

Faces vs. Interfaces

- No Queues (data comes from Content Store)
- No ARP or ES-IS (no layer-3 abstraction to bind to layer-2 address)
- Have to do Suppression (see SRM or NORM) on multi-user broadcast links but can amortise to low cost by content-based prioritisation plus adaptive affinity.

QoS

- In the current Internet, problems that require QoS are highly localized.
- Roughly half the problem is caused by the serial dependencies created by queues.
- The other half is caused the lack of receiver based control of bottleneck links.
- Unlike IP, CCN is local, doesn't have queues and receivers have complete, fine-grained control.
- But it does aggregate traffic and has face-specific controls on the aggregation.

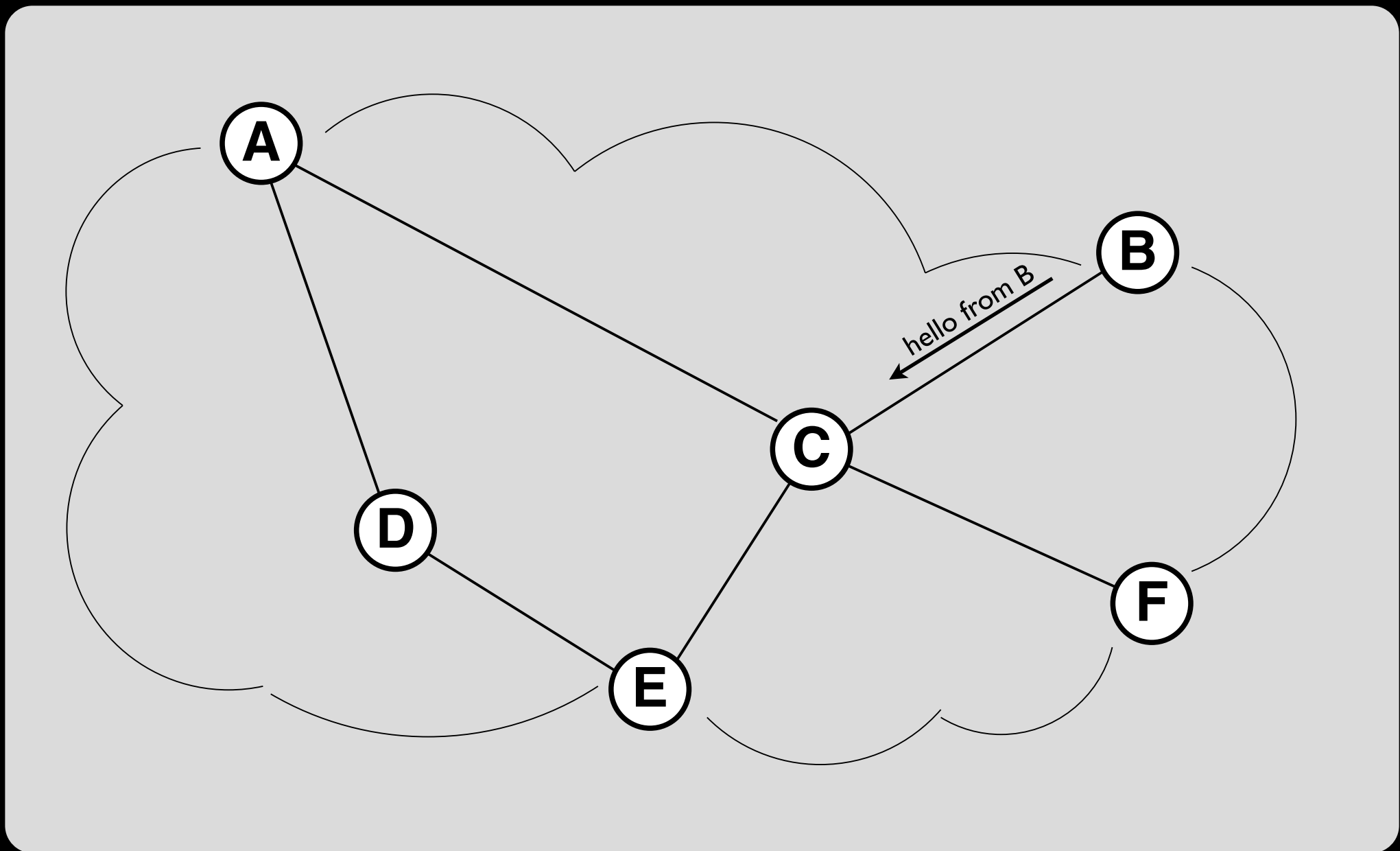
- History and motivation
- Content Model
- Security Model
- Node Model
- Routing
- Transport

- There are many (emerging) ways to do routing, e.g., Small Worlds, Geographic Hyperbolic, Pseudo-potential Gradient, Epidemic percolation.
- In general they're easier to implement and work better for CCN than for IP:
 - no looping data \Rightarrow no convergence issues.
 - multi-destination \Rightarrow state can be approximate (false positives ok).
 - CCN transport model matches routing's and adds security.

- I'm just going to talk about embedding CCN in existing Internet routing.
 - This is an easy evolutionary path (it allows for immediate, incremental deployment).
 - It offers some intuitions on scaling (same scaling as IP routing).
 - The basics are the same for any routing scheme.

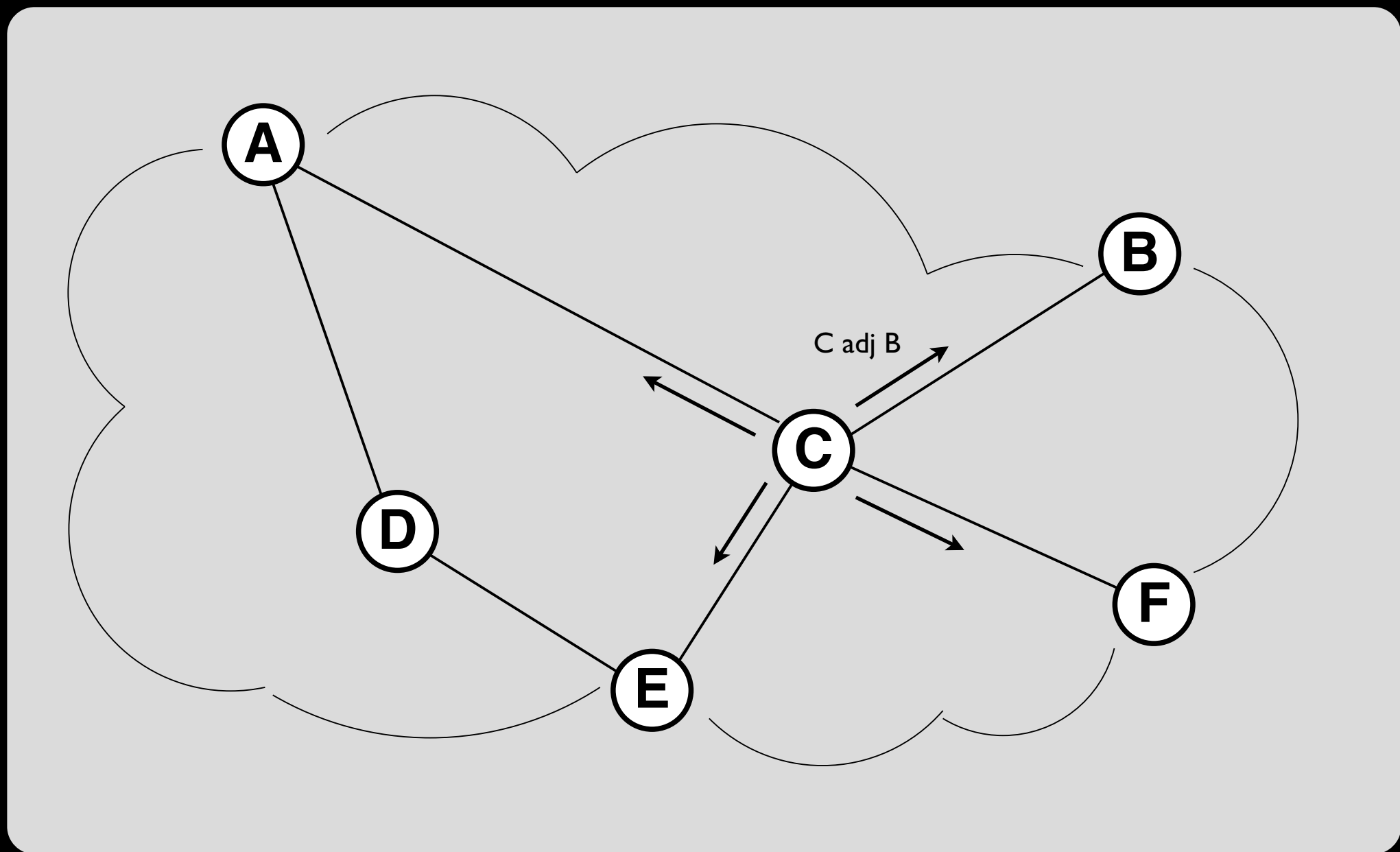
IGP routing

B sends a 'hello' out all its links



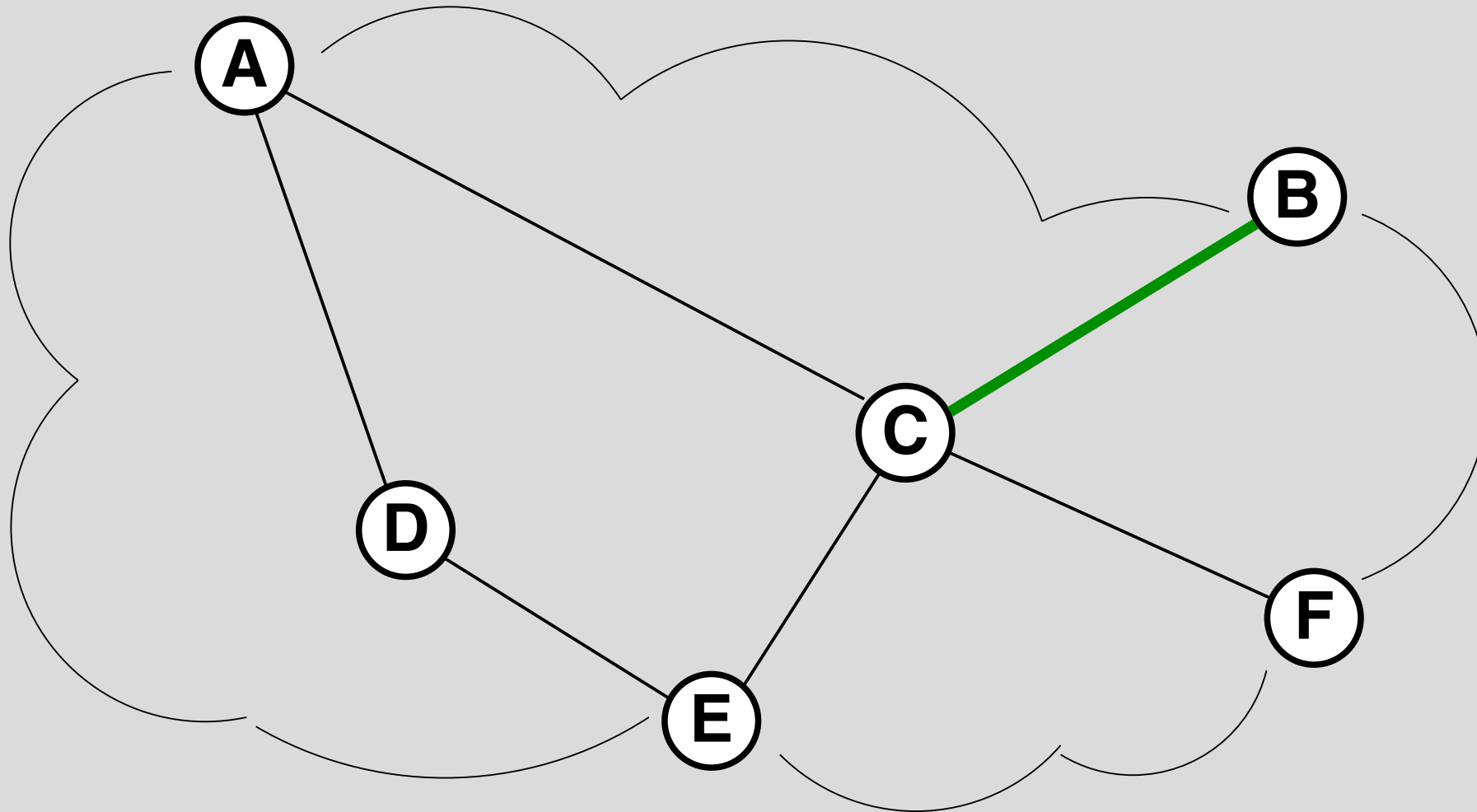
IGP routing

C floods that it's adjacent to B



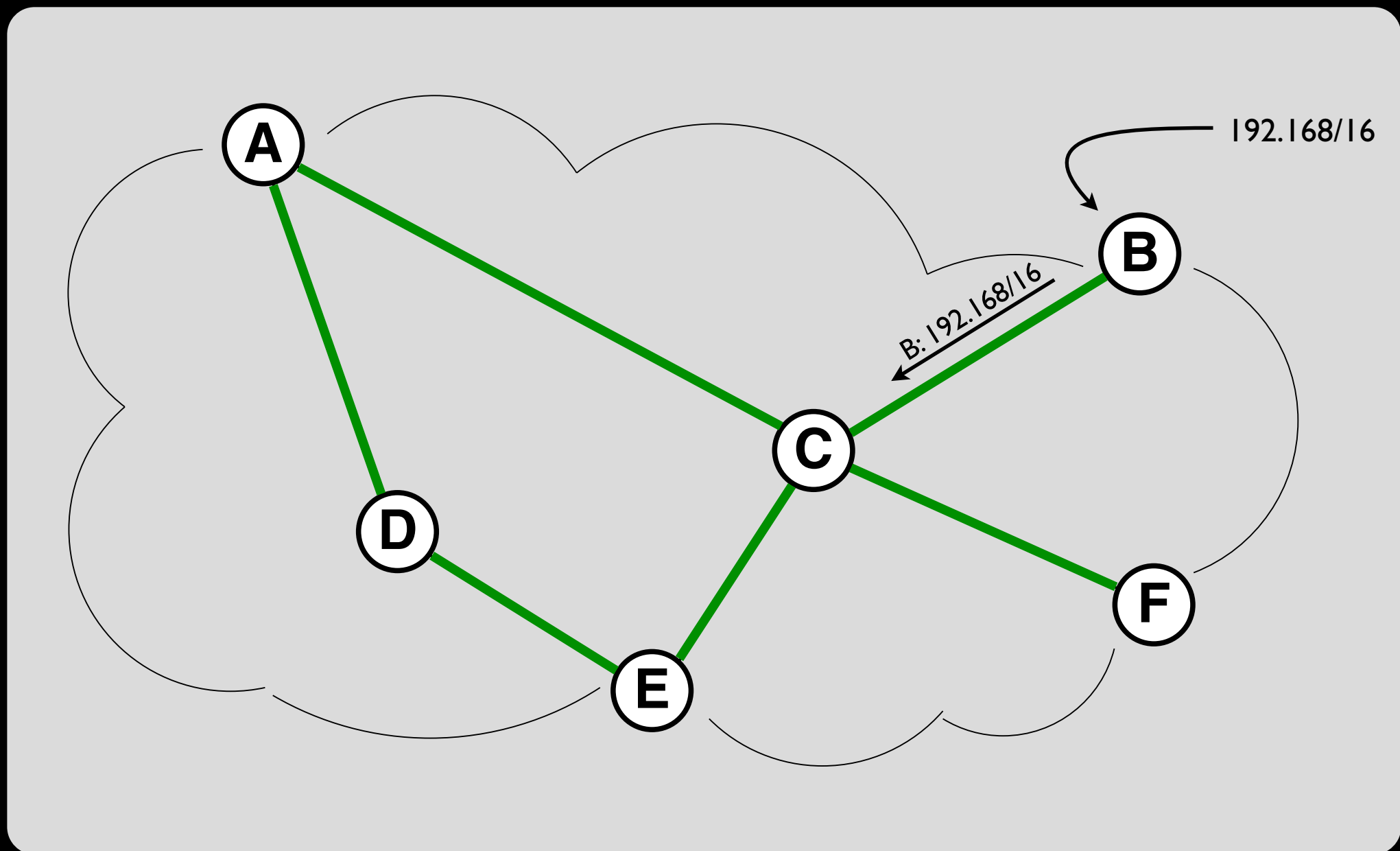
IGP routing

Same thing happens in $C \rightarrow B$ direction. When 'B adj C' announcement flooded, everyone adds B-C link to map.

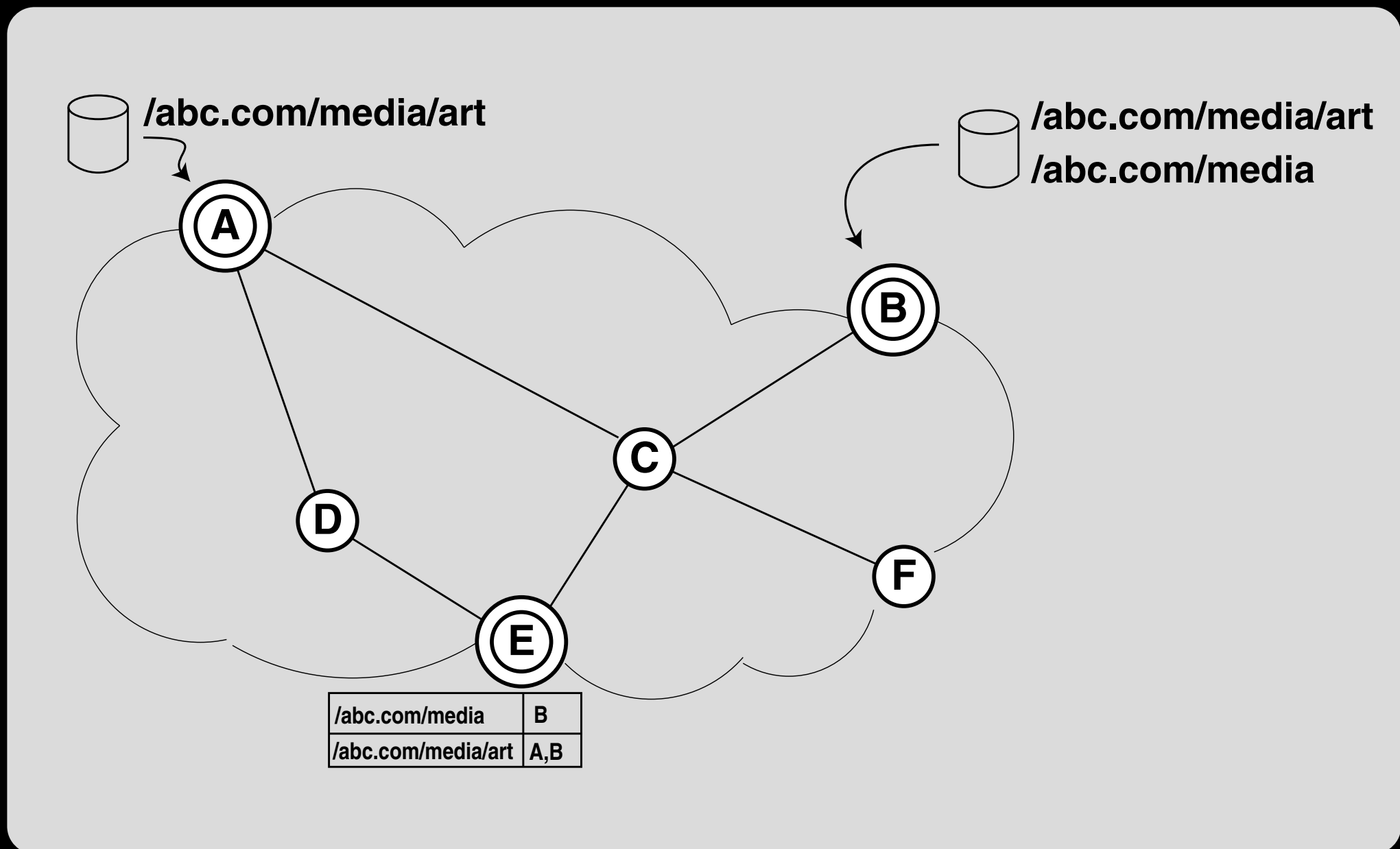


IGP routing

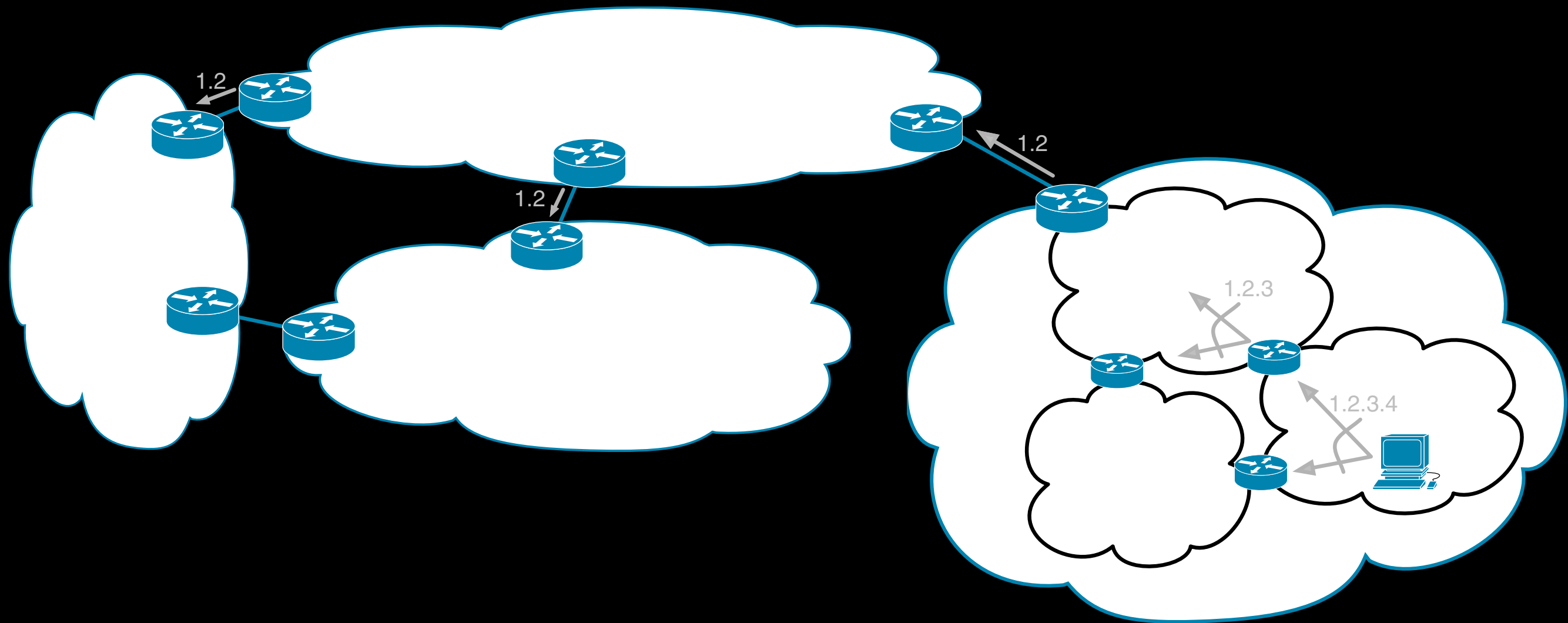
Some 'external' (non-IGP) agent injects a 'prefix announcement' which B floods to all other IGP nodes



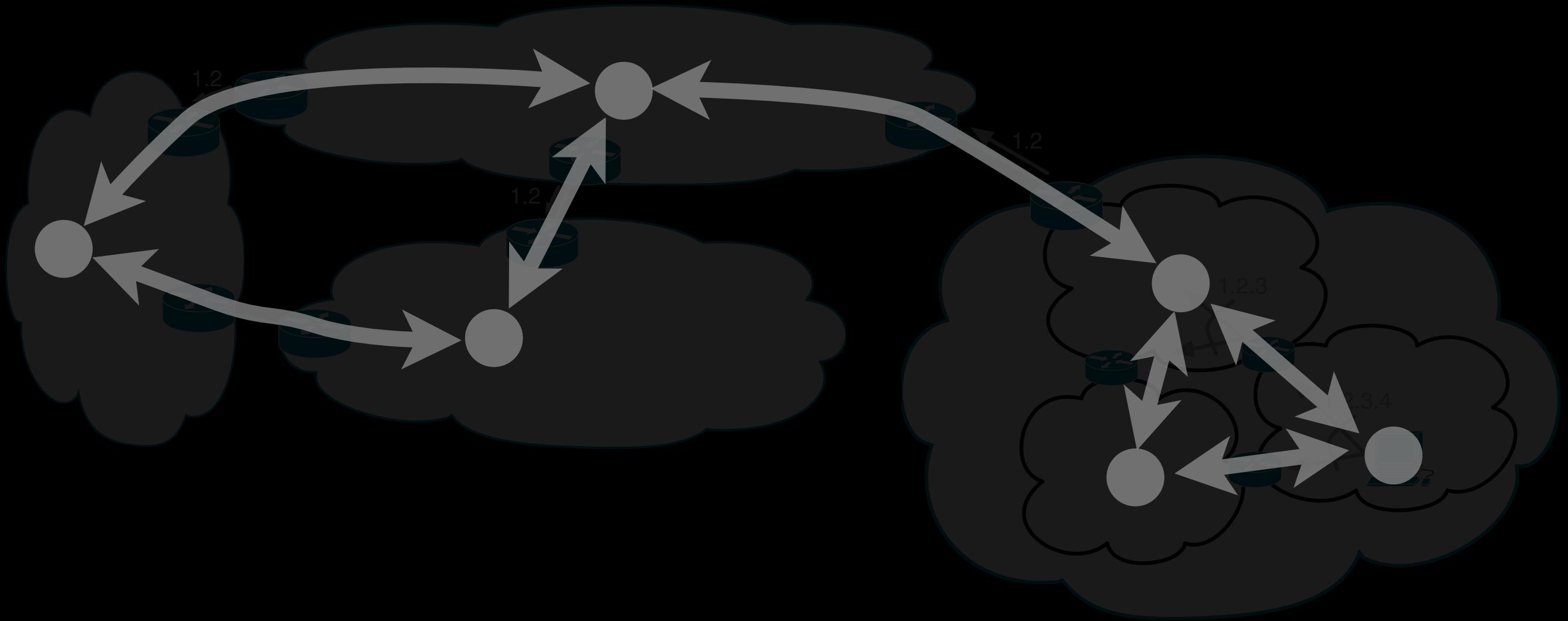
Existing link-state routing protocols can be used, unmodified, to construct a CCN FIB



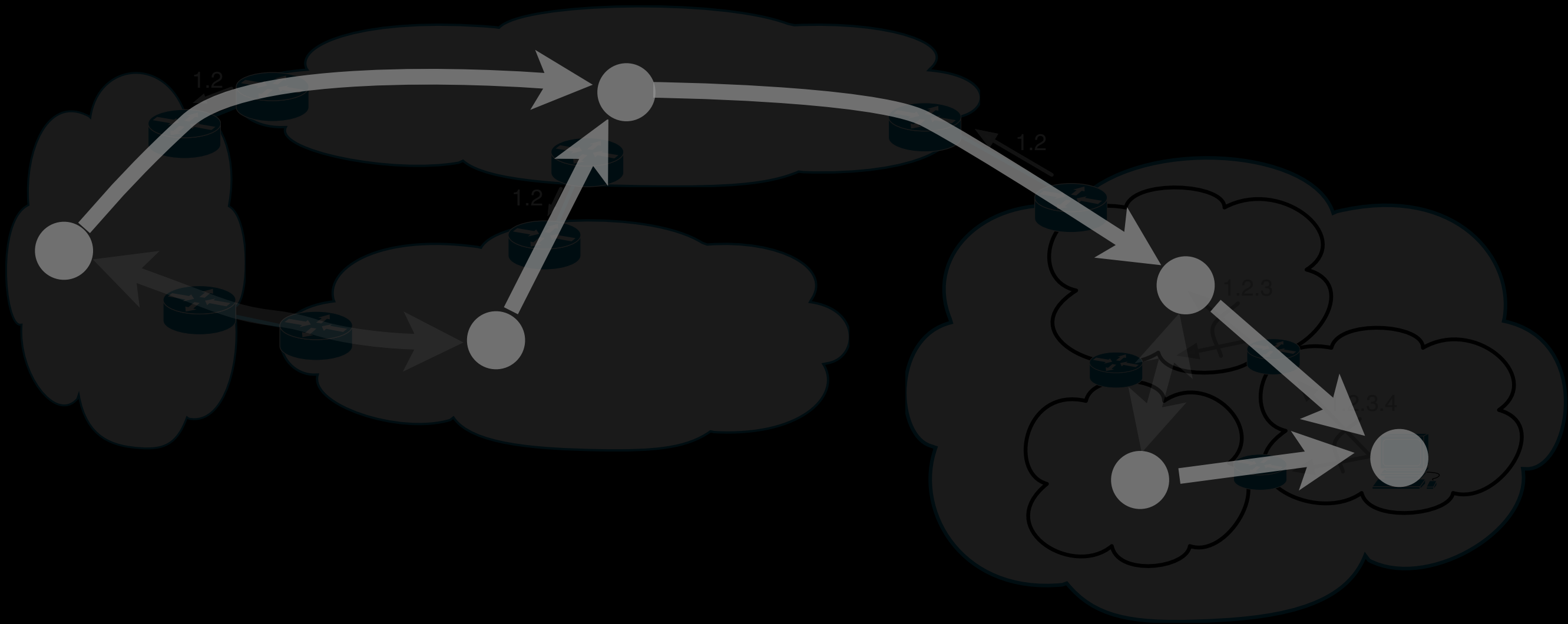
Architectural issues with IP routing



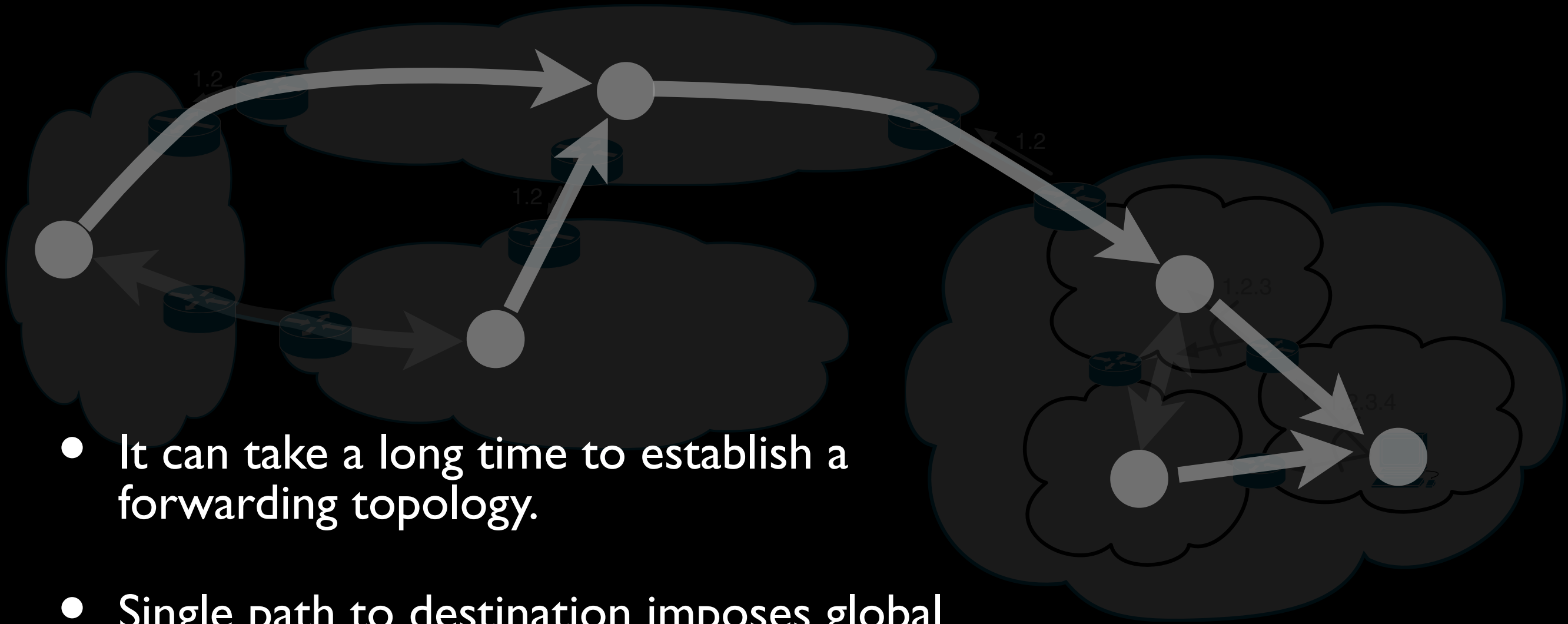
Architectural issues with IP routing



Architectural issues with IP routing



Architectural issues with IP routing



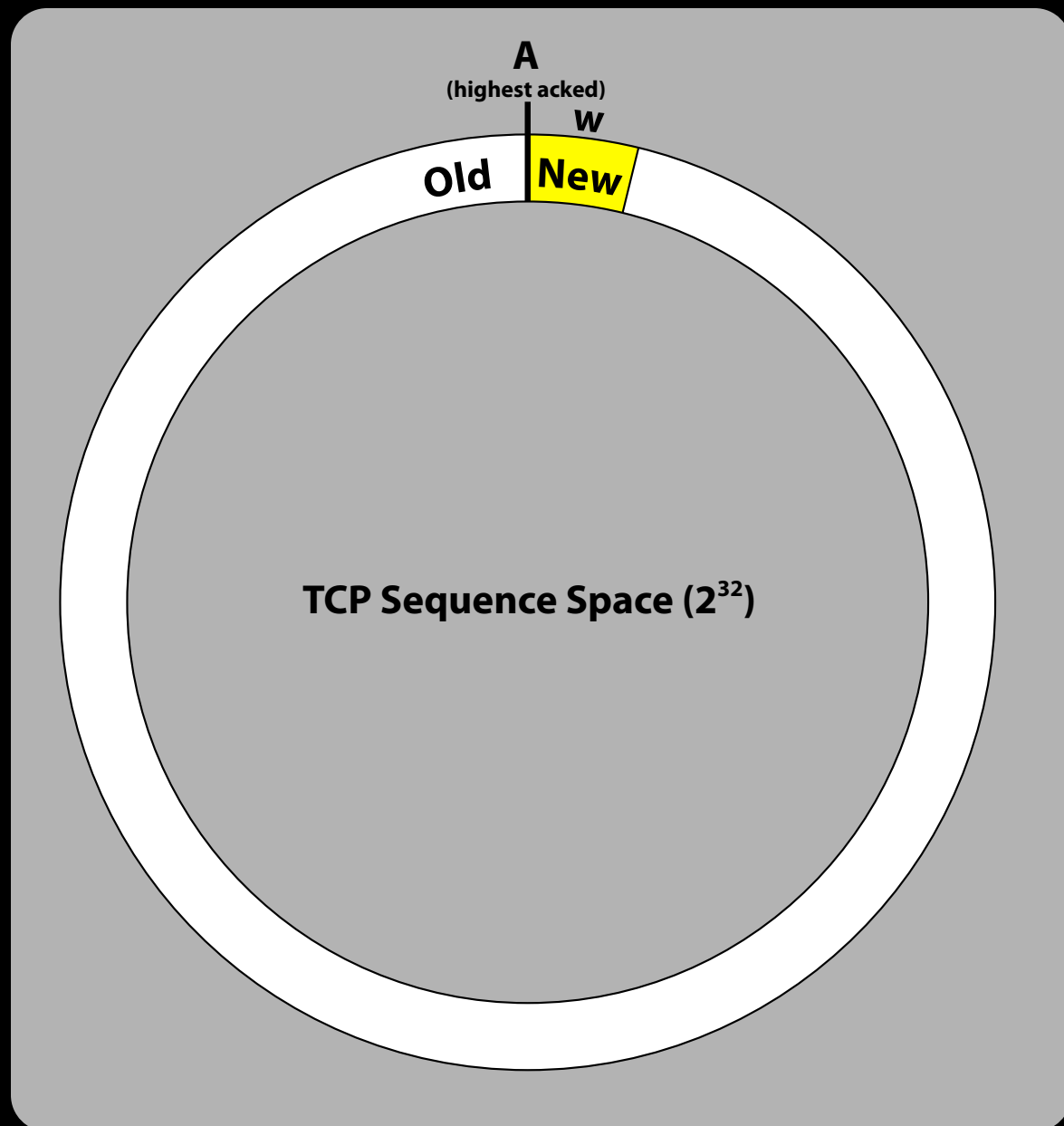
- It can take a long time to establish a forwarding topology.
- Single path to destination imposes global constraints on local forwarding decision.

CCN doesn't need topology

- Content model suppresses duplicates so nothing can loop.
- Data bandwidth use is near theoretical min.
- Topology can reduce interest bandwidth use but factor is small - $O(\text{diameter} * \text{avg.degree})$.
- With no topology, interests will always find (dynamic) shortest path to source.
- CCN works well with sloppy topology and performance-based interest re-expression.

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

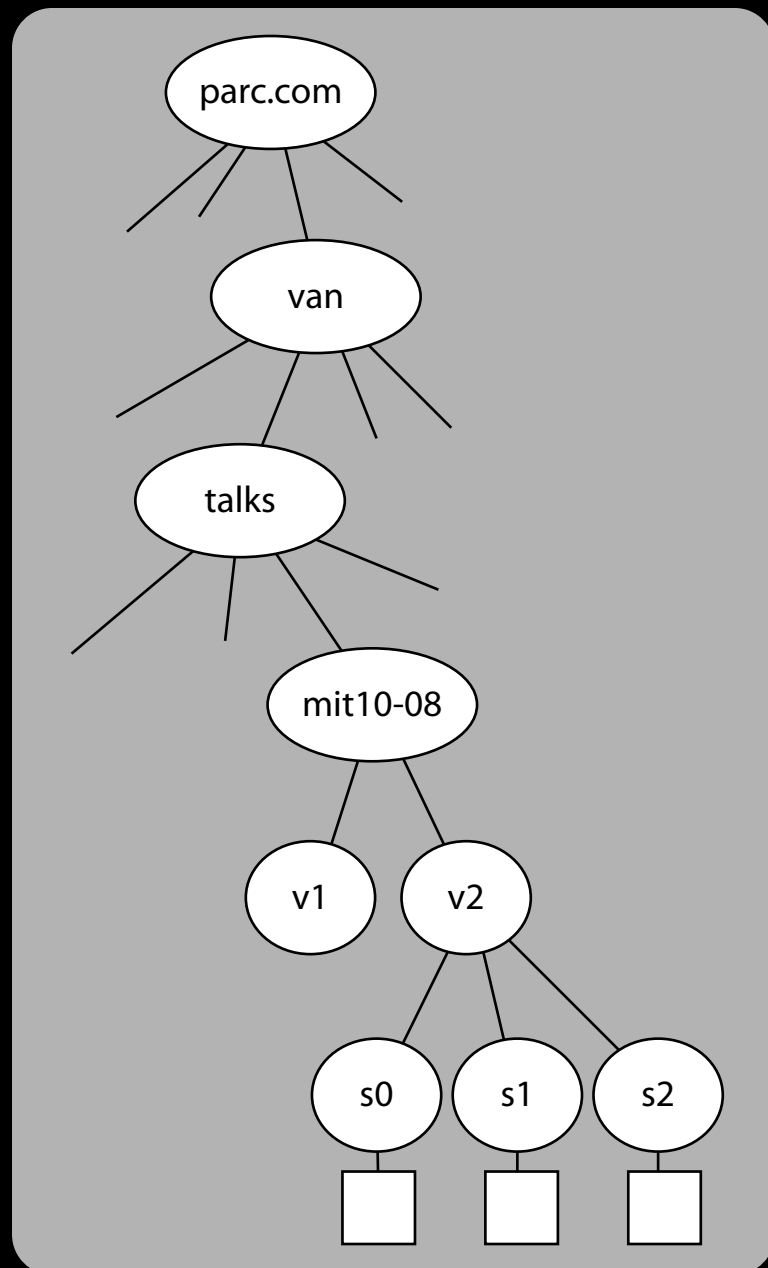
Transport State



Conversation transport state is very compact:

- One dynamic state variable (tcp sequence / ack number) conveys what ends know.
- Additional static variable (tcp window) conveys what they want.

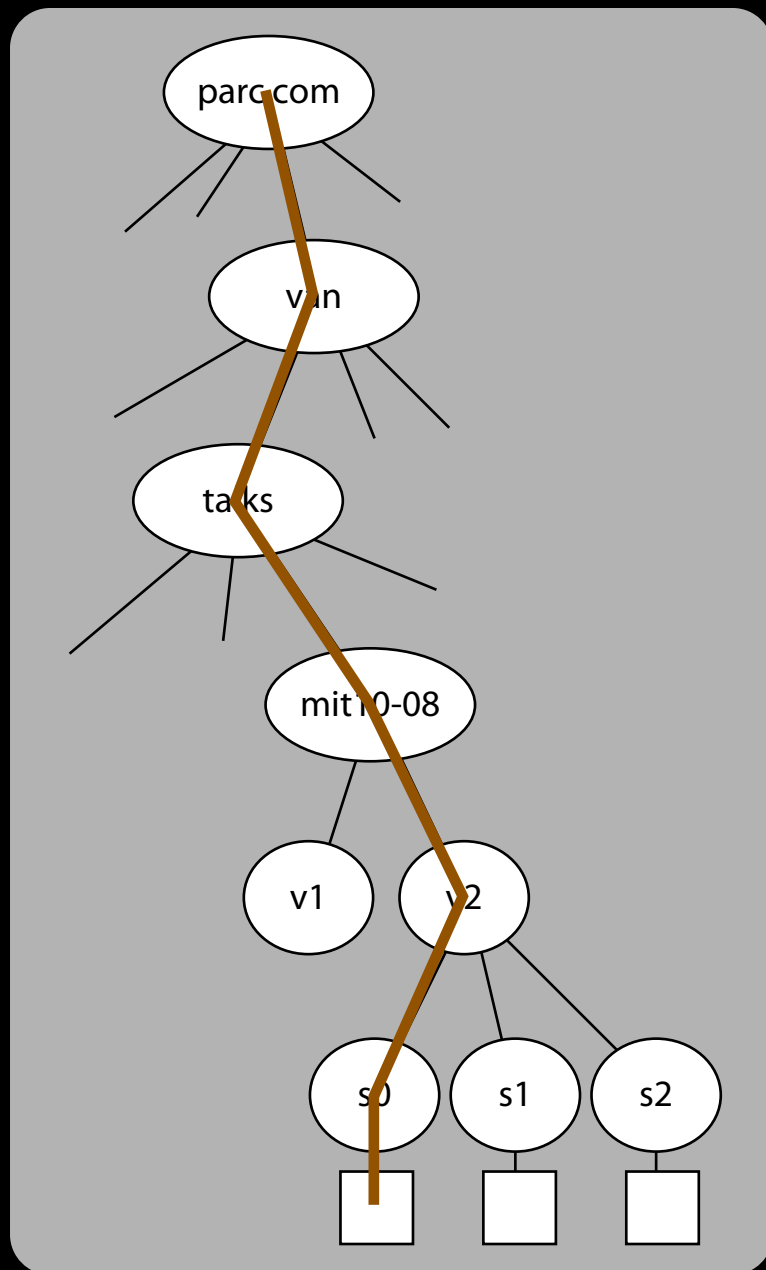
Annotated path in CCN name tree serves as transport state



Conventions:

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

Annotated path in CCN name tree serves as transport state

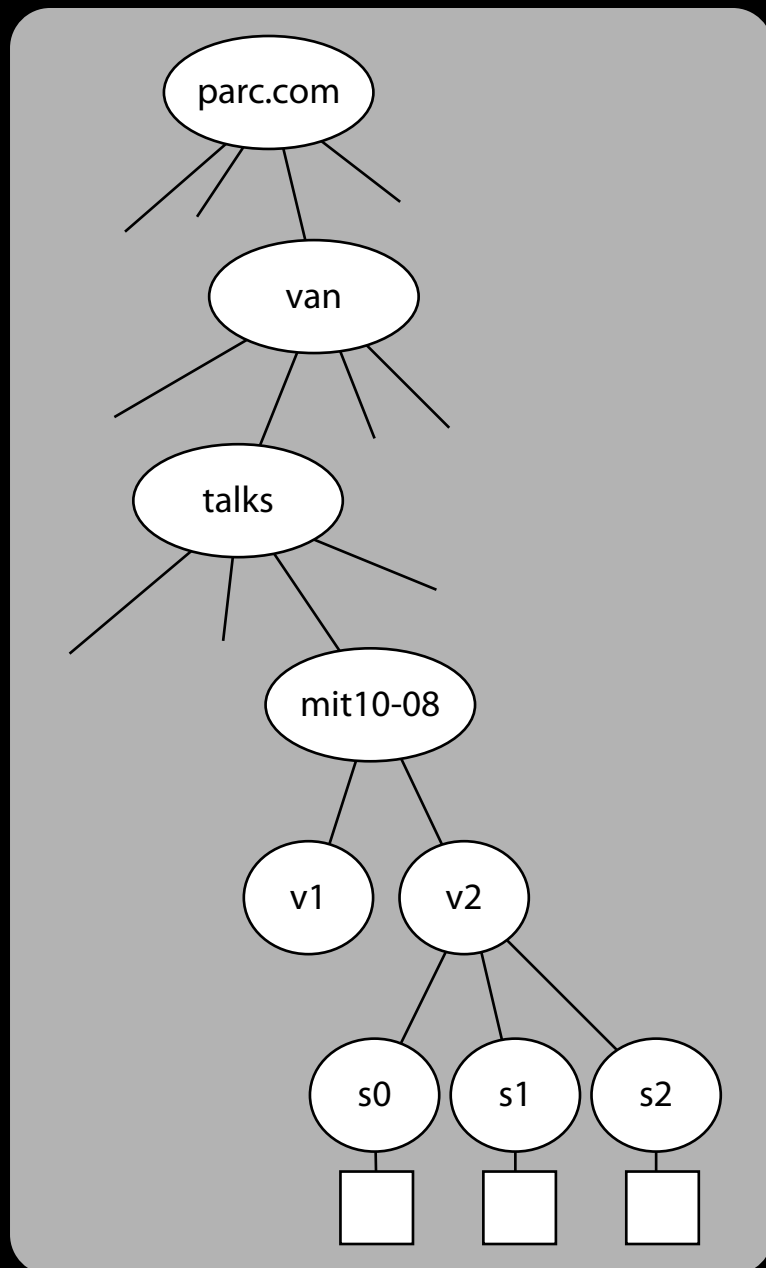


Most recent version of slides for this talk:
parc.com/van/talks/mit10-08 <rightmost child>

Conventions:

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

Annotated path in CCN name tree serves as transport state



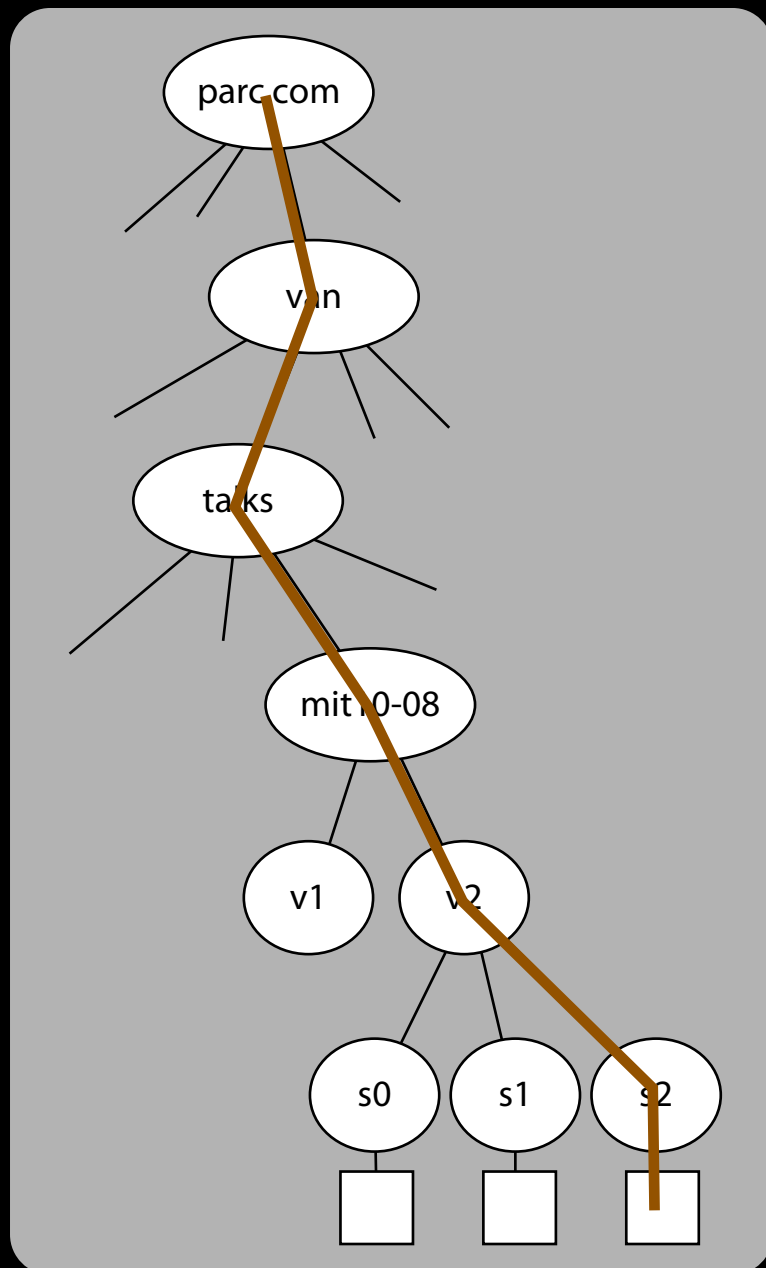
Most recent version of slides for this talk:
parc.com/van/talks/mit10-08 <rightmost child>

Newest version after v2:
parc.com/van/talks/mit10-08/v2 <rightmost sibling>
(fails since there is no newer version)

Conventions:

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

Annotated path in CCN name tree serves as transport state



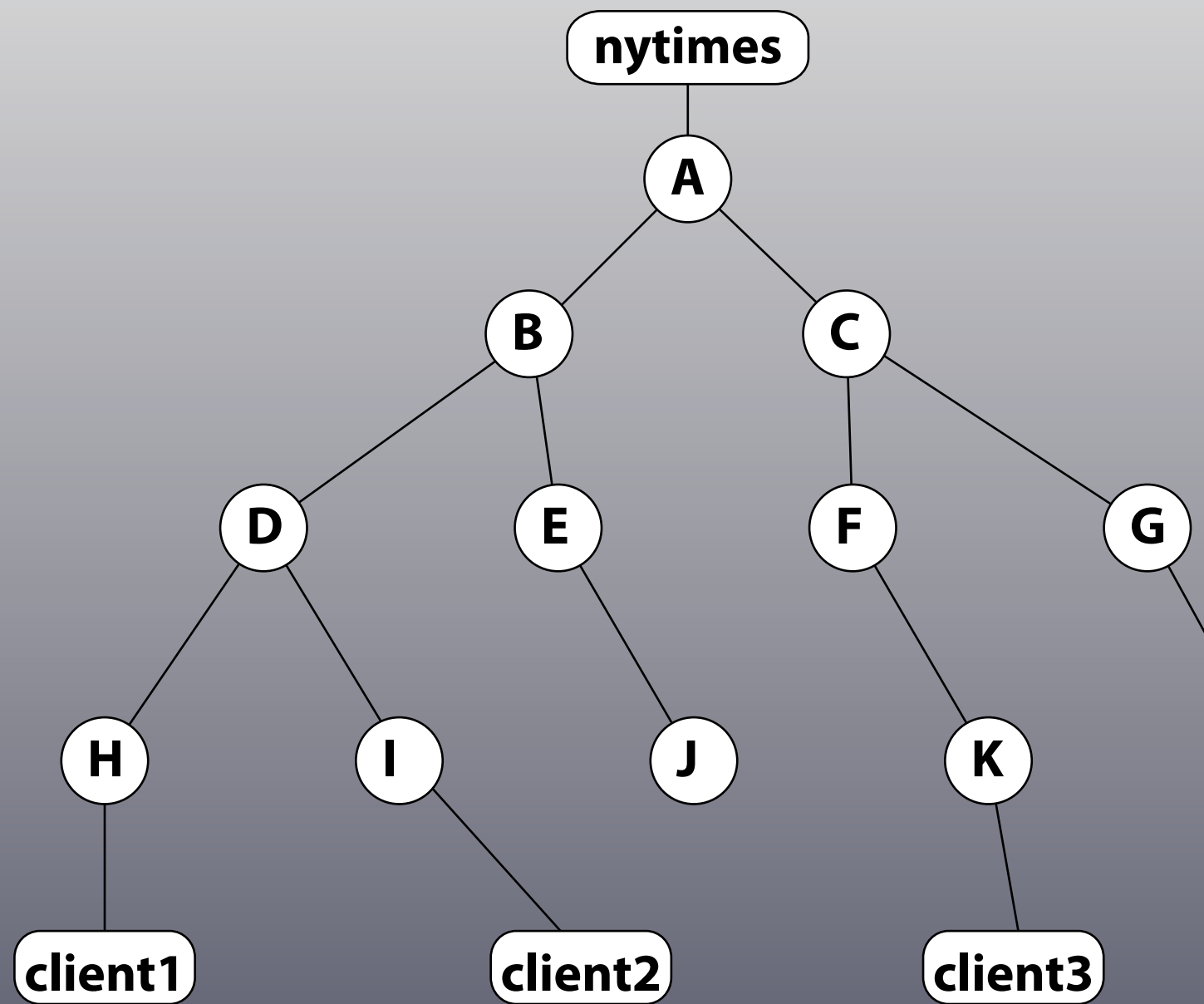
Most recent version of slides for this talk:
`parc.com/van/talks/mit10-08` <rightmost child>

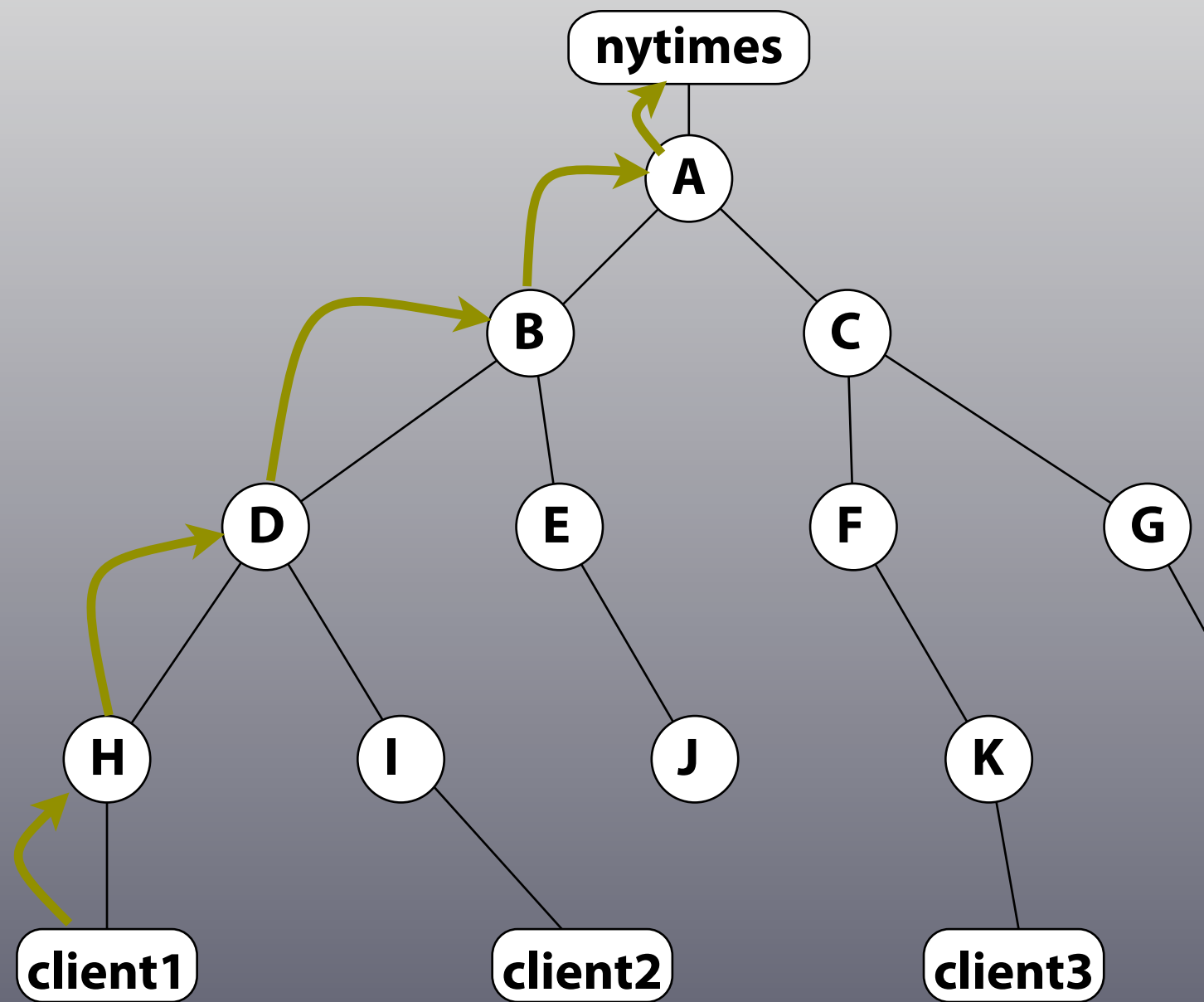
Newest version after v2:
`parc.com/van/talks/mit10-08/v2` <rightmost sibling>
(fails since there is no newer version)

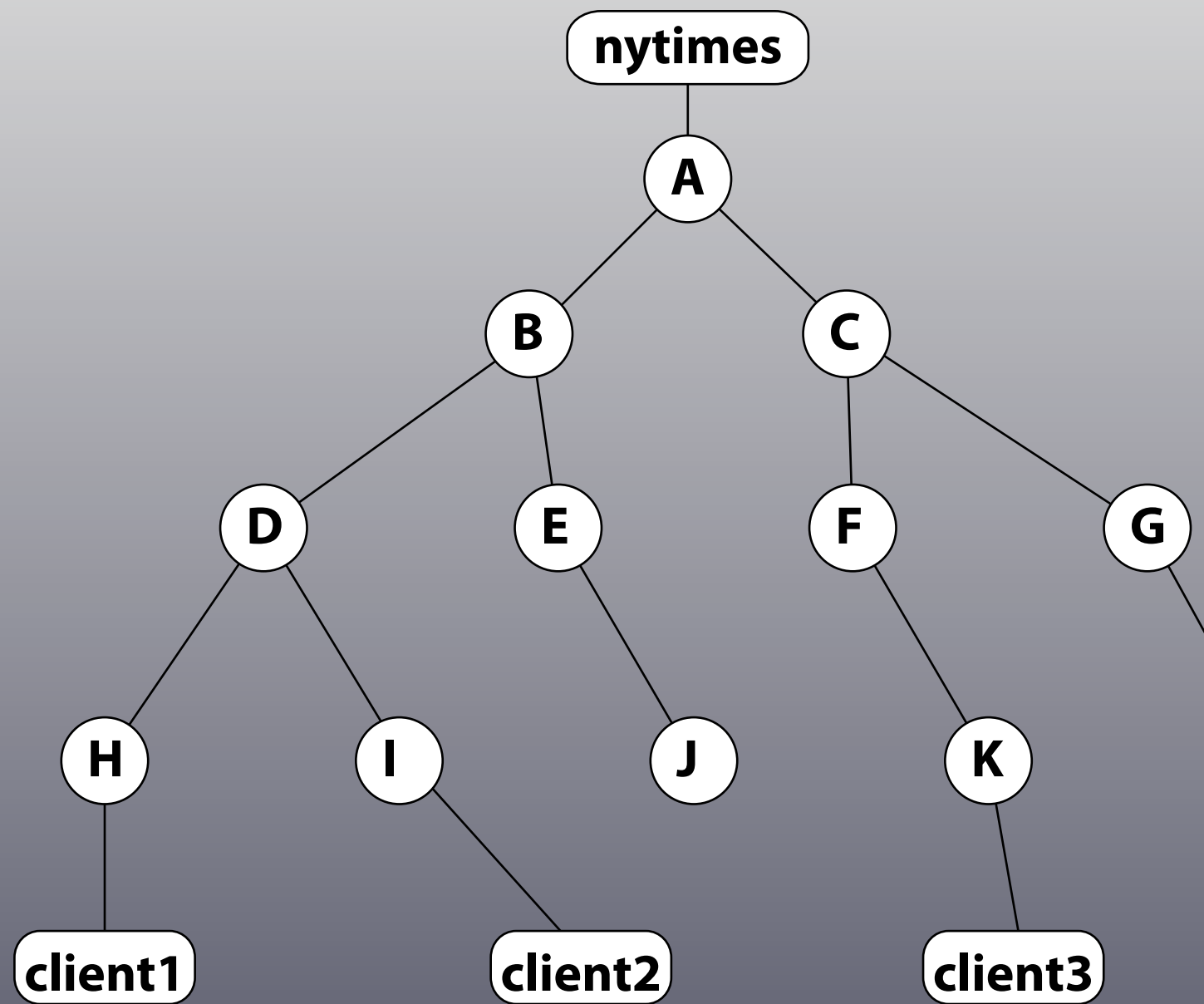
Next available chunk after s1:
`parc.com/van/talks/mit10-08/v2/s1`

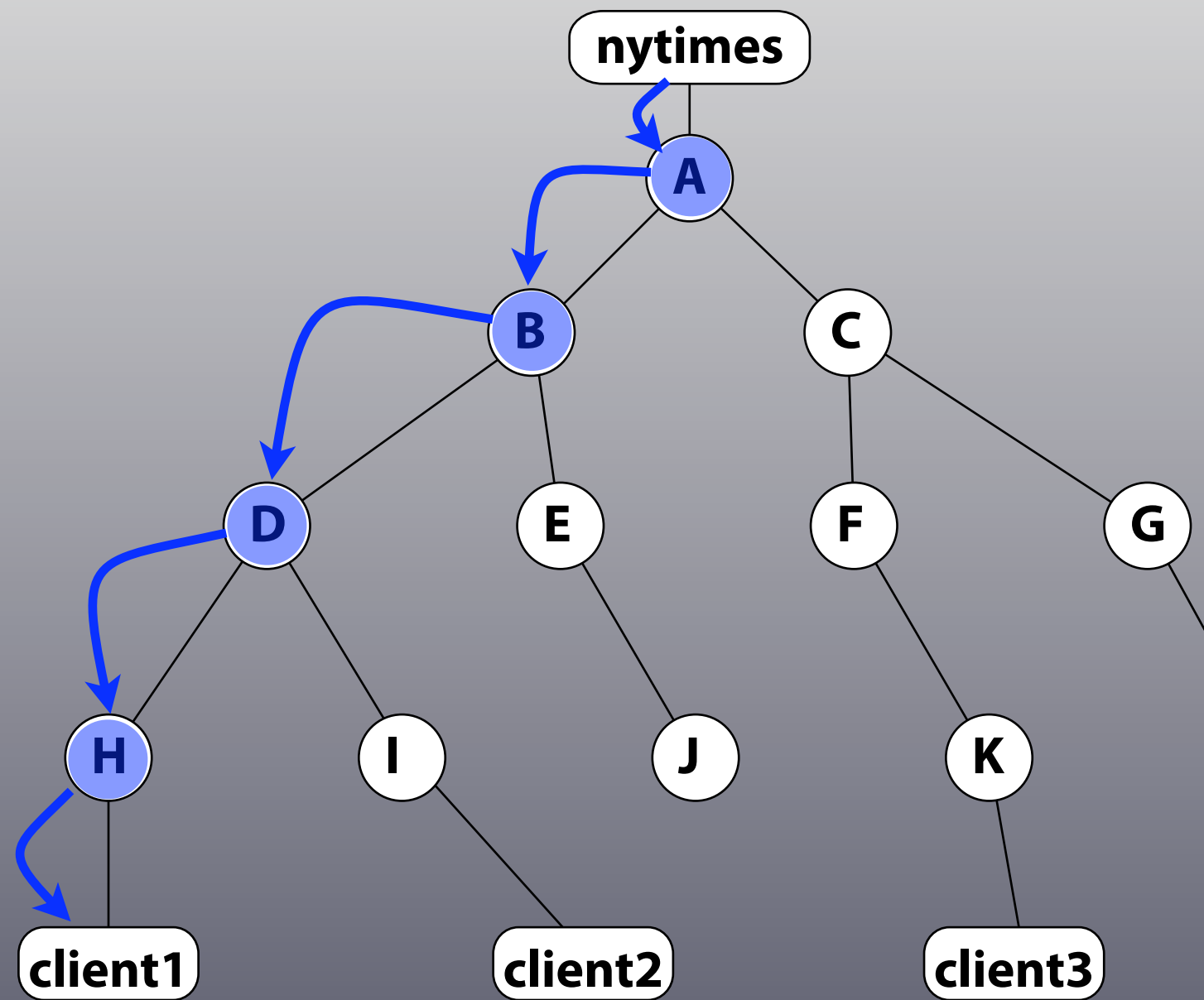
Conventions:

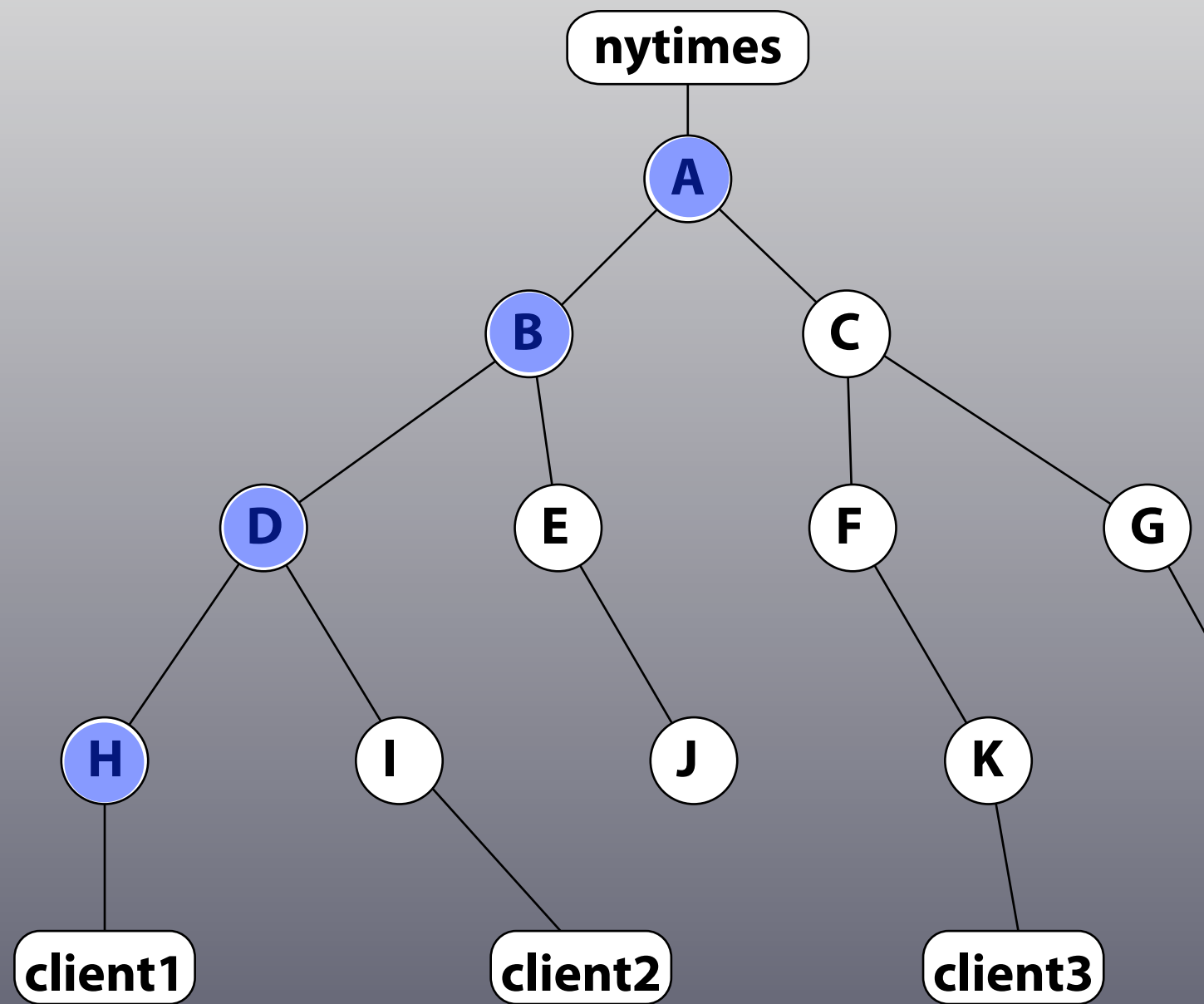
- name tree child nodes are lexically ordered
- <next> assumed if no relationship specified

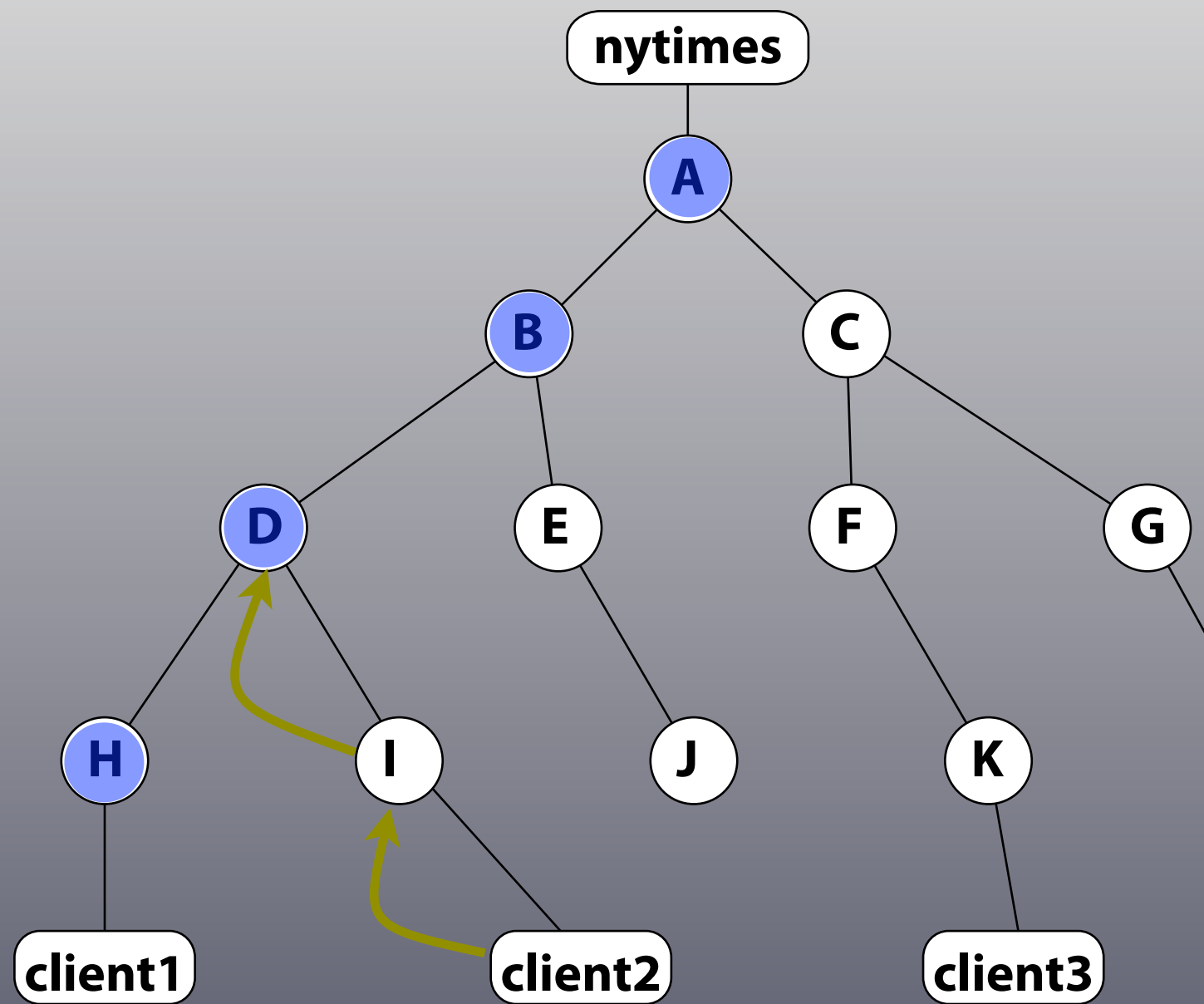


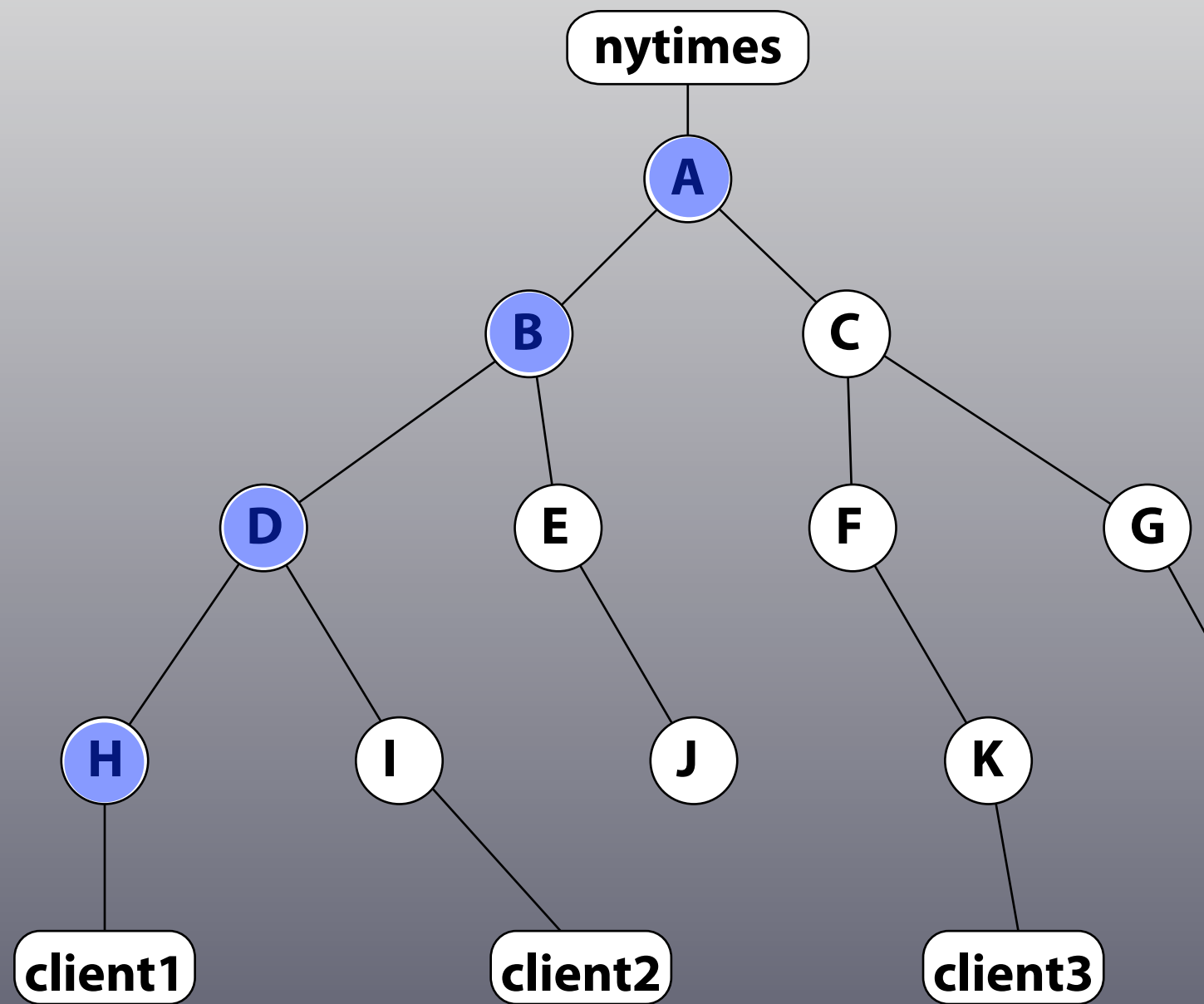


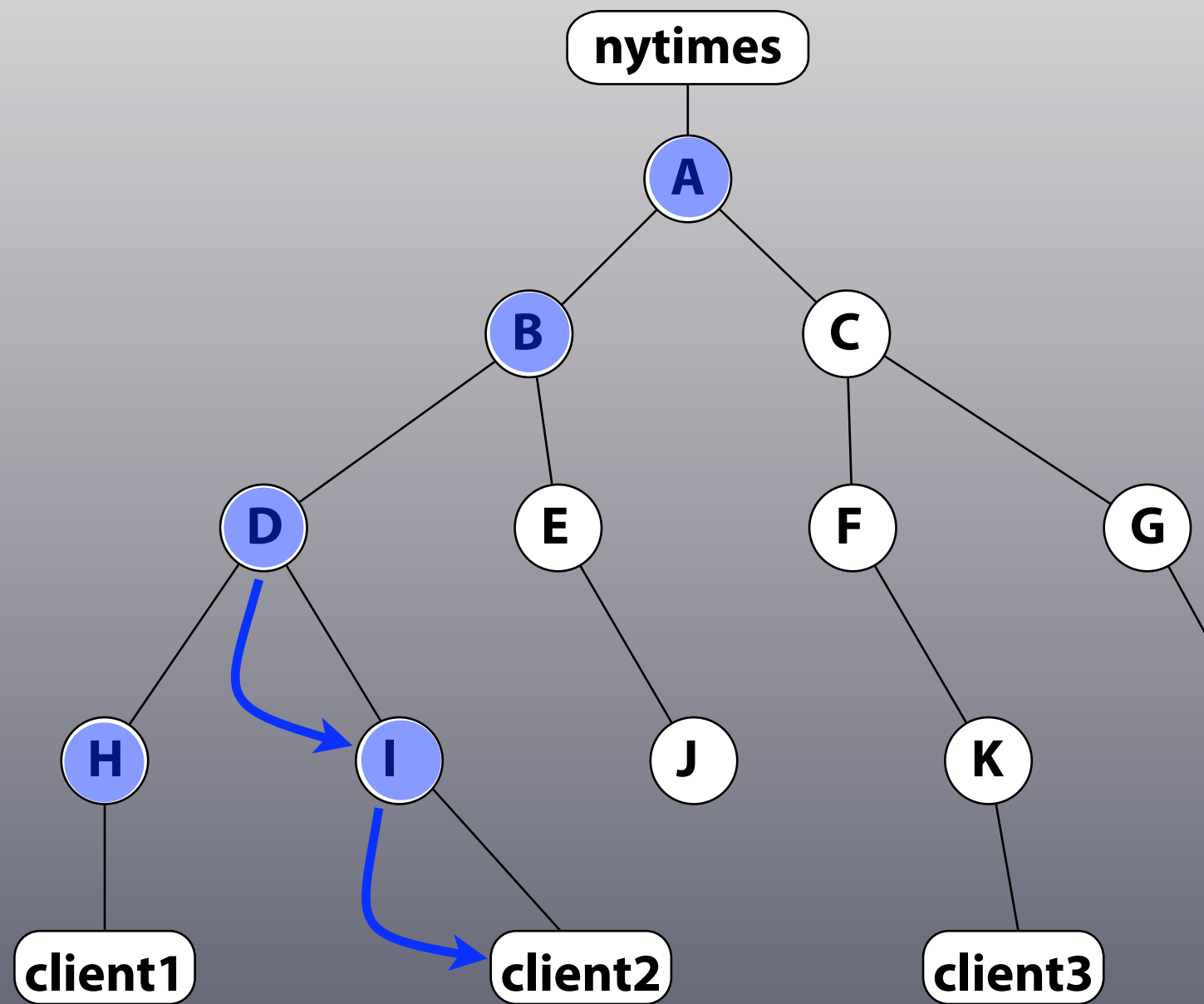


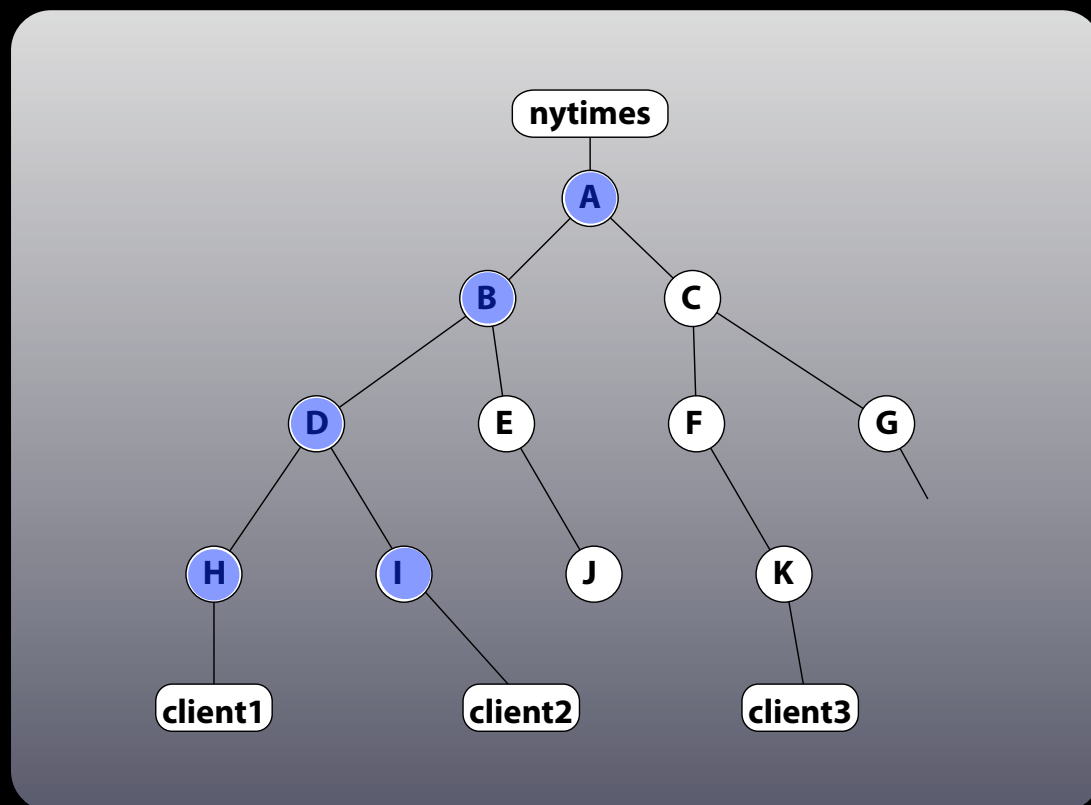






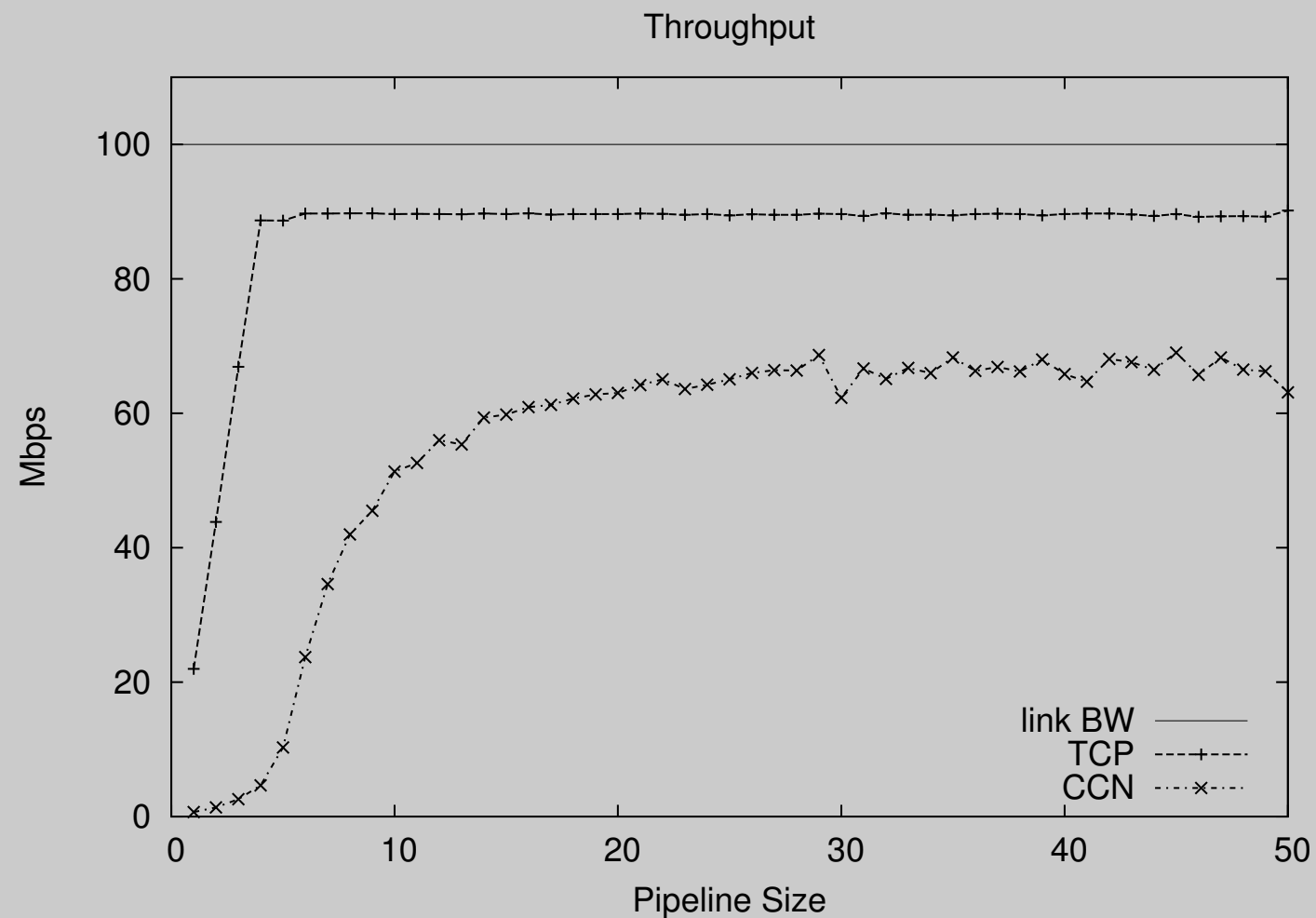




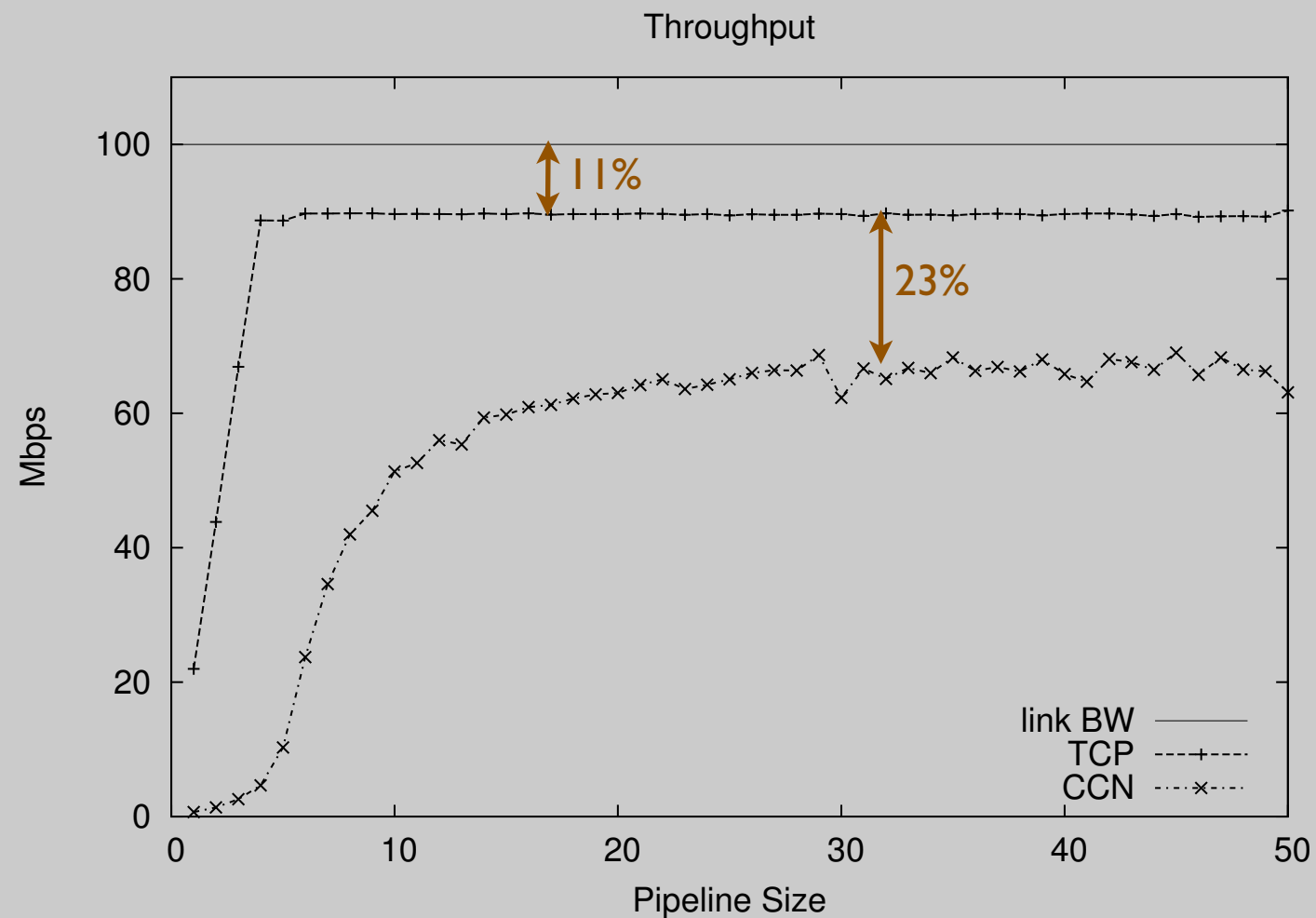


- Content goes only where there's interest.
- It takes at most one trip across any link.
- Average latency is minimized.
- Total bandwidth is minimized.
- There's no routing or control traffic associated with the replicas.

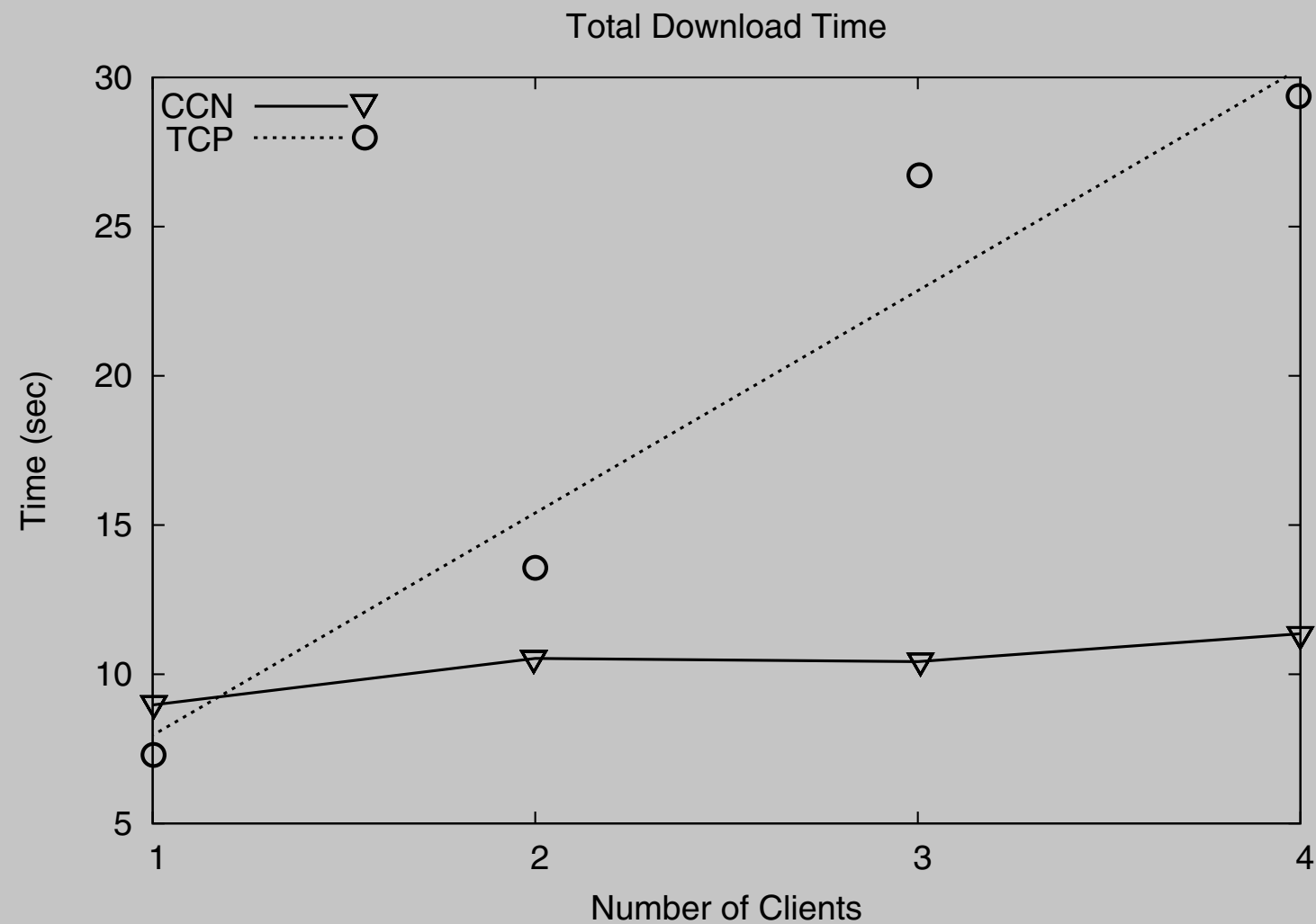
Bulk-data transfer performance comparison



Bulk-data transfer performance comparison



Shared-content performance comparison



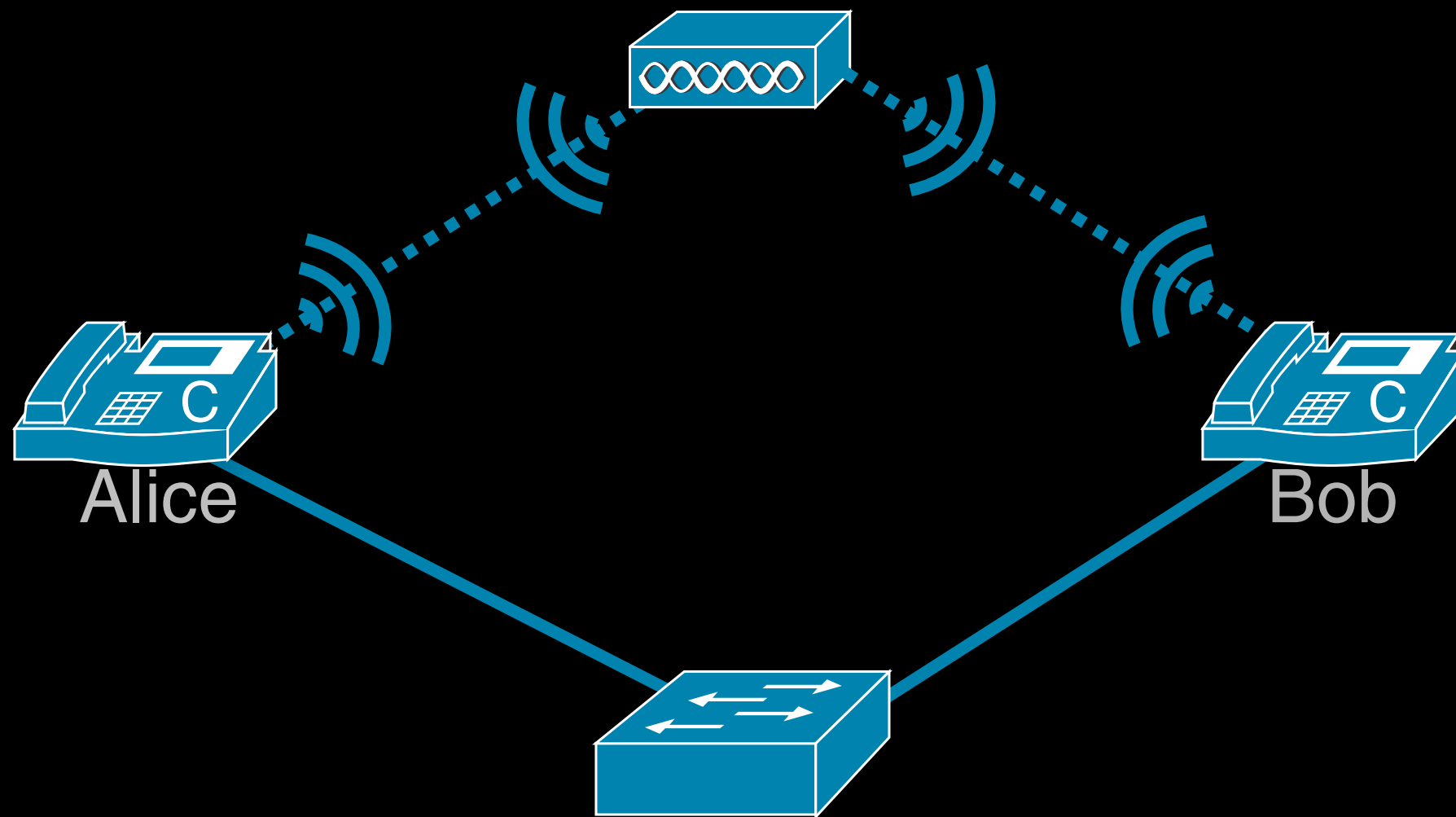
Random transport notes

- Unsatisfied interests are timed out.
Consumer is responsible for re-expressing interest if they still care (fate-sharing soft-state model).
- Adaptive suppression (as in SRM or NORM) avoids response implosions on broadcast faces.
- Basic CCN behaves as an efficient, secure, serverless, scalable distributed pub-sub plus intentional names.

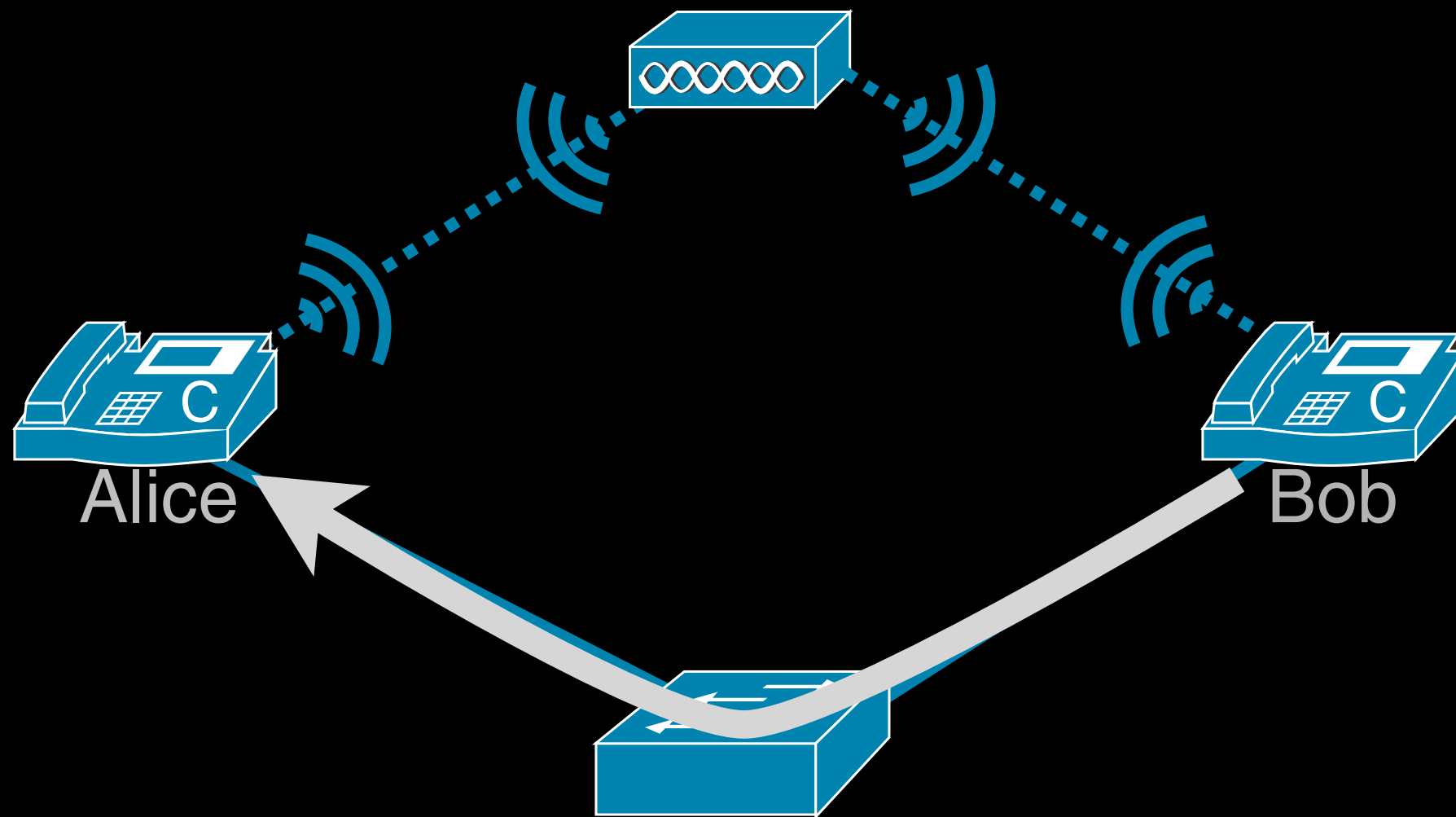
Strategy layer (mobility management)

- If you don't care who you're talking to, you don't care if they change.
- If you can only ask for a few small pieces at a time, it doesn't matter much if one gets dropped.
- If you can use any and all your links simultaneously, it's easy for the stack to run experiments.
- 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



A New Layering

