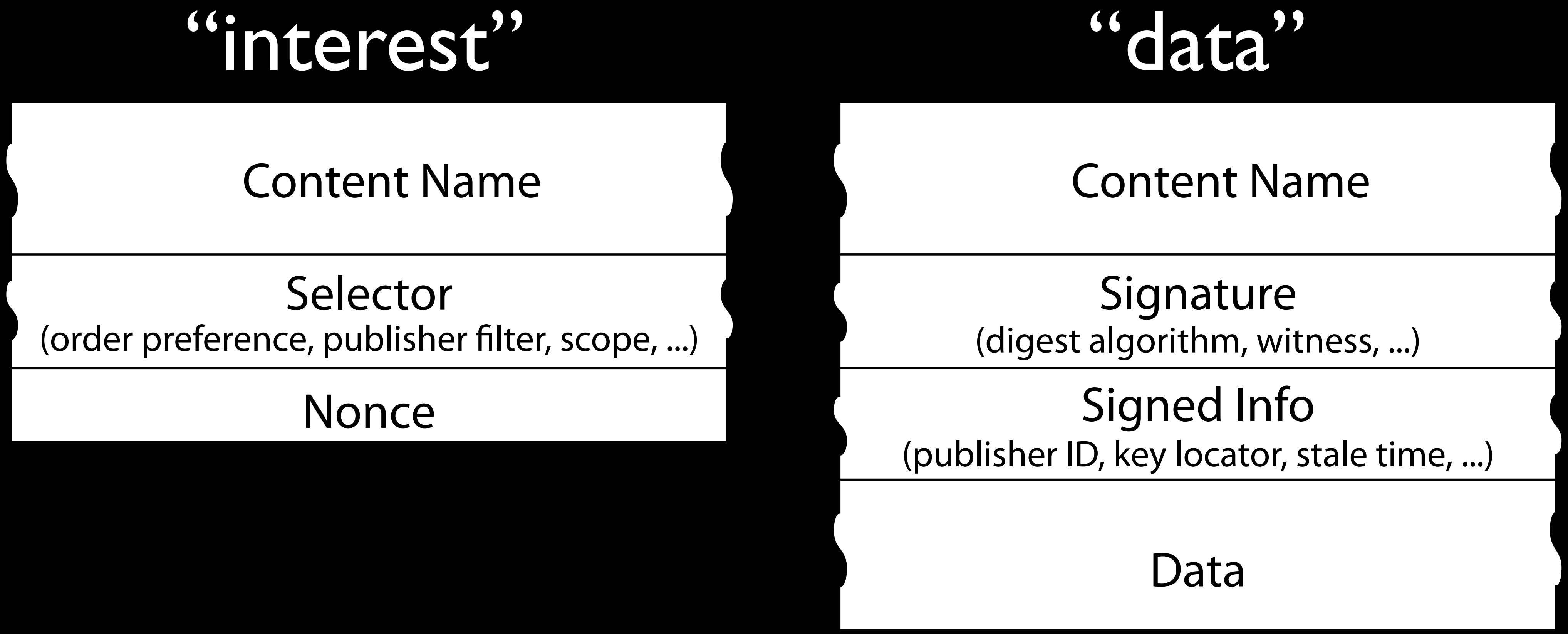
- History and motivation
- Content Model
- Security Model
- **Node Model**
- Routing
- Transport

CCN packets



“interest”

Content Name

Selector
(order preference, publisher filter, scope, ...)

Nonce

“data”

Content Name

Signature
(digest algorithm, witness, ...)

Signed Info
(publisher ID, key locator, stale time, ...)

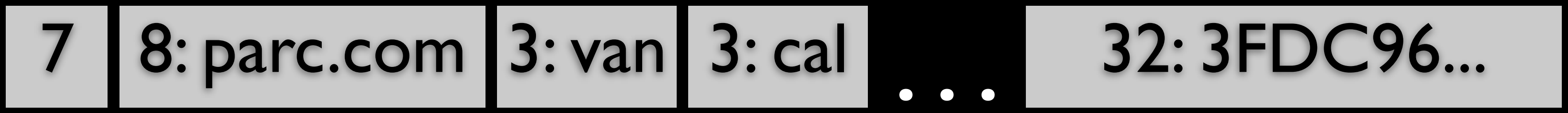
Data

There are just two CCN packet types - *interest* (similar to http “get”) and *data* (similar to http response). Both are encoded in an efficient binary XML.

Internally, CCN names are opaque, structured byte strings

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

is represented as a component count
then, for each component, a byte count
followed by that many bytes:



7 | 8: parc.com | 3: van | 3: cal | . . . | 32: 3FDC96...

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

Using Names

- The hierarchical structure is used to do ‘longest match’ lookups (similar to IP prefix lookups) which helps guarantee log(n) state scaling for globally accessible data.
- Although CCN names are longer than IP identifiers, their *explicit structure* allows lookups as efficient as IP’s.

(see hashing work by Rasmus Pagh and Martin Dietzfelbinger)

Names and meaning

- Like IP, a CCN node imposes no semantics on names — meaning comes from application, institution and global conventions reflected in prefix forwarding rules.

  For example,

  `/parc.com/people/van/presentations/FISS09`

  might be the name of a presentation's data and

  `/thisRoom/projector`

  the name of the projector it should display on.

- The former is a globally meaningful name leveraging the DNS global naming structure. The latter is local and context sensitive—it refers to different objects depending on the room you're in.

Basic CCN forwarding

- Consumer ‘broadcasts’ an ‘interest’ over any & all available communications media:

```
Want ‘/parc.com/van/presentation.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 that data:

```
HereIs ‘/parc.com/van/presentation.pdf/p1’ <data>
```

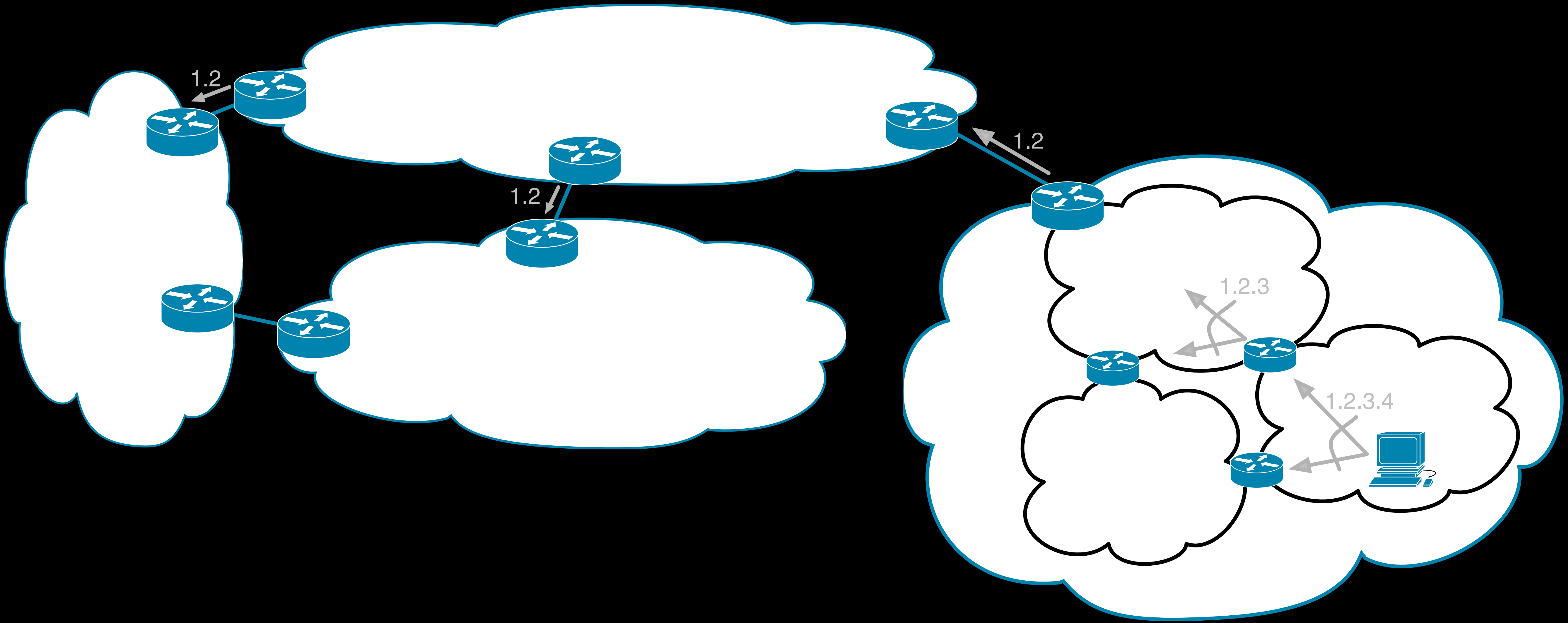
This isn’t google - it’s how IP works at the packet level

- Node announces (via ‘promiscuous ARP’) interest in packets sent to its IP address:

  ```
  arp 1.2.3.4 is-at 02:07:01:00:01:c4
  ```

- Interest is propagated via IP routing (with host-level granularity aggregated to subnet then net-level as interest gets farther away).
- Packets bound for node follow trail of interests (routes) back to it.

IP route propagation and aggregation



Basic CCN transport

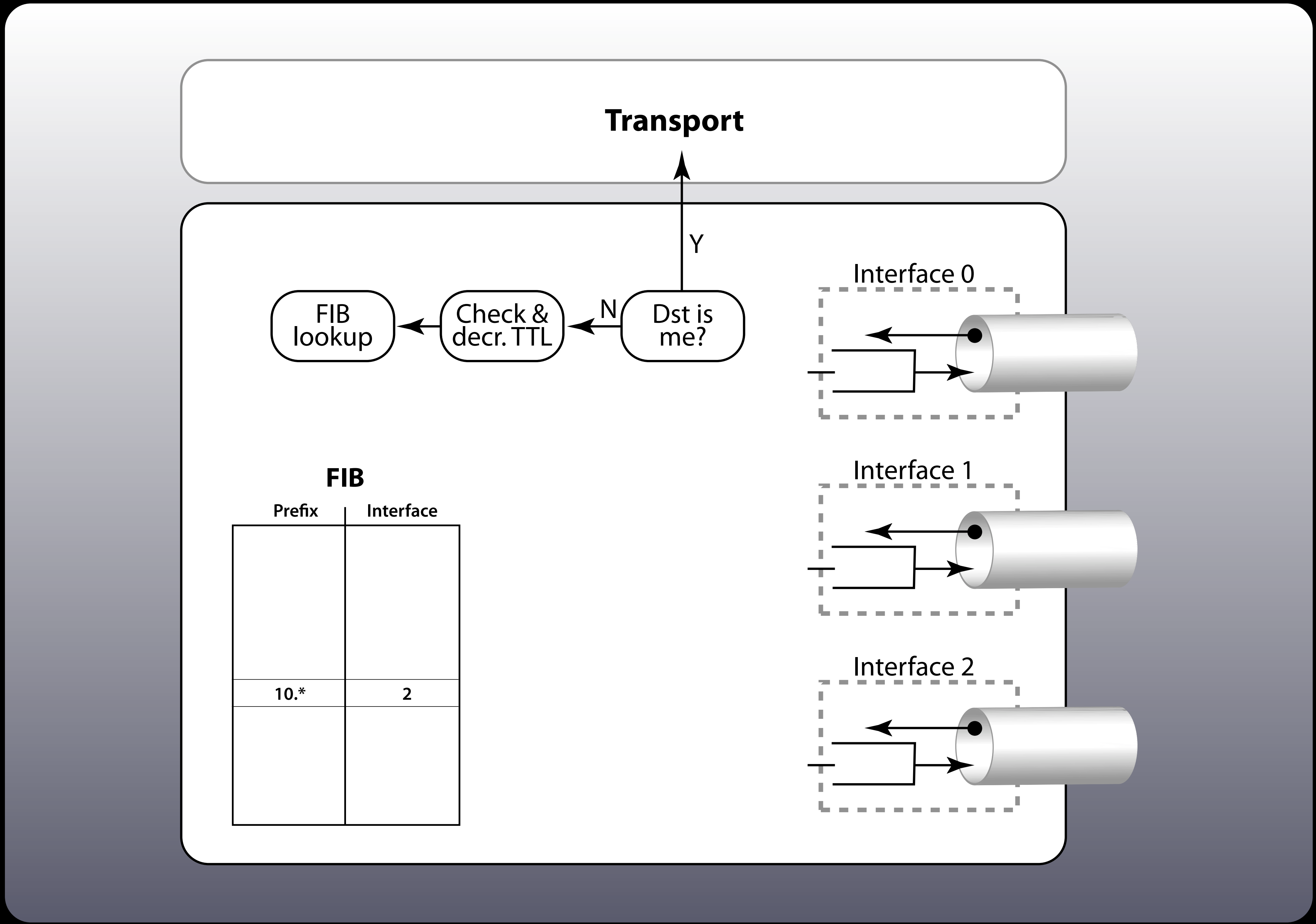
- Data that matches an interest ‘consumes’ it.
- Interest must be re-expressed to get new data. (Controlling the re-expression allows for traffic management and environmental adaptation.)
- Multiple (distinct) interests in same collection may be expessed (similar to TCP window).

TCP works this way at the packet level

- a TCP Ack expresses interest in new data on a connection.
- Sending new data consumes the Ack and another must generated to get more data.
- Multiple Acks (and associated data packets) may be in transit simultaneously.
- System always operates in bounded flow balance regime which is robust and stable under arbitrary aggregate demand.

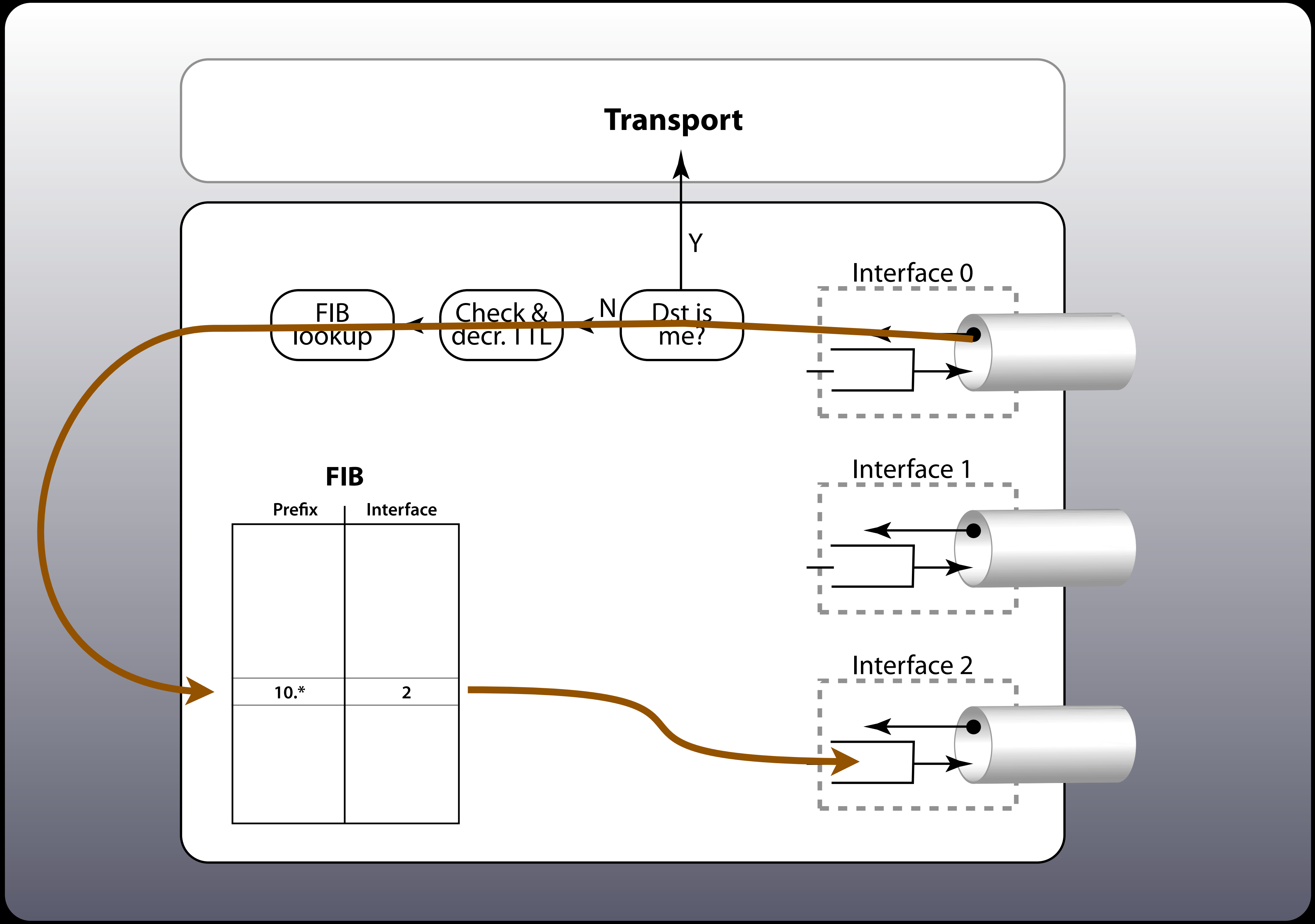
(see “Reversibility and Stochastic Networks”, Frank Kelly, 1979)

IP node model



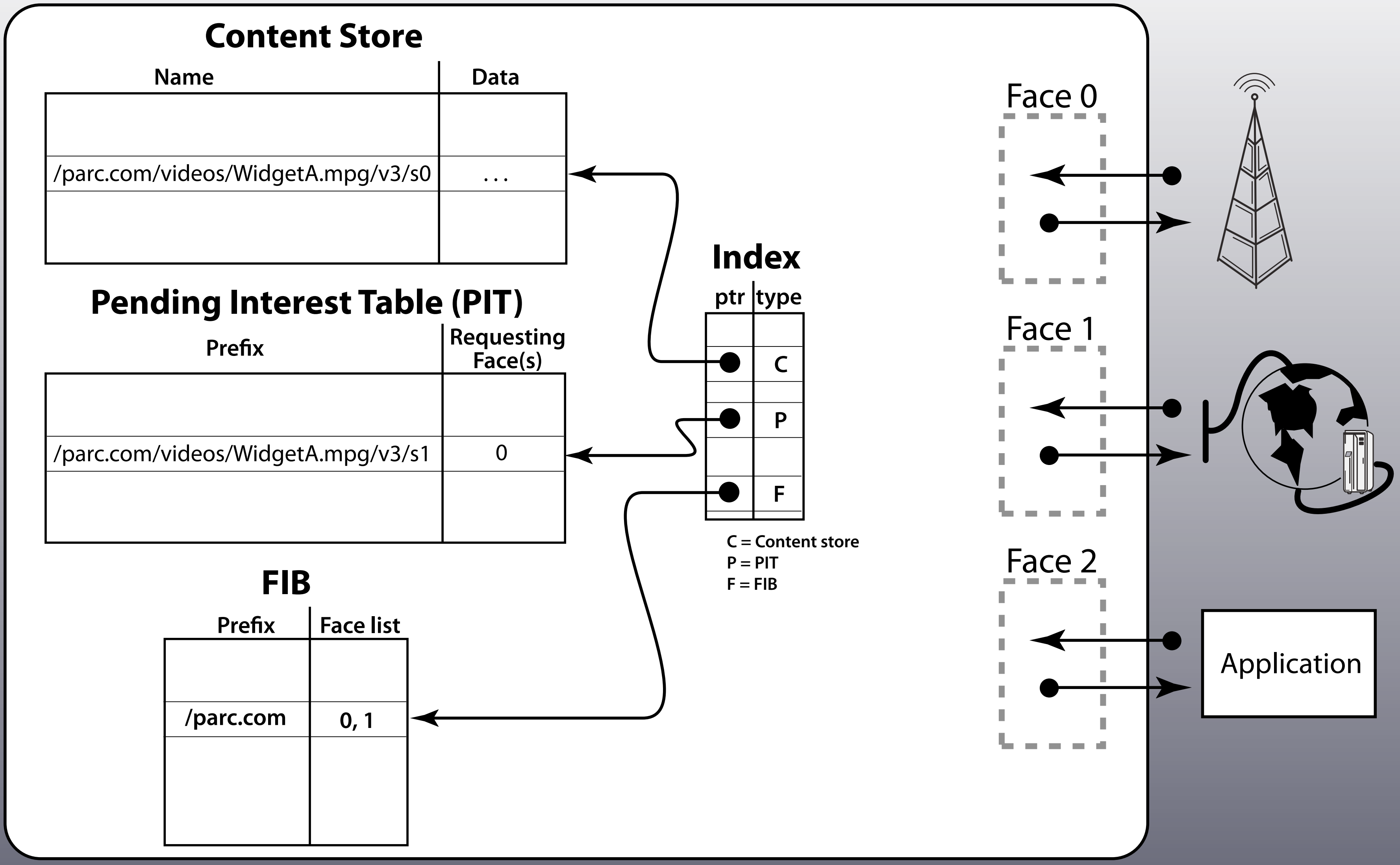
Prefix	Interface
10.*	2

IP node model



Prefix	Interface
10.*	2

CCN node model



Content Store

Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

Pending Interest Table (PIT)

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

FIB

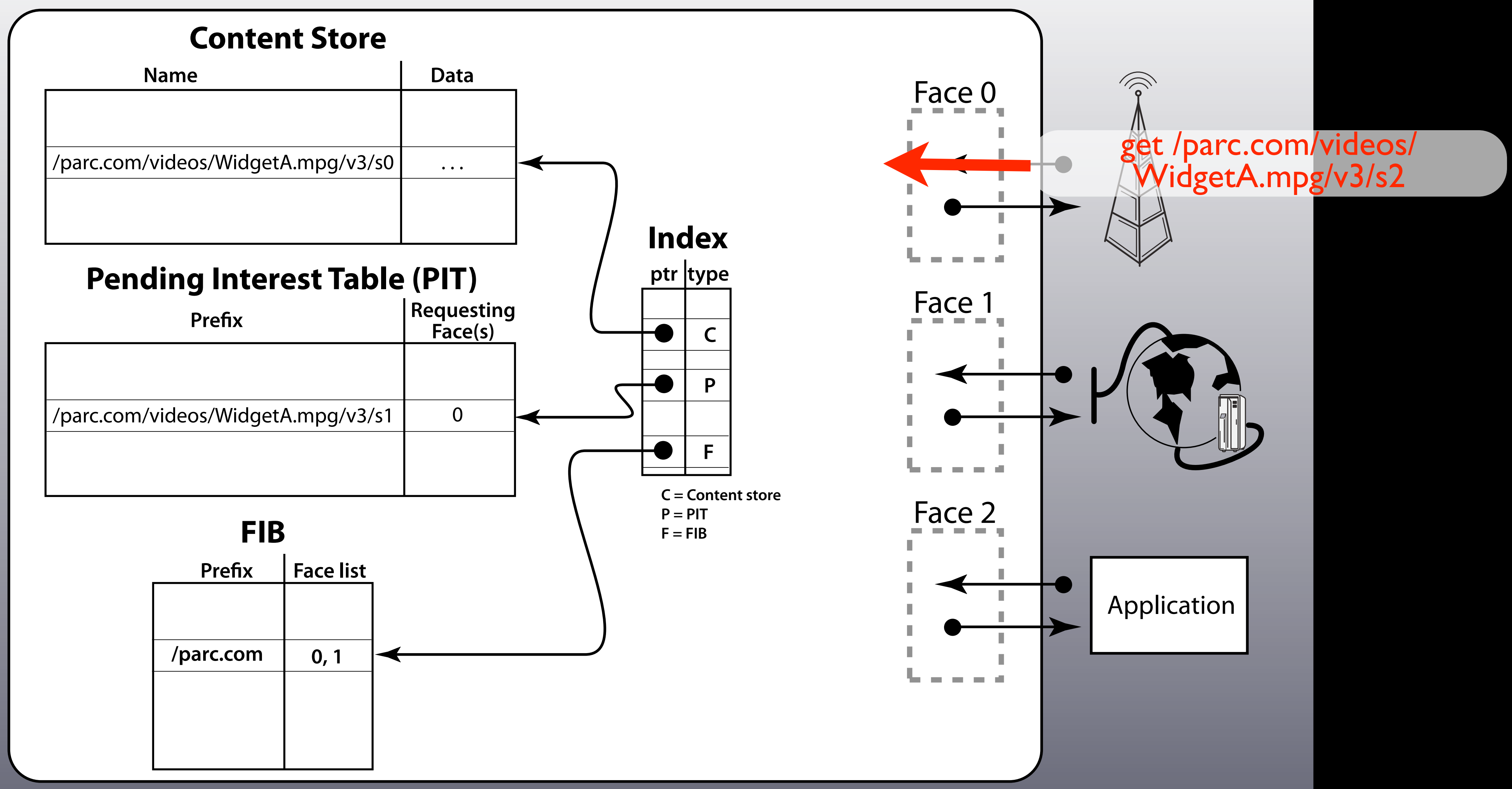
Prefix	Face list
/parc.com	0, 1

Index

ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB

CCN node model



Content Store

Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

Pending Interest Table (PIT)

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

FIB

Prefix	Face list
/parc.com	0, 1

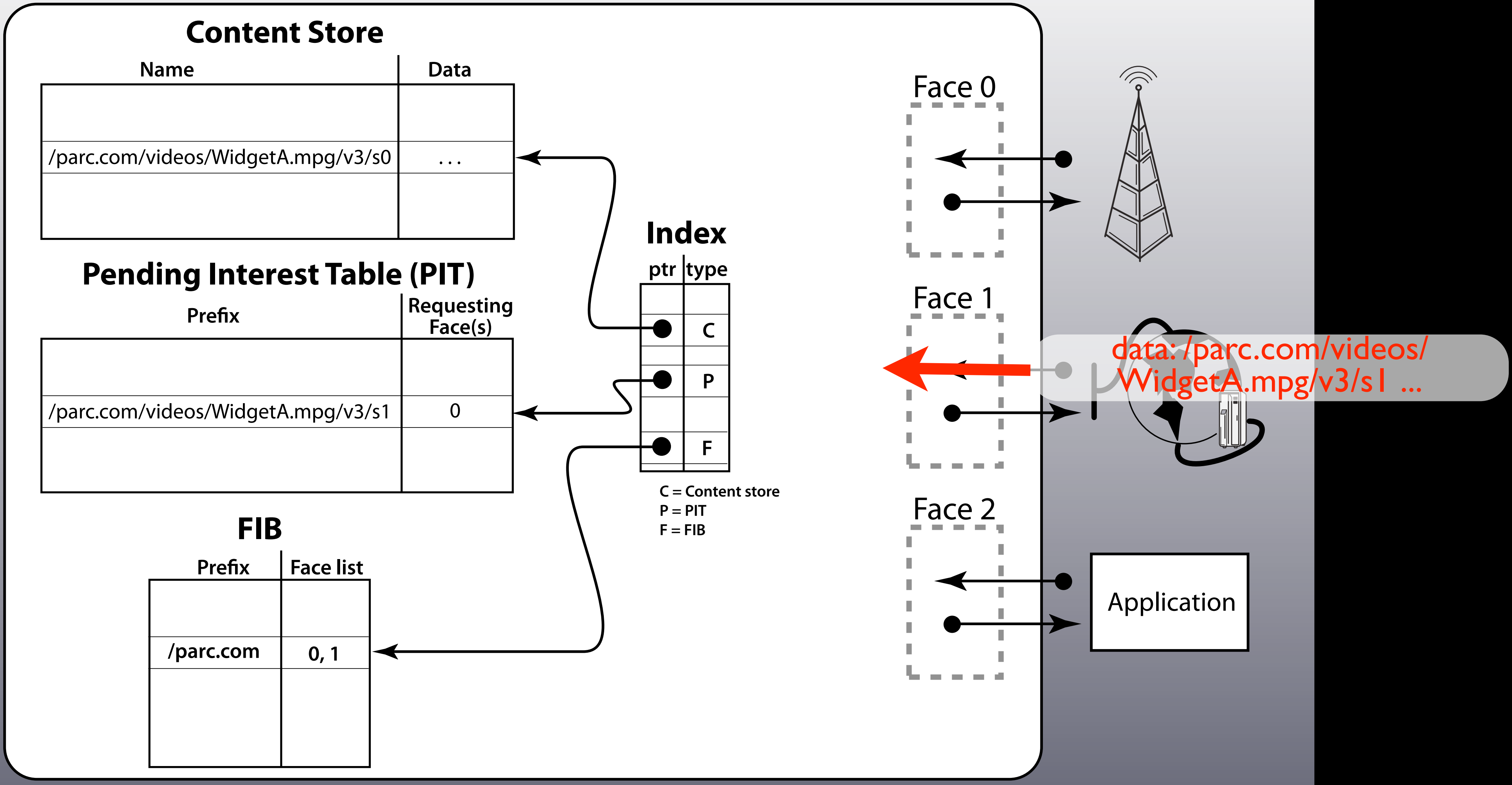
Index

ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB

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

CCN node model



Content Store

Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

Pending Interest Table (PIT)

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

FIB

Prefix	Face list
/parc.com	0, 1

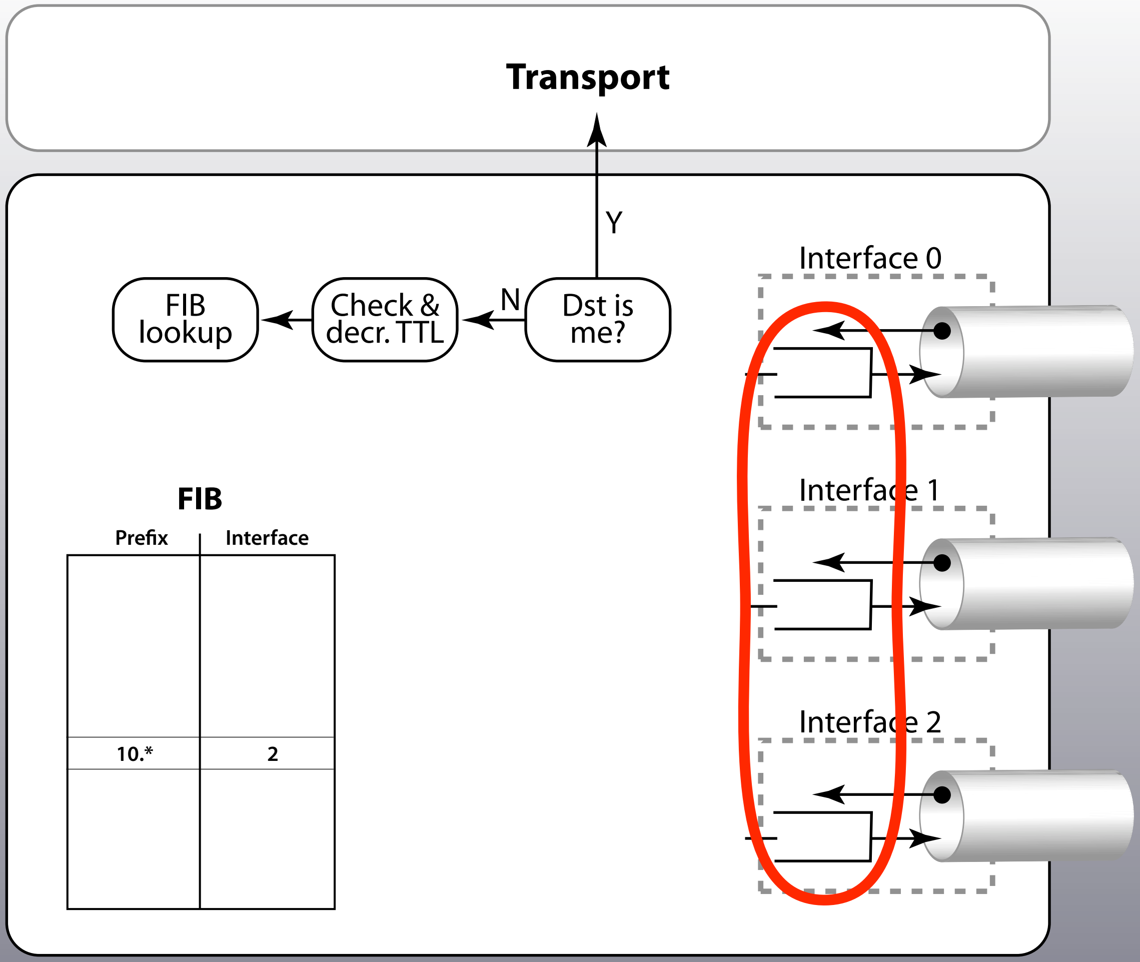
Index

ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB

data: /parc.com/videos/WidgetA.mpg/v3/s1 ...

Comparison



Transport

Y

N

FIB lookup

Check & decr. TTL

Dst is me?

Interface 0

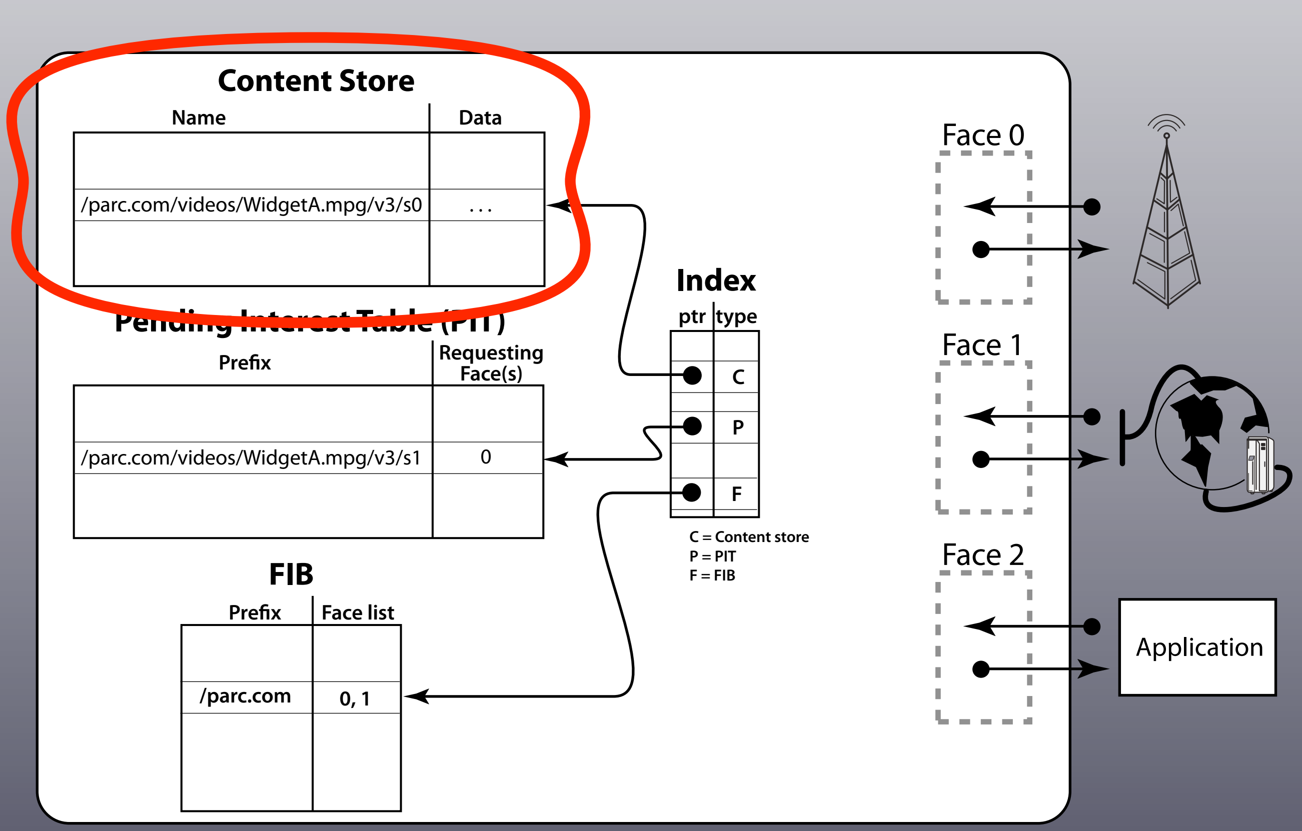
Interface 1

Interface 2

FIB

Prefix	Interface
10.*	2

Content Store is same as buffer memory - same contents, different replacement policy.



Content Store

Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

Pending Interest Table (PIT)

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

FIB

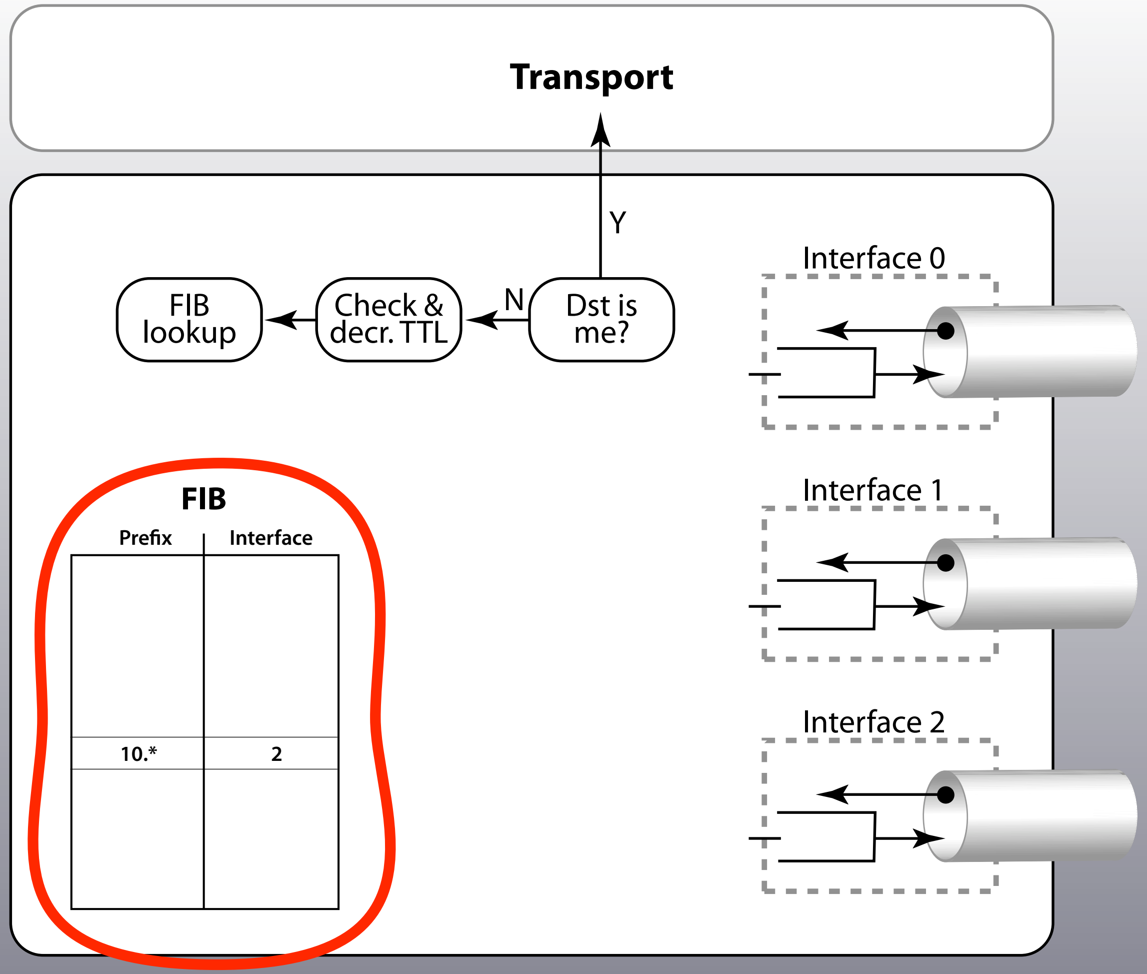
Prefix	Face list
/parc.com	0, 1

Index

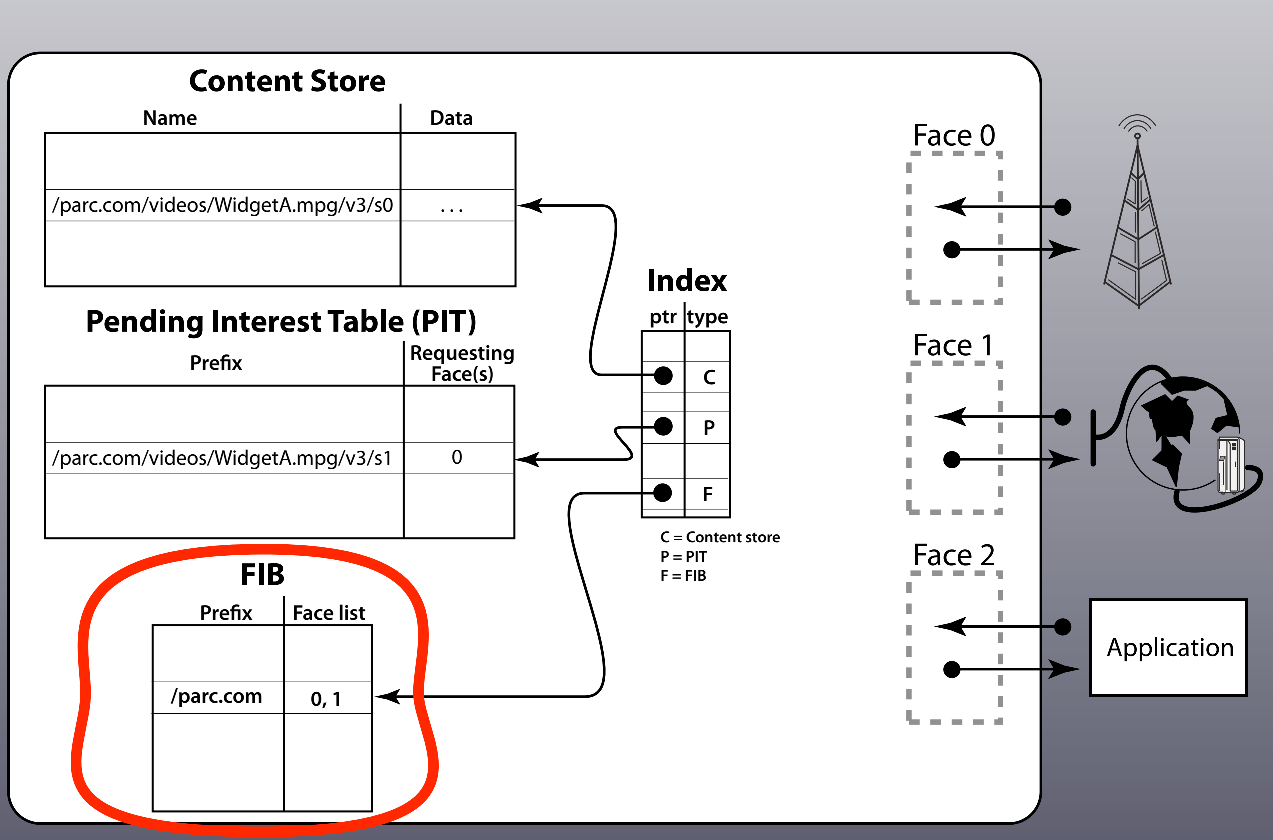
ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB

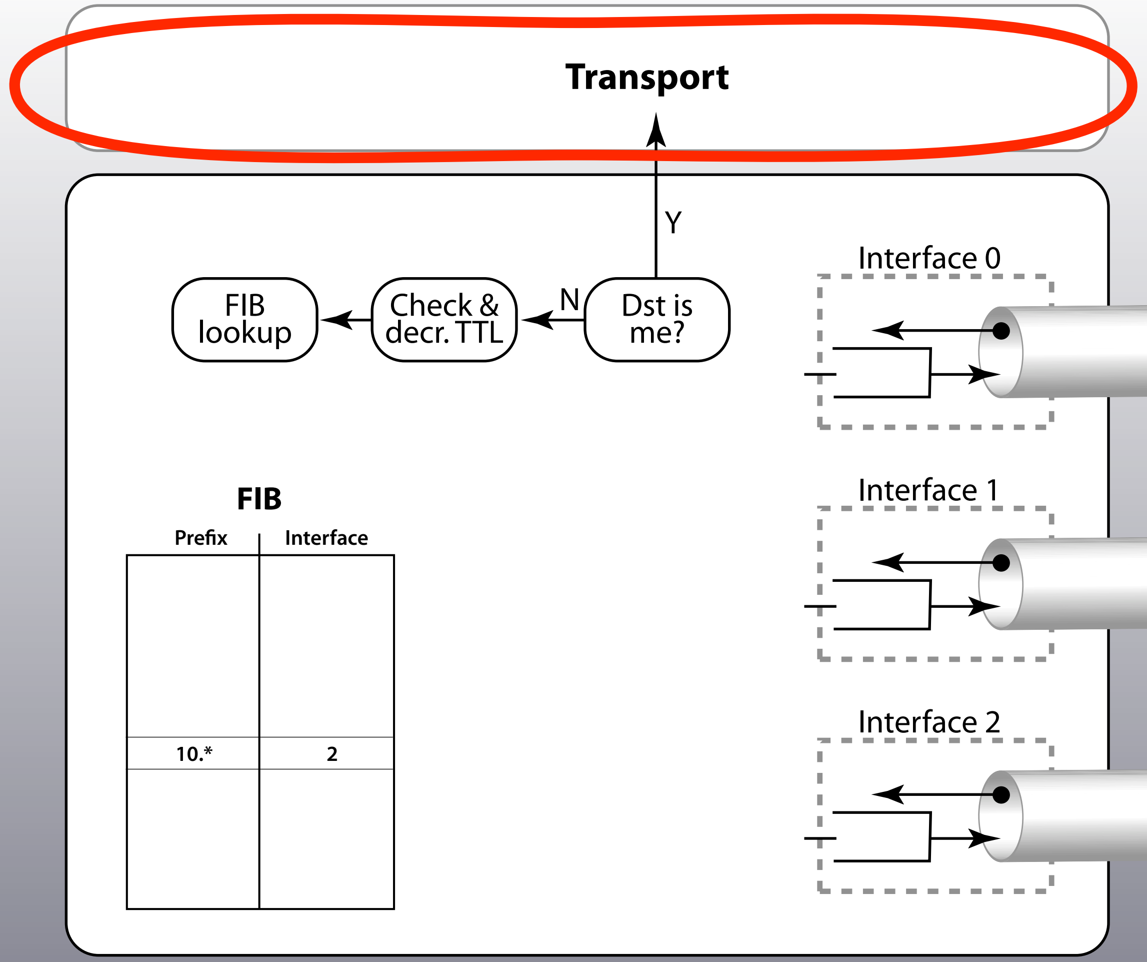
Comparison



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

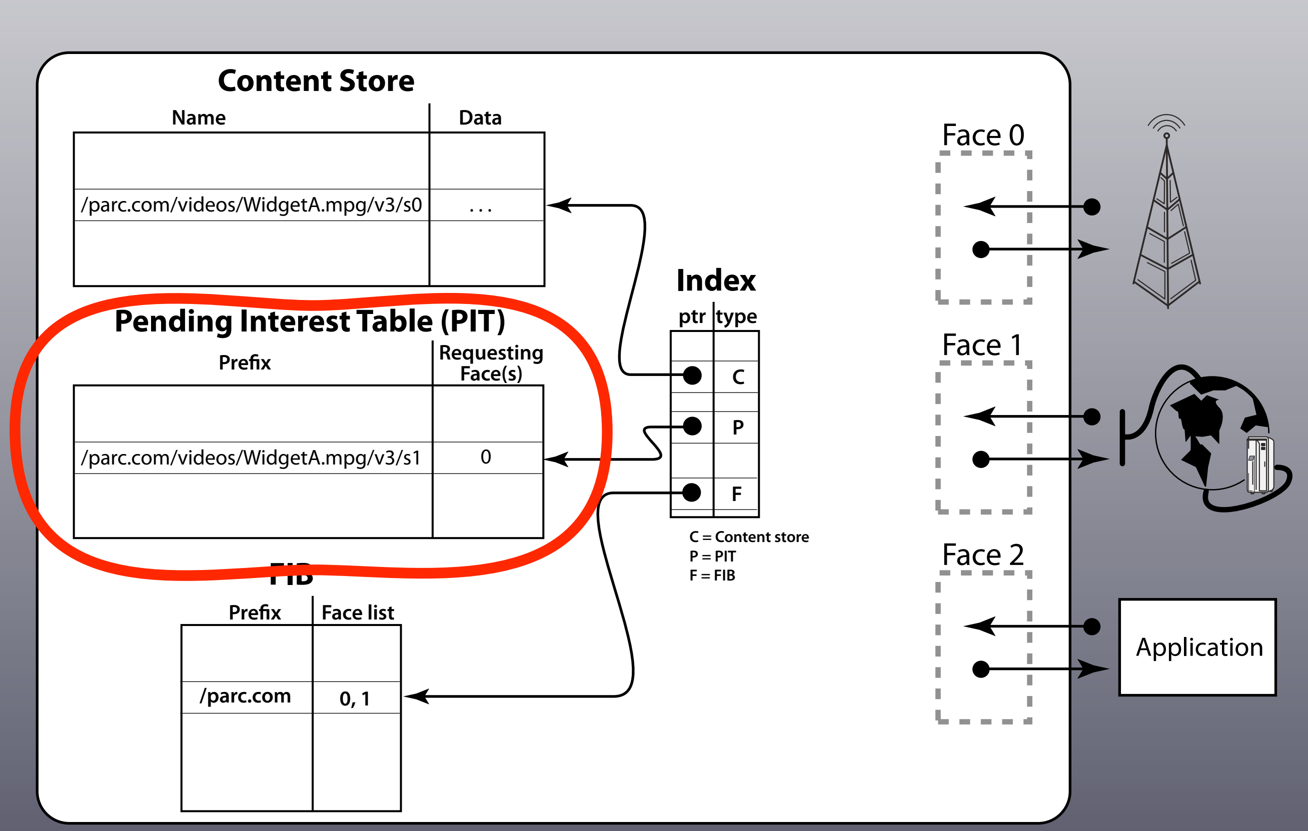


Comparison



Prefix	Interface
10.*	2

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



Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

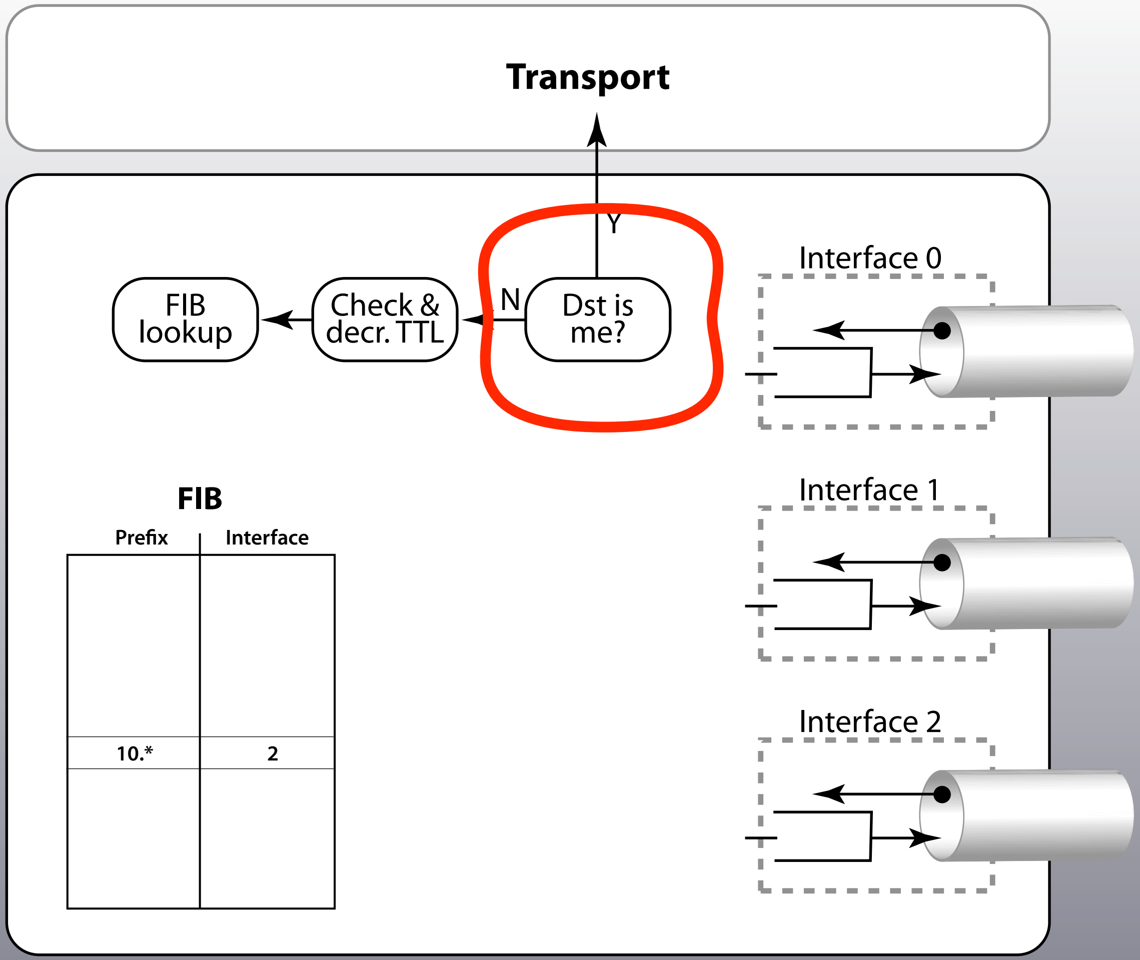
Prefix	Requesting Face(s)
/parc.com/videos/WidgetA.mpg/v3/s1	0

Prefix	Face list
/parc.com	0, 1

ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB

Comparison



Transport

Y

N

FIB lookup

Check & decr. TTL

Dst is me?

Interface 0

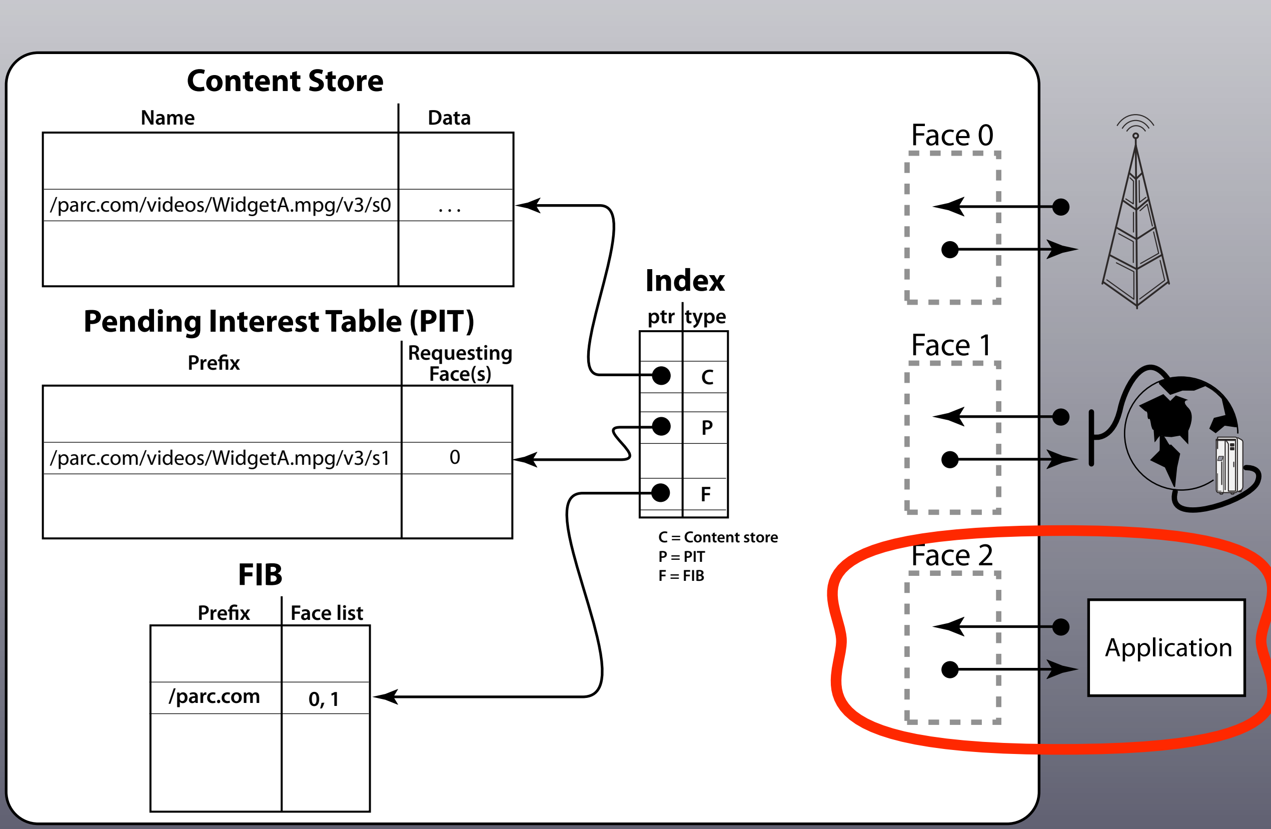
Interface 1

Interface 2

FIB

Prefix	Interface
10.*	2

There's no "me" test since CCN demuxing & forwarding are the same



Content Store

Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

Pending Interest Table (PIT)

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

FIB

Prefix	Face list
/parc.com	0, 1

Index

ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB

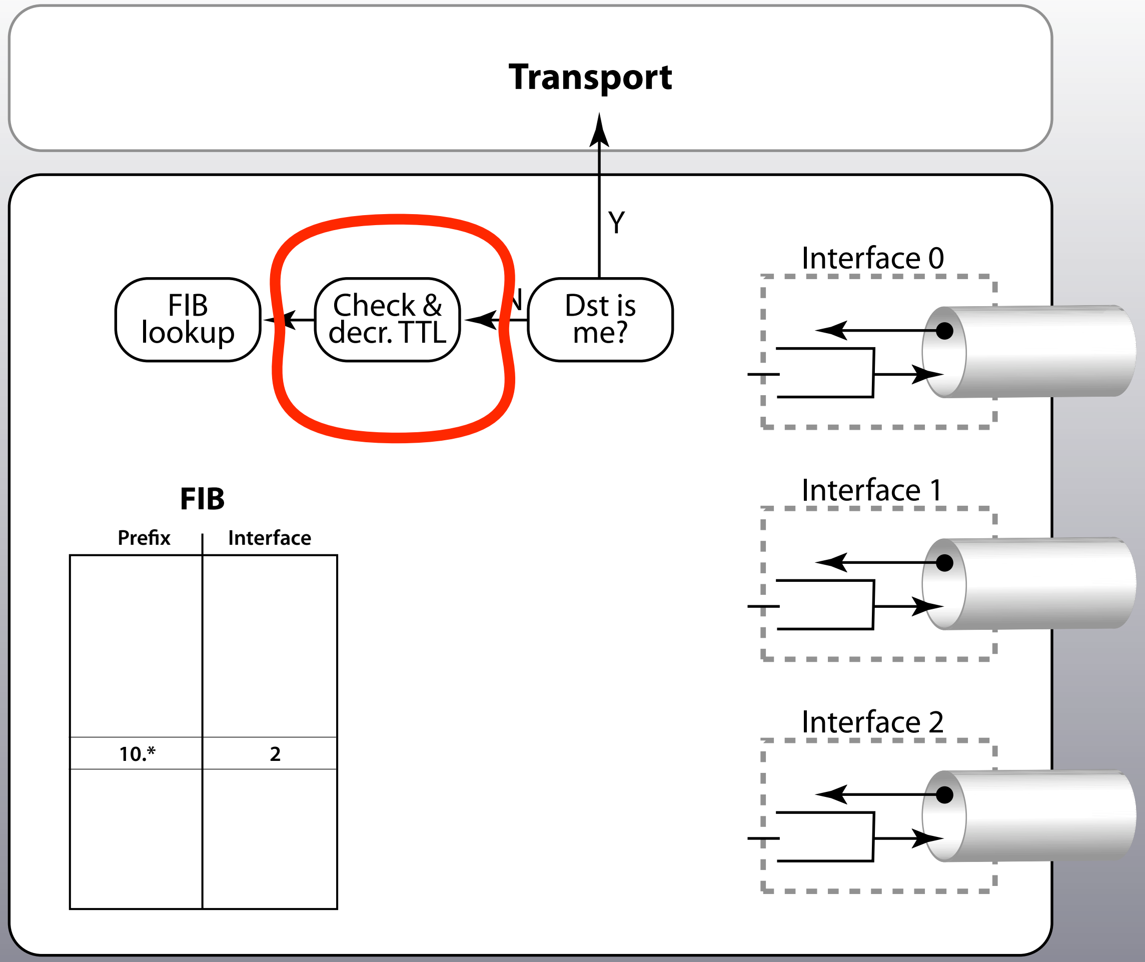
Face 0

Face 1

Face 2

Application

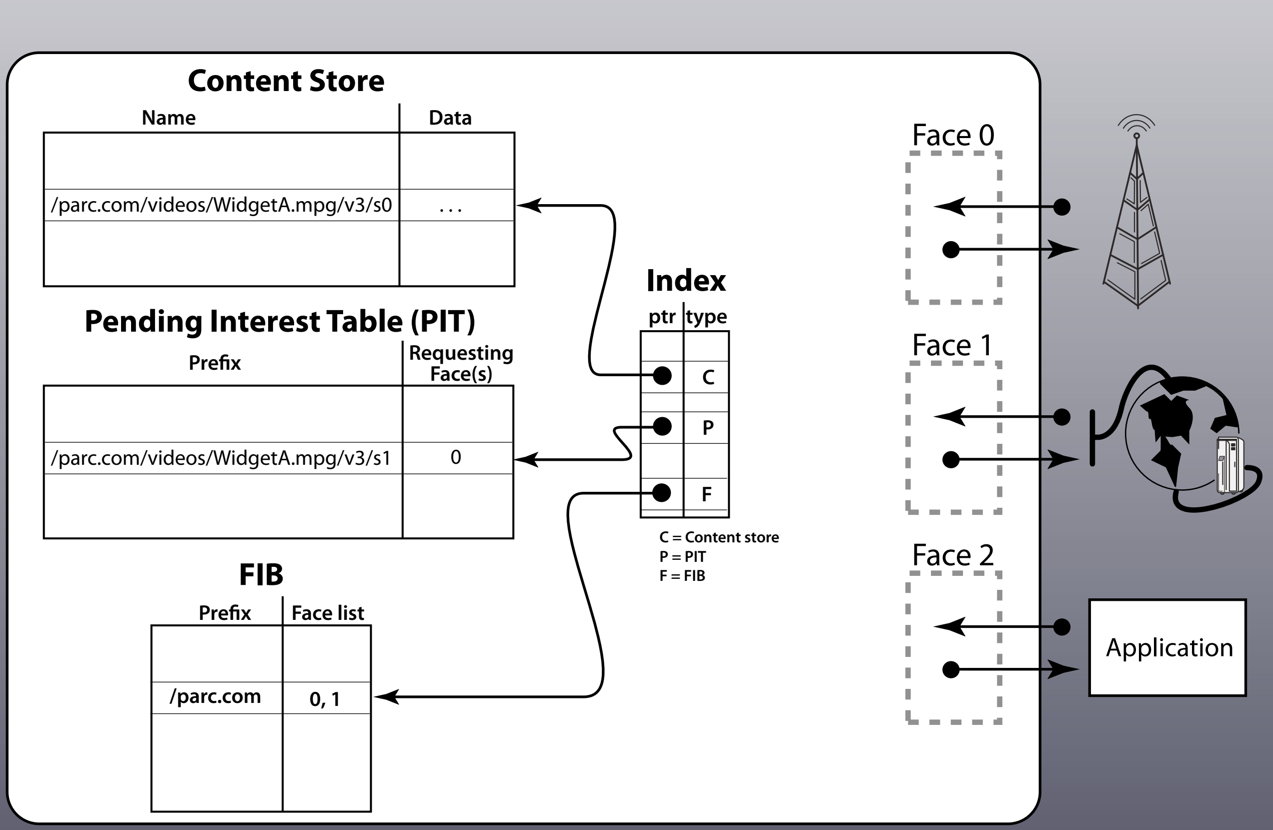
Comparison



FIB

Prefix	Interface
10.*	2

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



Content Store

Name	Data
/parc.com/videos/WidgetA.mpg/v3/s0	. . .

Pending Interest Table (PIT)

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

FIB

Prefix	Face list
/parc.com	0, 1

Index

ptr	type
●	C
●	P
●	F

C = Content store
P = PIT
F = FIB