Content Centric Networking

Van Jacobson
Palo Alto Research Center (PARC)

IETF77 ISOC Internet Researchers meeting
Anaheim, CA
24 March 2010

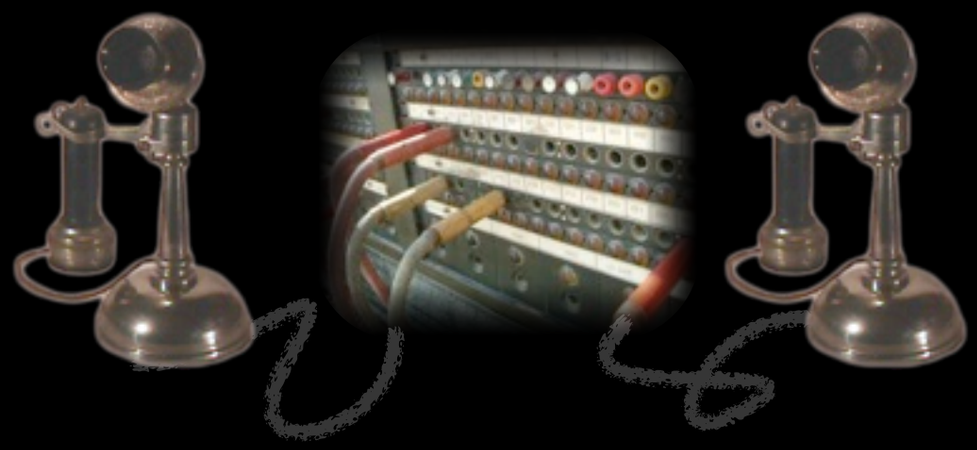
This talk describes ongoing PARC work on CCN (Content-centric Networking) by:

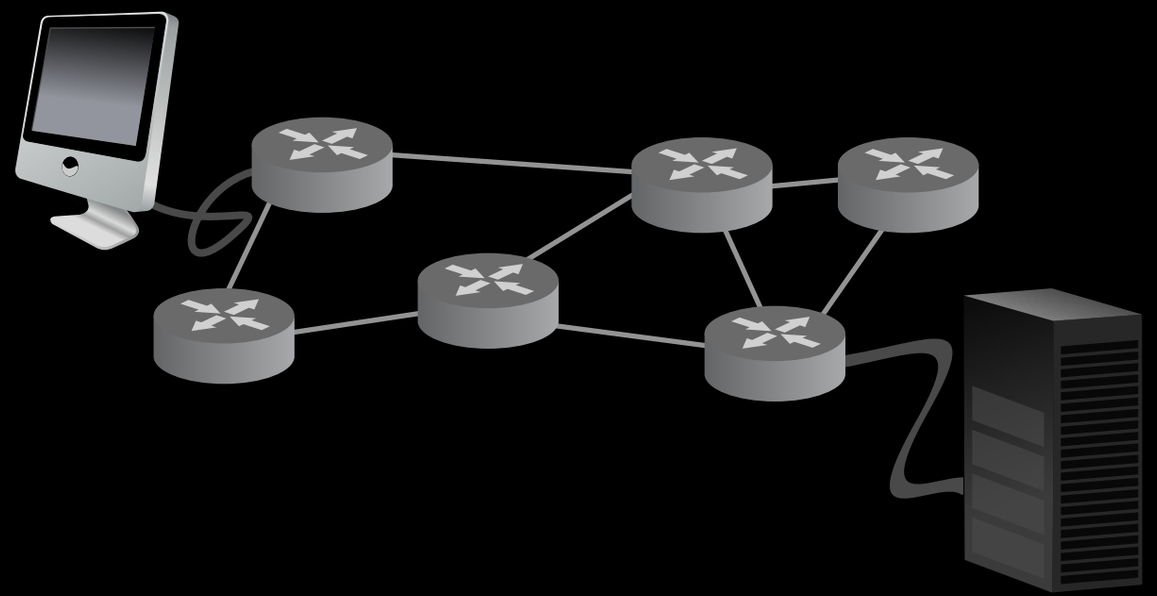
- Jim Thornton
- Diana Smetters
- Nick Briggs
- Michael Plass
- Rebecca Braynard
- Tim Diebert
- Elaine Shi
- Simon Barber
- Ignacio Solis
- Mark Mosko
- Philippe Golle
- and me

CCN offers ...

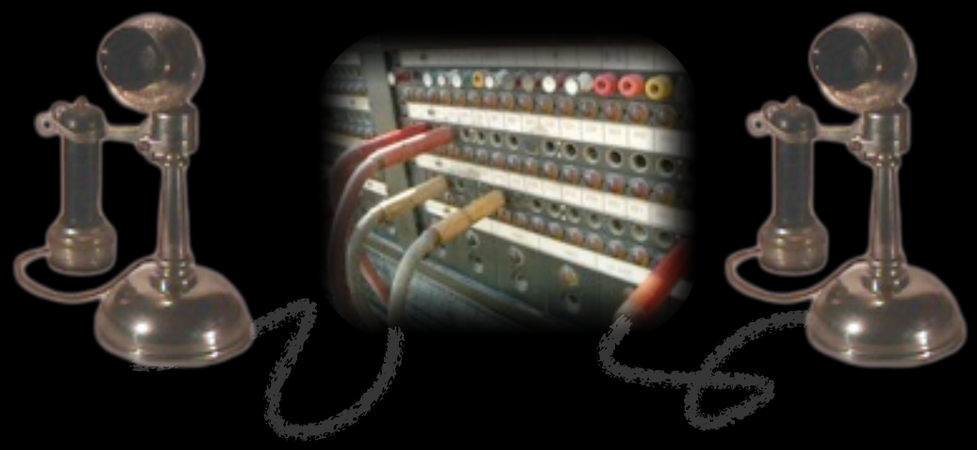
- (provably) optimal content distribution
- painless mobility, wireless, virtualization, ...
- same scalability & efficiency as TCP/IP
- simple, secure, robust configuration
- an easy, incremental, evolutionary path
- *much* better security

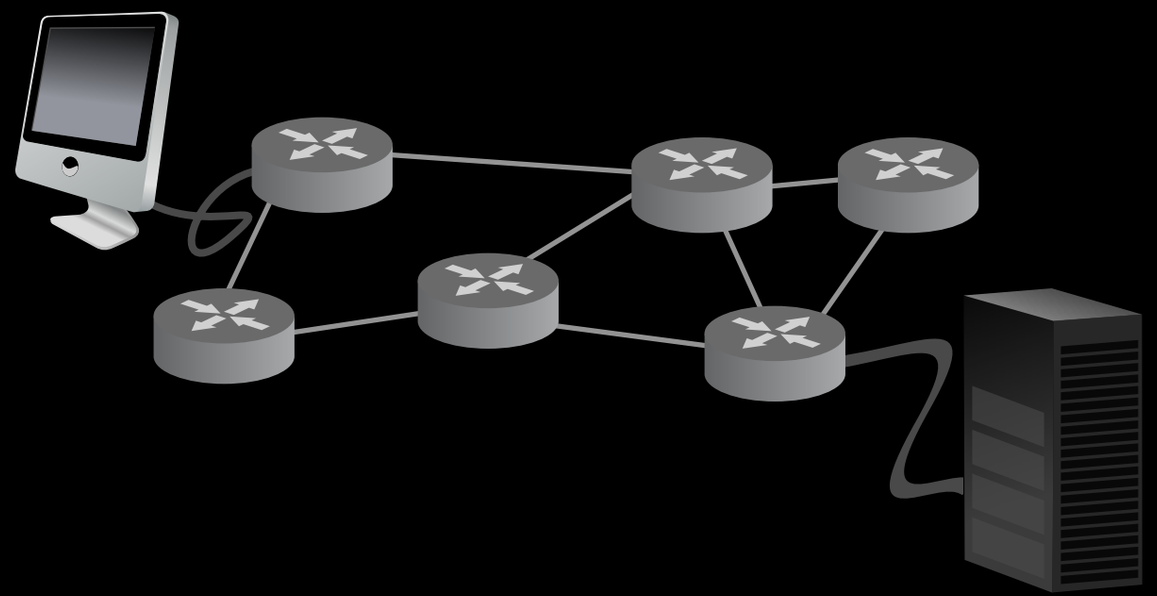
For 150 years ‘communication’ has meant a conversation over a wire connecting two devices.





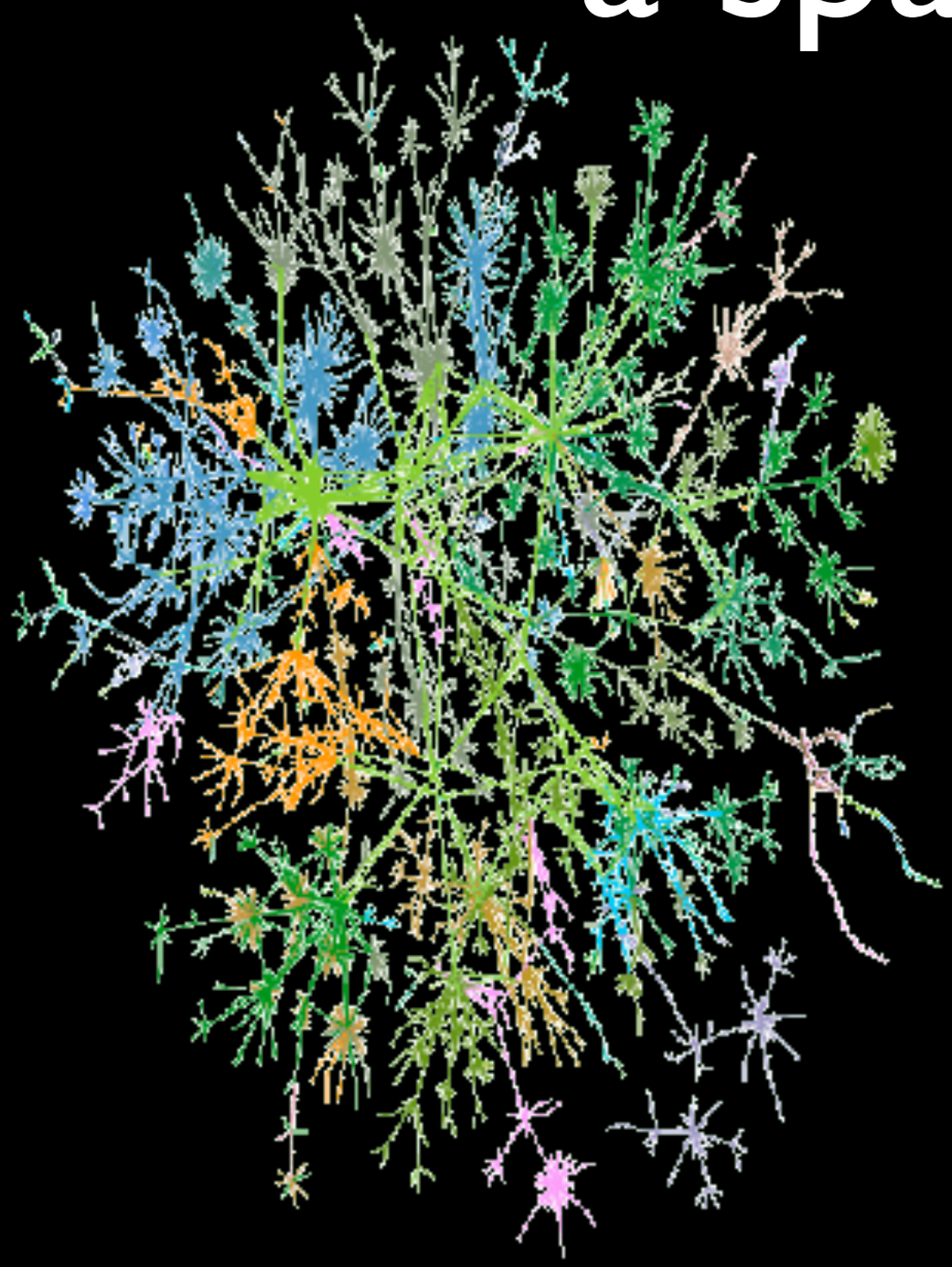
For 150 years ‘communication’ has meant a conversation over a wire connecting two devices.





For consumers, the Web forever changed that.

‘Circuit’ model requires a spanning tree



- Introduces global dependencies that remove local choice.
- Acts as a lens to magnify attacks.
- Makes load near source scale with popularity.

Today, wires and memories solve complimentary aspects of the *same problem*:

- ‘wires’ move information in space
- ‘memories’ move information in time

We need a communications architecture that unifies both aspects.

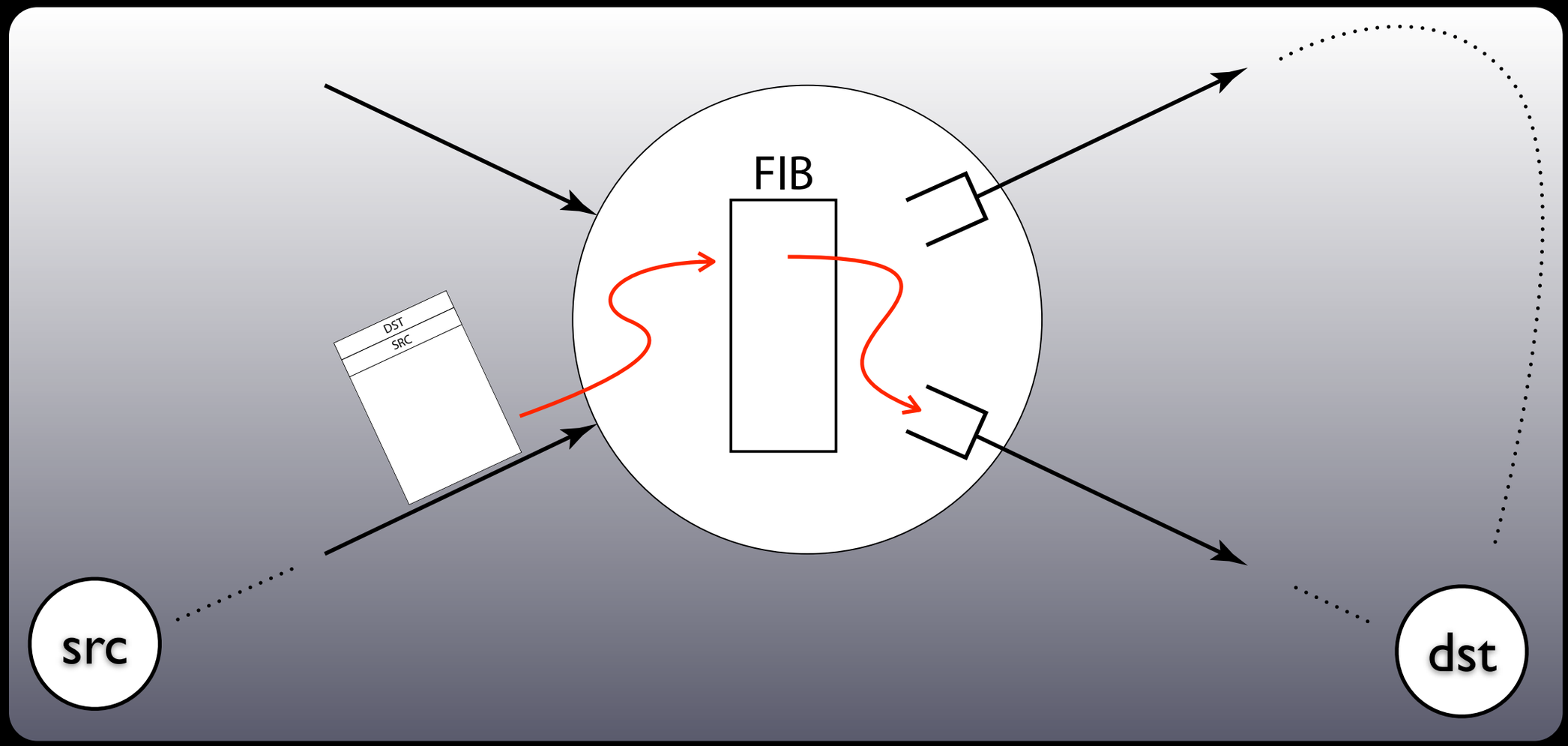
What has to change?

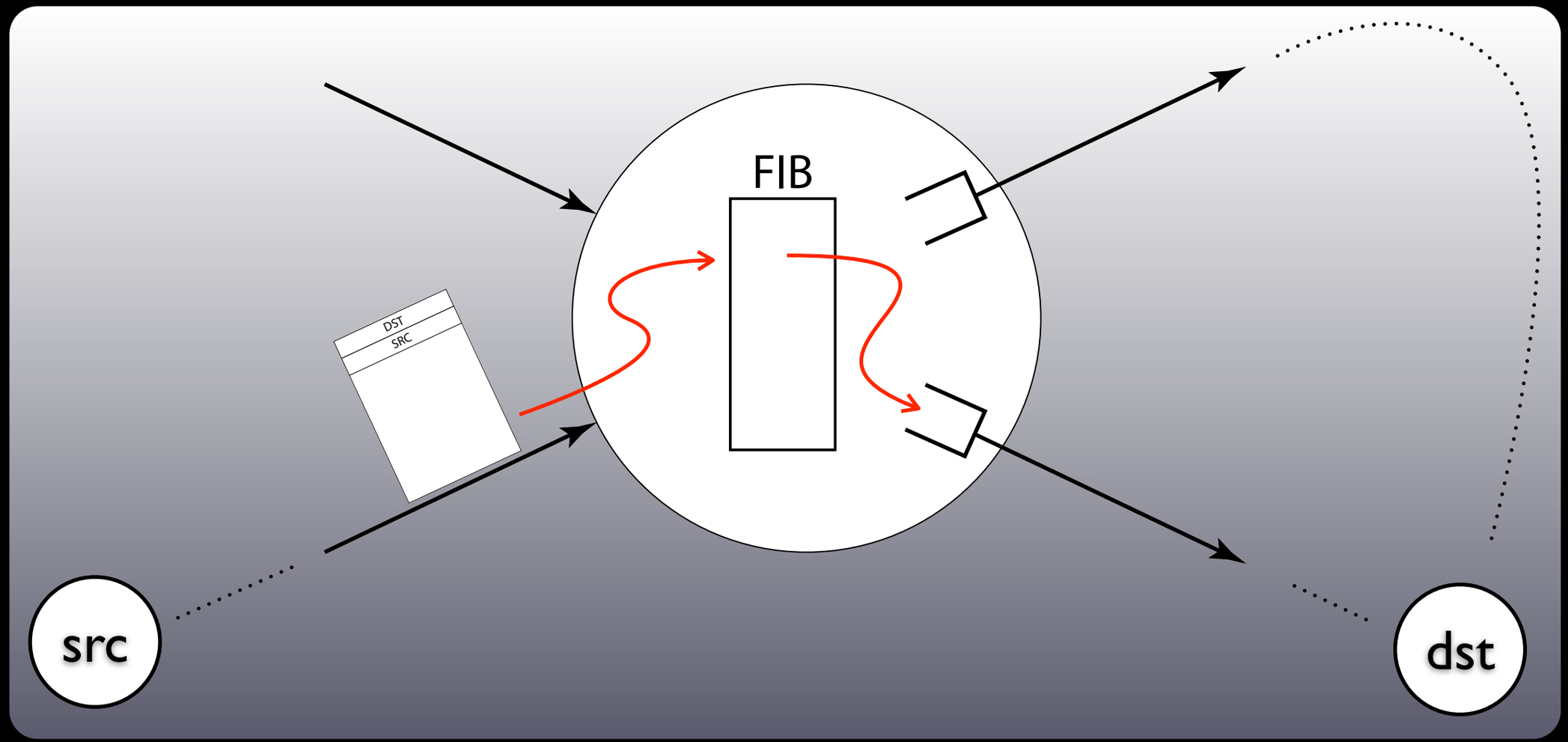
- For sharing, packets must describe what you want, not the process of getting it.
  An address has to name data, not conversation endpoints.

- For asynchrony (time-shifting, intermittent connectivity, mobility, ...) memory has to be explicit in the communication model.

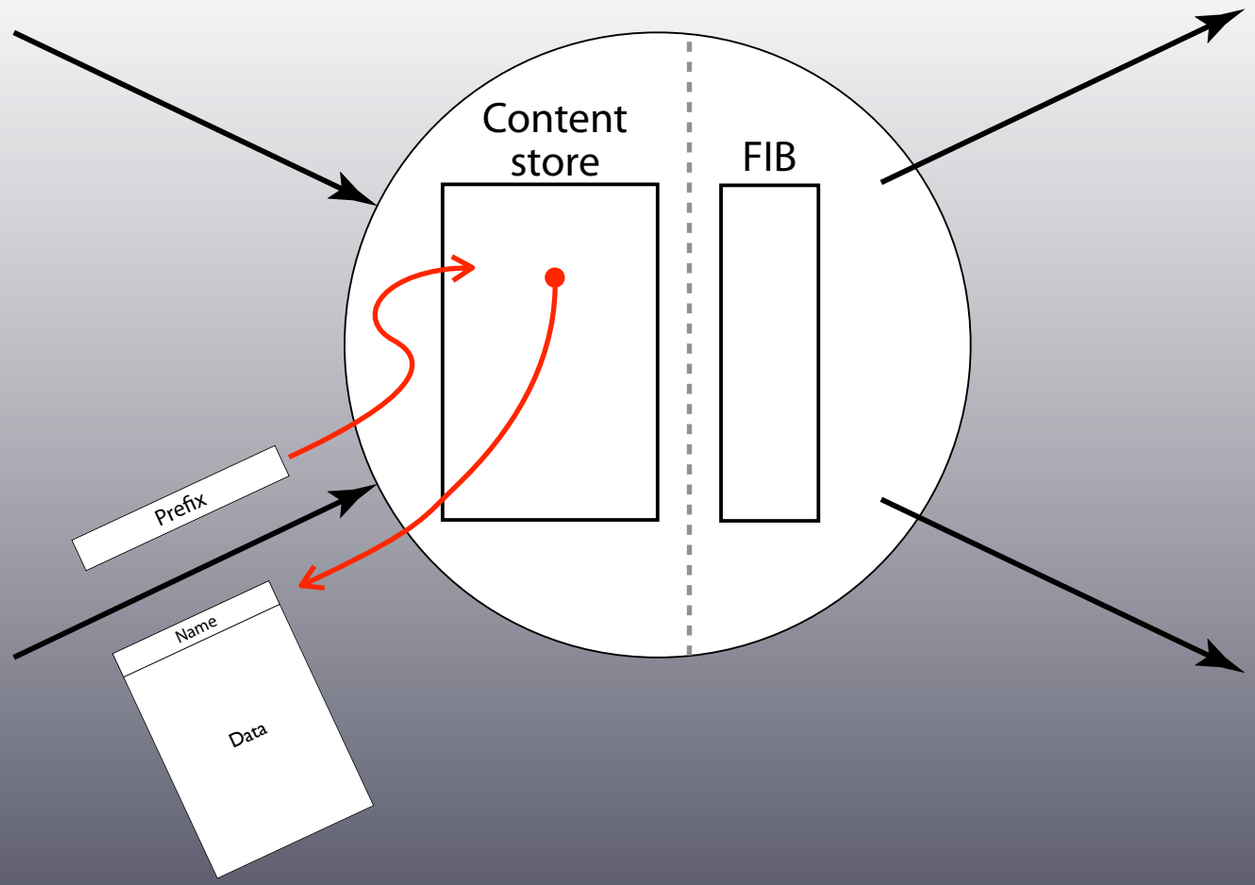
- Have to stop pretending container-based security can work and start securing data.

the IP model

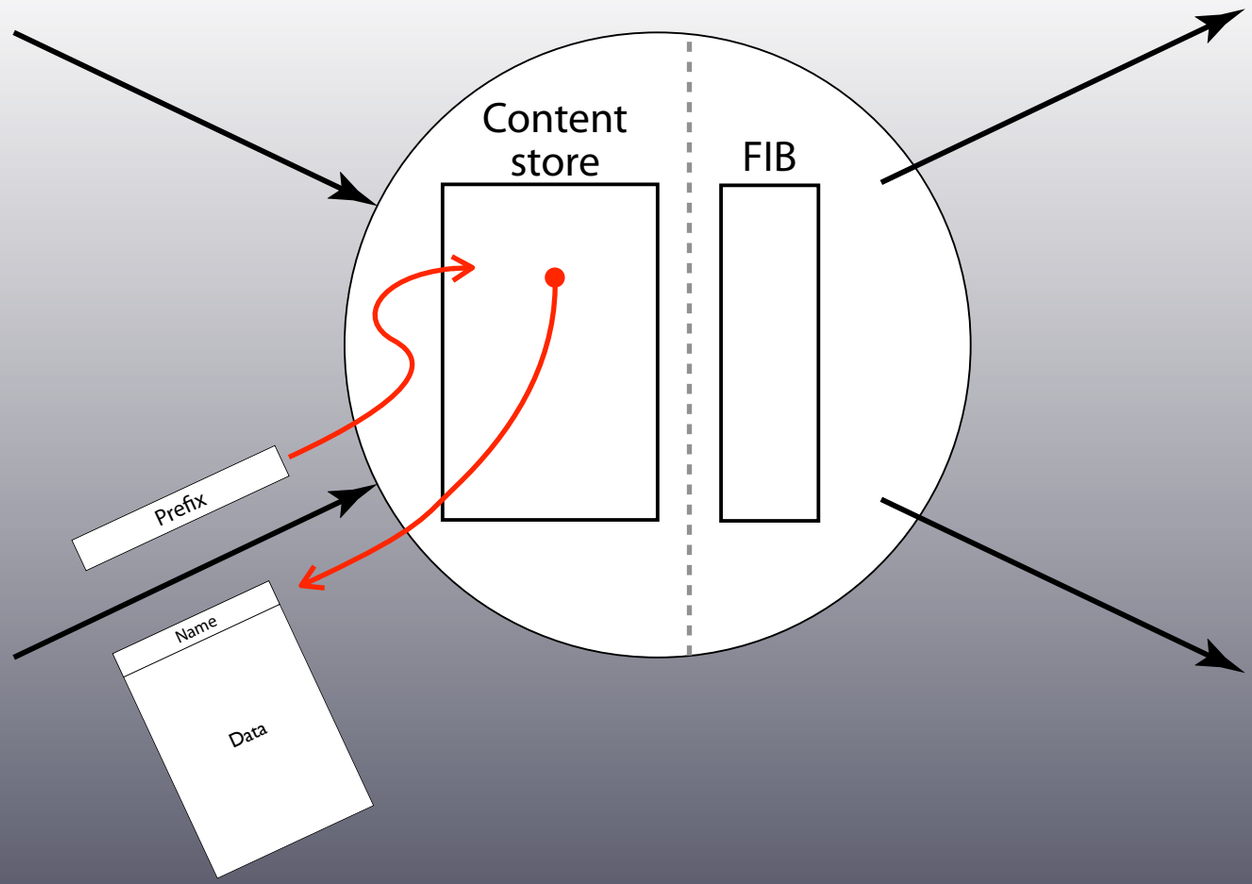




- Intermediate nodes are invisible
- Intermediate nodes can’t choose.
- Intermediate nodes can’t measure success

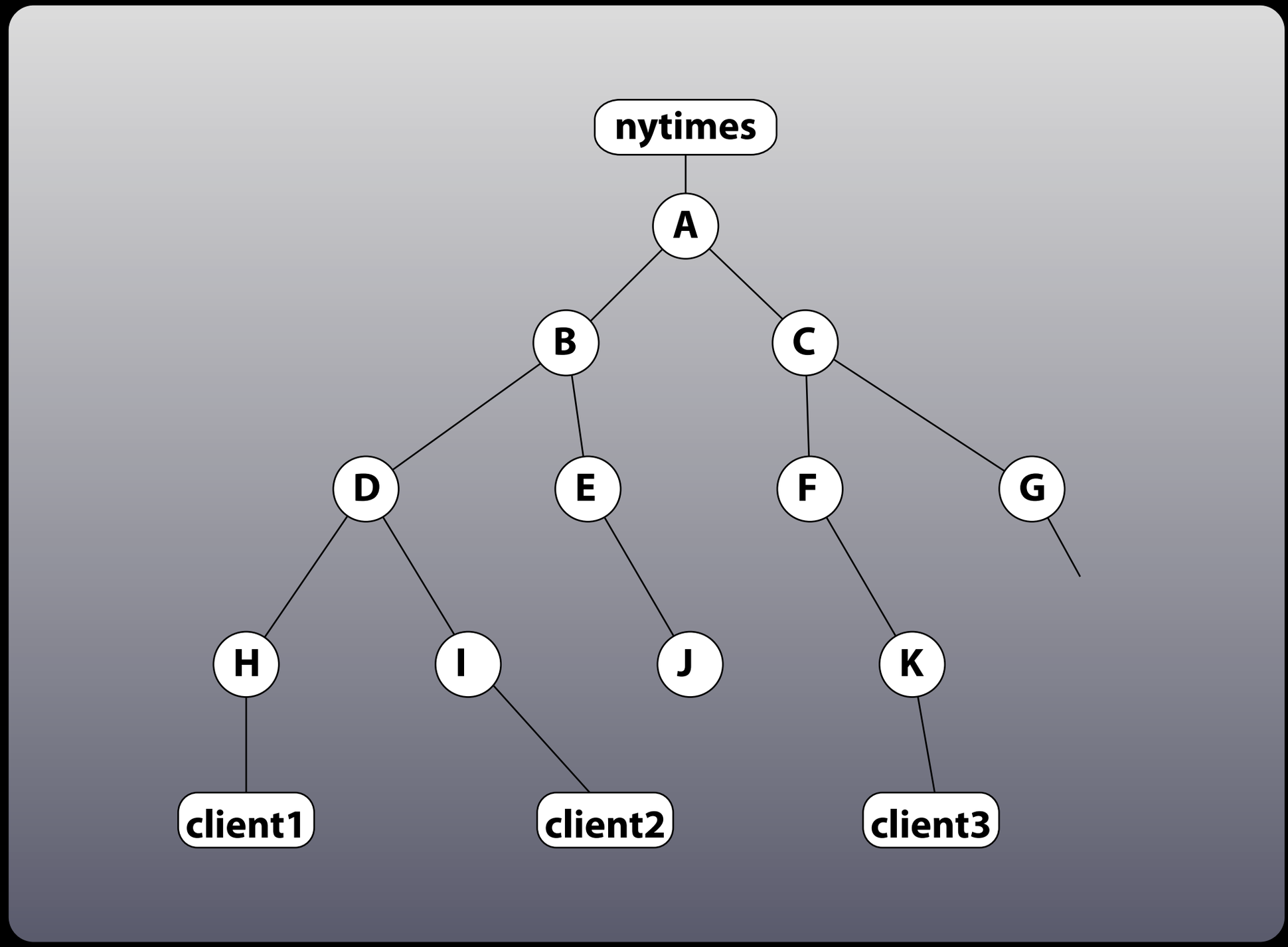


- Packets say ‘what’ not ‘who’ (no src or dst)
- communication is local
- entire net runs in flow balance
- memory damps large scale dynamics

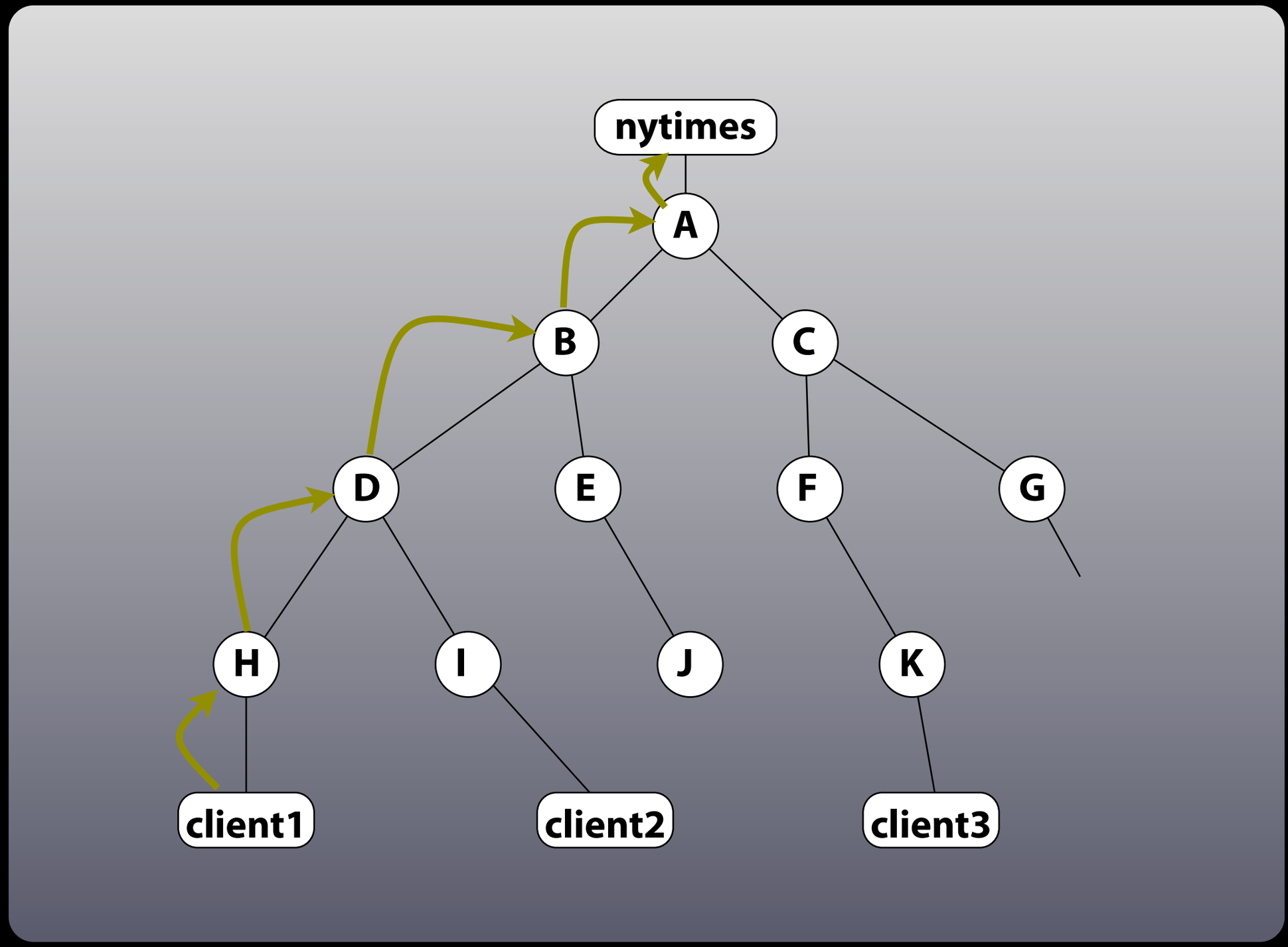


- topology is an optimization
- node gets fine-grained choice of upstream(s)
- upstream performance is measurable

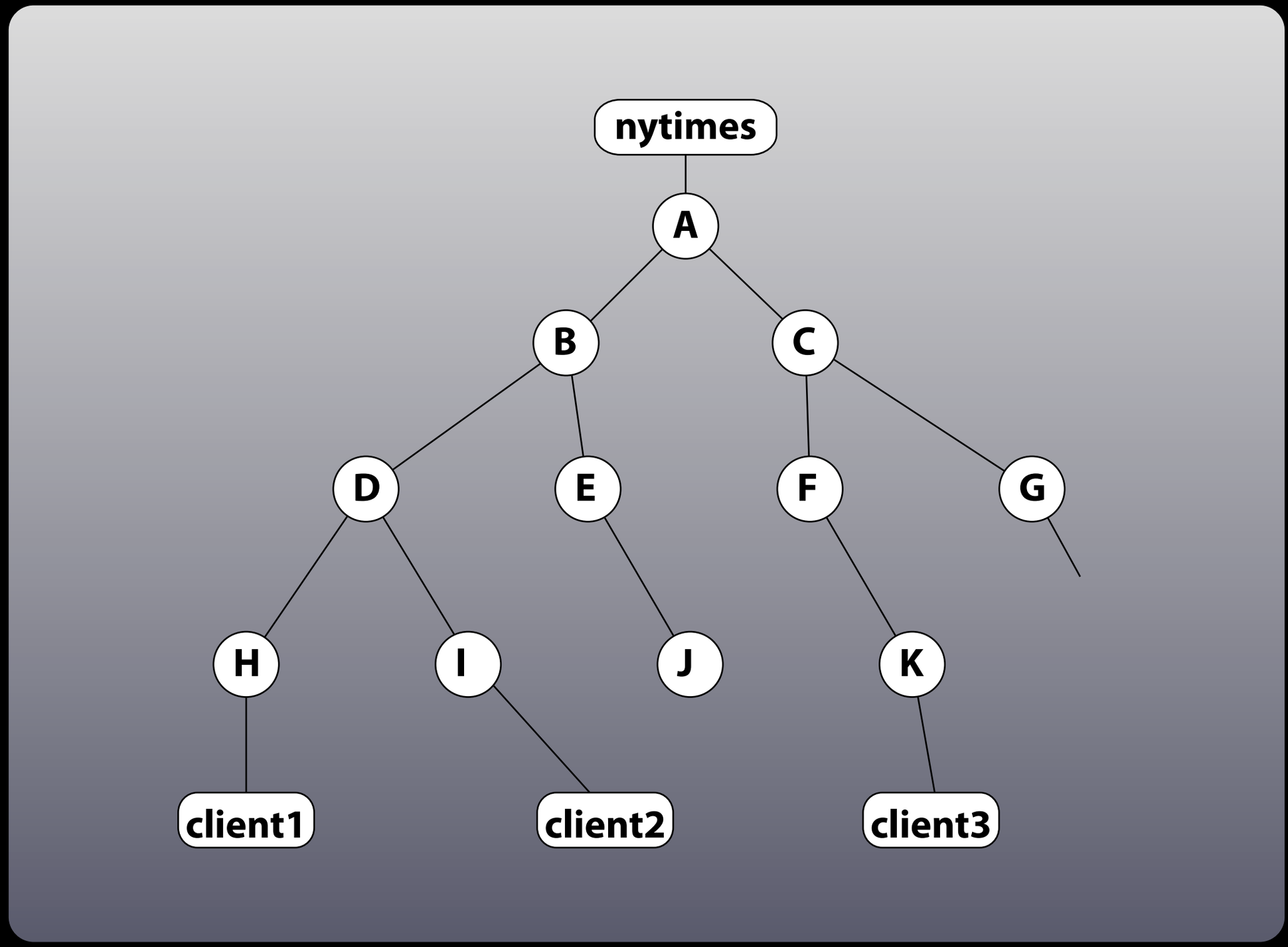
Example: Content Distribution



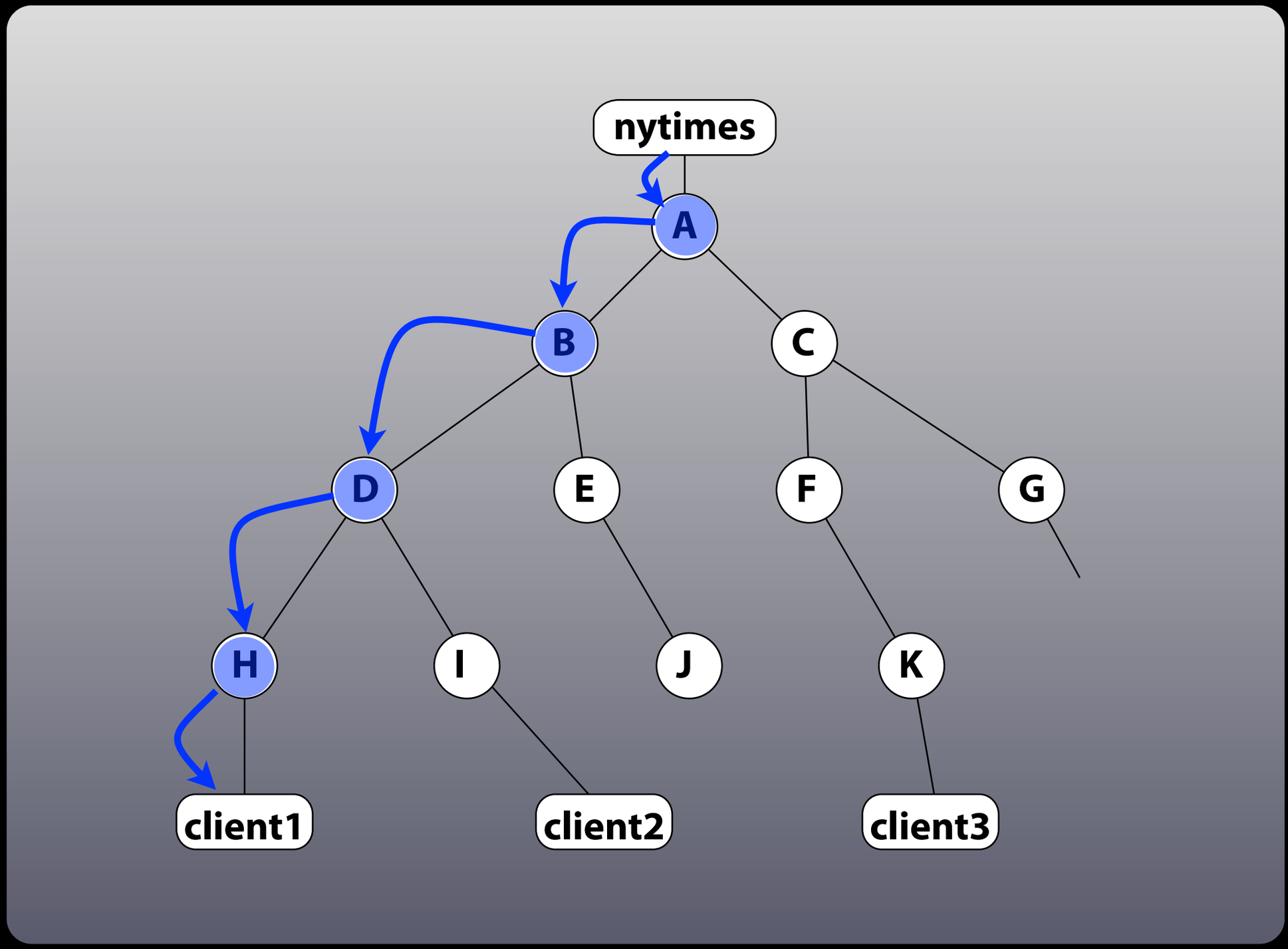
Example: Content Distribution



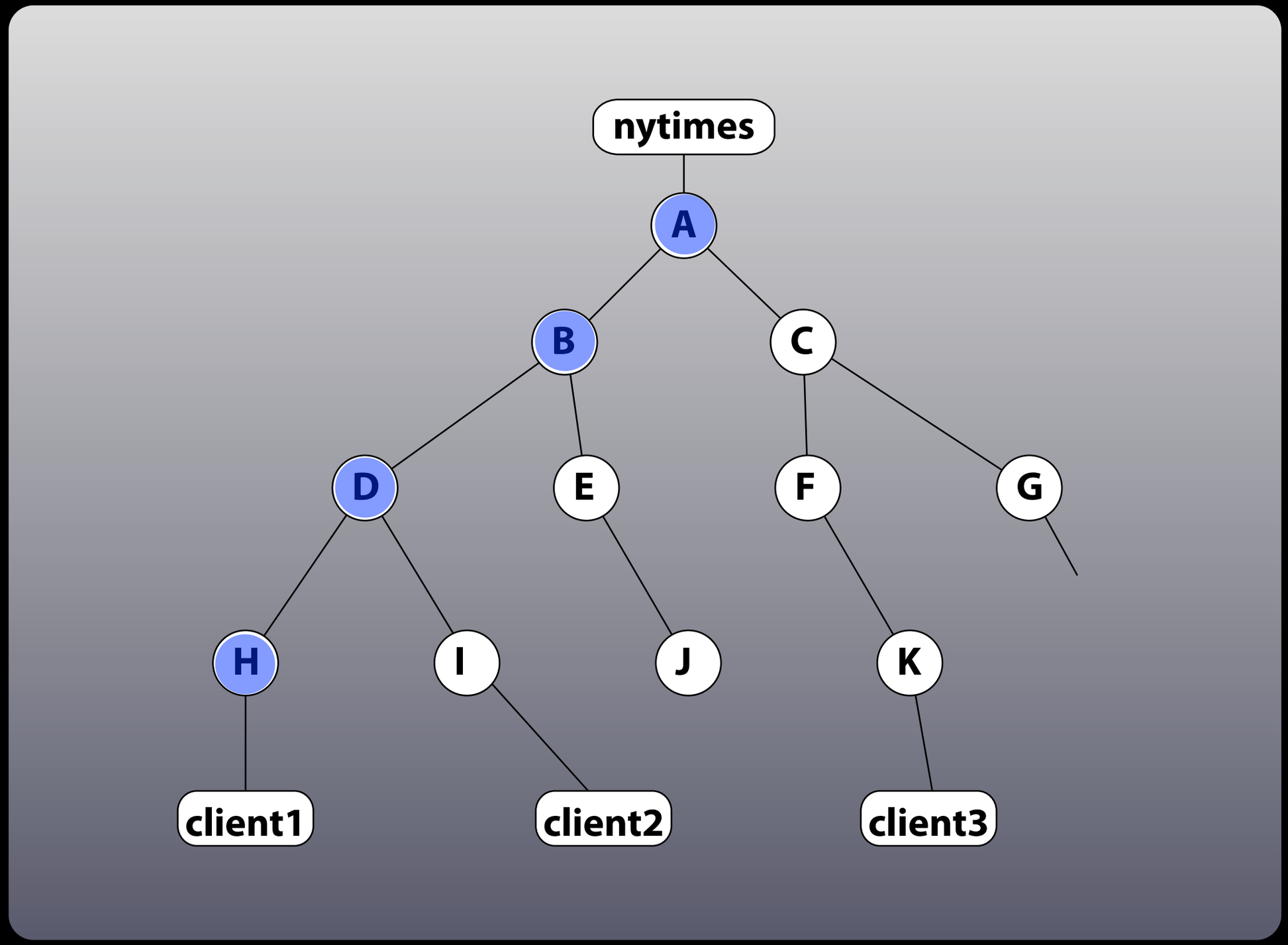
Example: Content Distribution



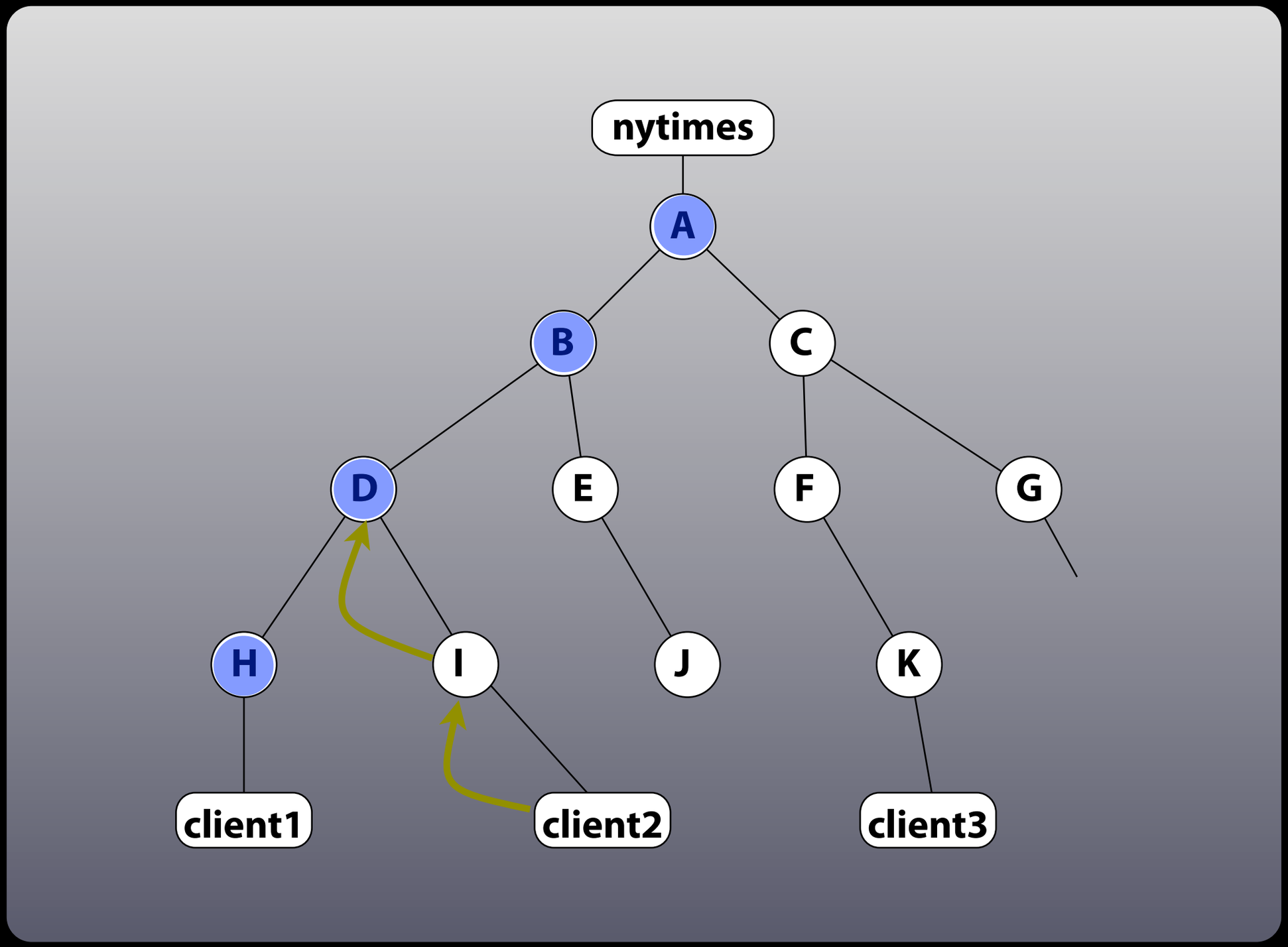
Example: Content Distribution



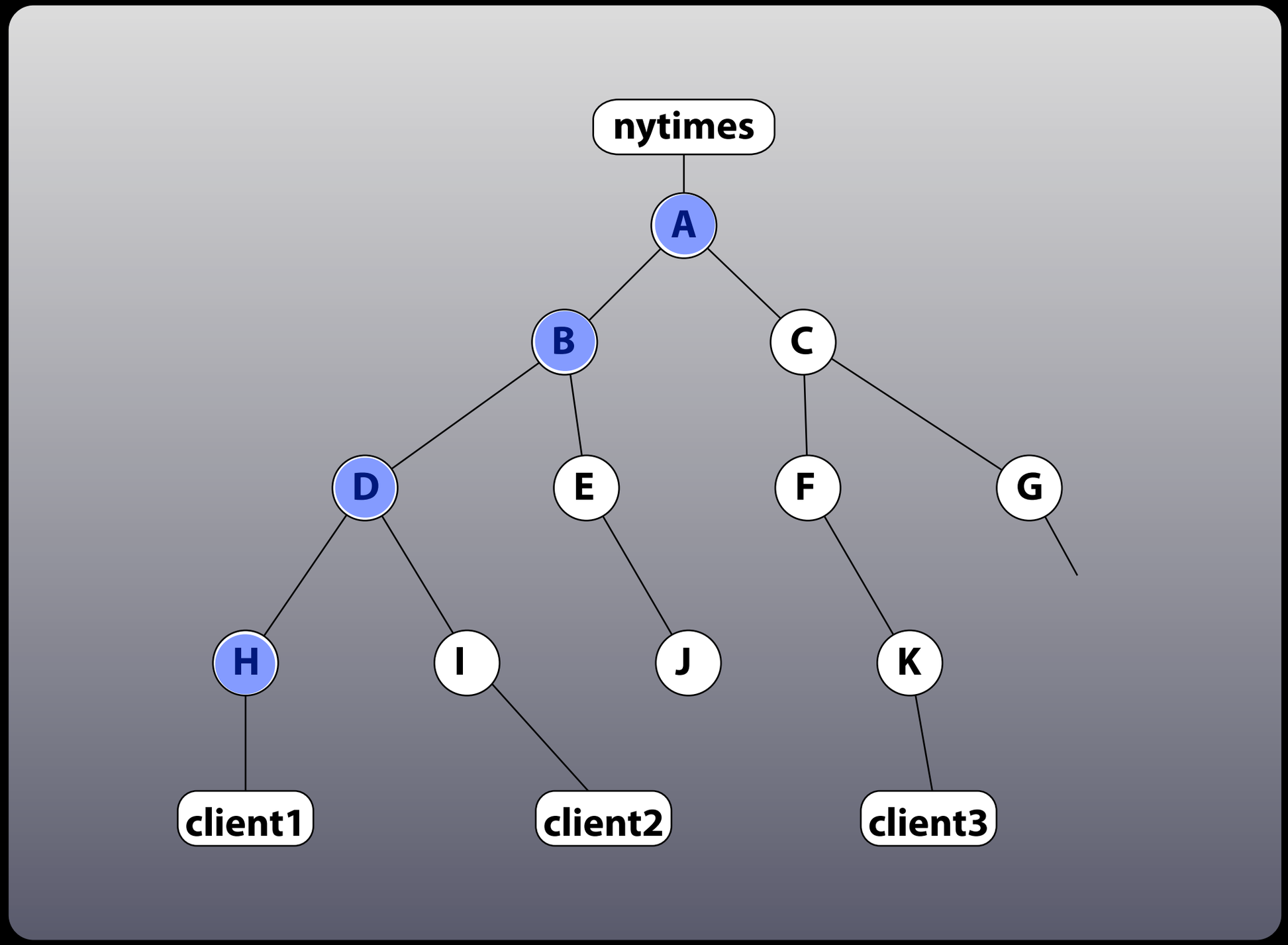
Example: Content Distribution



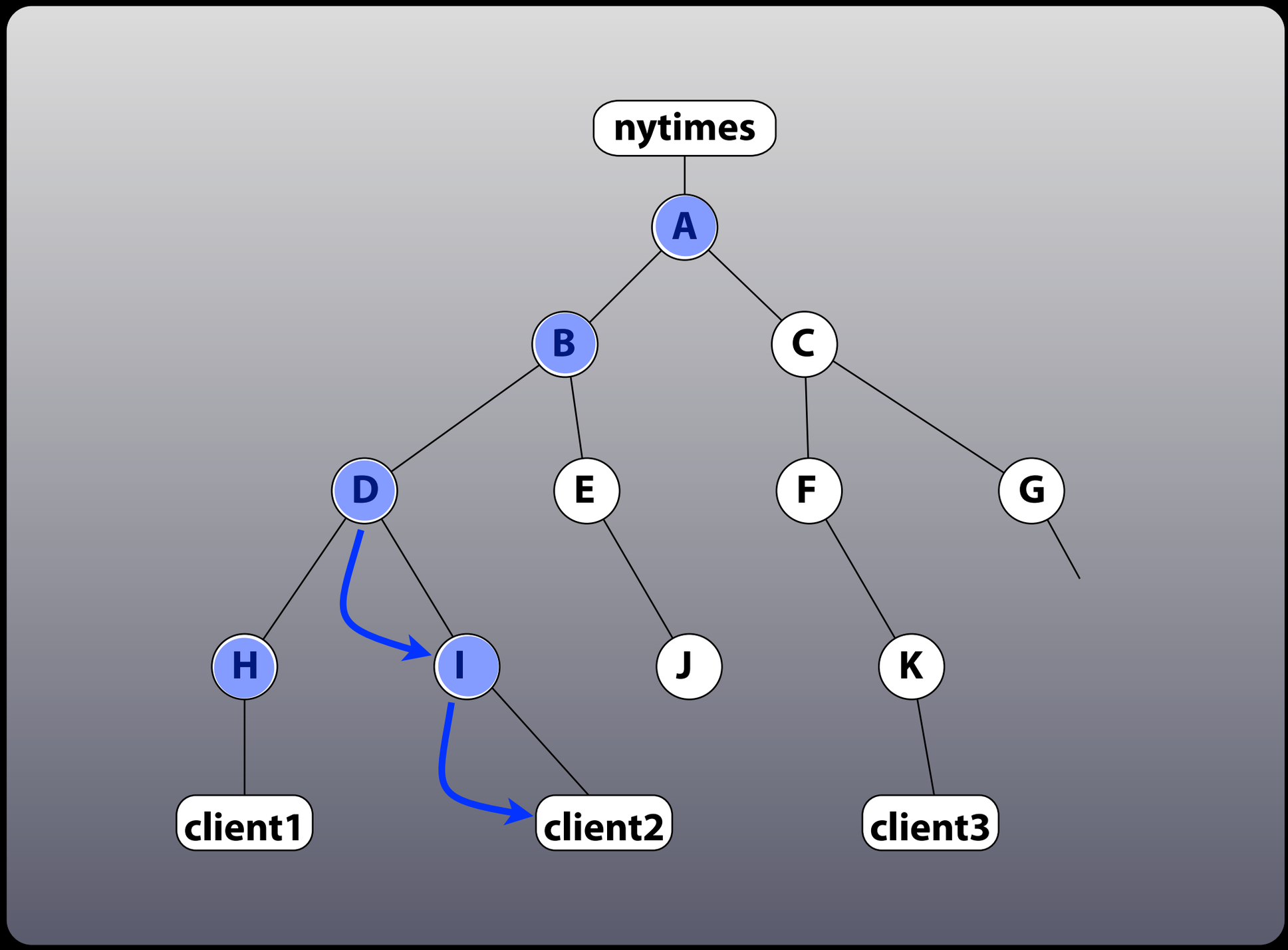
Example: Content Distribution

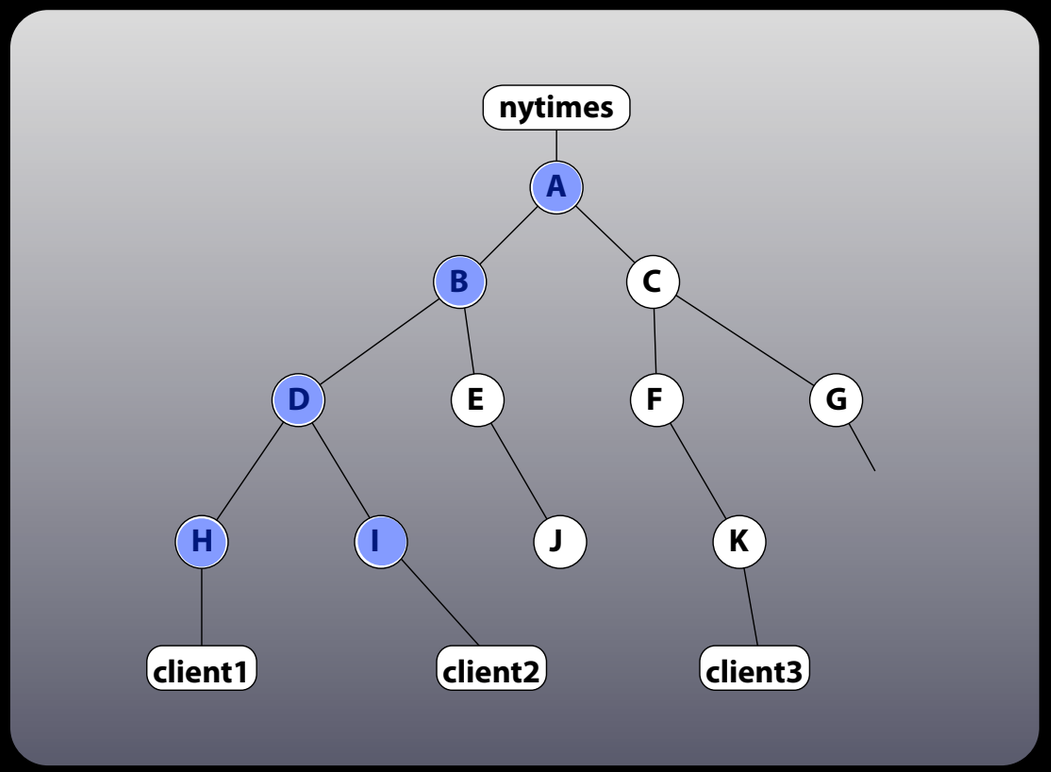


Example: Content Distribution



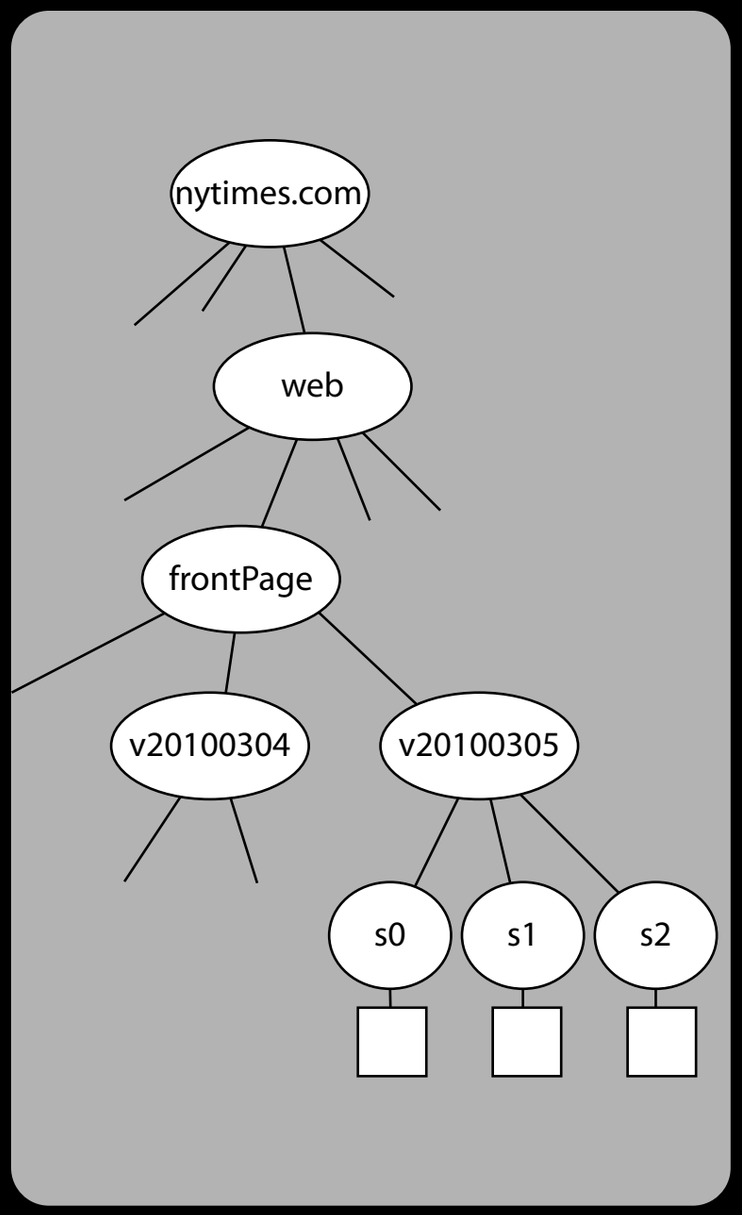
Example: Content Distribution





- Content goes only where there's interest.
- It takes at most one trip across any link.
- Average latency is minimized.
- Total bandwidth is minimized.
- There's no routing or control traffic associated with the replicas.

Name tree solves ‘discovery problem’



Newest nytimes:

nytimes.com/web/frontPage
<rightmost child>

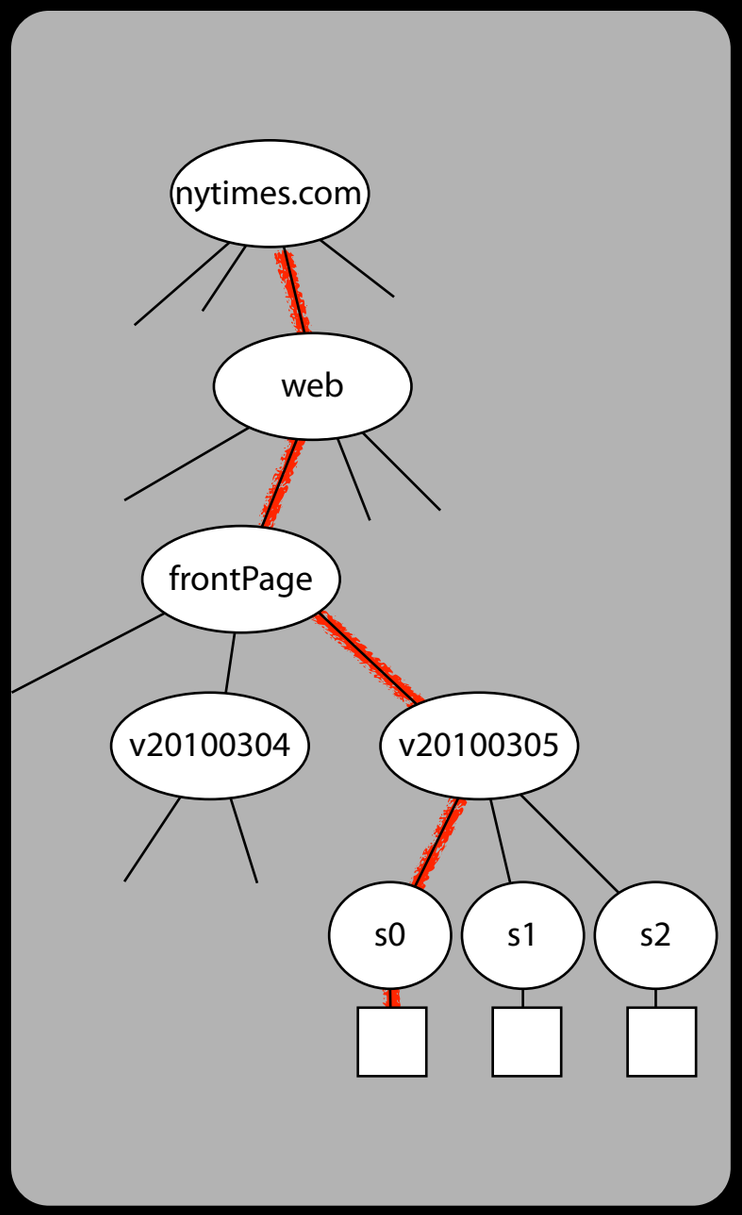
Newest that’s more recent than mine:

nytimes.com/web/frontPage/v20100301
<rightmost sibling>

Conventions:

- name tree child nodes are lexically ordered
- <next> assumed if no relationship specified

Name tree solves ‘discovery problem’



Newest nytimes:

nytimes.com/web/frontPage
<rightmost child>

Newest that’s more recent than mine:

nytimes.com/web/frontPage/v20100301
<rightmost sibling>

Conventions:

- name tree child nodes are lexically ordered
- <next> assumed if no relationship specified

Internet security sucks

This is *our problem*:

Communication ≡ Information + Trust

Files, hosts and network connections are *containers* for information

- A *secured perimeter* is the *only* way to secure containers.

- For today's network use, any realistic perimeter encloses the planet.

Forget containers – secure the content

Do it as the final production step to minimize attack surface.

Ron Rivest's SDSI has shown this can be done if any consumer can assess *solely from the data:*

- Integrity (is data intact and complete?)
- Relevance (what question does this answer?)
- Provenance (who asserts this is an answer?)

CCN data

/nytimes.com/web/frontPage/v20100305/s0/0x3fdc96a4...

signature

0x1b048347

key

nytimes.com/web/george/desktop public key

Signed by nytimes.com/web/george

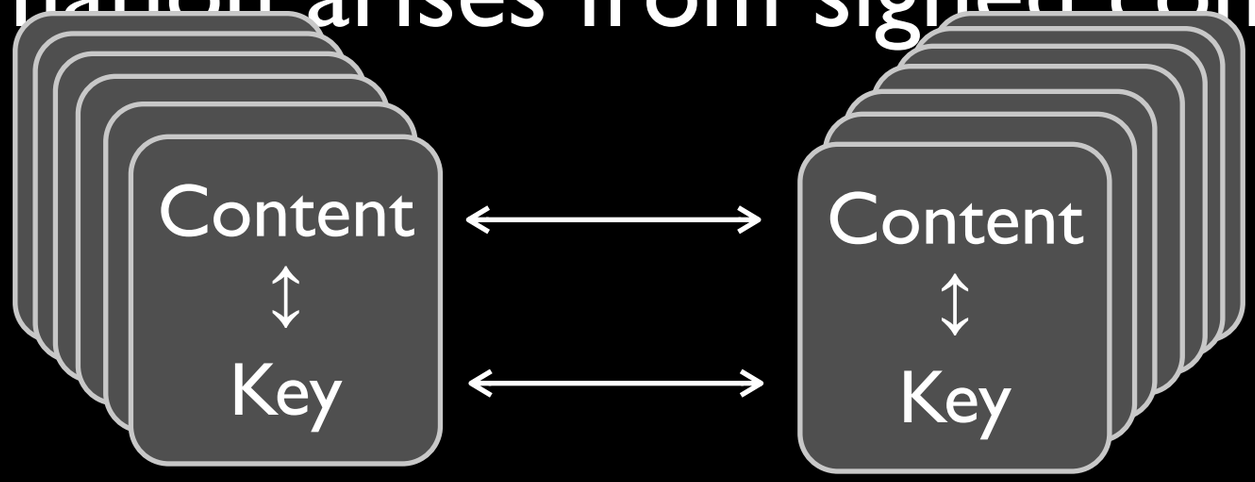
Signed by nytimes.com/web

Signed by nytimes.com

Note: Content networking has no key distribution problem since keys are content.

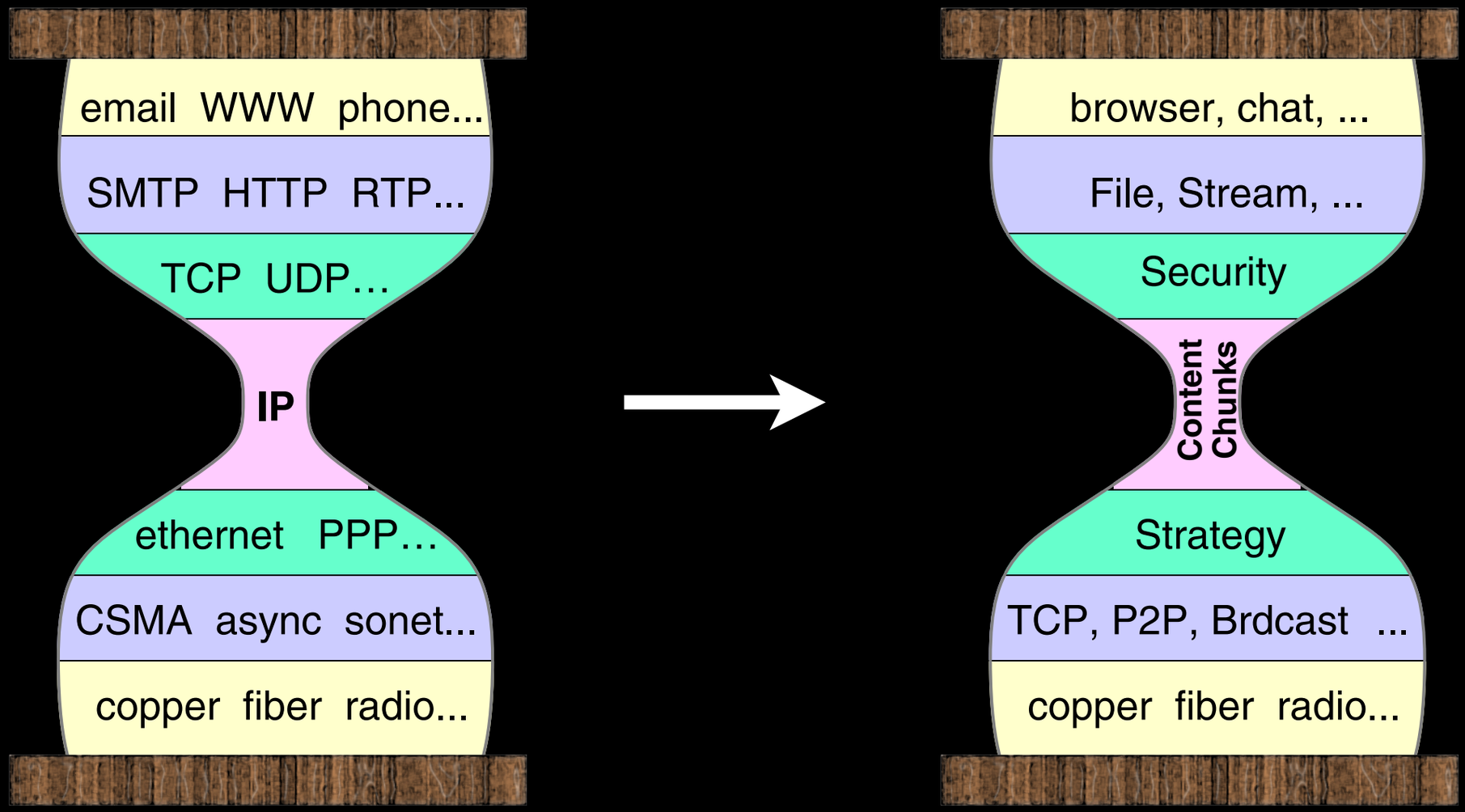
Evidentiary Trust

- Rich web of trustable, interconnected information arises from signed content:



- Attacks have to be consistent with information and links – get exponentially harder as information base grows.

New Layering



Information on CCN is available at

www.ccnx.org

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

Naming

Names and meaning

- Like IP, CCN 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.

Scaling

Names Route Interests

- FIB lookups are longest match (like 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.

- Since nothing can loop, state can be approximate (e.g., bloom filters).

"But ...

"this doesn't handle conversations or realtime.

Yes it does - see ReArch VoCCN paper.

"this is just Google.

This is IP-for-content. We don't search for data, we route to it.

"this will never scale.

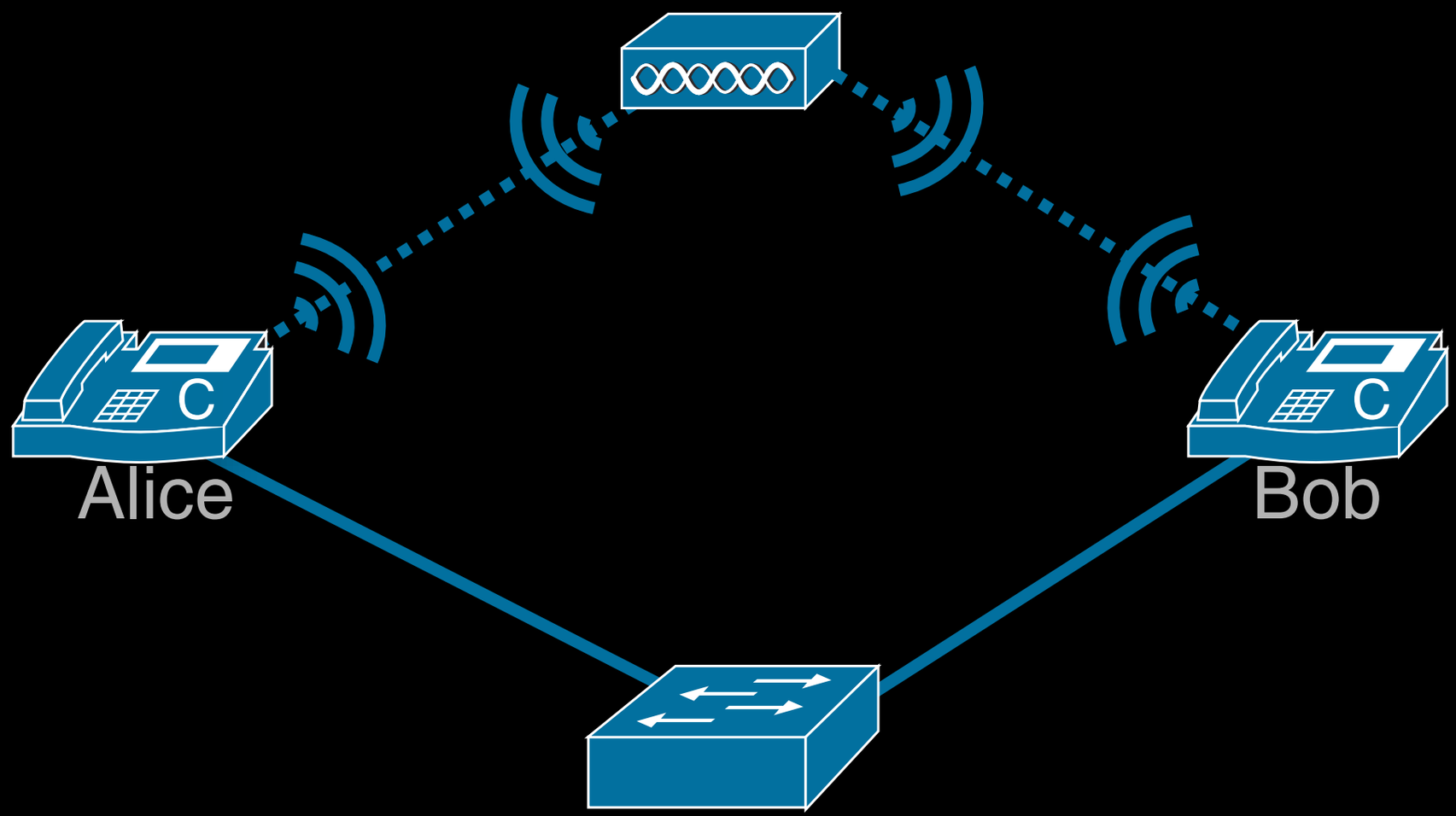
Hierarchically structured names give same log(n) scaling as IP but CCN tables can be *much* smaller since multi-source model allows inexact state (e.g., Bloom filter).

Strategy

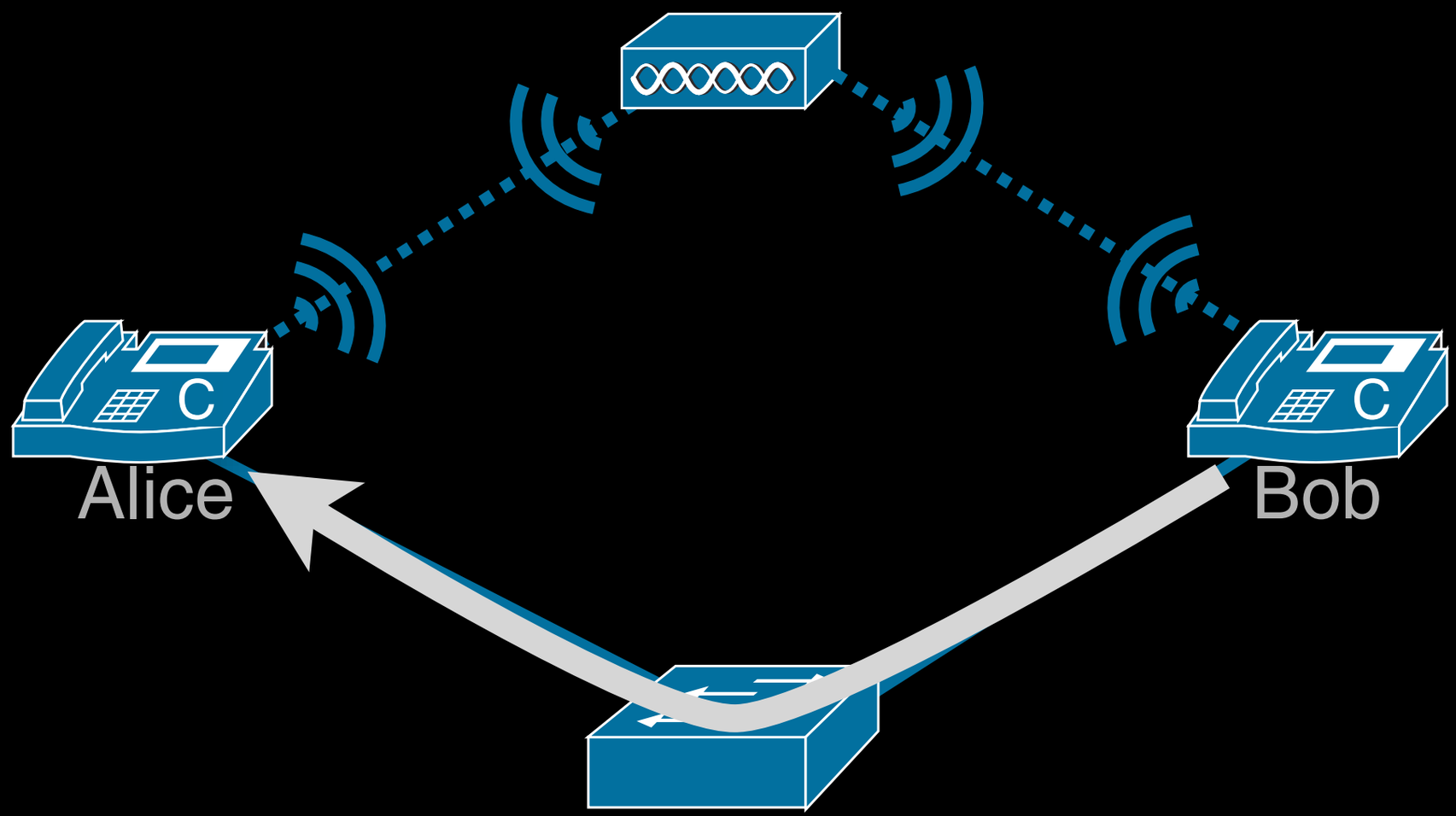
Strategy layer (mobility management)

- If you don't care who you're talking to, you don't care if they change.
- If you ask for a few pieces at a time, it's ok if one gets delivered where you used to be.
- If you can use all your links simultaneously it's easy for the stack learn what's best.
- If all communication is flow balanced, you know exactly what's working and how well.

Performance-based interest re-expression

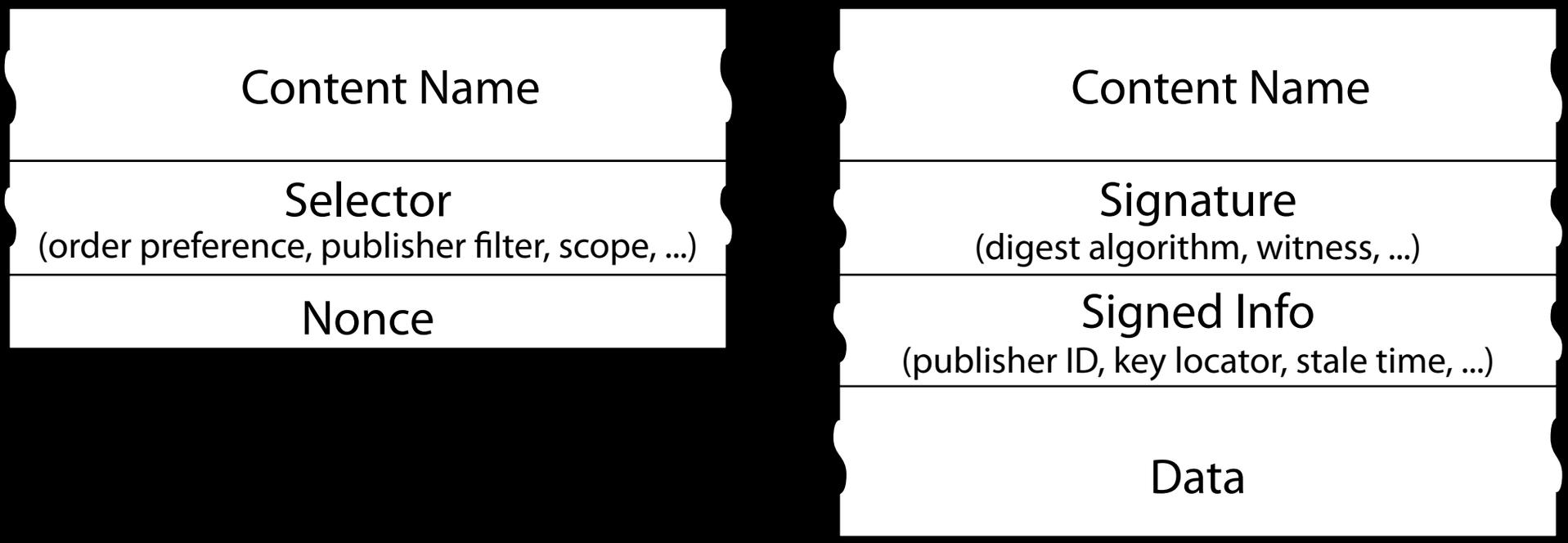


Performance-based interest re-expression



Basics

CCN packets

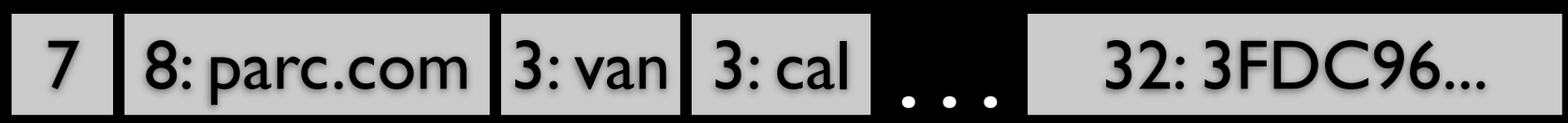


There are two CCN packet types: *interest* (a question) and *data* (an answer). Both are encoded in an efficient binary XML.

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:



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/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 it:

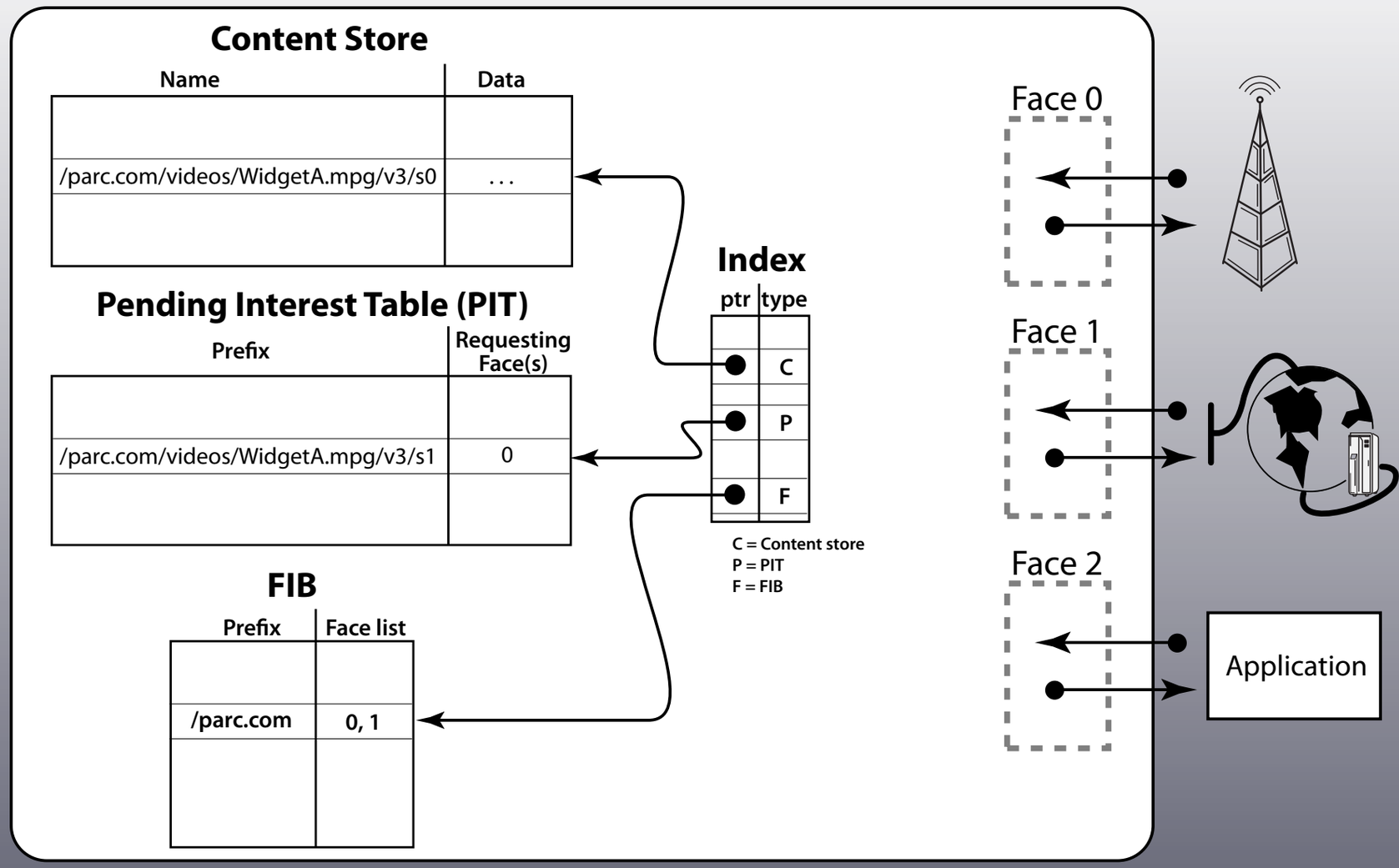
```
HereIs '/parc.com/van/presentation.pdf/p1' <data>
```

Basic CCN transport

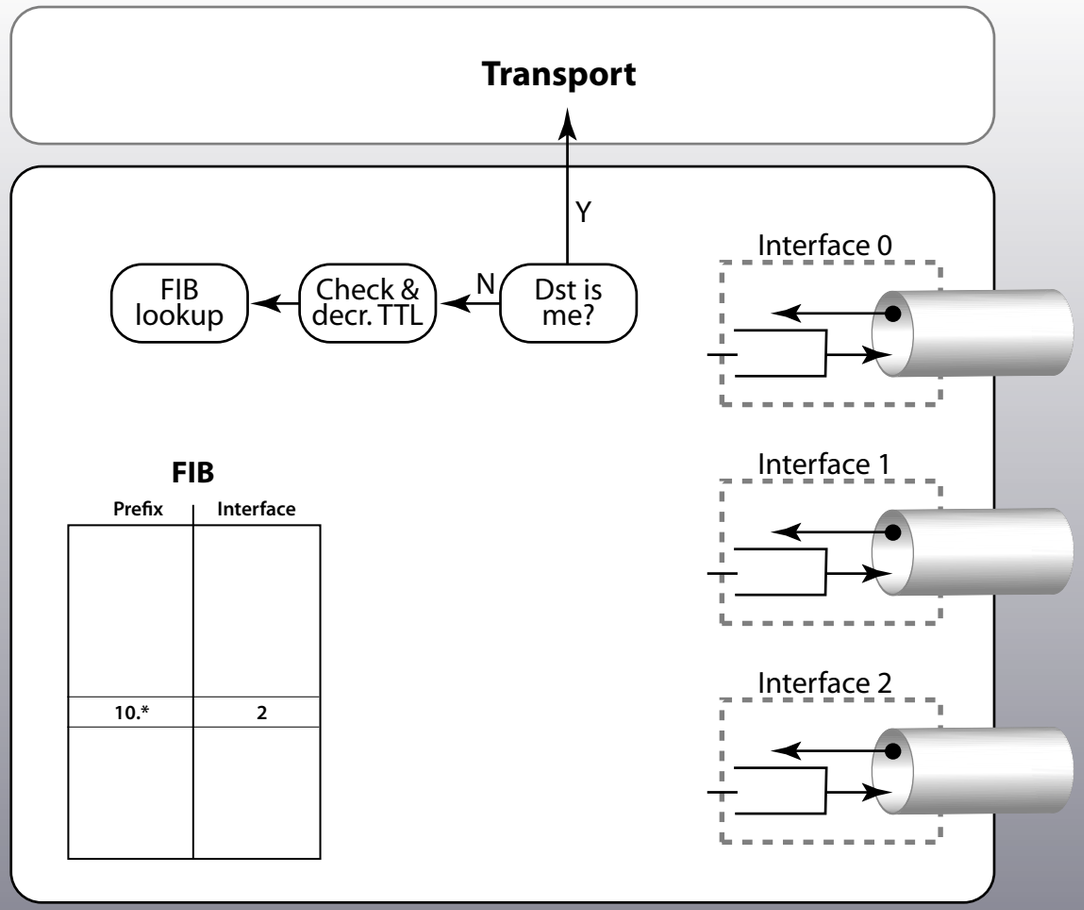
- Data that matches an interest 'consumes' it. (network always operates in flow balance.)
- Interest must be re-expressed to get new data. (Controlling the re-expression allows for traffic management and environmental adaptation.)
- Multiple (distinct) interests in same collection may be expessed (similar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

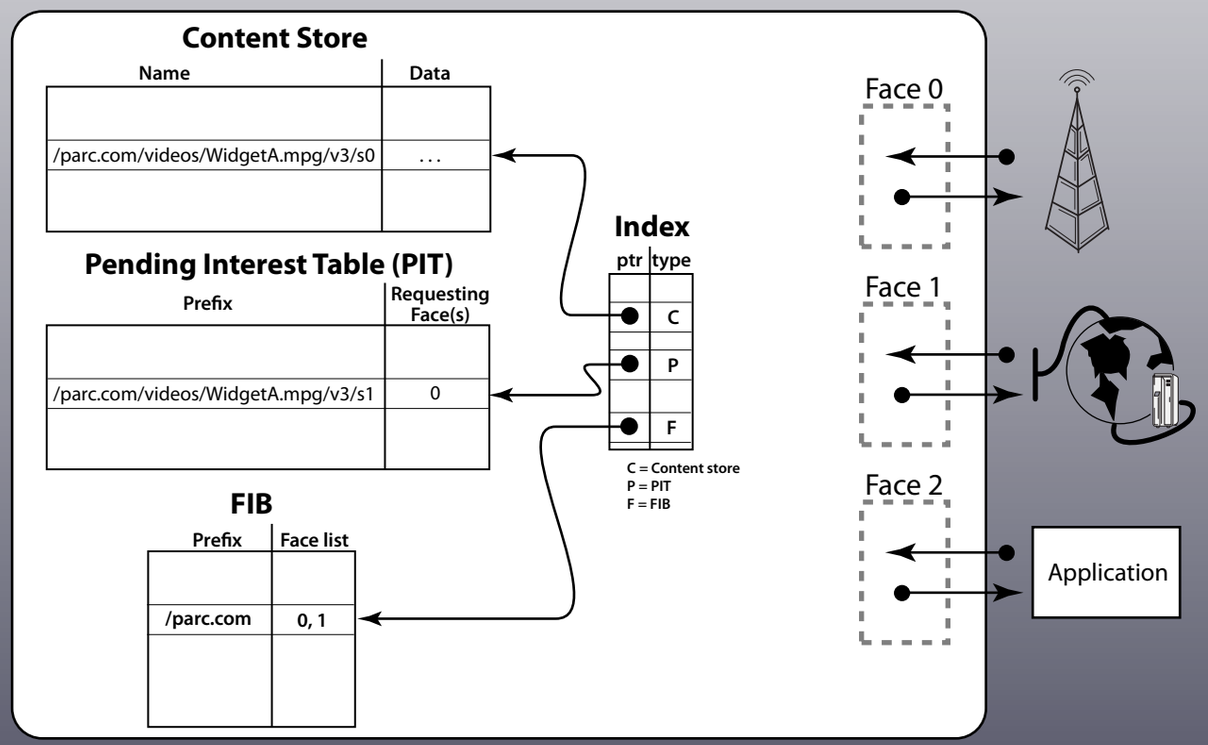
Node Model

CCN node model

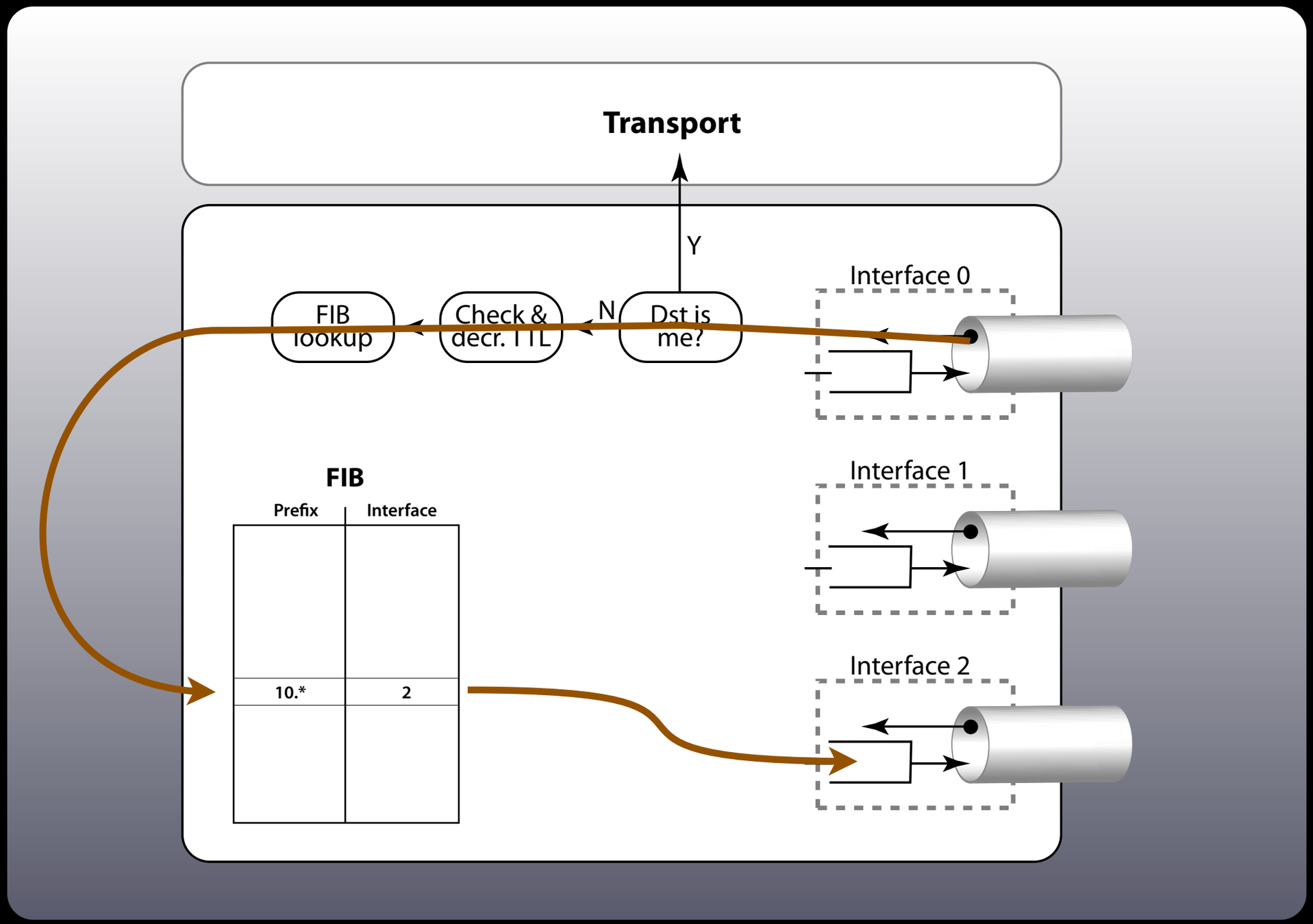


Comparison


Transport
Y
FIB lookup
Check & decr. TTL
N
Dst is me?
Interface 0
Interface 1
Interface 2
FIB
Prefix
Interface
10.*
2


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

IP node model



CCN node model