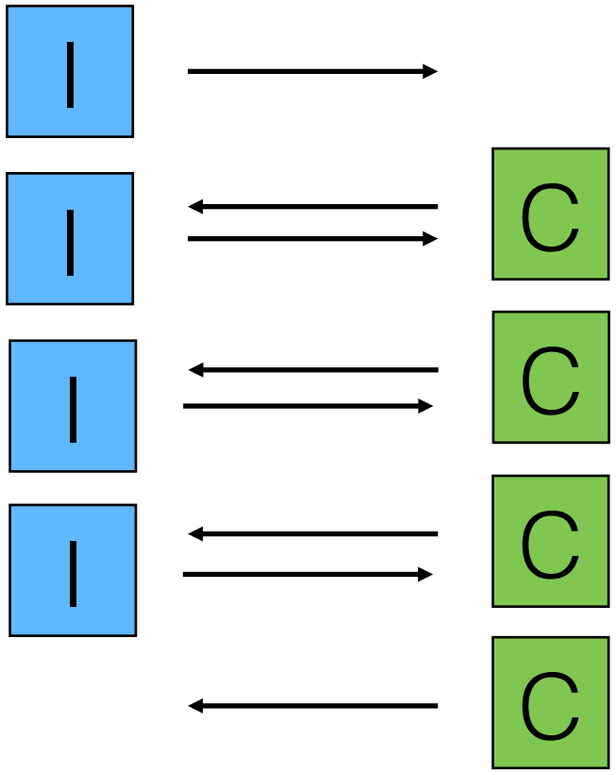


CCN creates a manifest describing the file

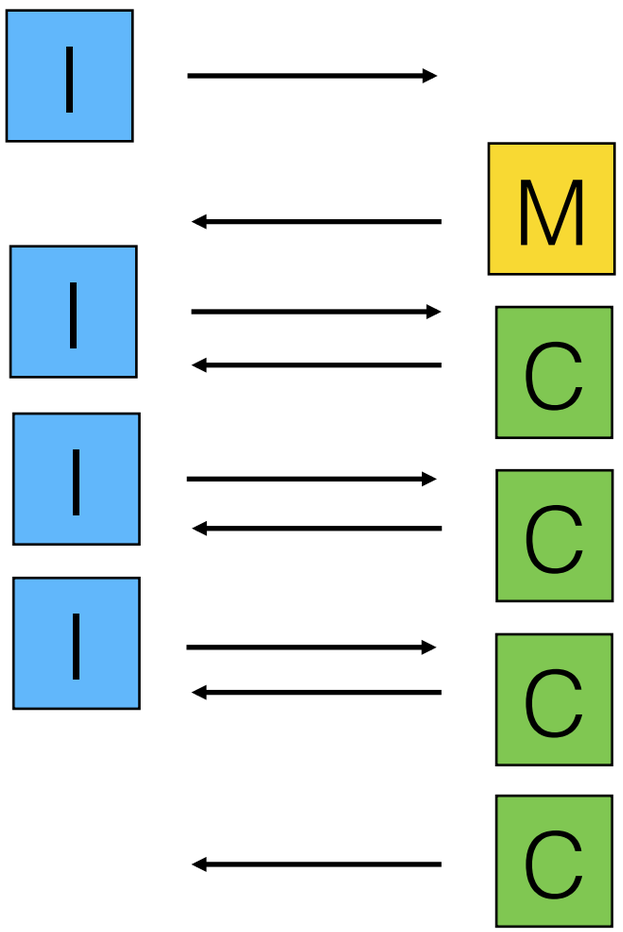
Transport Protocols



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

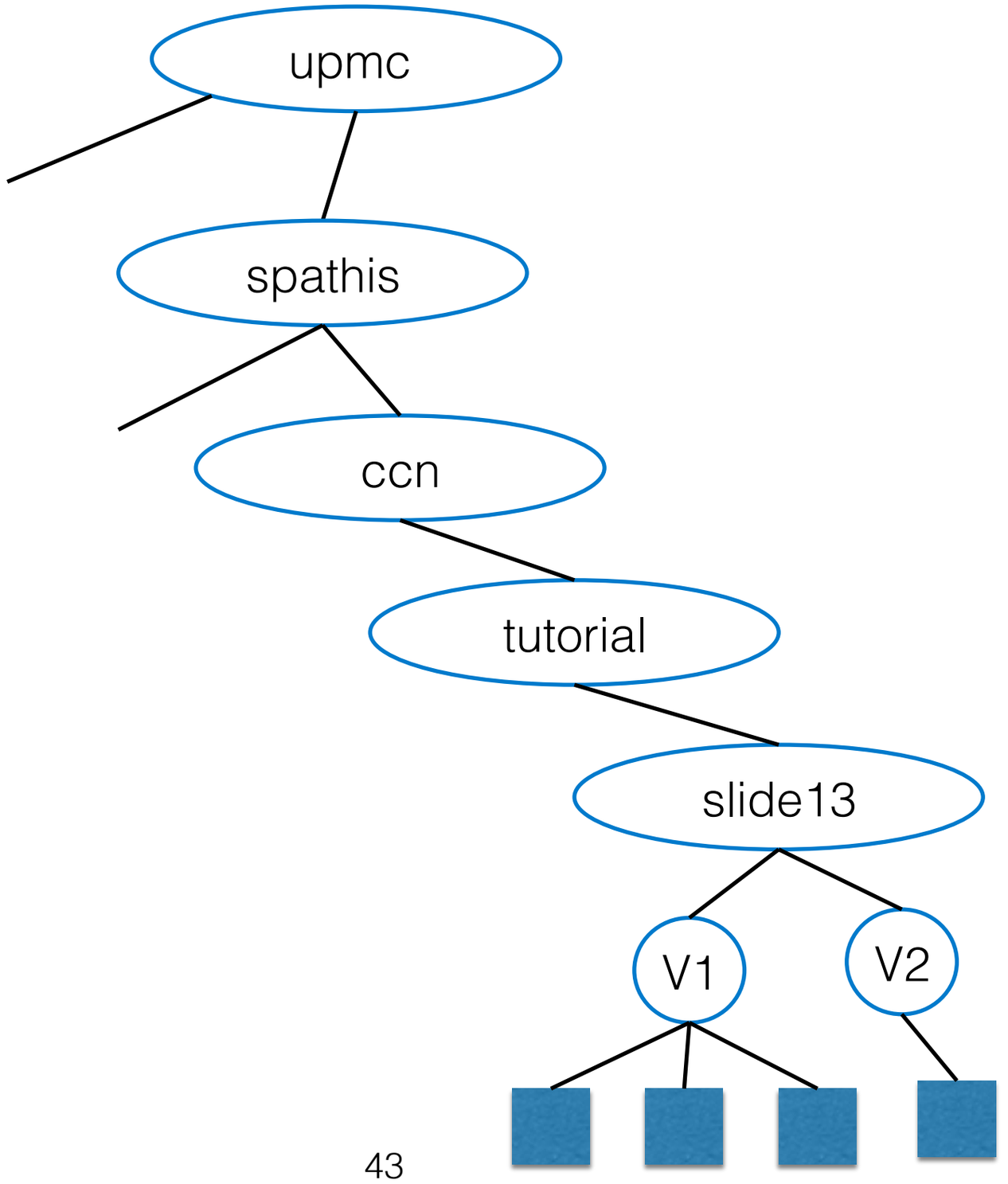
global routable name segments

application dependent name segments

protocol dependent name segments

Ordered labeled sequence of binary segments

Name tree traversal



RightmostChild

LeftmostChild

CCN Names Qualifier

Interests can identify content hash or publisher's key

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

ContentObjectHash

↓

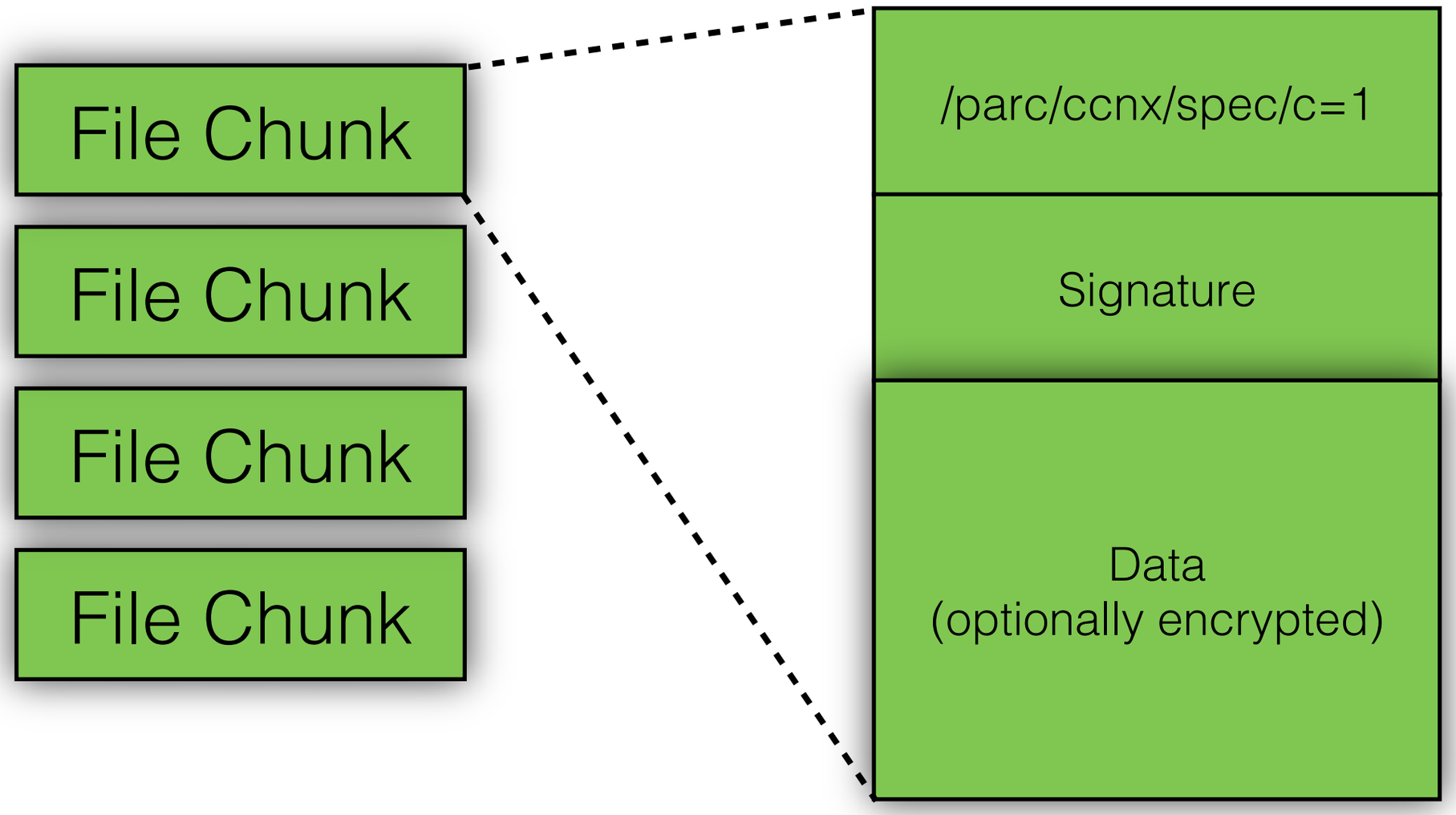
secure cryptographic hash of the Content Object message

KeyId

↓

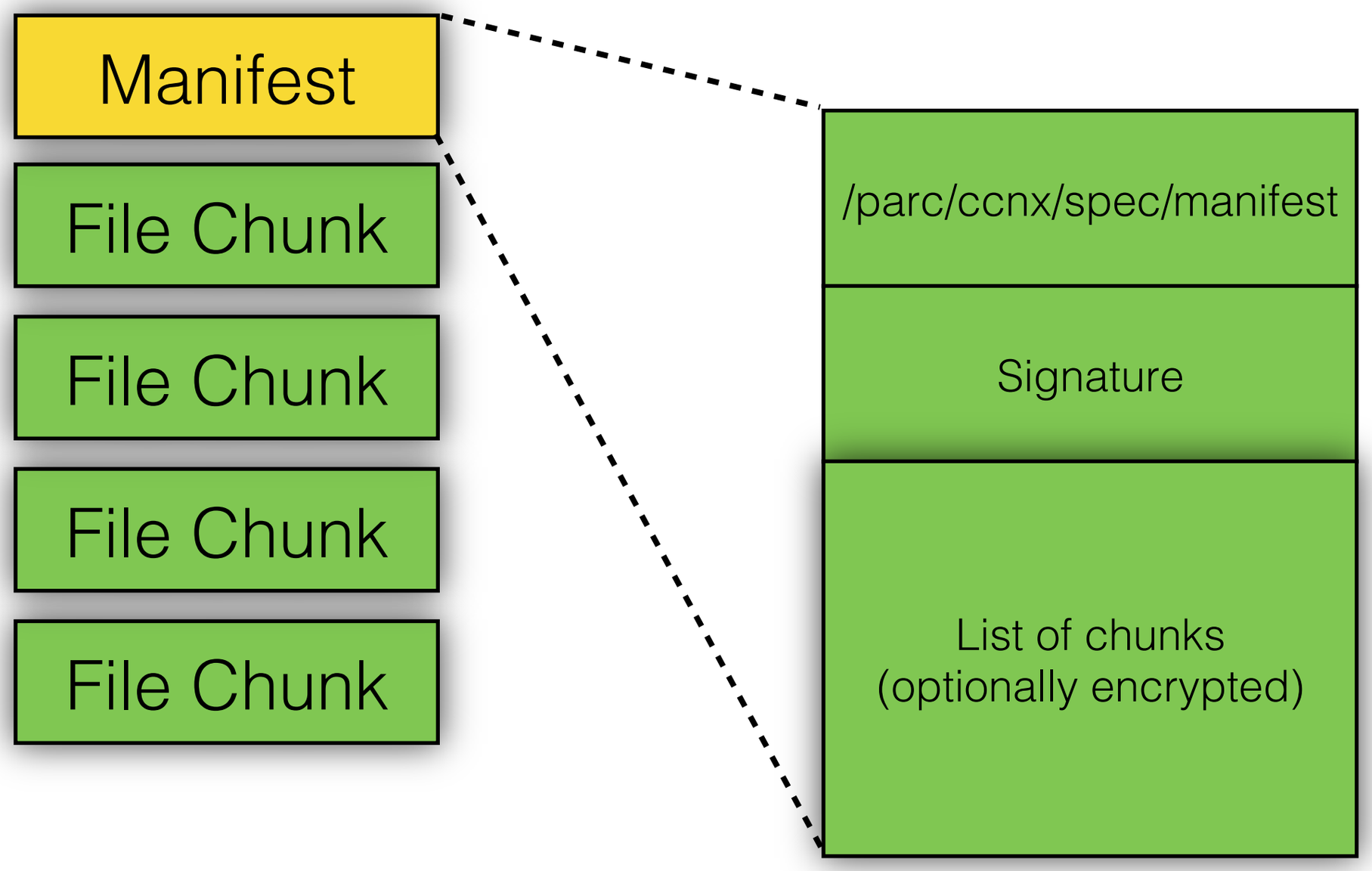
Identifier of the Content Objet publisher

Secure Single Chunk



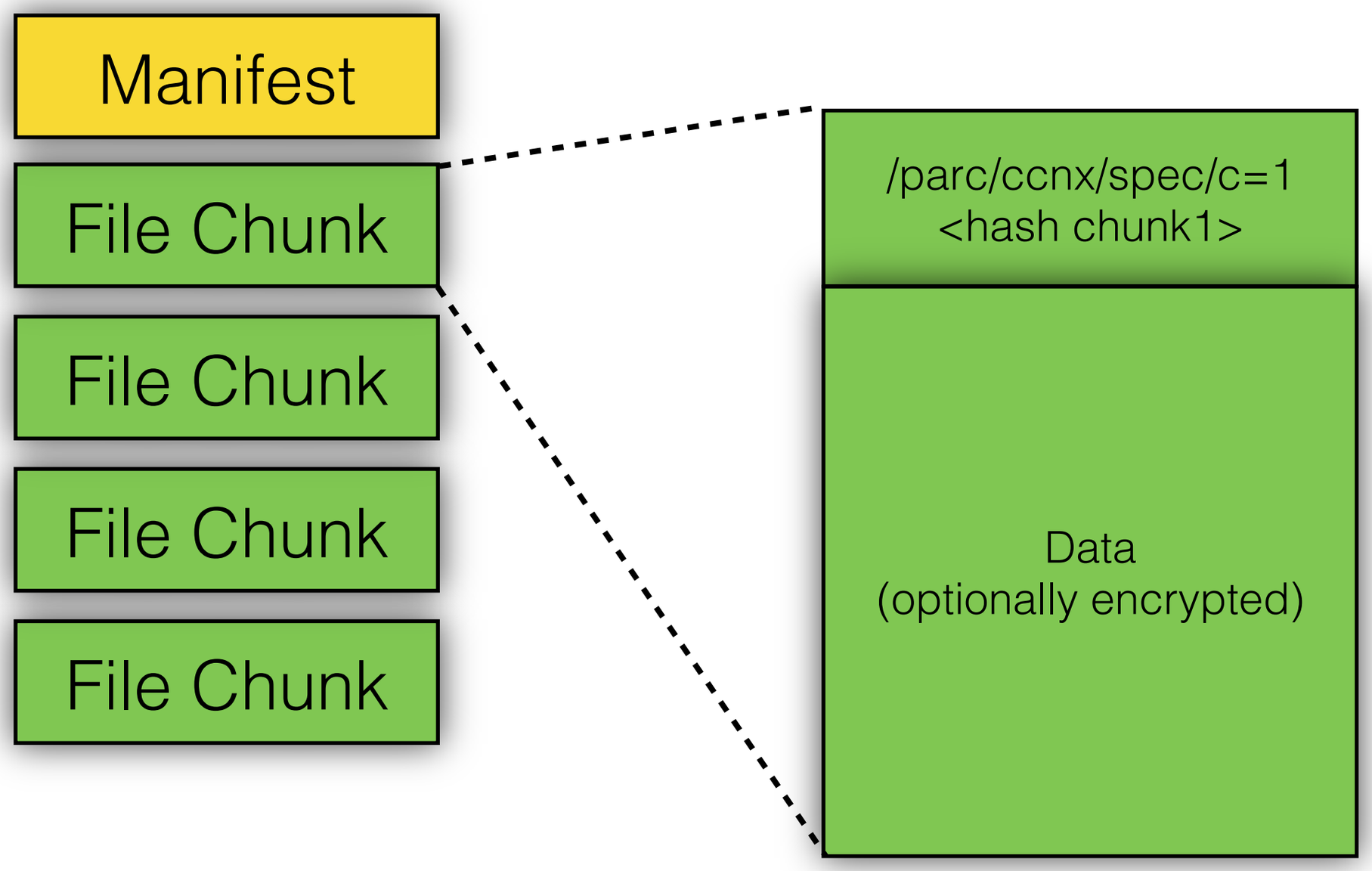
CCN names and signs every chunk

Secure Whole Object via Manifest



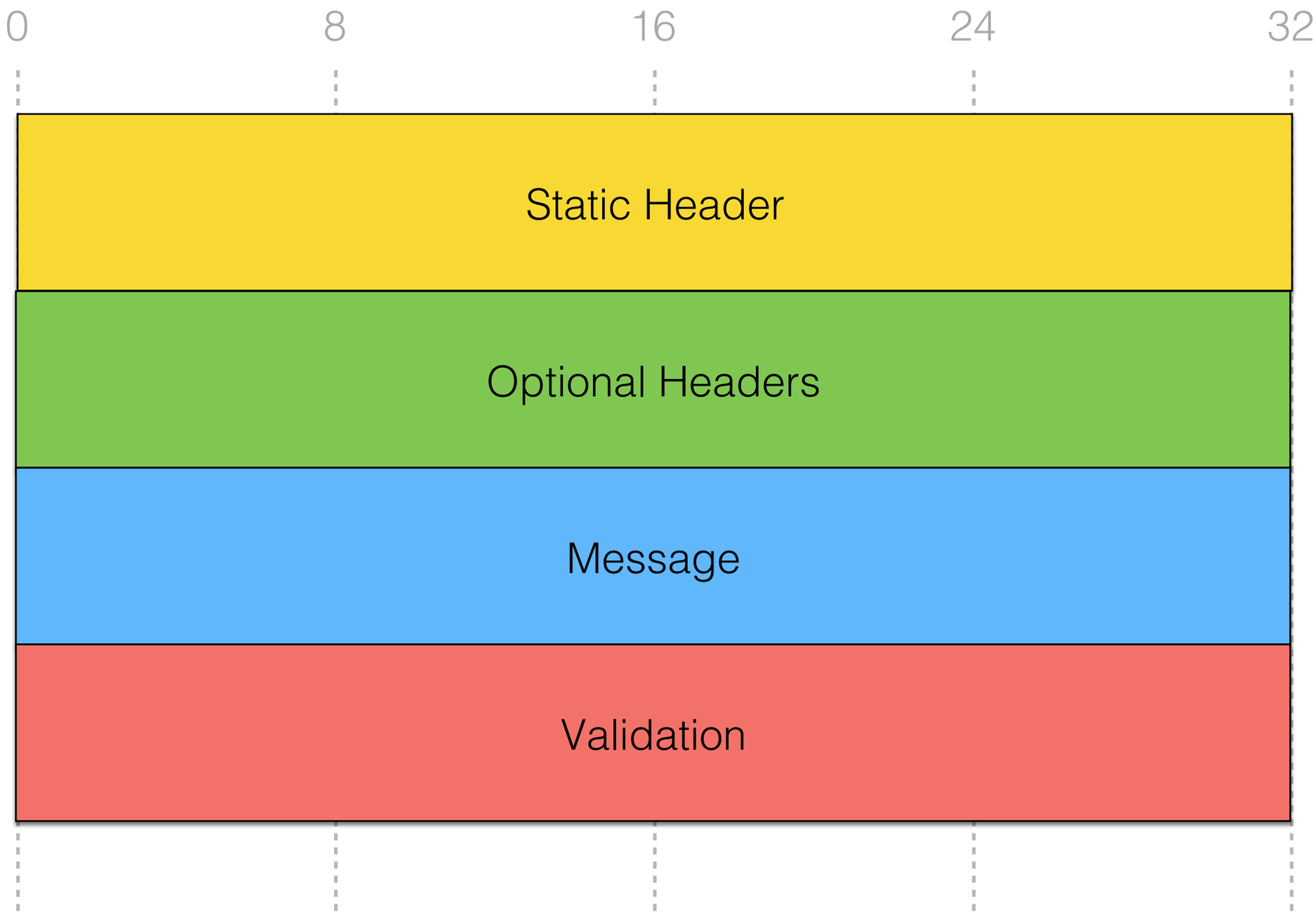
CCN names and signs the file via a manifest

Secure Whole Object via Manifest

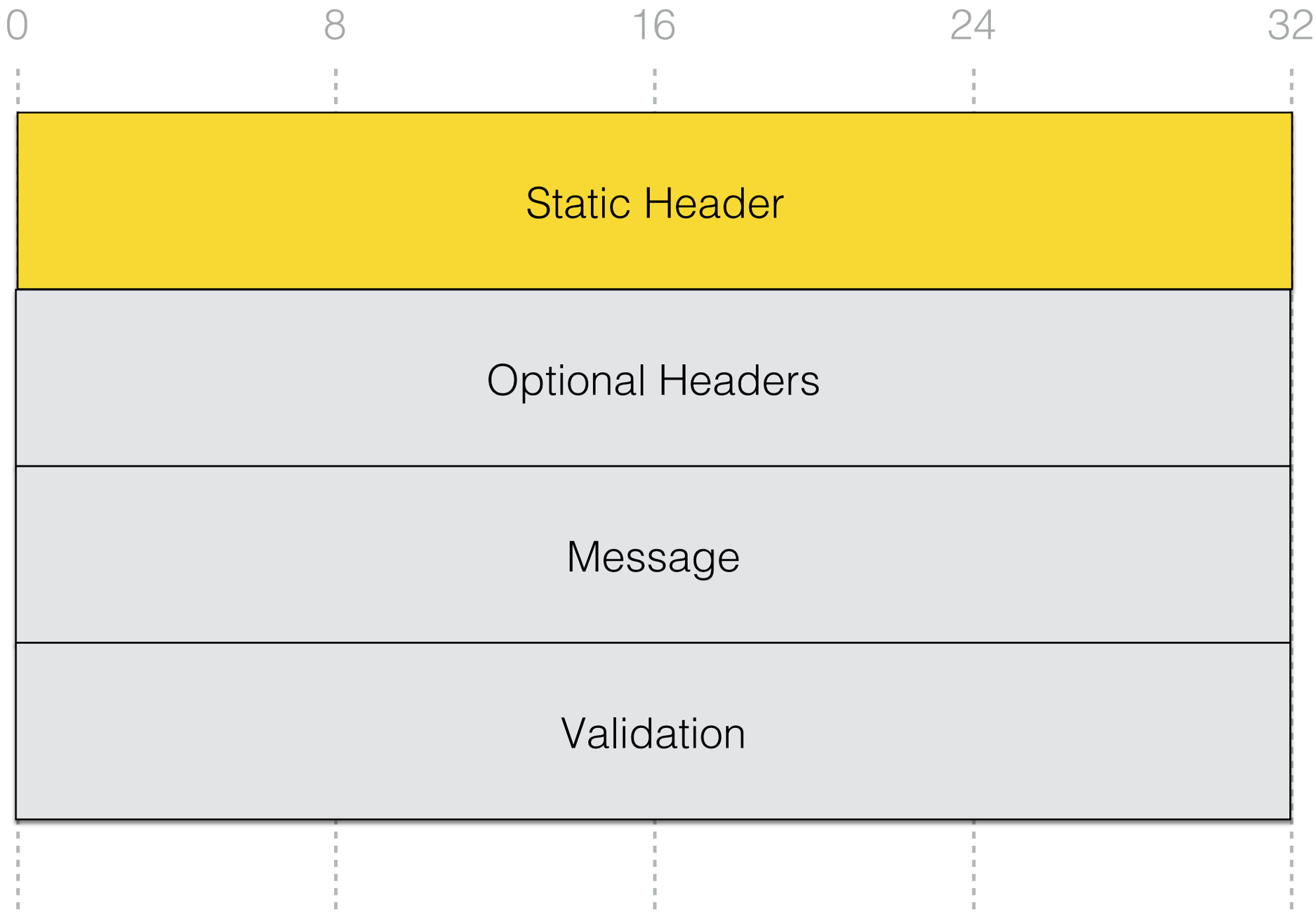


Indirectly sign every chunk through the manifest

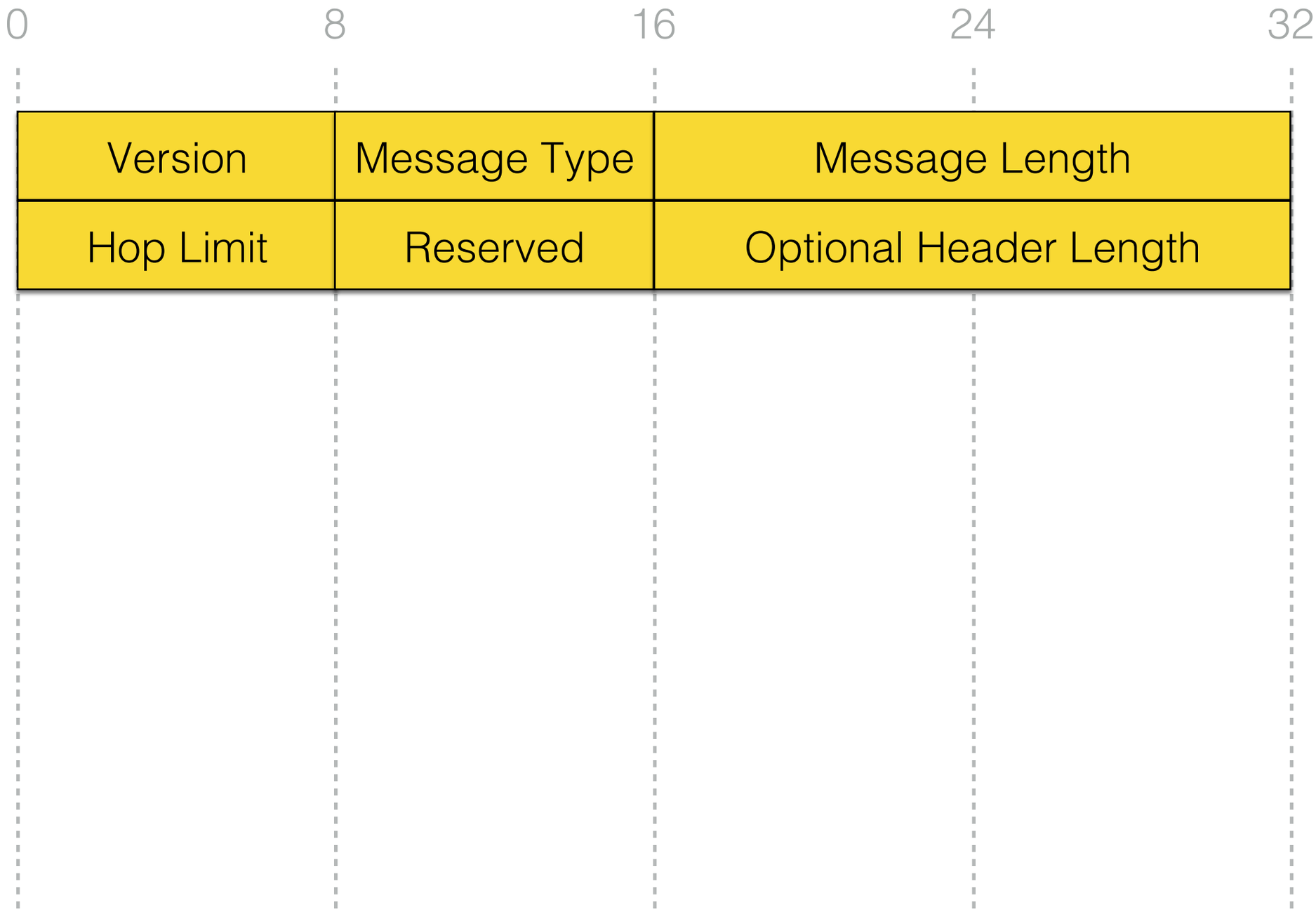
Packet Format



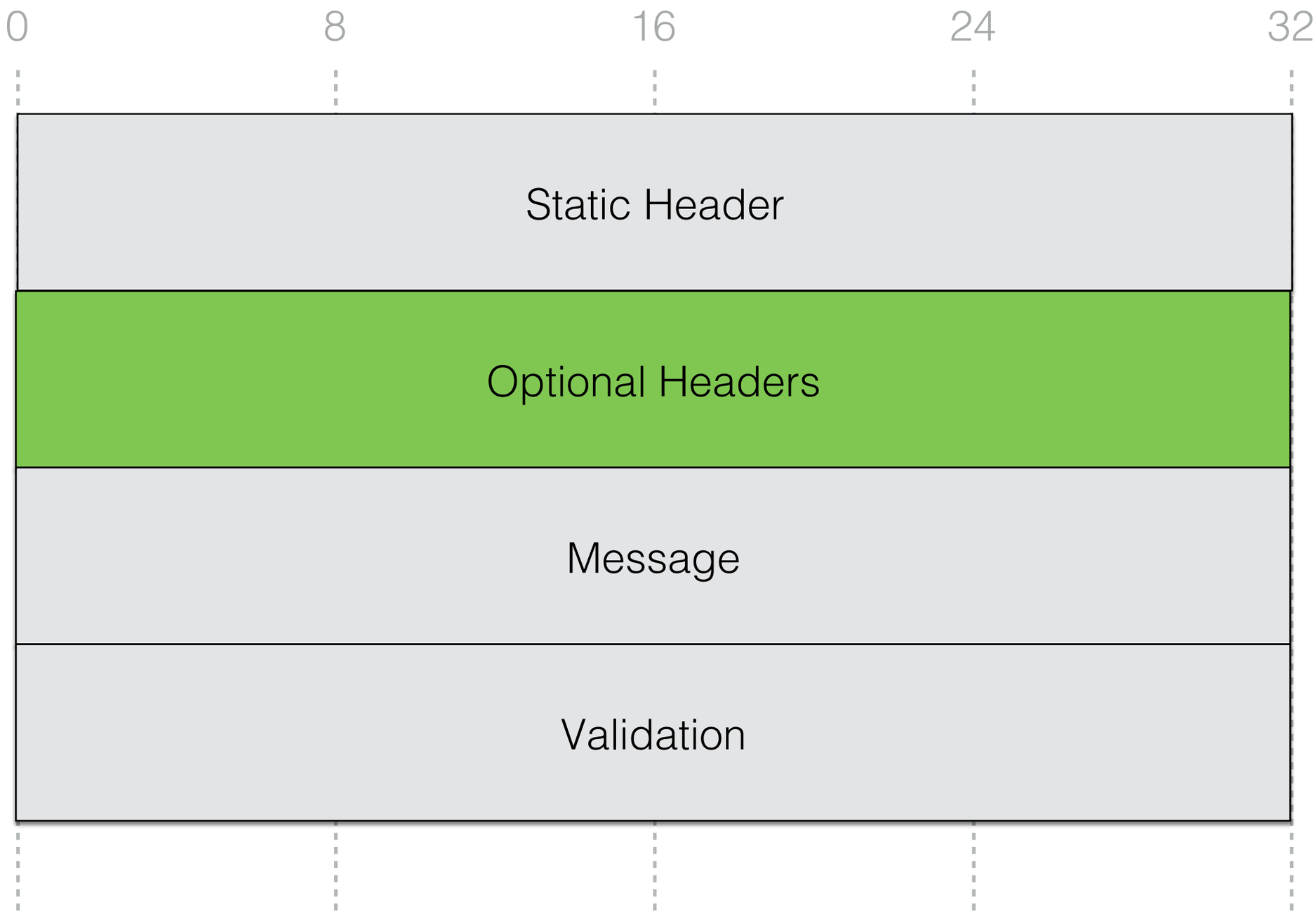
Static Header



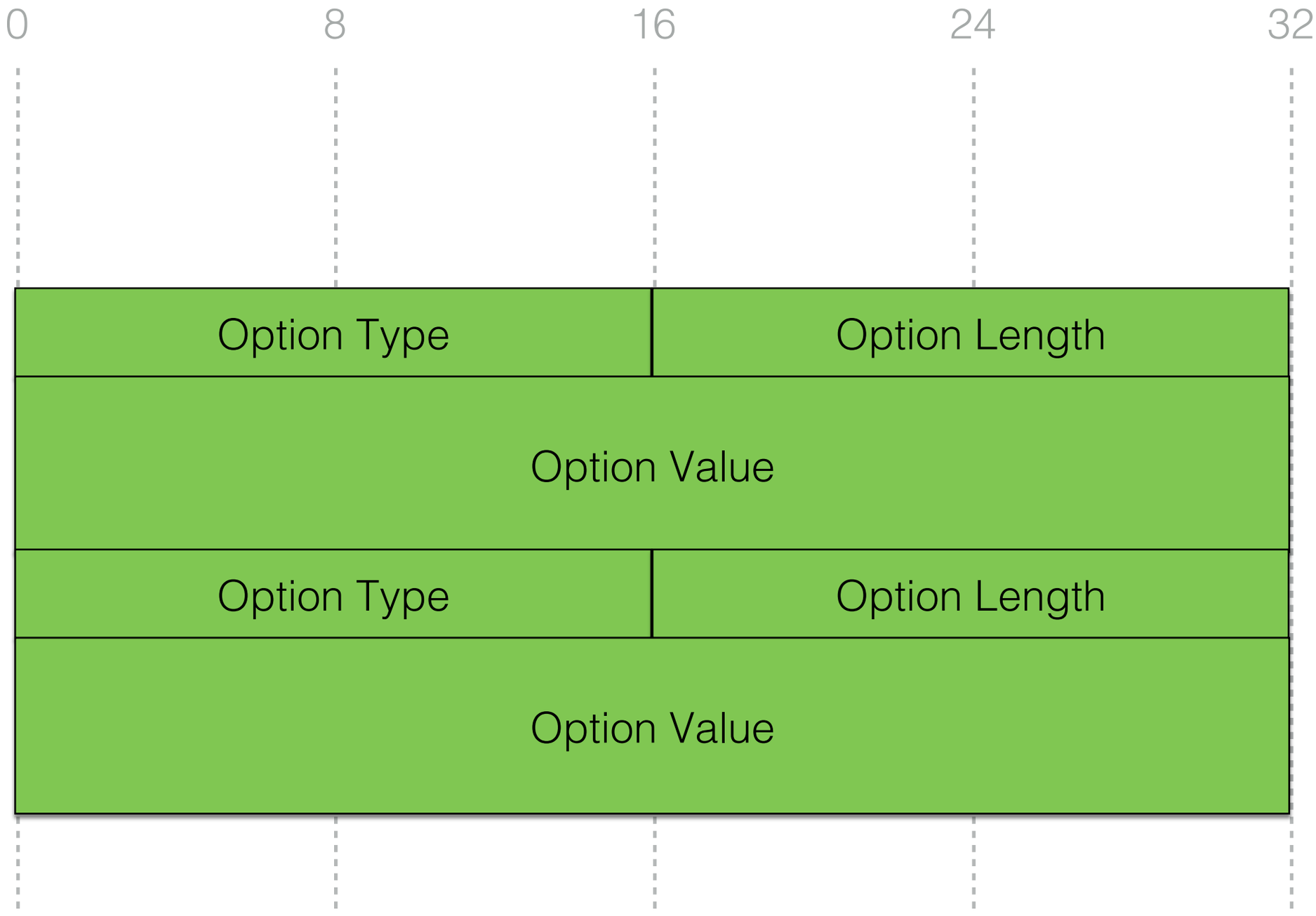
Static Header



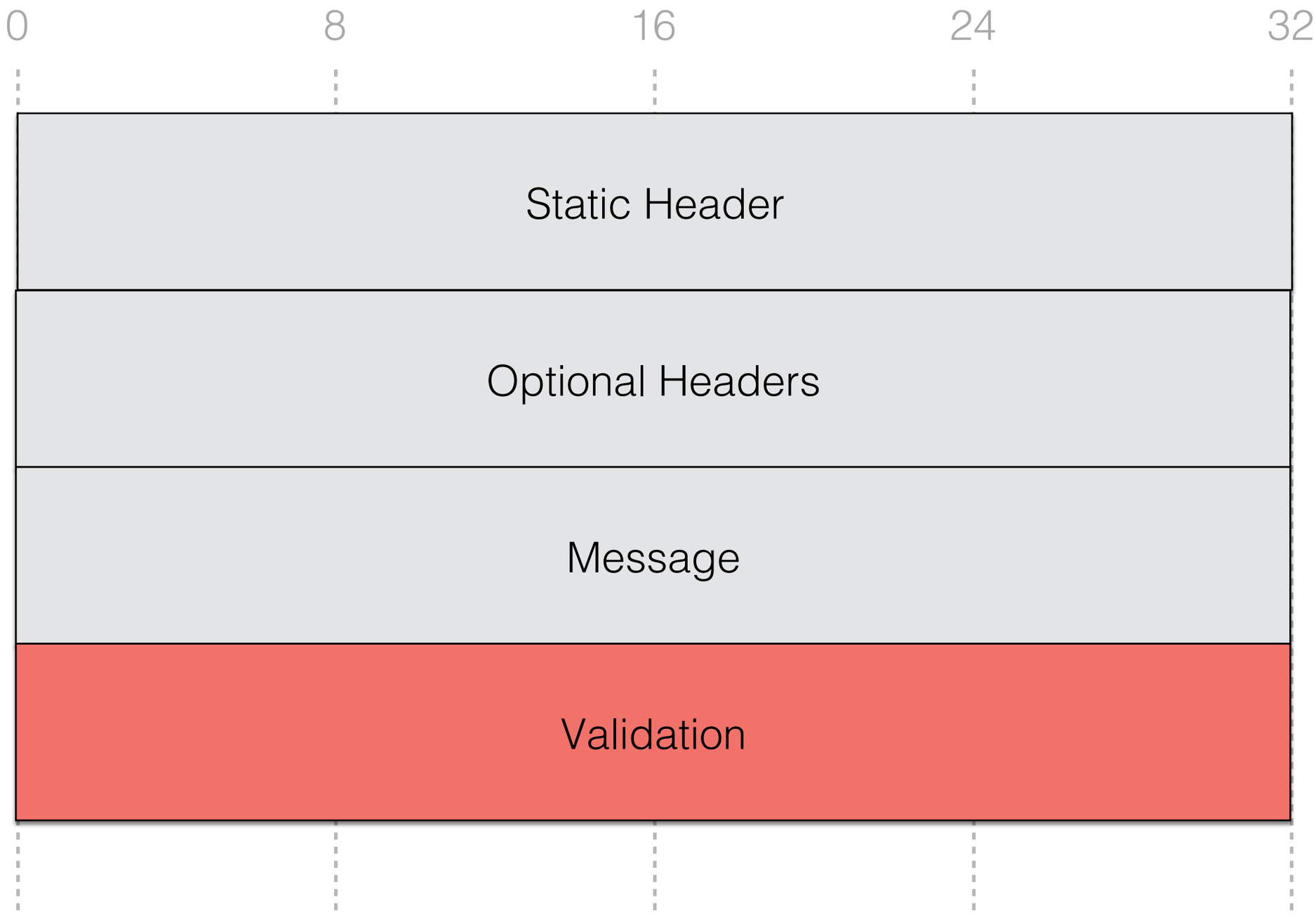
Packet Format



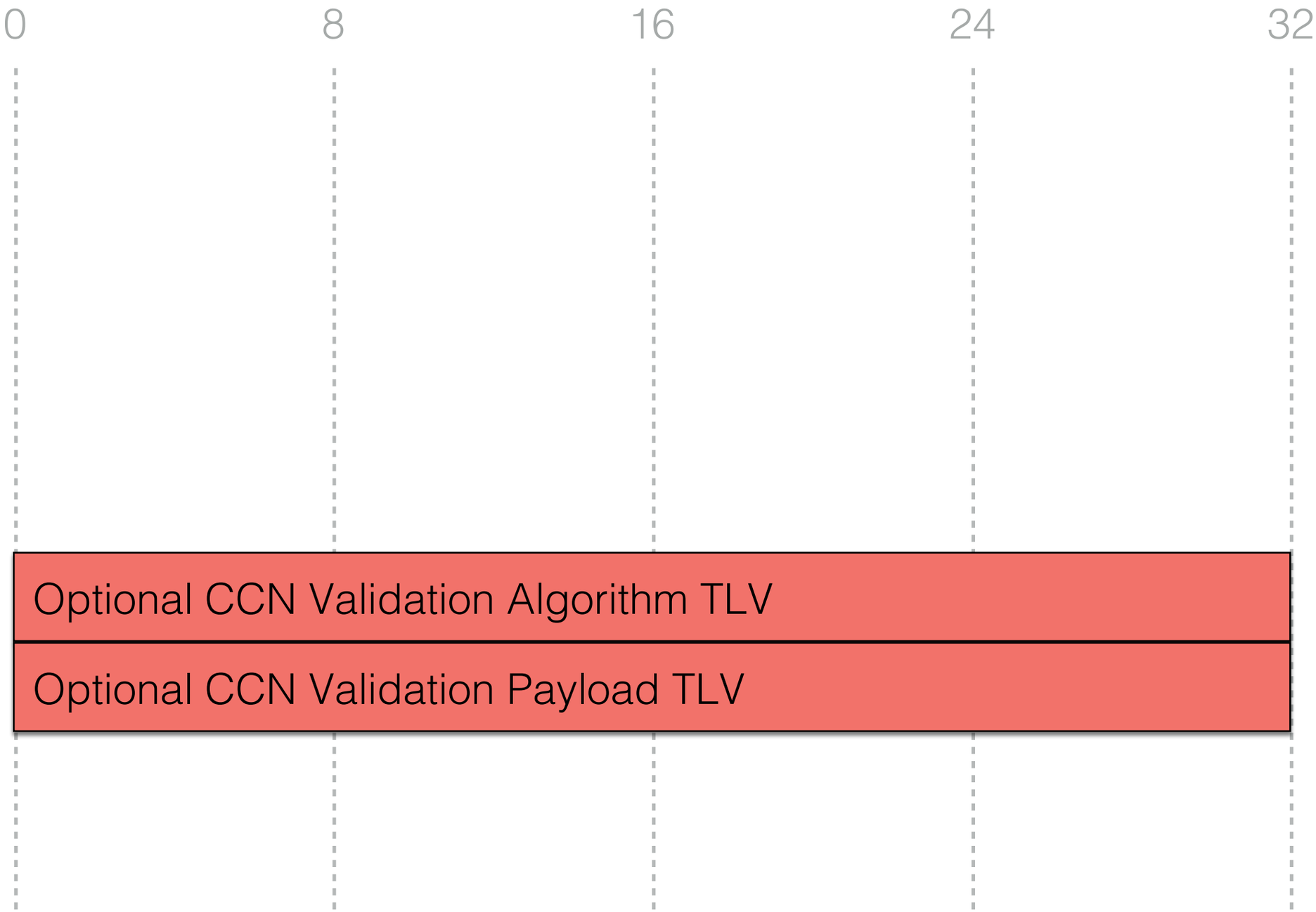
Optional Header



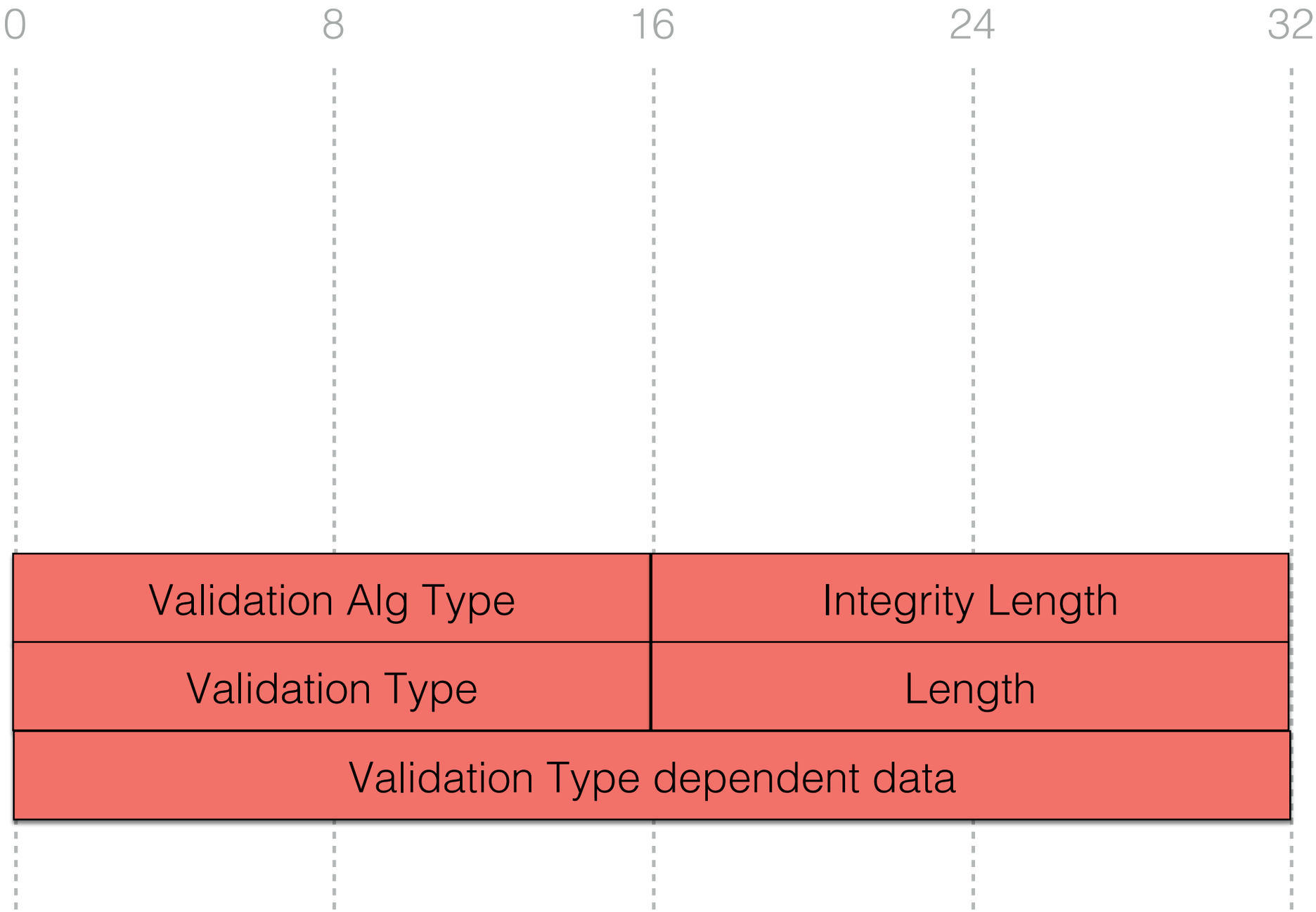
Packet Format



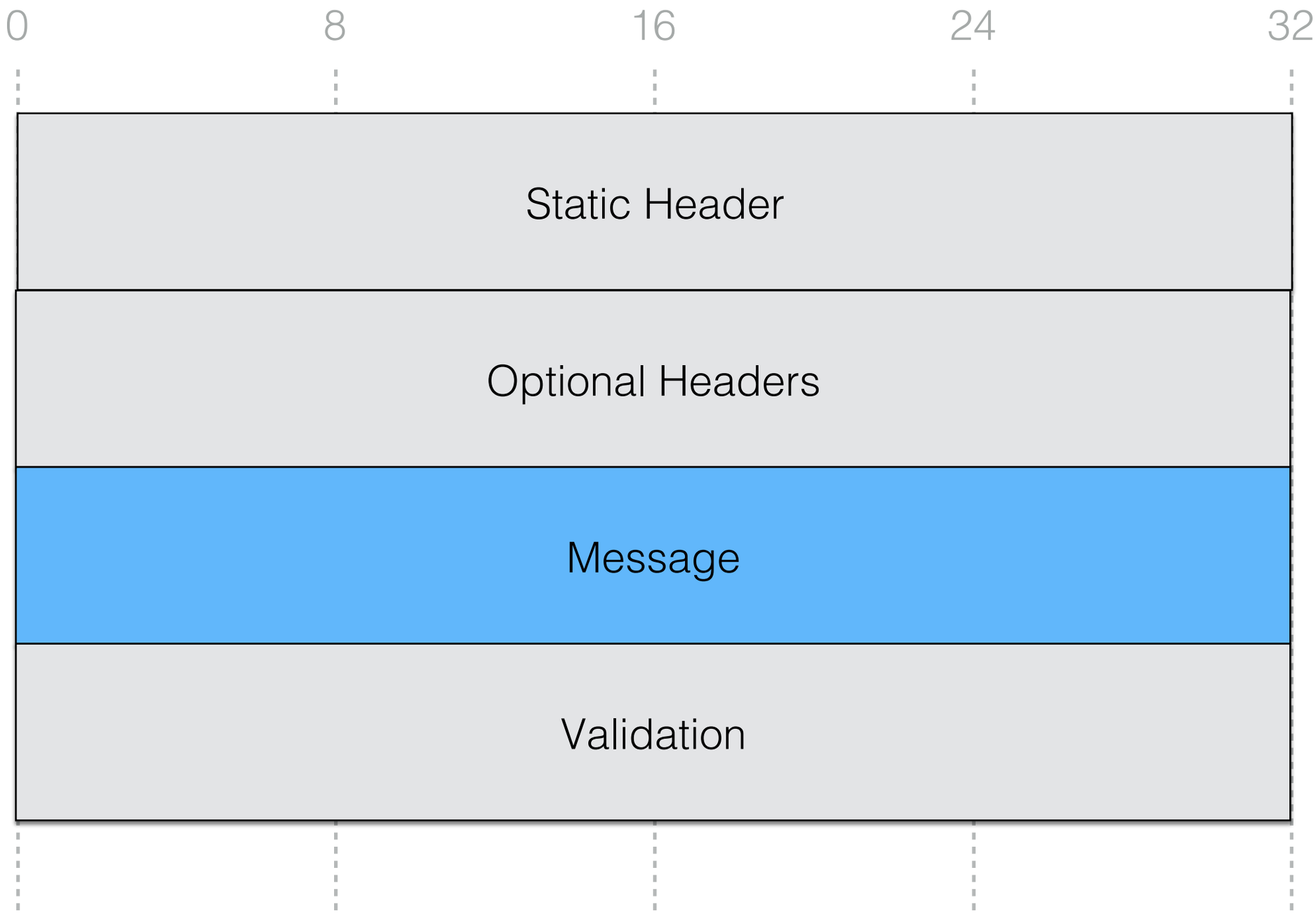
Validation


0
8
16
24
32
Optional CCN Validation Algorithm TLV
Optional CCN Validation Payload TLV

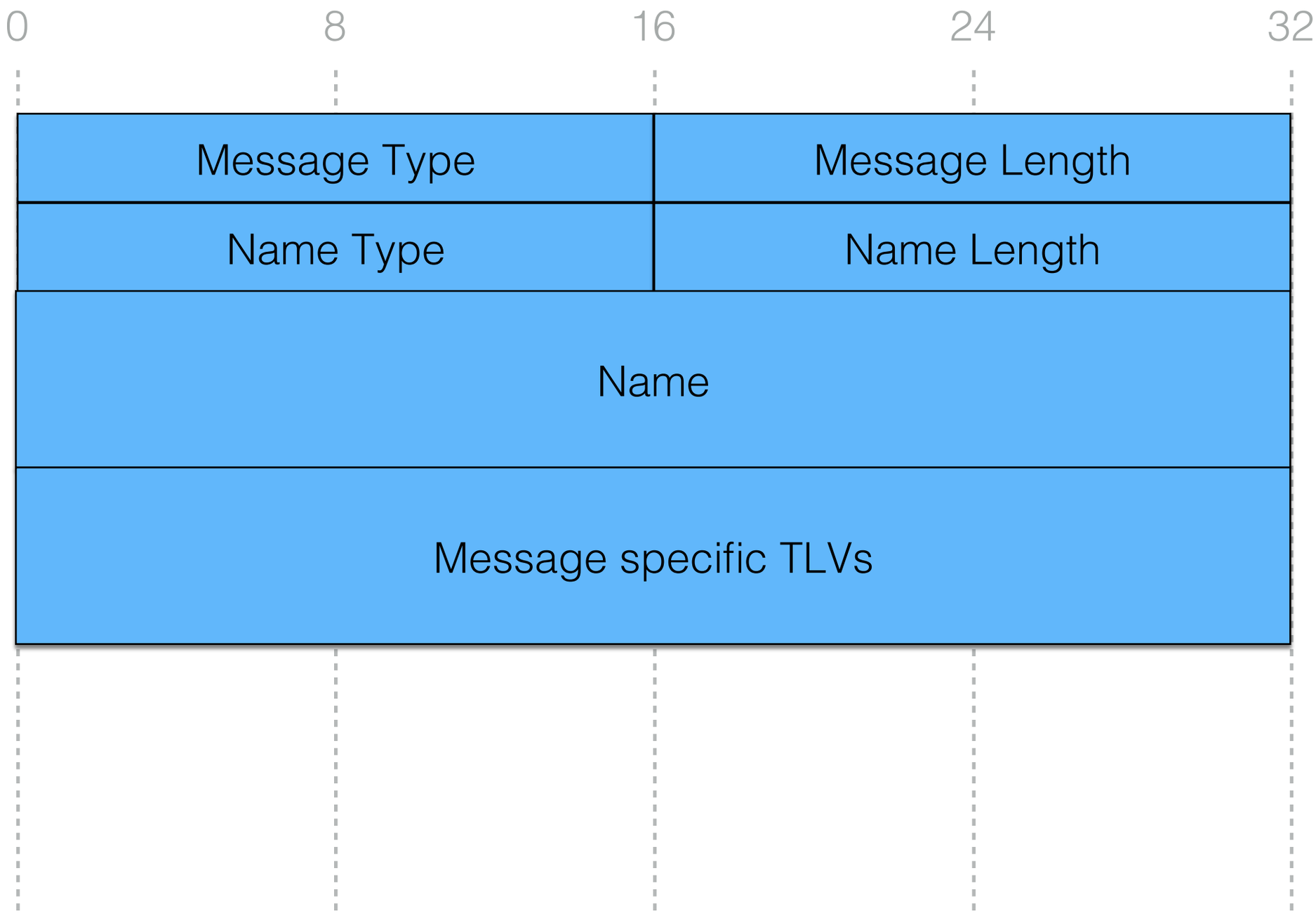
Validation


0
8
16
24
32
Validation Alg Type
Integrity Length
Validation Type
Length
Validation Type dependent data

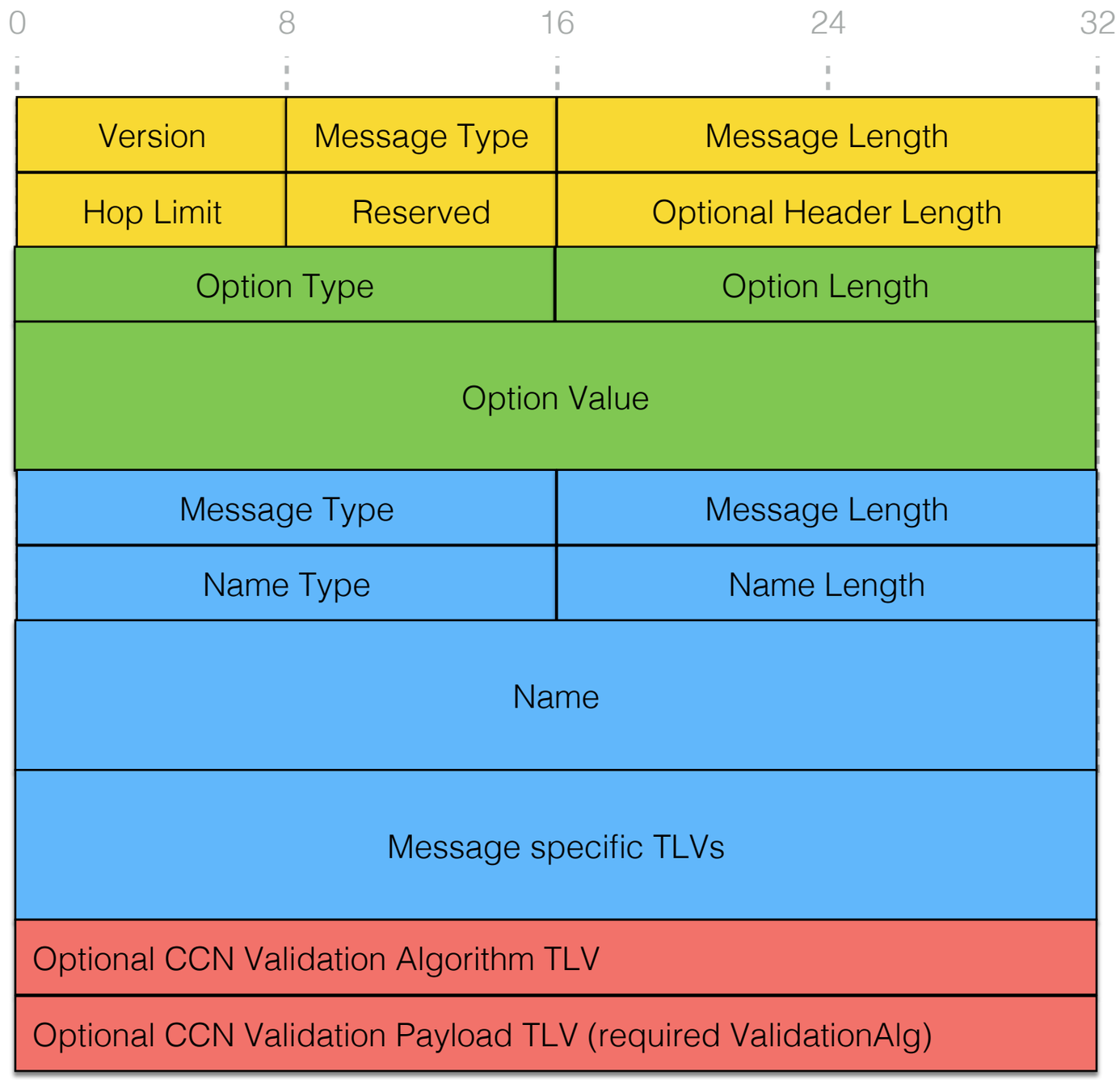
Packet Format



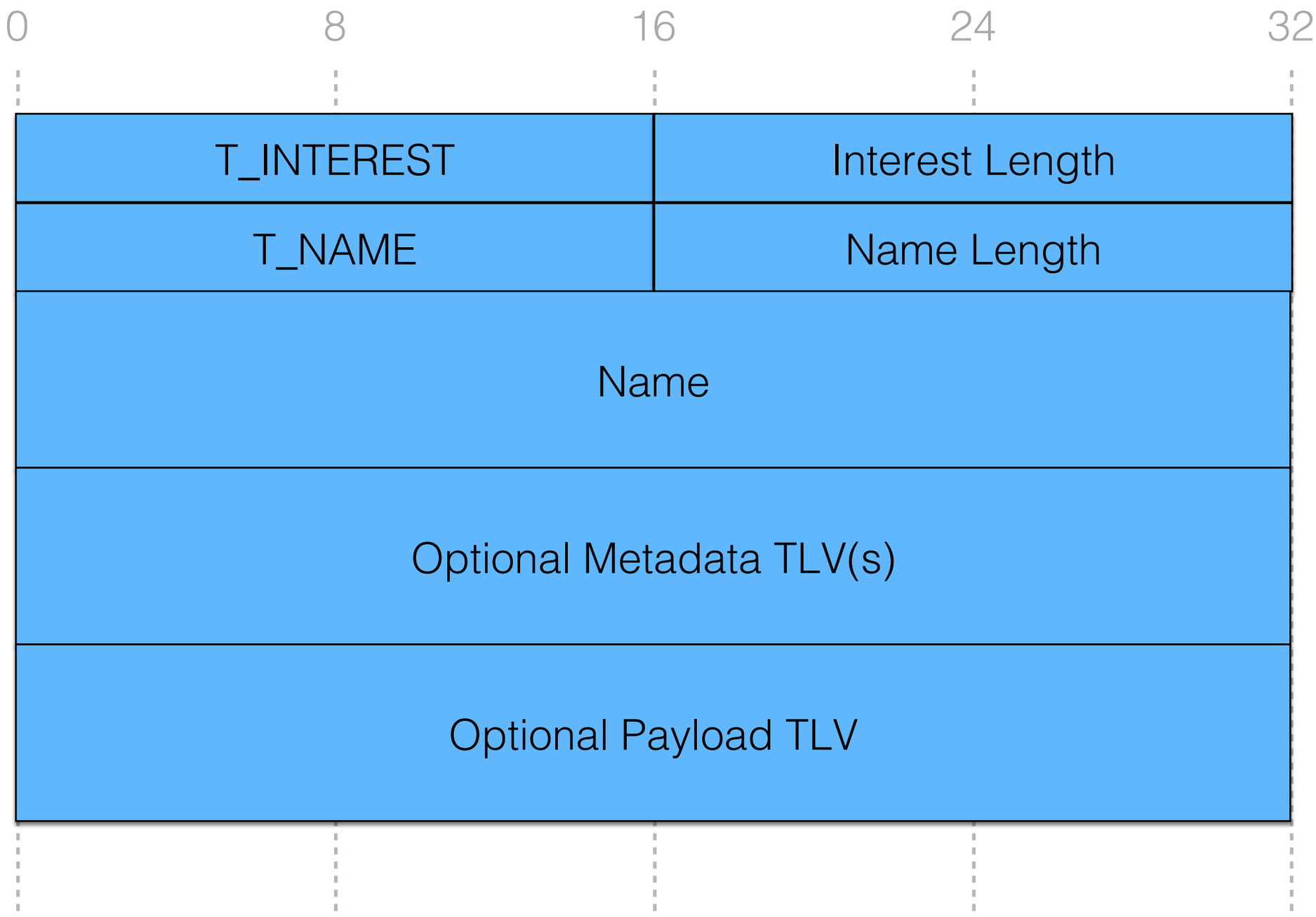
CCN Message


0
8
16
24
32
Message Type
Message Length
Name Type
Name Length
Name
Message specific TLVs

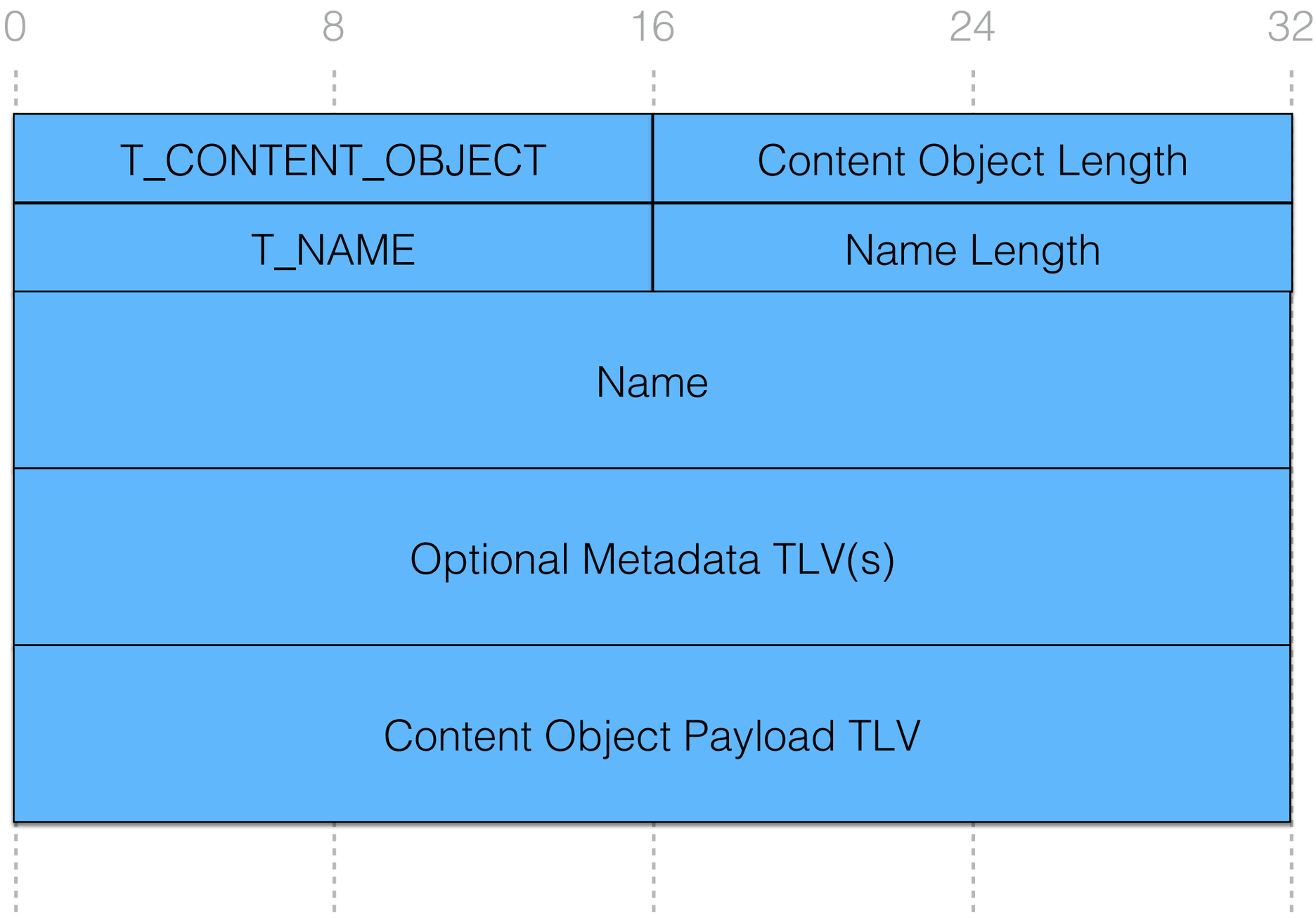
CCN Message


0
8
16
24
32
Version
Message Type
Message Length
Hop Limit
Reserved
Optional Header Length
Option Type
Option Length
Option Value
Message Type
Message Length
Name Type
Name Length
Name
Message specific TLVs
Optional CCN Validation Algorithm TLV
Optional CCN Validation Payload TLV (required ValidationAlg)

Interest Message



Content Object Message



Quick Example Break

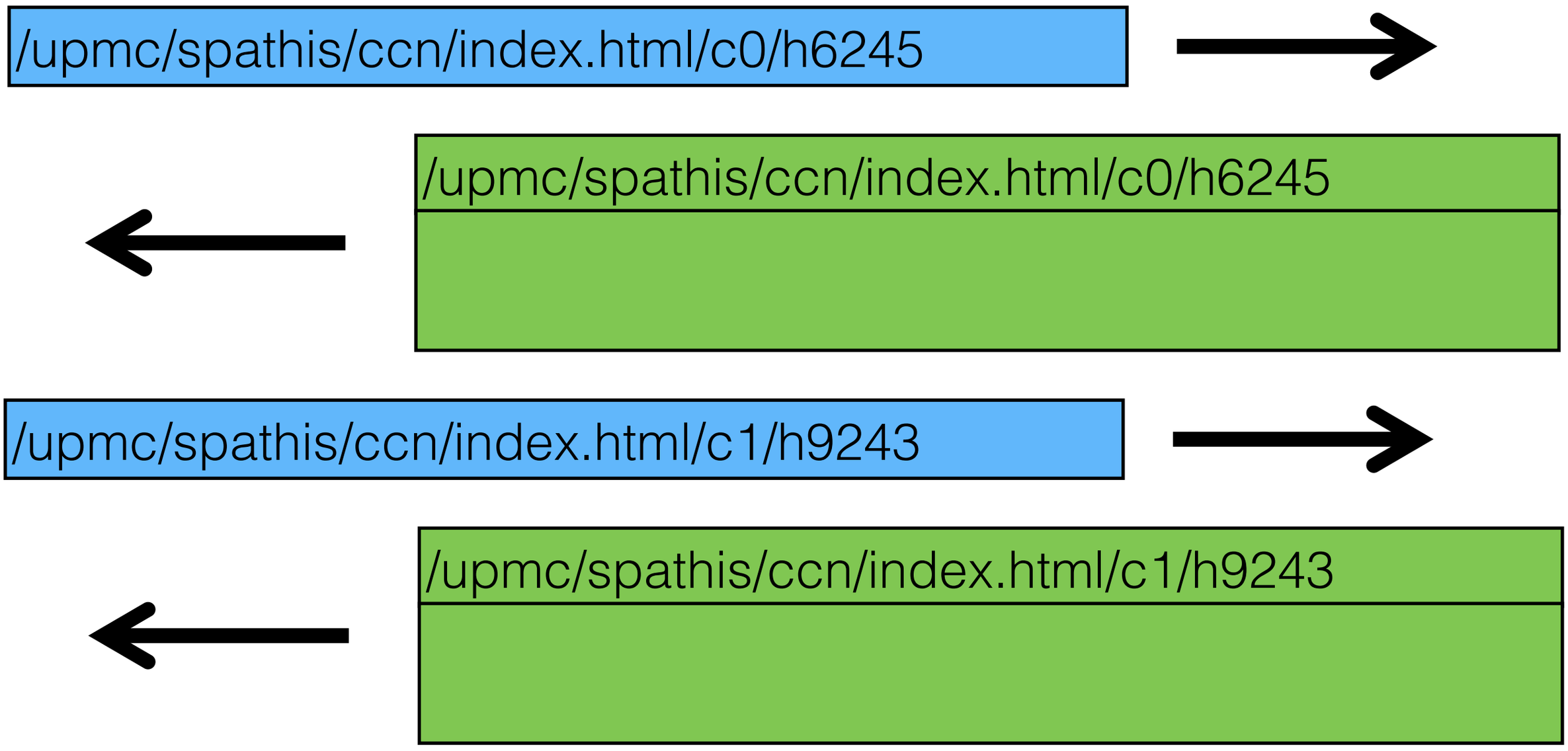
/upmc/spathis/ccn/index.html/manifest/c0 →

← /upmc/spathis/ccn/index.html/manifest/c0

c0/h6245
c1/h9243

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

/upmc/spathis/ccn/style.css/manifest/c0

/upmc/spathis/ccn/logo.gif/manifest/c0

/upmc/spathis/ccn/tutorial.ppt/manifest/c0

/upmc/spathis/ccn/style.css/manifest/c0

…

Retrieve references

CCN Security

- Authenticated mappings from names to content
- Each CCN packet is signed:

Signature(Name, Content, SignInfo)
SignInfo includes : cryptographic digest or fingerprint of publisher's key, key or key location

Content can be authenticated by every node (public key signatures)

Different signature algorithm available (security vs performance)

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

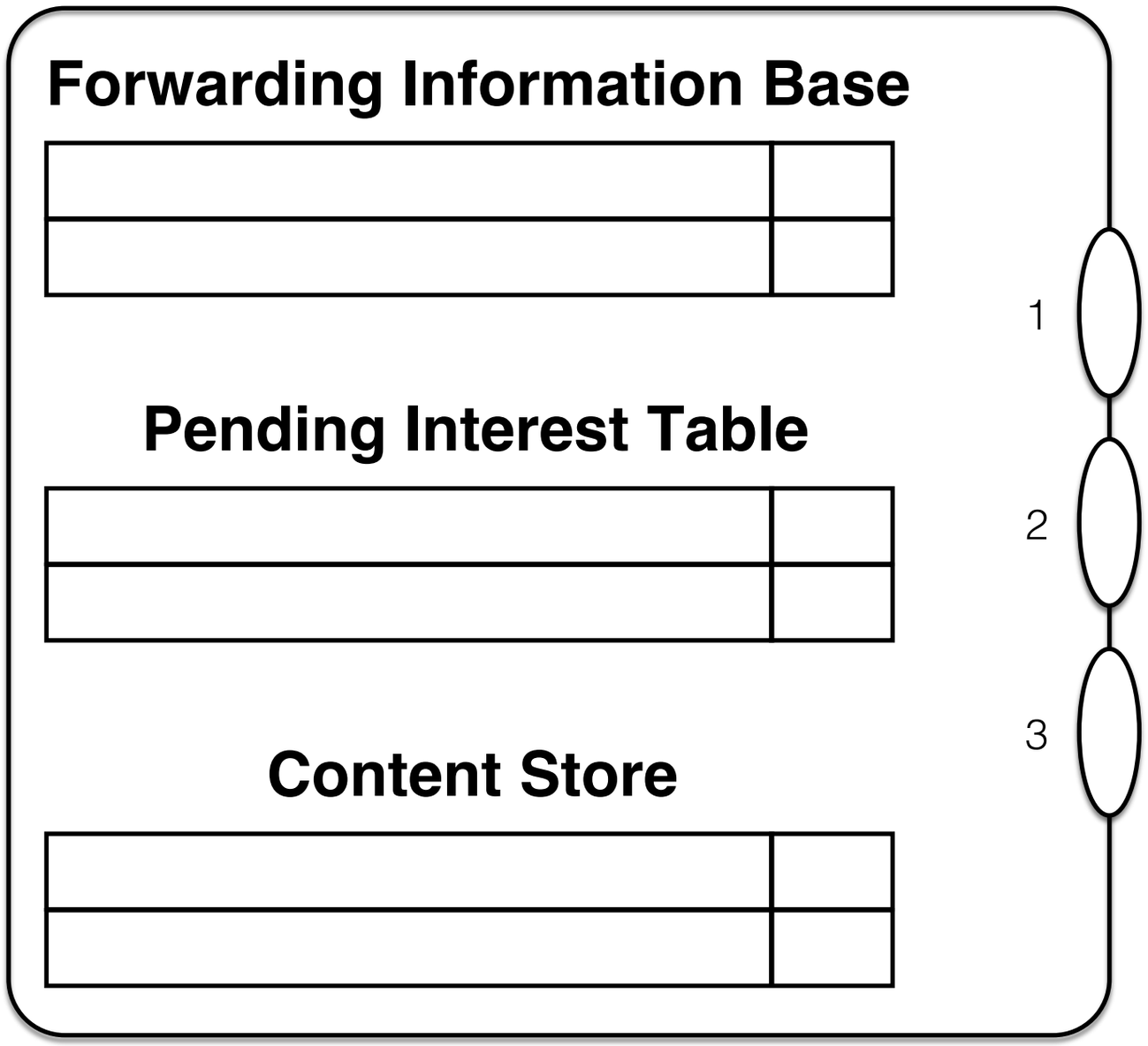
- Strategy

  How to use the alternatives

- Forwarding

  Processing a packet based on a strategy

CCN Forwarder

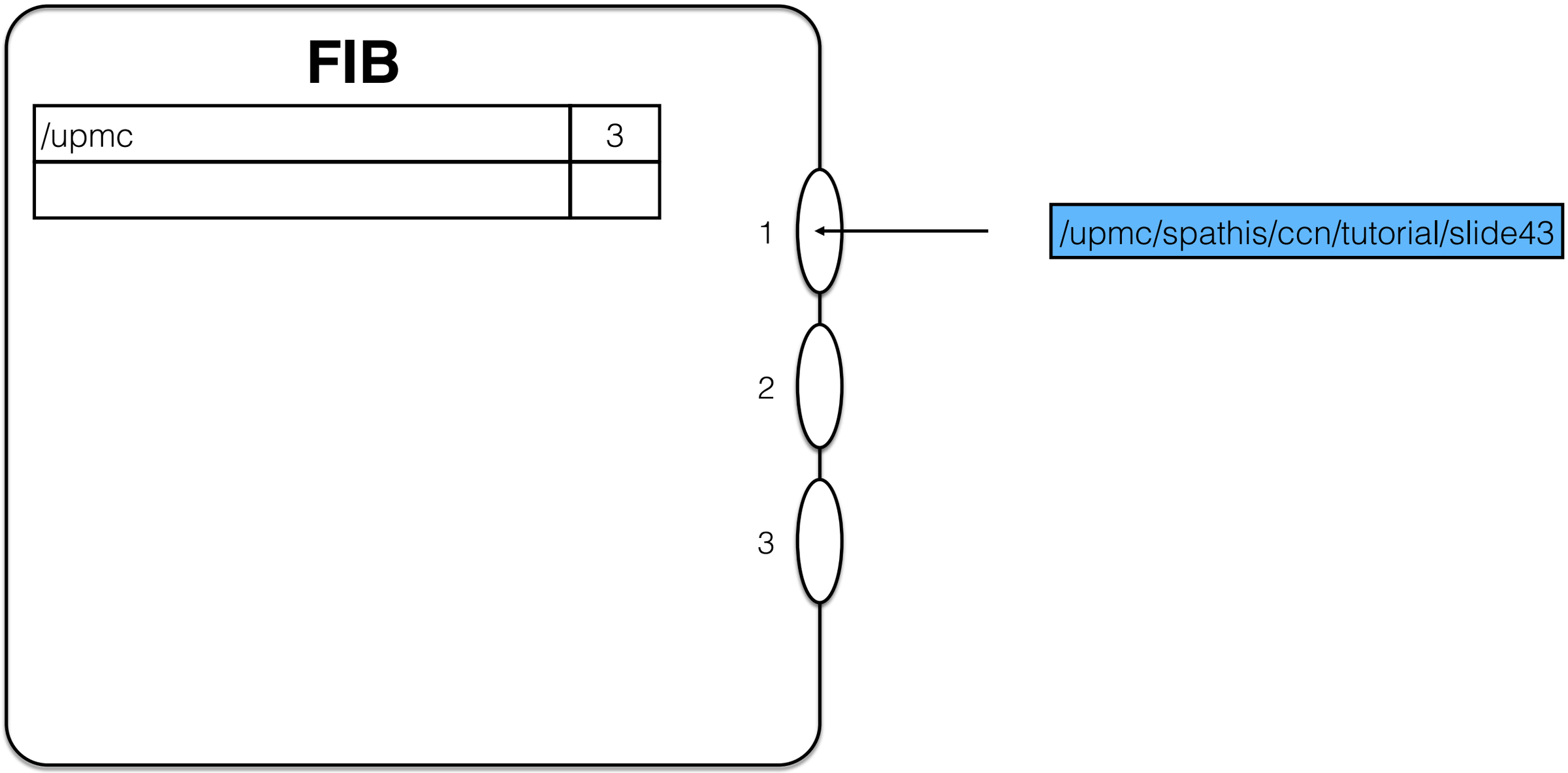


Store information about what face to follow to find a given name

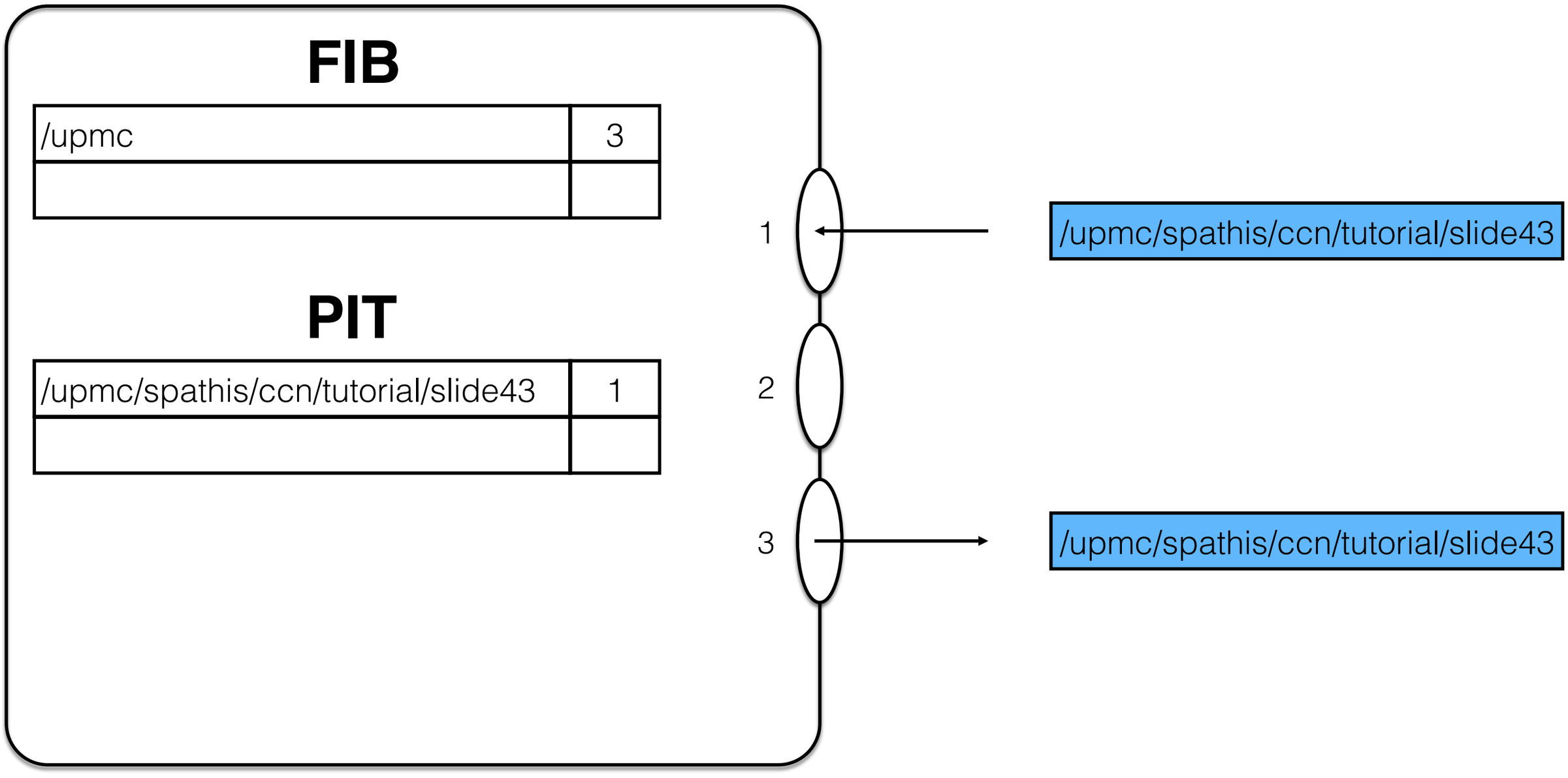
Store information about what interests are pending

Buffer content objects for potential reuse

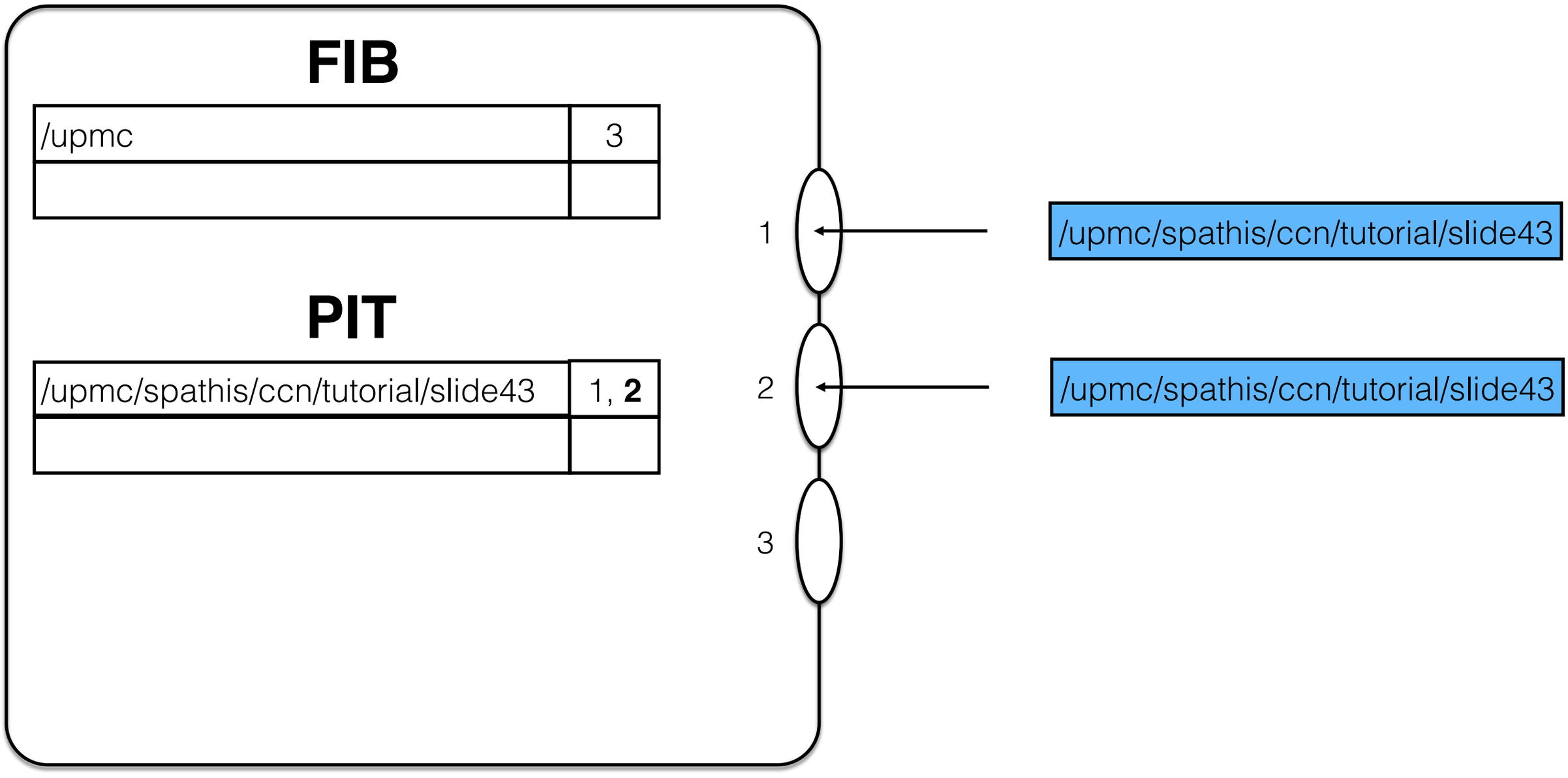
CCN Forwarder



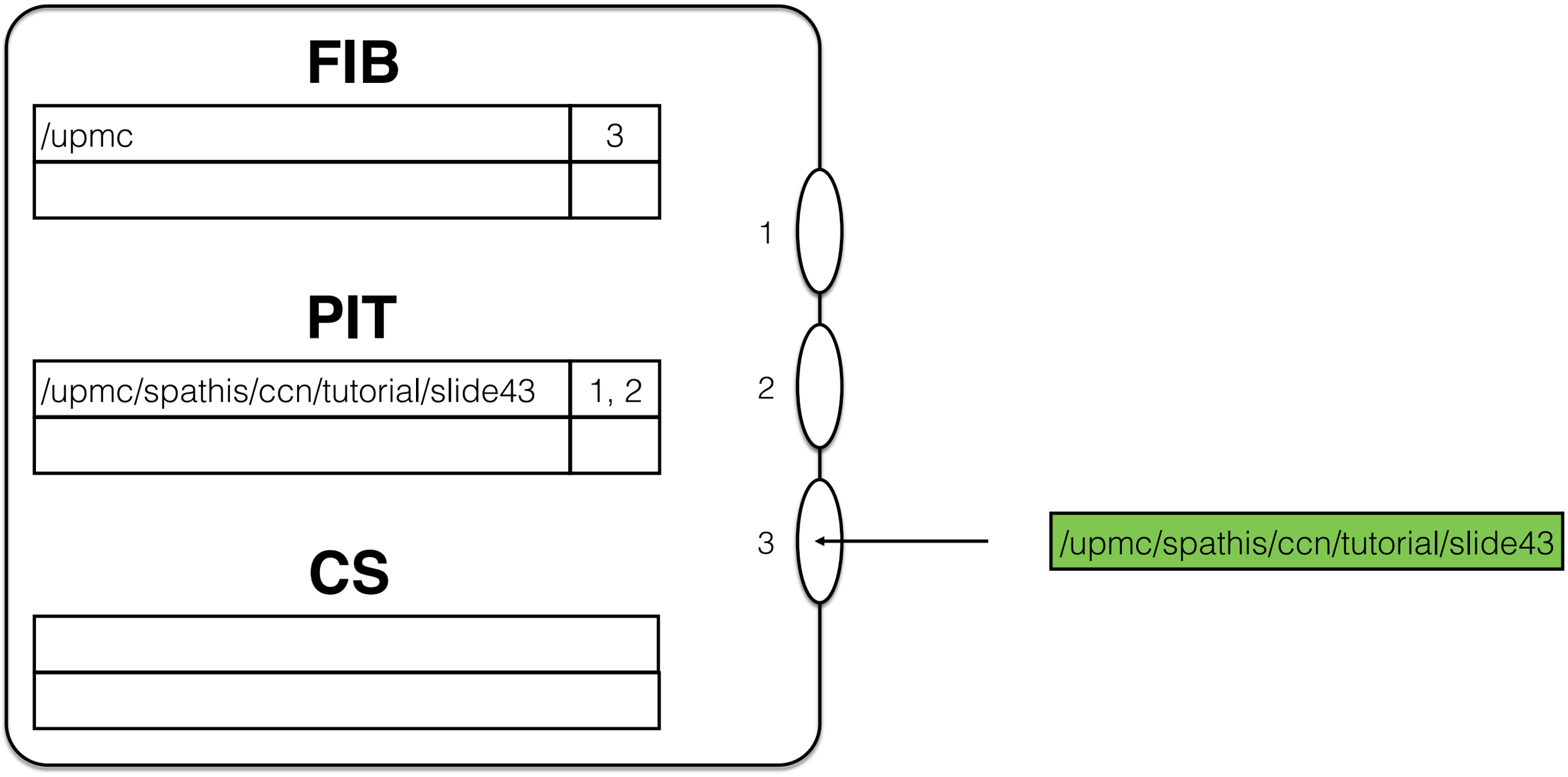
CCN Forwarder



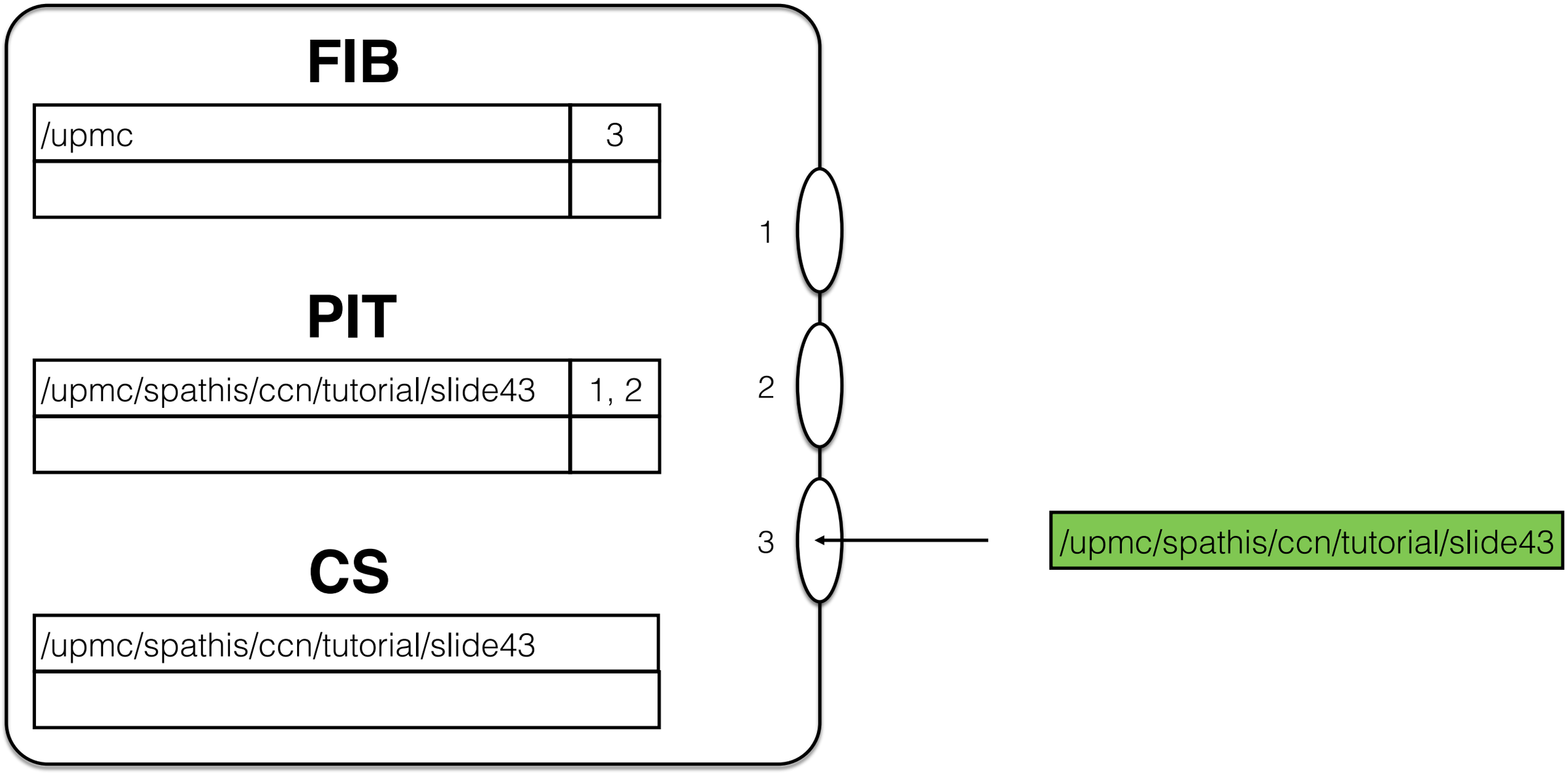
CCN Forwarder



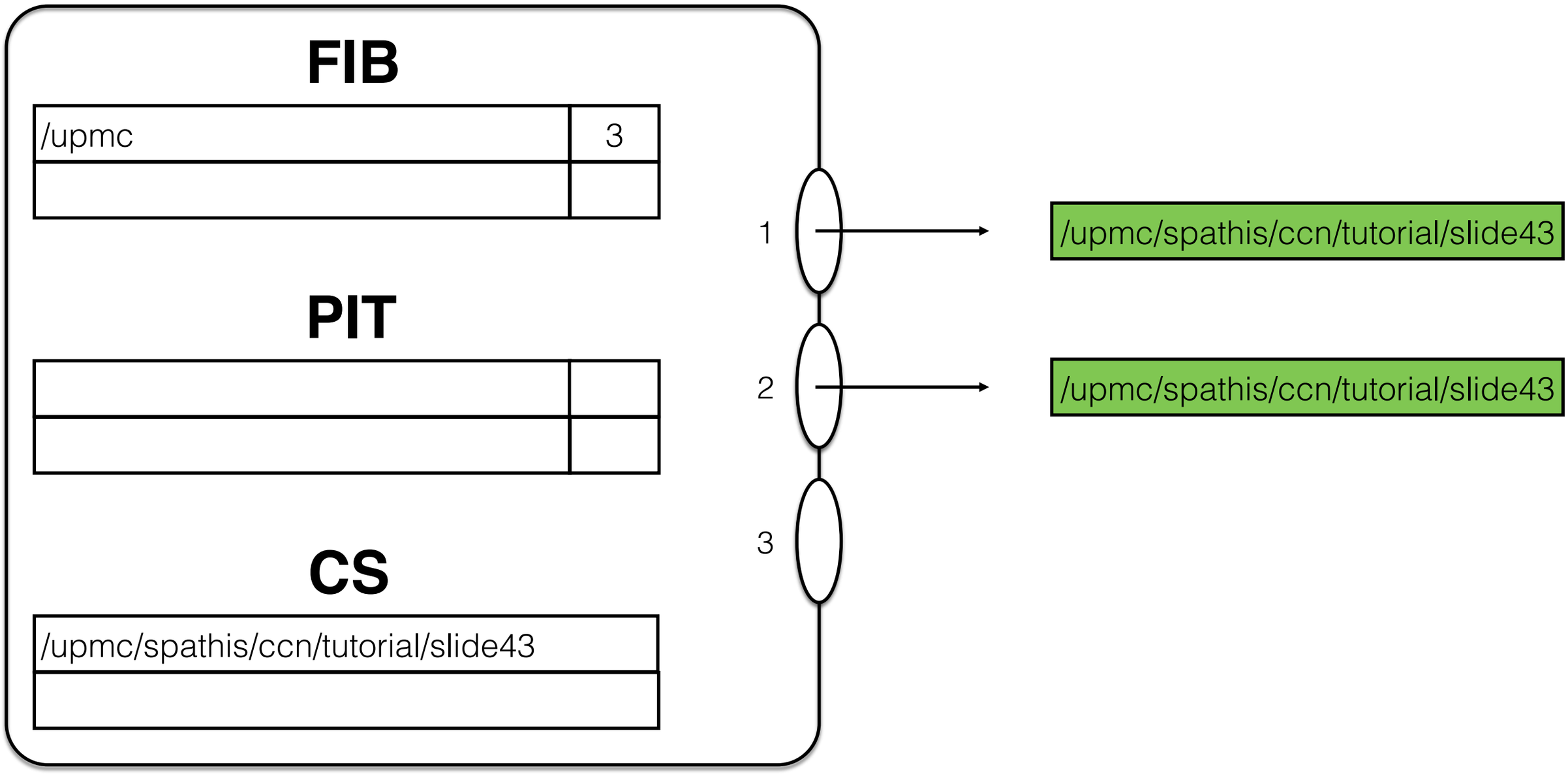
CCN Forwarder



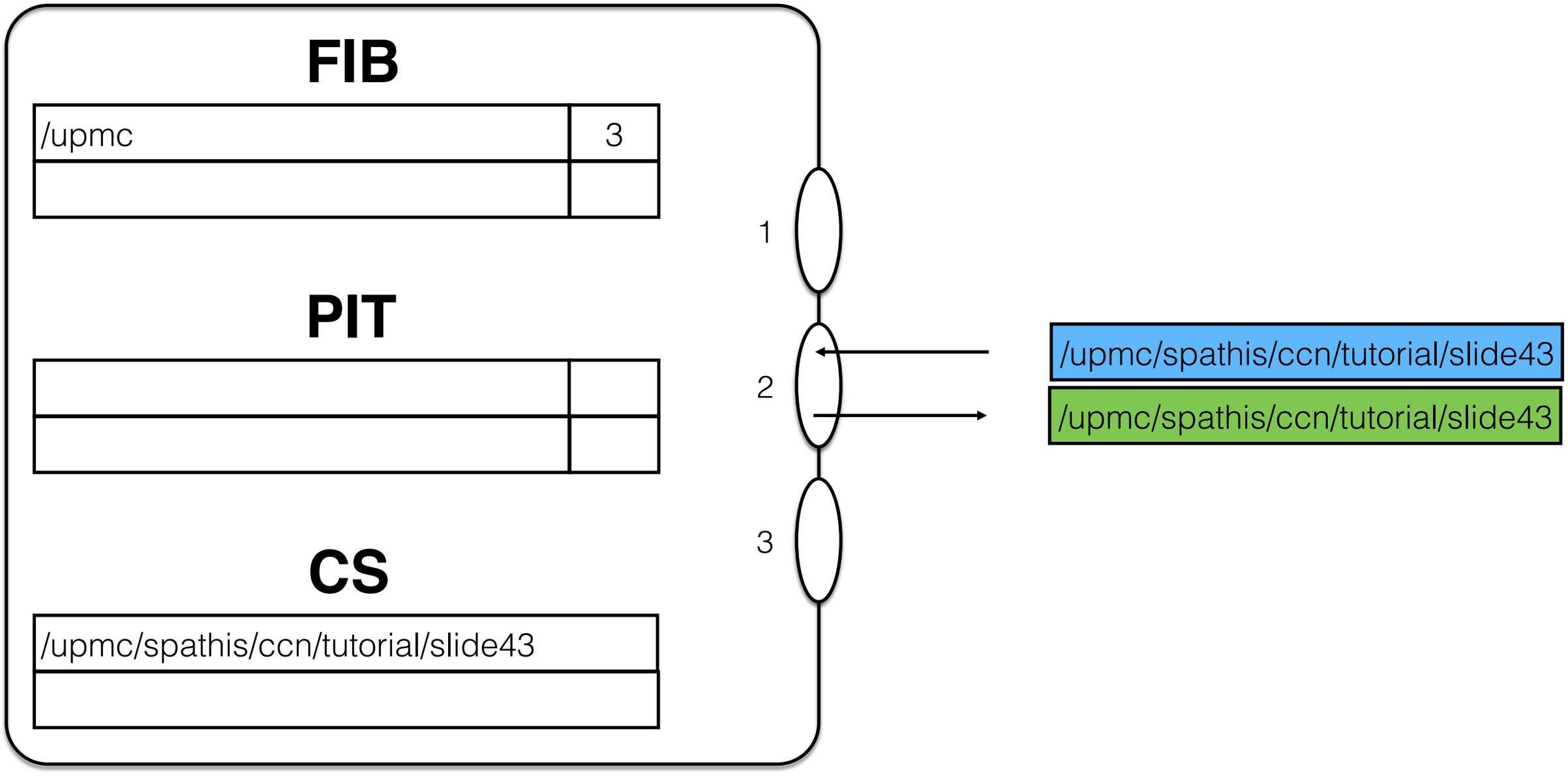
CCN Forwarder



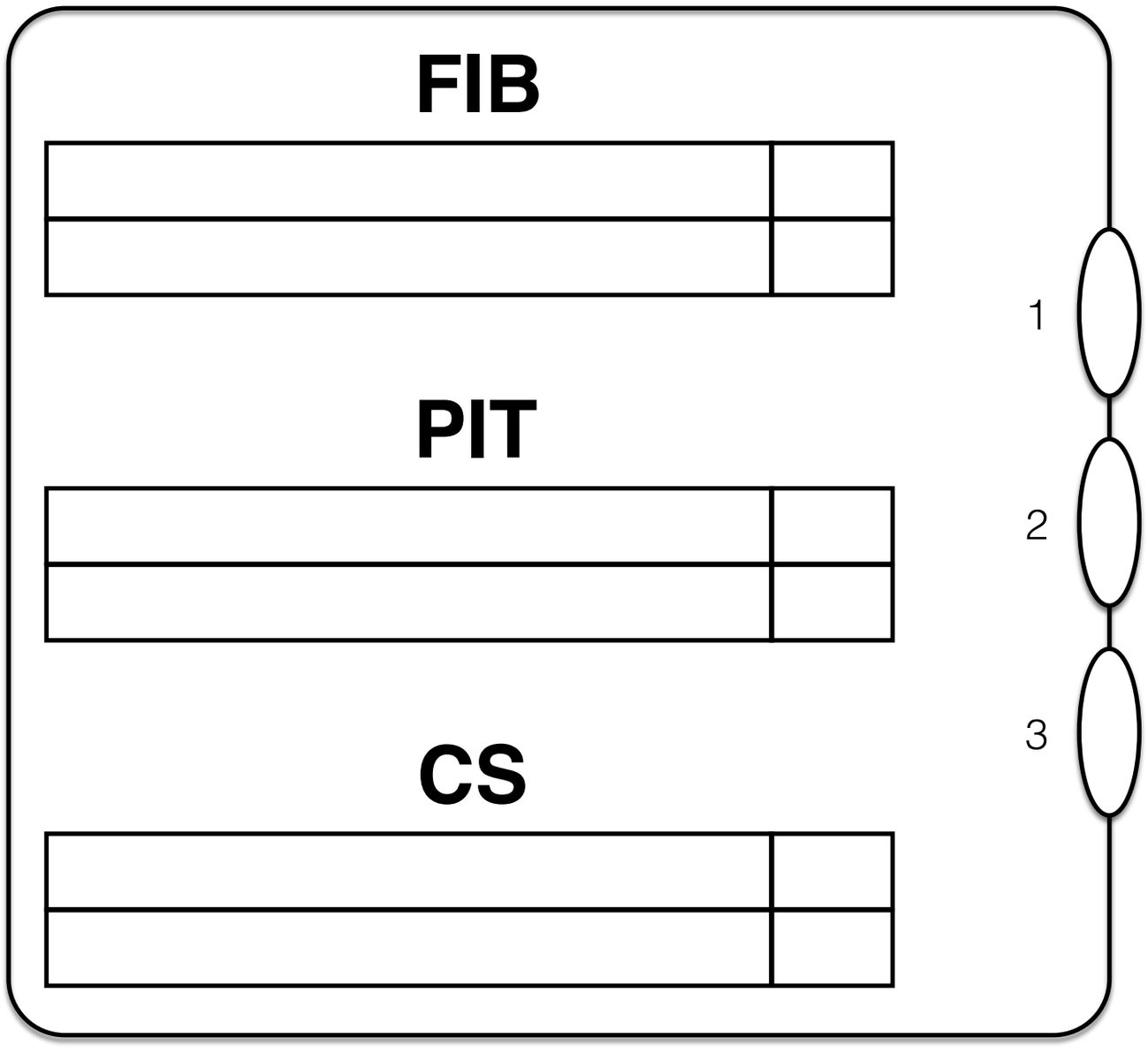
CCN Forwarder



CCN Forwarder



CCN Node Model

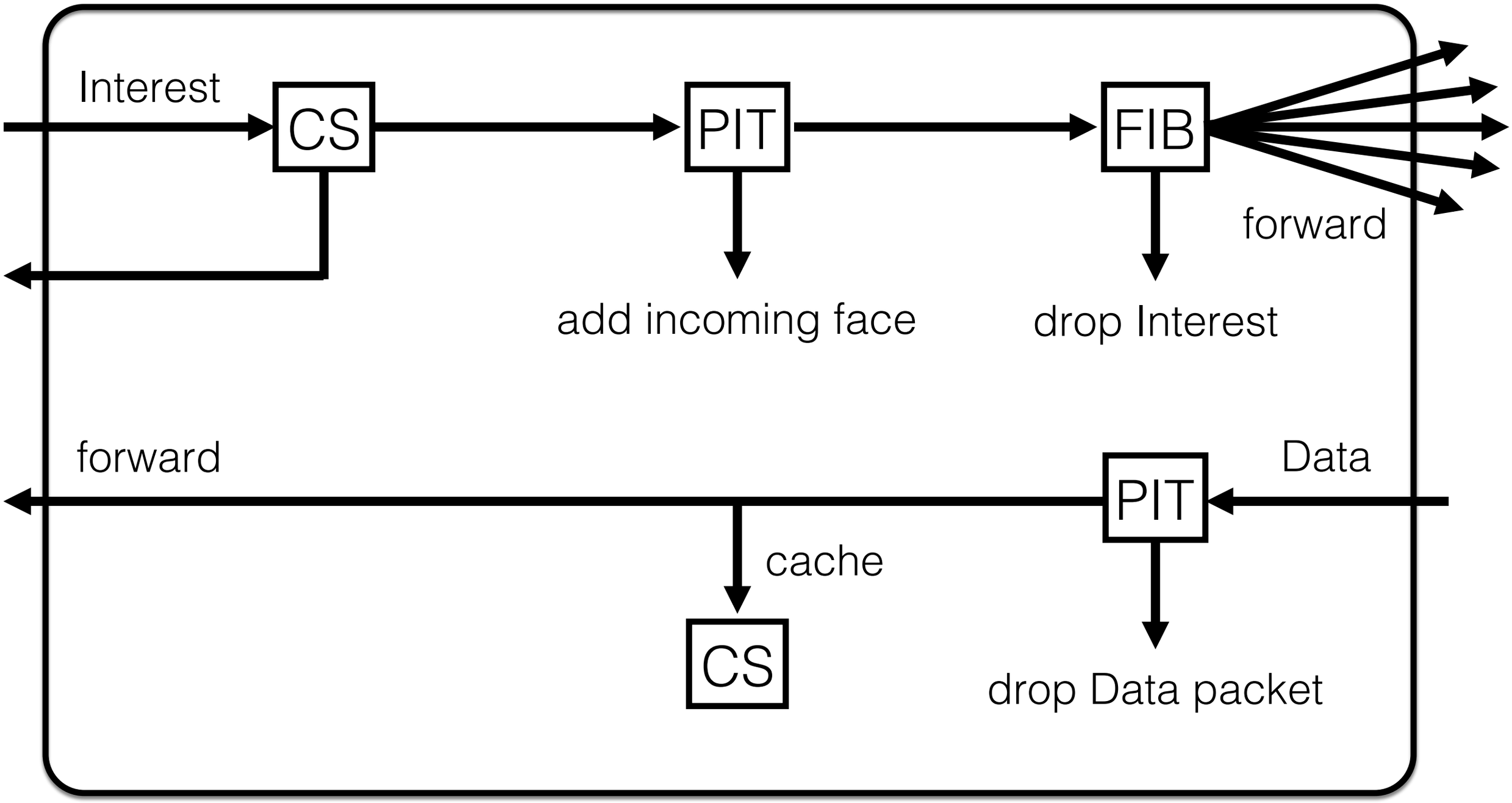


Longest matching prefix on names
Managed by external routing protocols

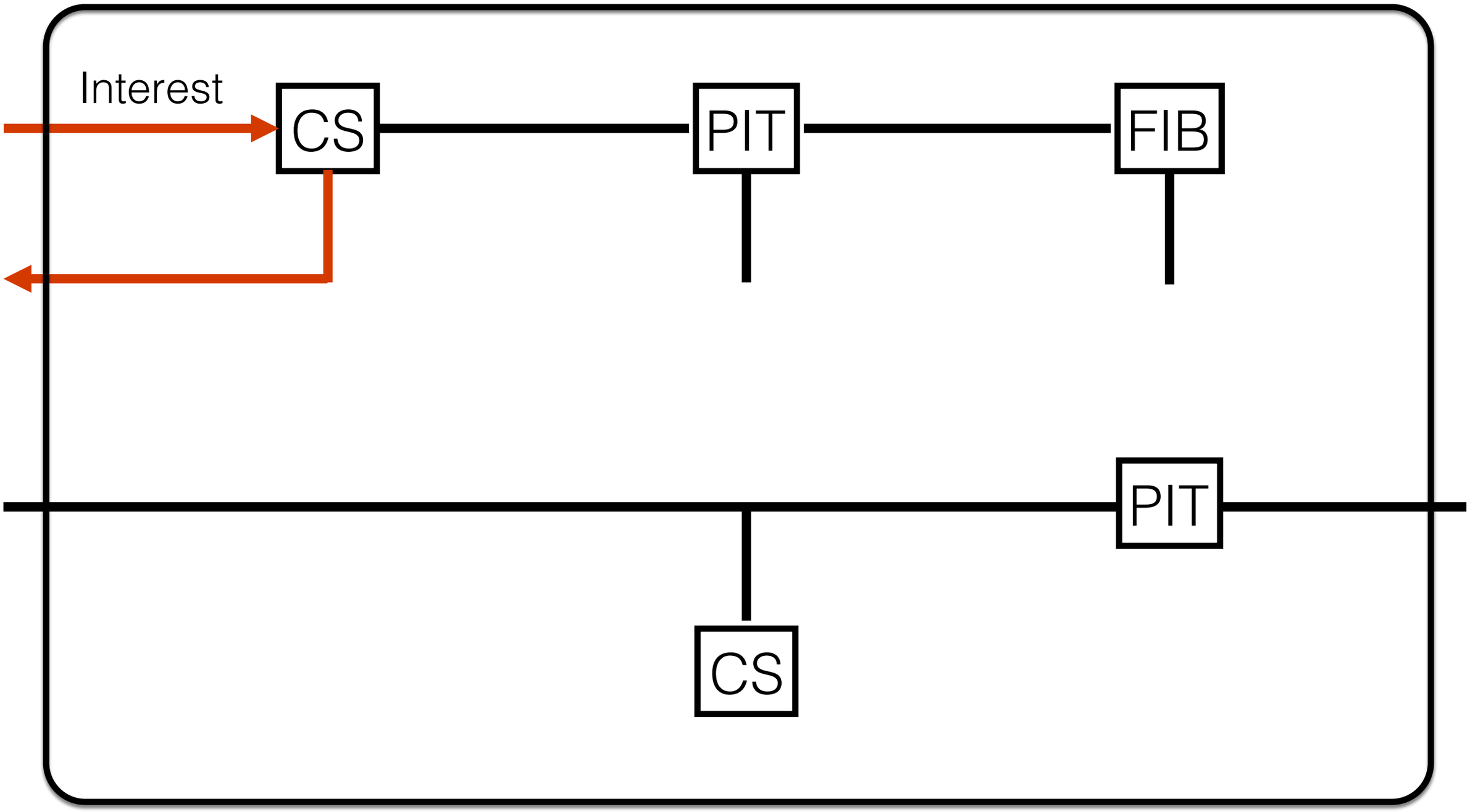
State for reverse-path forwarding

Long-term packet memory

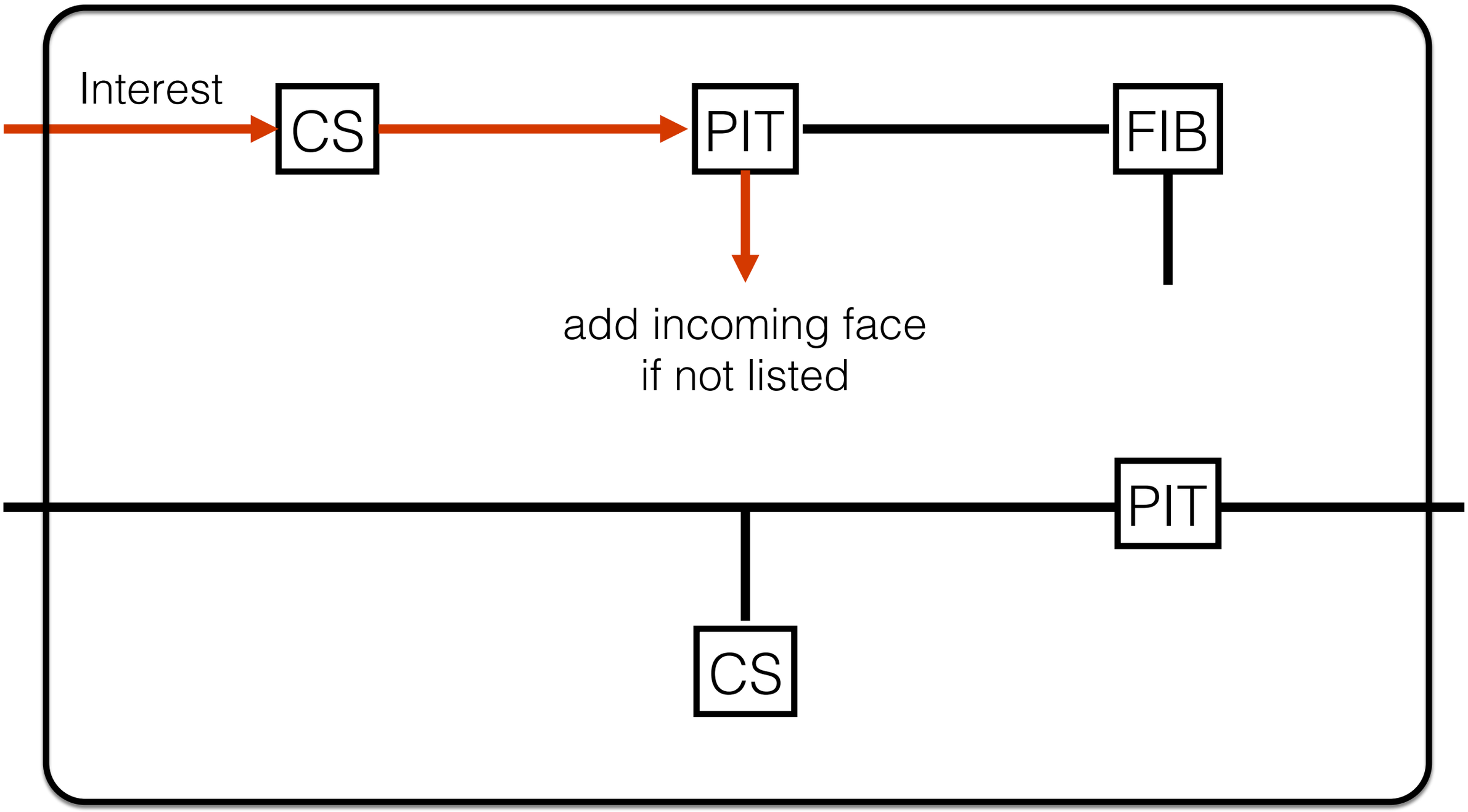
CCN Node Model



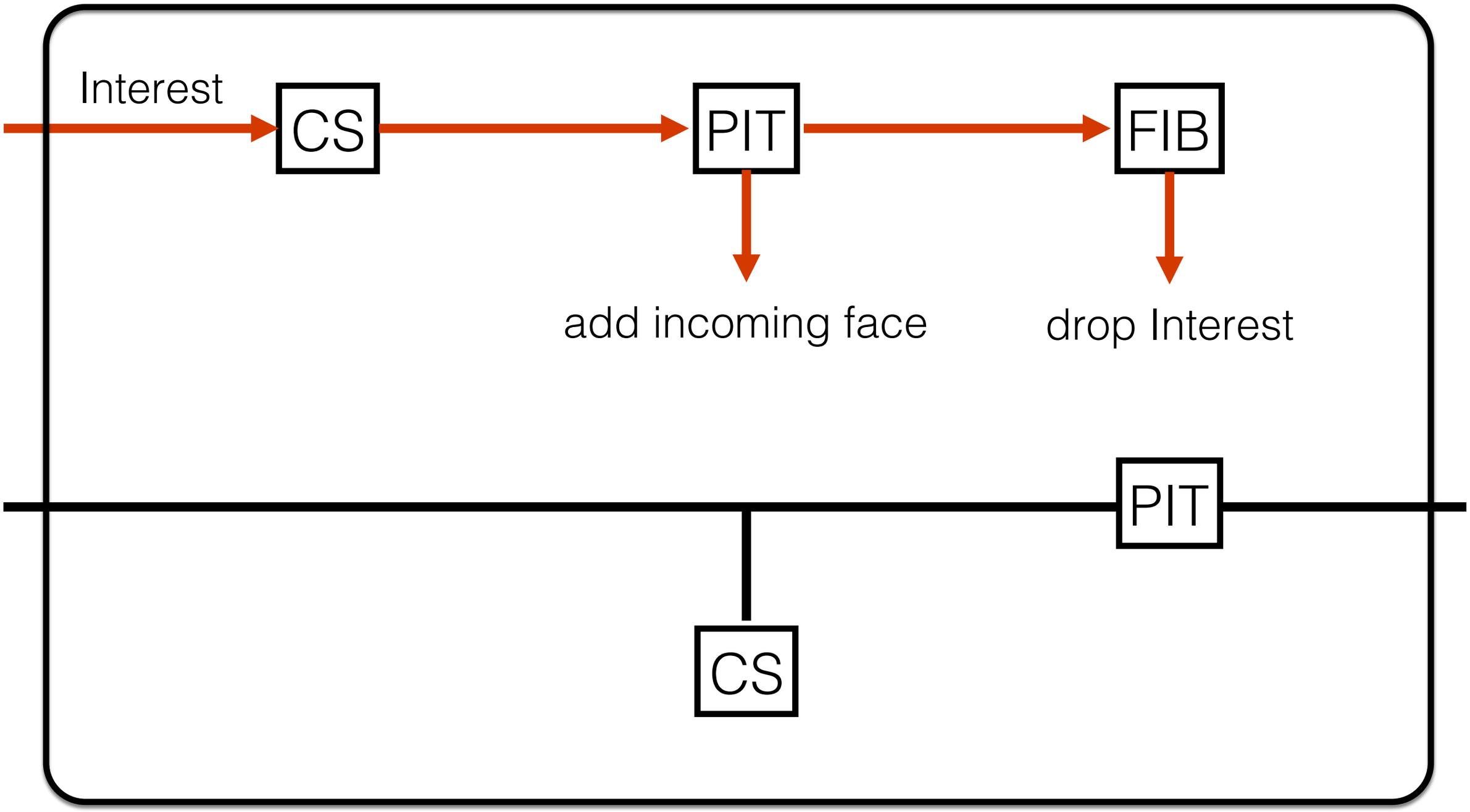
Interest satisfied by CS



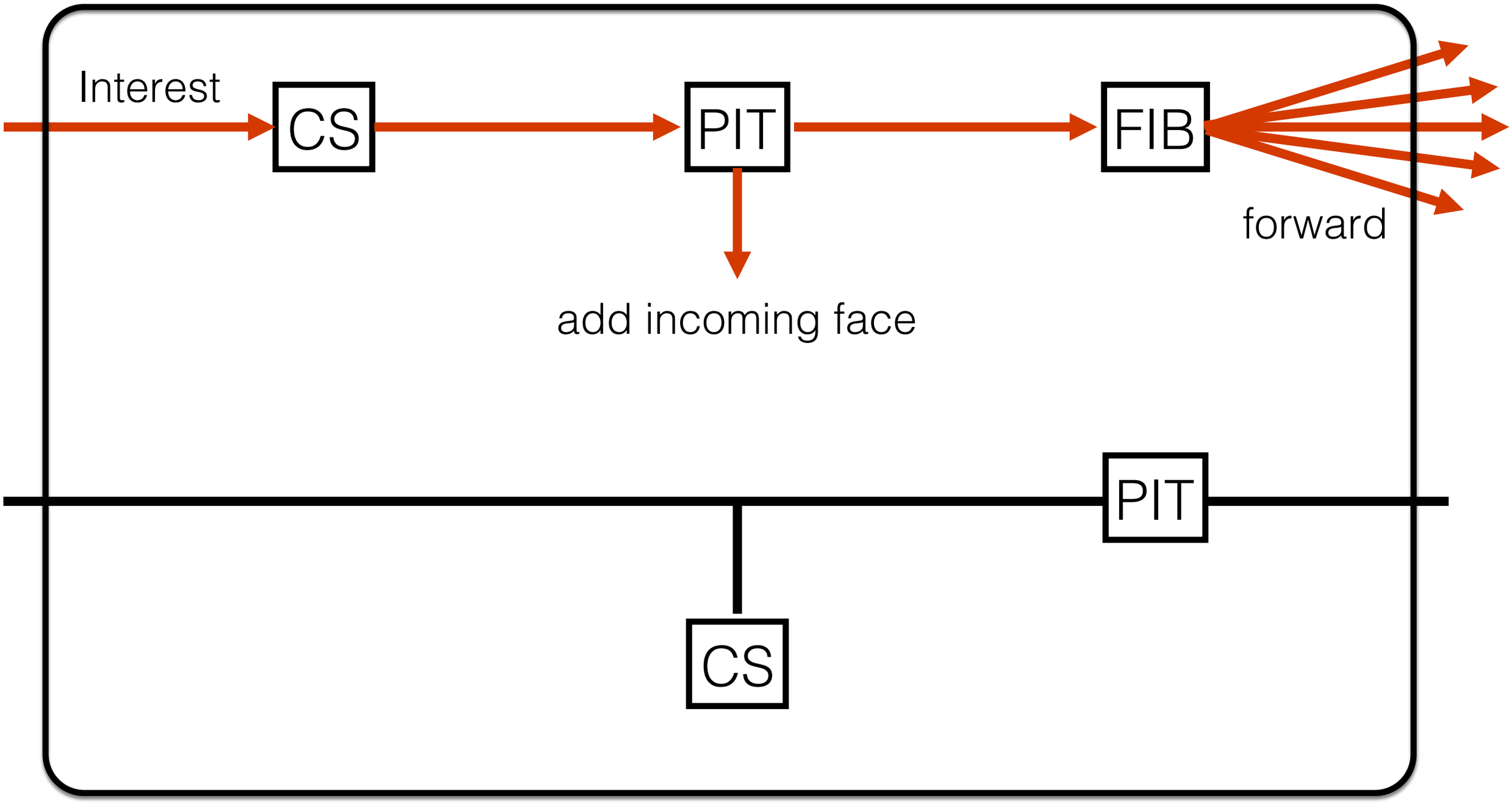
CS miss - Interest already in PIT



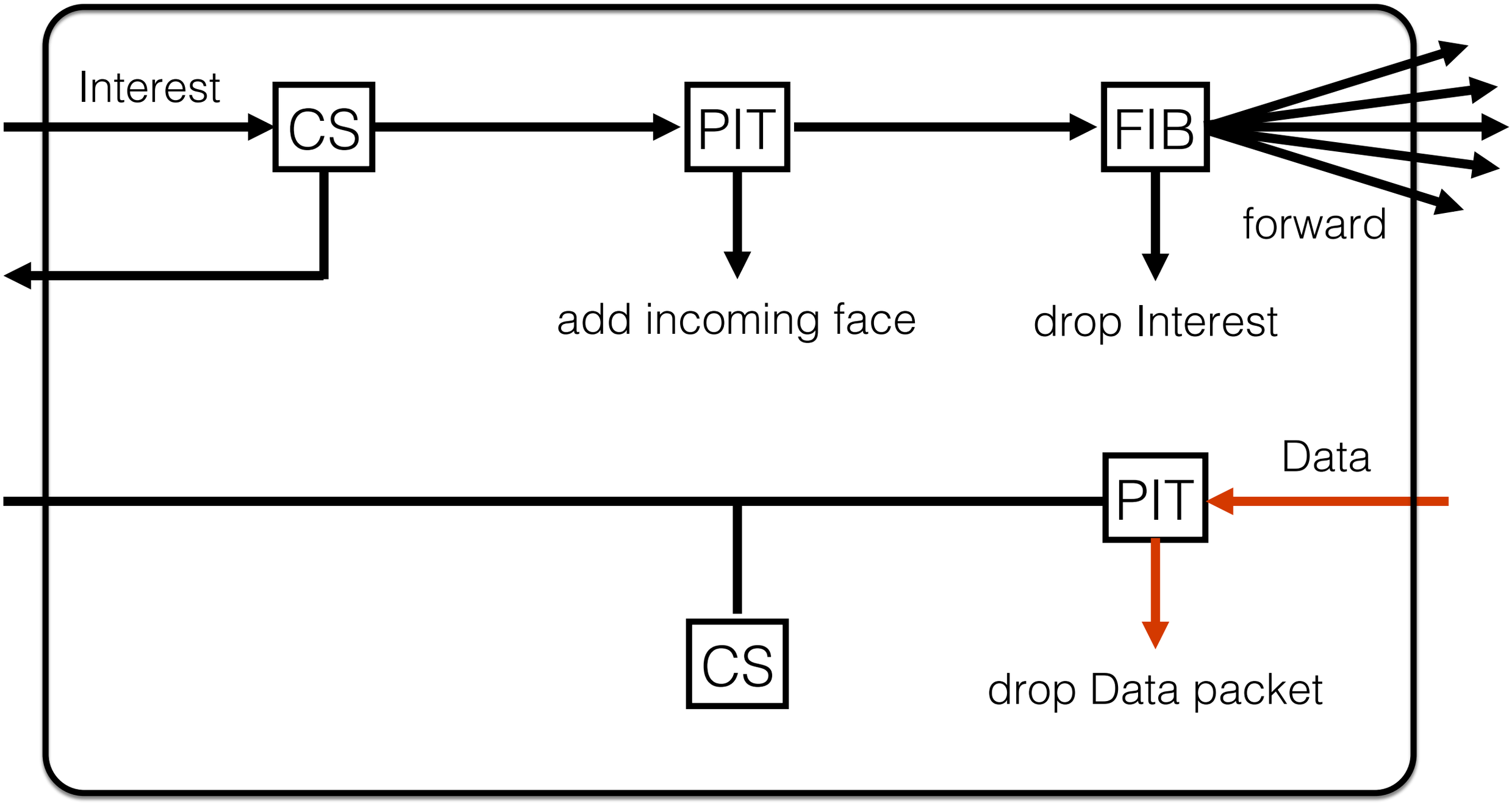
CS miss, PIT miss, no route in FIB for Interest



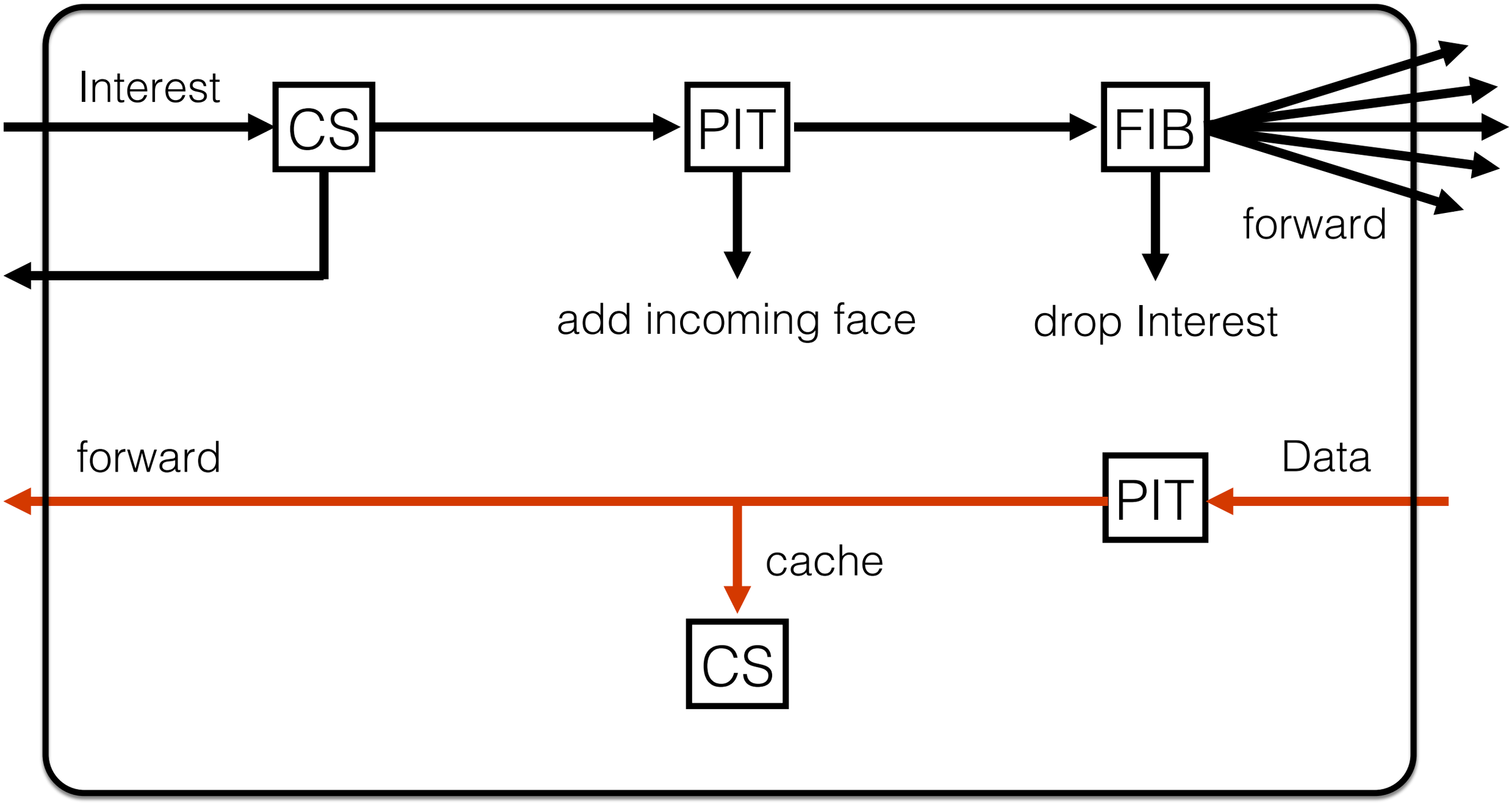
CS miss, PIT miss, route match in FIB - Forward via Strategy



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 Forwarding Strategy

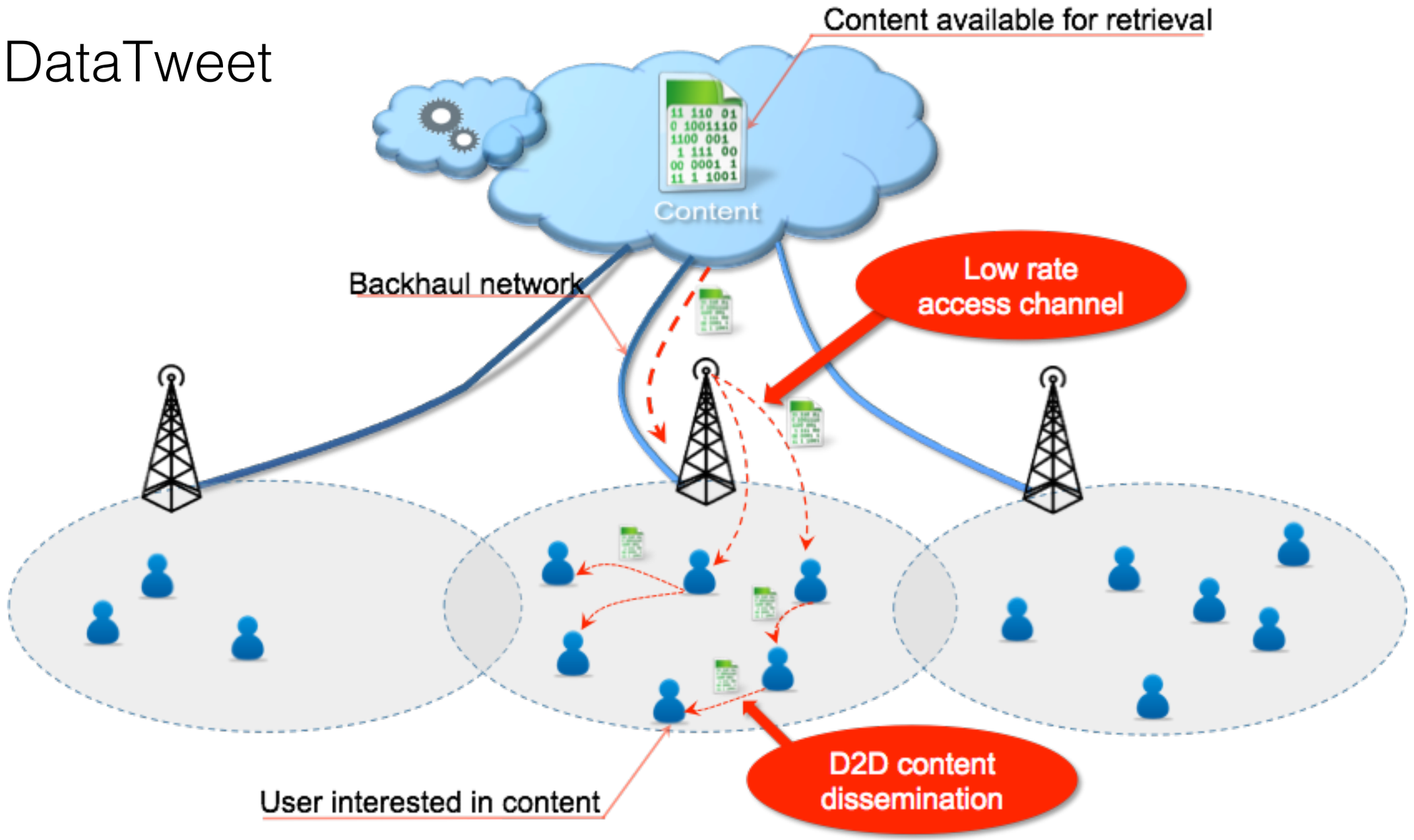
- Each entry can have a list of output interfaces
- Need for forwarding policies

FIB

/upmc	1,3,5

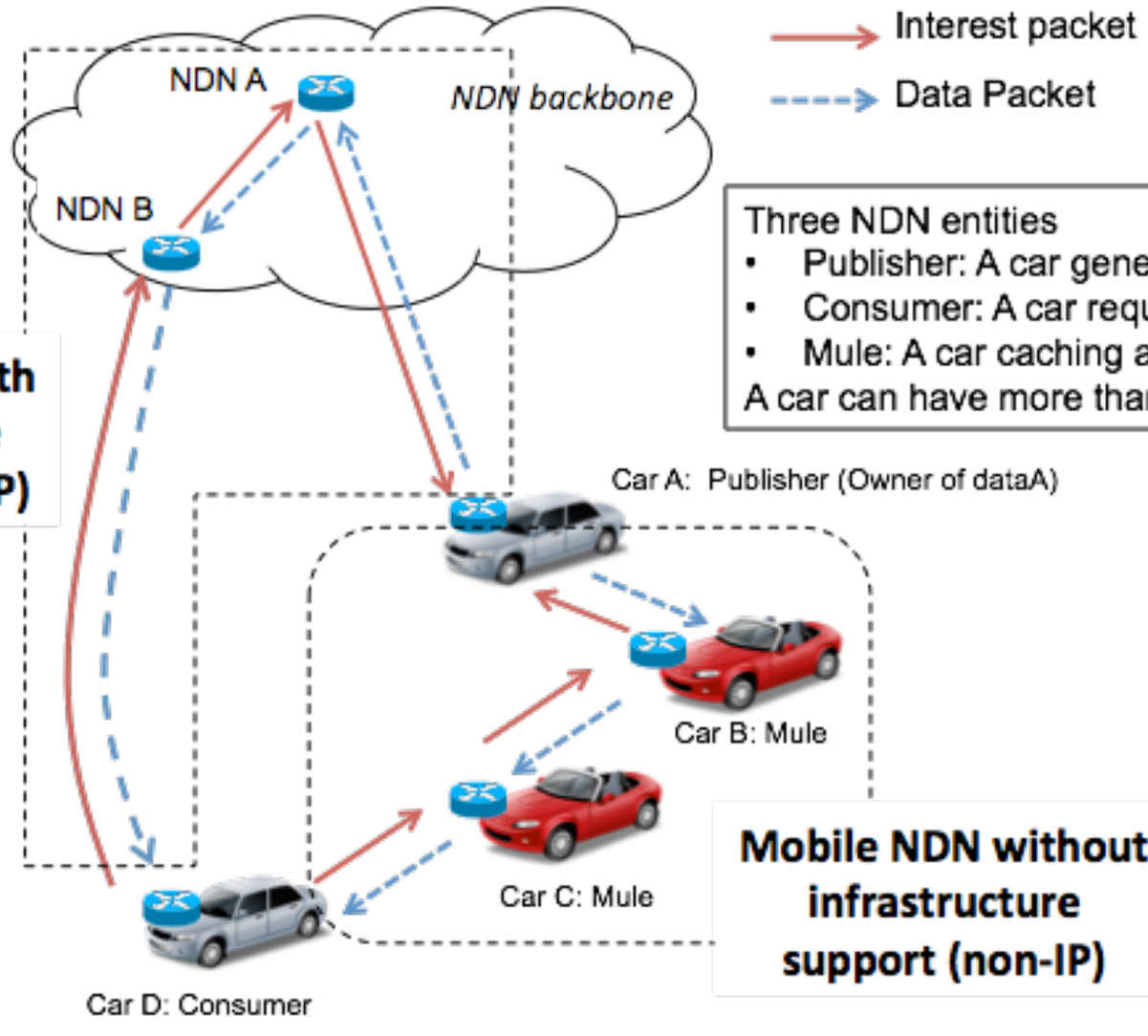
CCN at UPMC (2): Content Centricity in Constrained Cellular-Assisted D2D Communications

ANR DataTweet



CCN at UPMC (1): Vehicular Named Data Networking

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

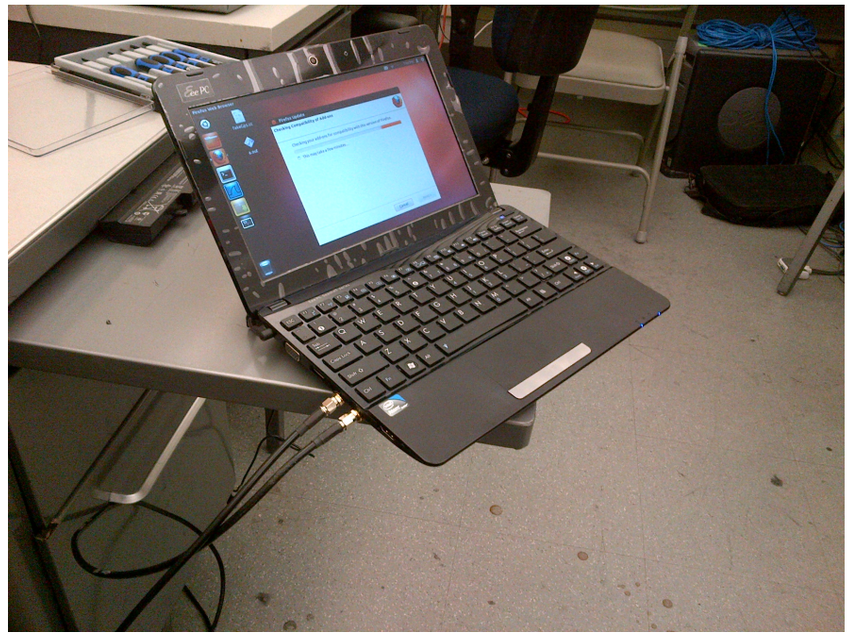


Three NDN entities
- Publisher: A car generating data
- Consumer: A car requesting data
- Mule: A car caching and forwarding data

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

Mobile NDN with infrastructure support (over IP)

Mobile NDN without infrastructure support (non-IP)



Takeaways

- CCN node is as simple as an IP node:

  - same memory requirements

  - same computational requirements (with option to increase security)

- CCN does near optimal content distribution.

- Security, delivery efficiency, mobility and disruption tolerance than TCP/IP

- CCNx, NDN: Tools for experimenting new apps in emerging environments

Thank You !!!

Information Centric Networking

MSc in Computer Networking
January 20, 2017

Prométhée Spathis
promethee.spathis@upmc.fr

Salah-Eddine Belouanas
salah-eddine.belouanas@lip6.fr