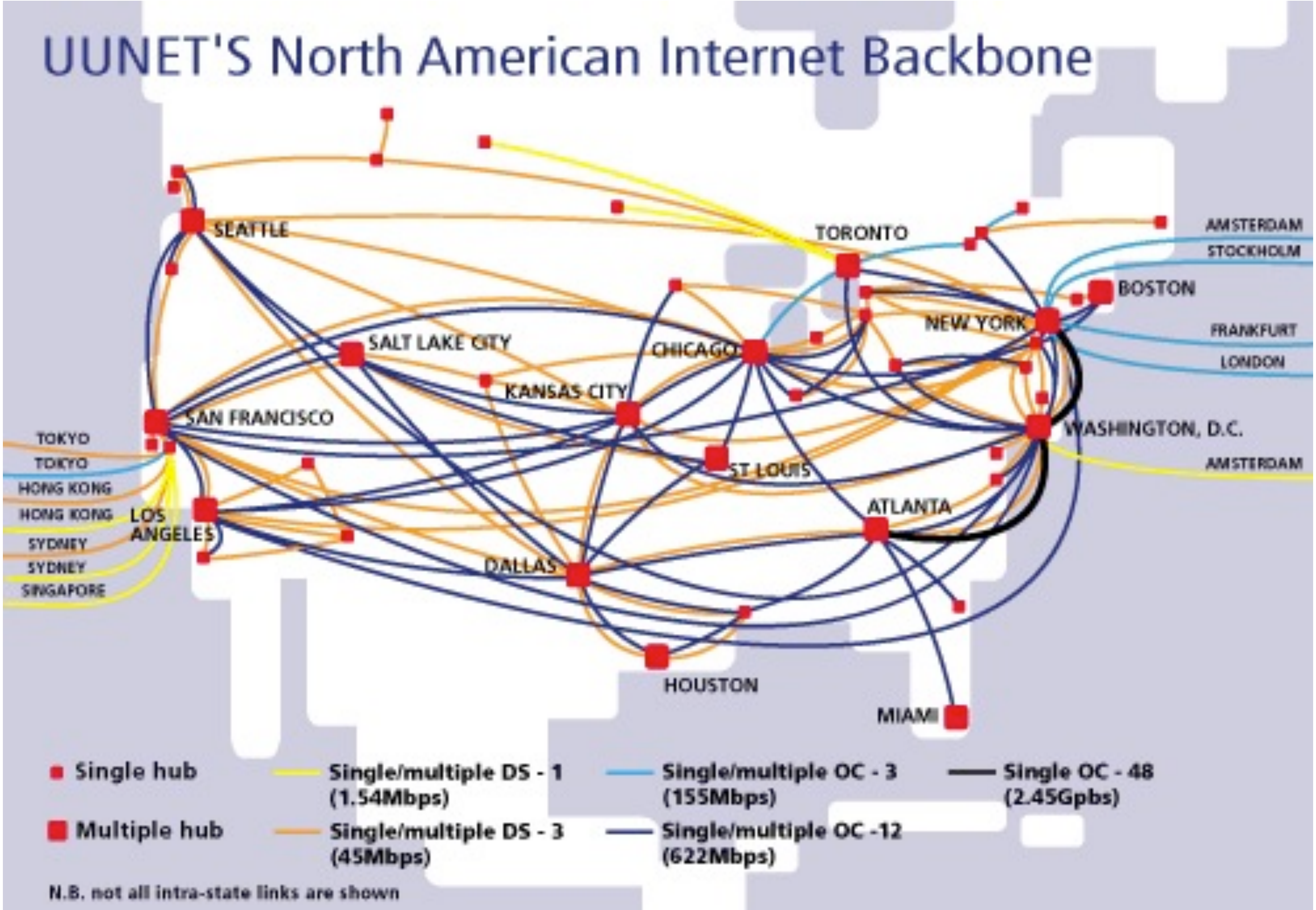
CS4700/CS5700
Fundamentals of Computer Networks

Lecture 22: Overlay networks

Slides used with permissions from Edward W. Knightly,
T. S. Eugene Ng, Ion Stoica, Hui Zhang

Abstract View of the Internet

- A collection of IP routers and point-to-point physical links connecting routers
- Point-to-point links between two routers are physically as direct as possible
  - A copper wire, a coax cable or a fiber laid from one router to another


UUNET'S North American Internet Backbone
SEATTLE
TORONTO
AMSTERDAM
STOCKHOLM
BOSTON
NEW YORK
FRANKFURT
LONDON
SALT LAKE CITY
CHICAGO
KANSAS CITY
SAN FRANCISCO
WASHINGTON, D.C.
TOKYO
TOKYO
HONG KONG
HONG KONG
SYDNEY
SYDNEY
SINGAPORE
LOS ANGELES
ST LOUIS
AMSTERDAM
ATLANTA
DALLAS
HOUSTON
MIAMI
Single hub
Multiple hub
Single/multiple DS - 1 (1.54Mbps)
Single/multiple DS - 3 (45Mbps)
Single/multiple OC - 3 (155Mbps)
Single/multiple OC -12 (622Mbps)
Single OC - 48 (2.45Gpbs)
N.B. not all intra-state links are shown

Reality

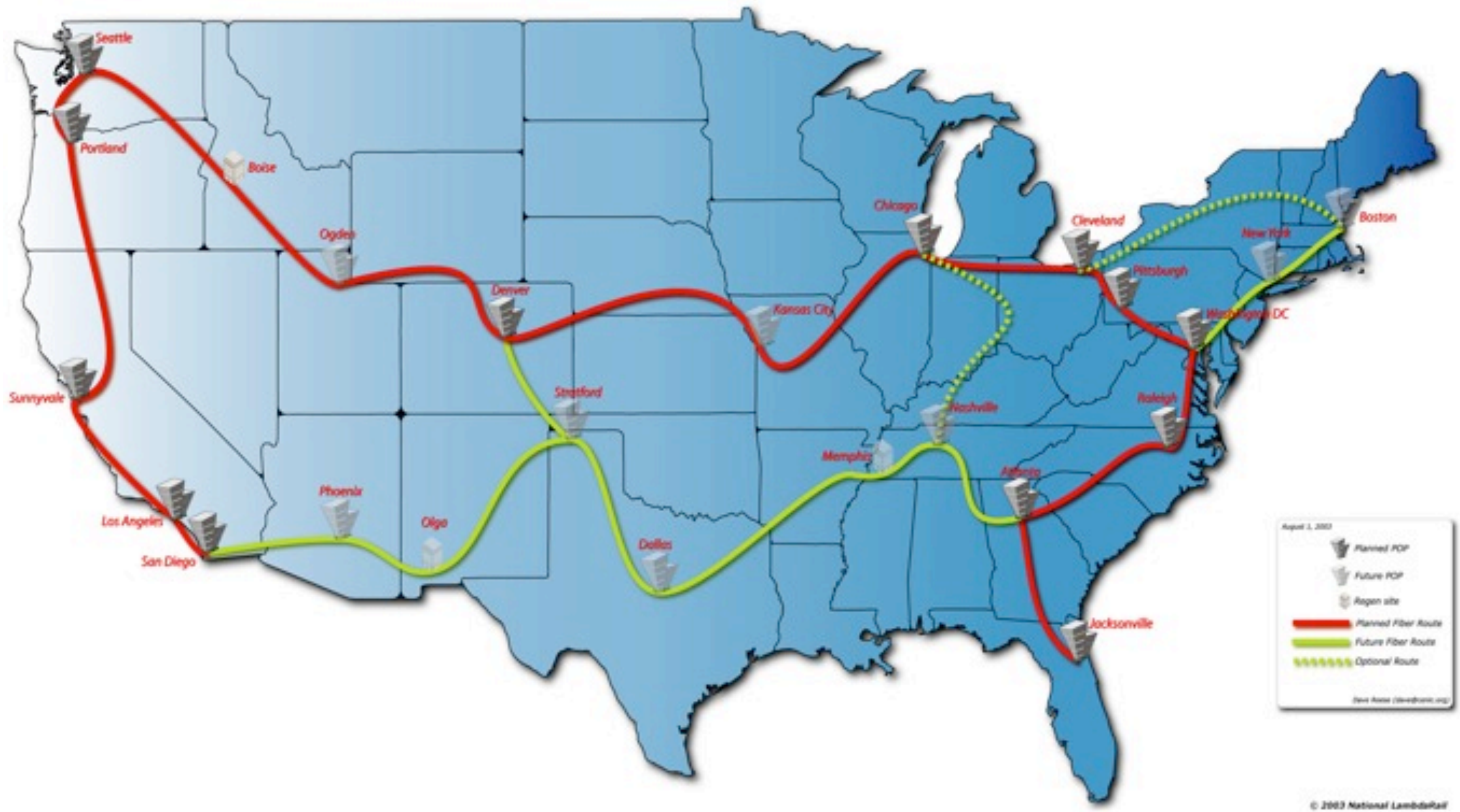
- Fibers and wires are laid with tremendous physical constraints
  - You can't just dig up the ground everywhere and lay fibers
  - Right-of-way issue
  - Most fibers are laid along railroads
- Physical fiber topology often very far from the topology you want

- IP Internet is over-laid on top of this physical fiber topology
- IP Internet topology is only logical!

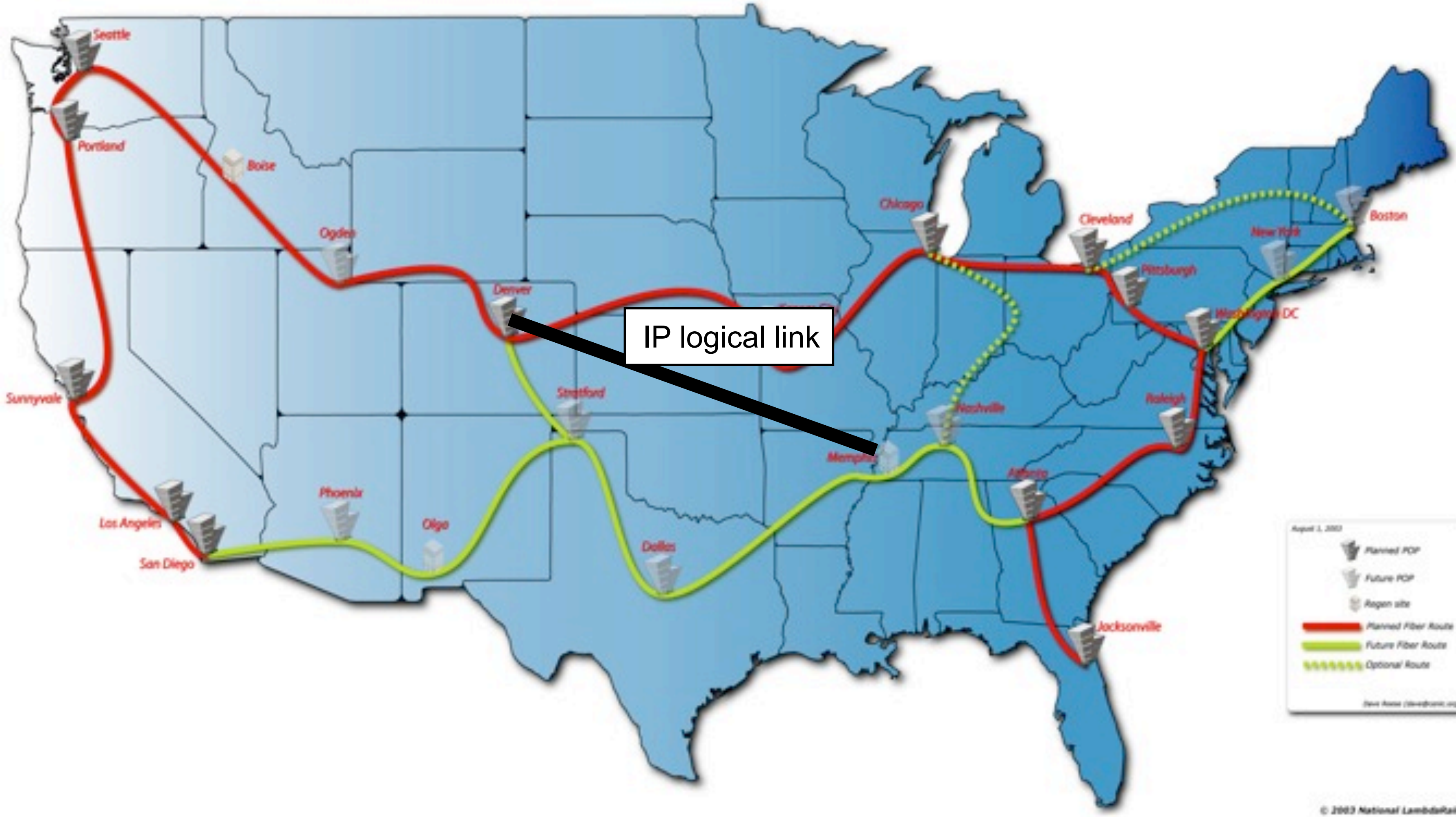
- Concept: IP Internet is an overlay network

E.g. National Lambda Rail Project – Fiber Topology

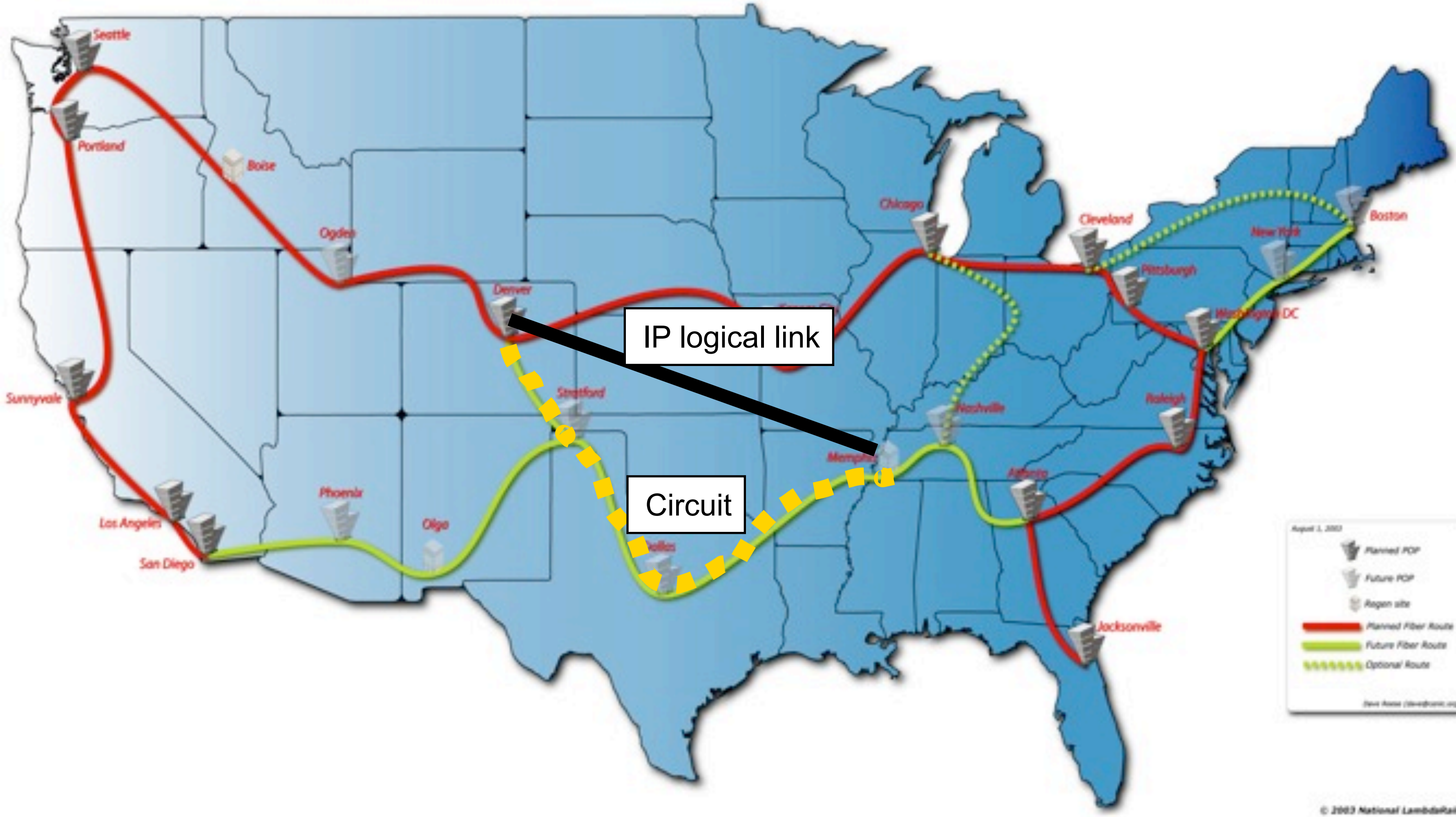
E.g. An IP logical link overlaid on a circuit



E.g. An IP logical link overlaid on a circuit

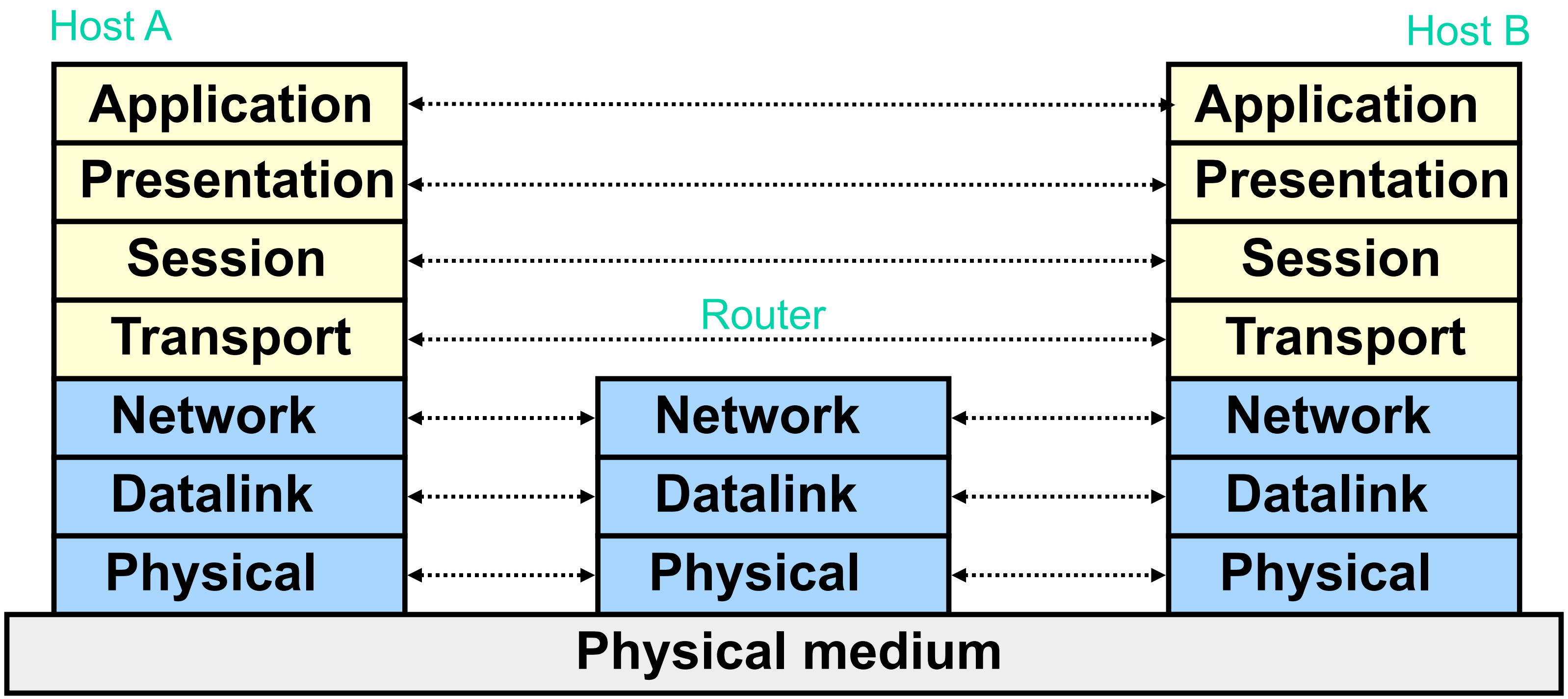


E.g. An IP logical link overlaid on a circuit



Made Possible by Layering

- Layering hides the detail of lower layer from higher layer
- IP operates on datalink layer (say ATM or SONET) logical topology
- ATM/SONET creates point-to-point circuits on the fibers



Overlay

- Overlay is clearly a general concept
  - You can keep overlaying one network on another, it's all logical
- IP Internet overlays on top of physical topology
  - Why stop here?
- Something else can overlay on top of IP Internet
  - Use IP tunnels to create yet another logical topology
  - E.g. VPNs

Advanced Reasons to Overlay On IP Internet

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?
  - Multicast

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?
  - Multicast
  - Reliable performance-based routing

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?
  - Multicast
  - Reliable performance-based routing
  - More… e.g. content addressing and distribution

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?
  - Multicast
  - Reliable performance-based routing
  - More… e.g. content addressing and distribution
- Can you build can overlay network on IP Internet to provide QoS?

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?
  - Multicast
  - Reliable performance-based routing
  - More… e.g. content addressing and distribution
- Can you build can overlay network on IP Internet to provide QoS?
  - How?

Advanced Reasons to Overlay On IP Internet

- IP provides basic best effort datagram service
- Many things you may want in a network but not supported
- Like what?
  - Multicast
  - Reliable performance-based routing
  - More… e.g. content addressing and distribution
- Can you build can overlay network on IP Internet to provide QoS?
  - How?
  - Overlay links must have guaranteed performance characteristics, otherwise, the overlay network cannot guarantee anything!

Unicast Routing Overlay

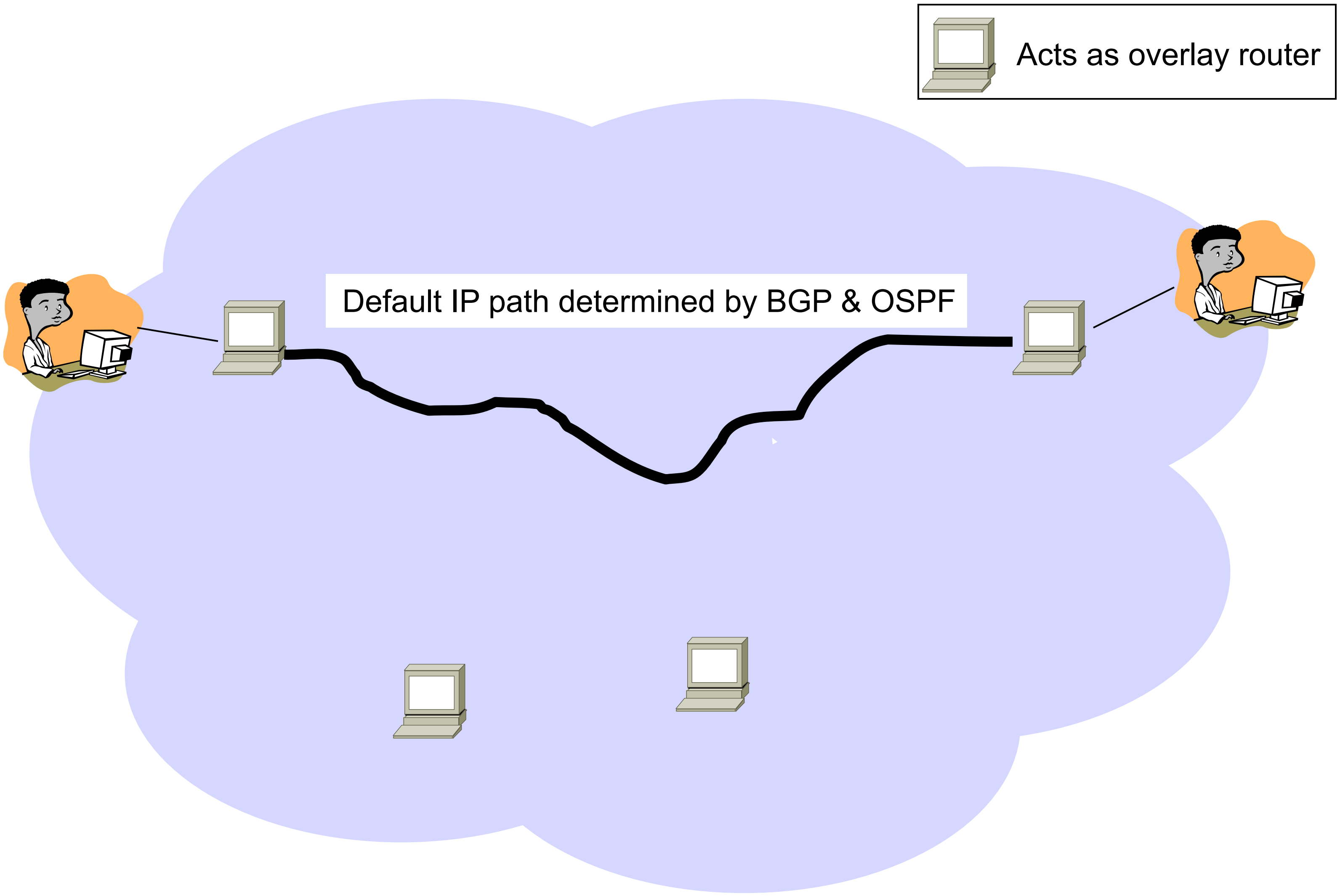
- Internet routing is built upon Intra-domain and Inter-domain router protocols
  - OSPF/RIP; BGP
- OSPF/RIP routing based on shortest link weight routing
  - Link weights are typically very static
  - Does not necessarily give you best performance path (delay, throughput, loss rate)
- BGP routing based mostly on policy
  - Policy may have nothing to do with performance
  - BGP very slow to react to failure (no reaction to high loss rate, e.g.)

Resilient Overlay Network (RON)

- Install N computers all over the place on the Internet
- Each computer acts as an overlay network router
  - Between each overlay router is a IP tunnel (logical link)
  - Logical overlay topology is all-to-all (N^2)
- Computers actively measure each logical link in real time for
  - Packet loss rate, latency, throughput, etc
- Route overlay network traffic based on measured characteristics

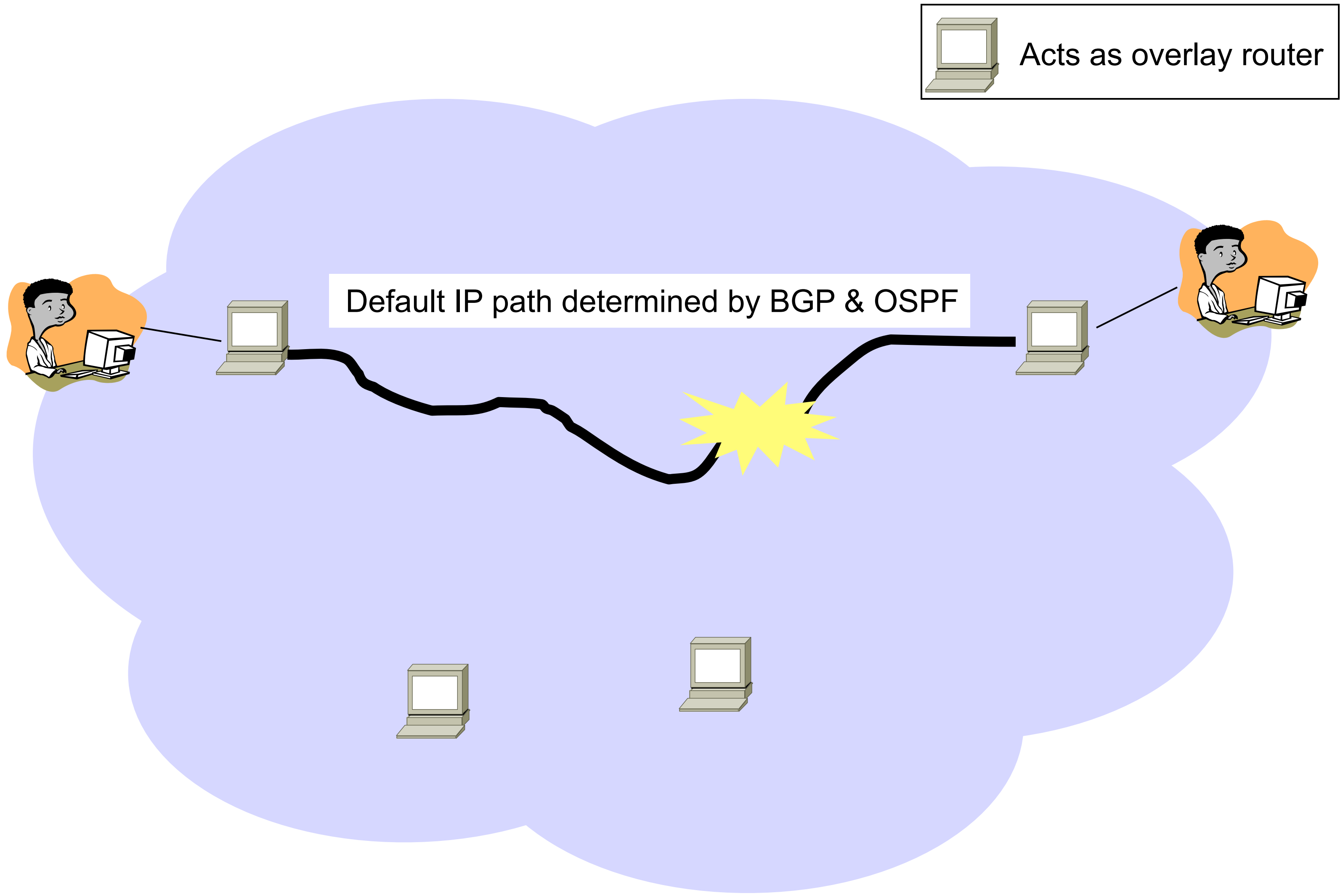
- Able to consider multiple paths in addition to the default IP Internet path given by BGP/OSPF

Example



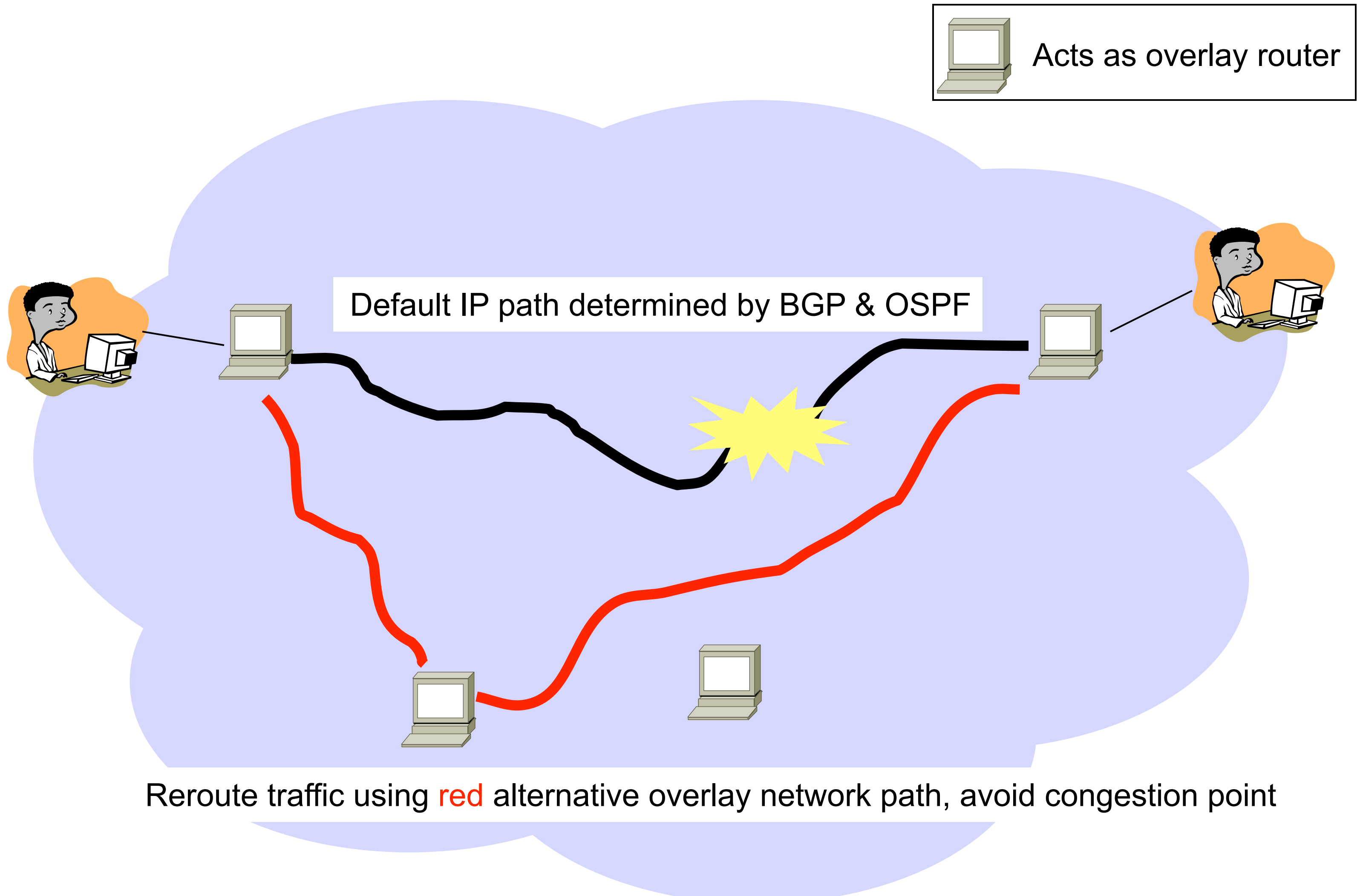
Acts as overlay router
Default IP path determined by BGP & OSPF

Example



Acts as overlay router
Default IP path determined by BGP & OSPF

Example


Acts as overlay router
Default IP path determined by BGP & OSPF
Reroute traffic using red alternative overlay network path, avoid congestion point

Potential Problems…

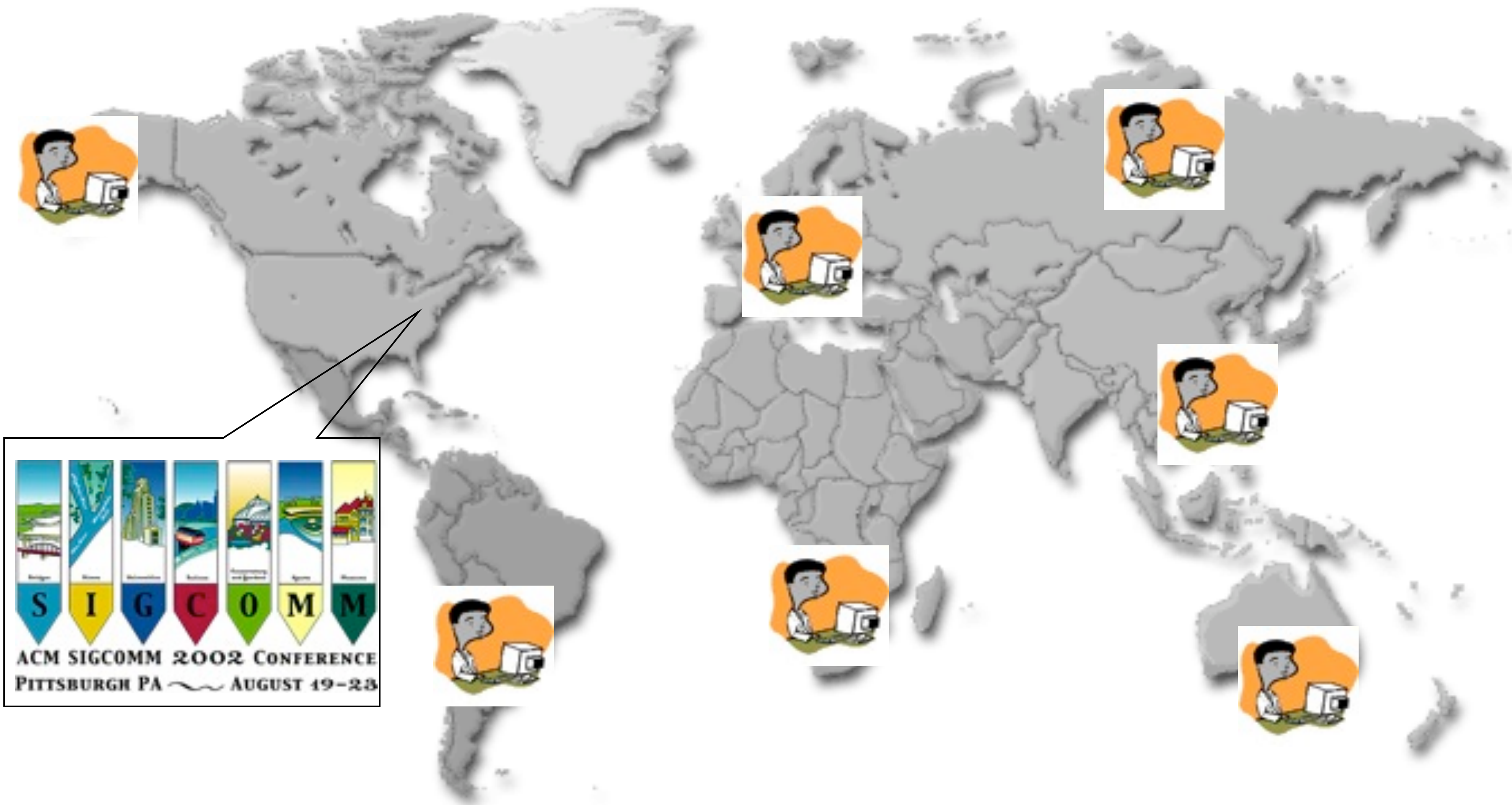
- Scalability of all these network measurements!
  - Overhead
  - Interference of measurements?
  - What if everyone has his/her own overlay network doing this?
- Stability of the network? Oscillation? Keep rerouting back and forth?
- How much can you really gain?
  - In delay/bandwidth, may not be that much
  - But is much faster to react to complete link failures than BGP

Multicast Overlay

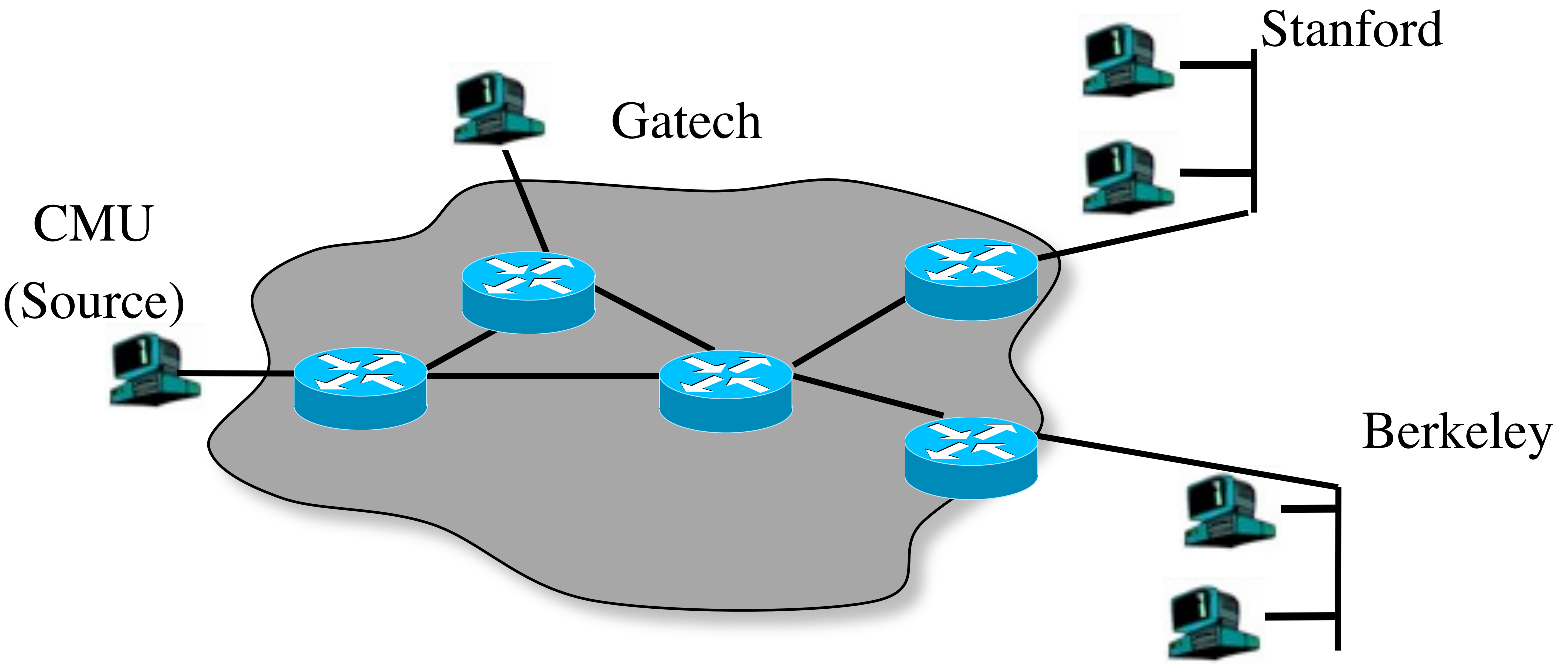
- IP multicast supposed to provide one-to-many packet delivery
- IP multicast routers supposed to maintain group membership, duplicate packets appropriately and send to all members

- Why "supposed"? In the Internet today, we have none of that

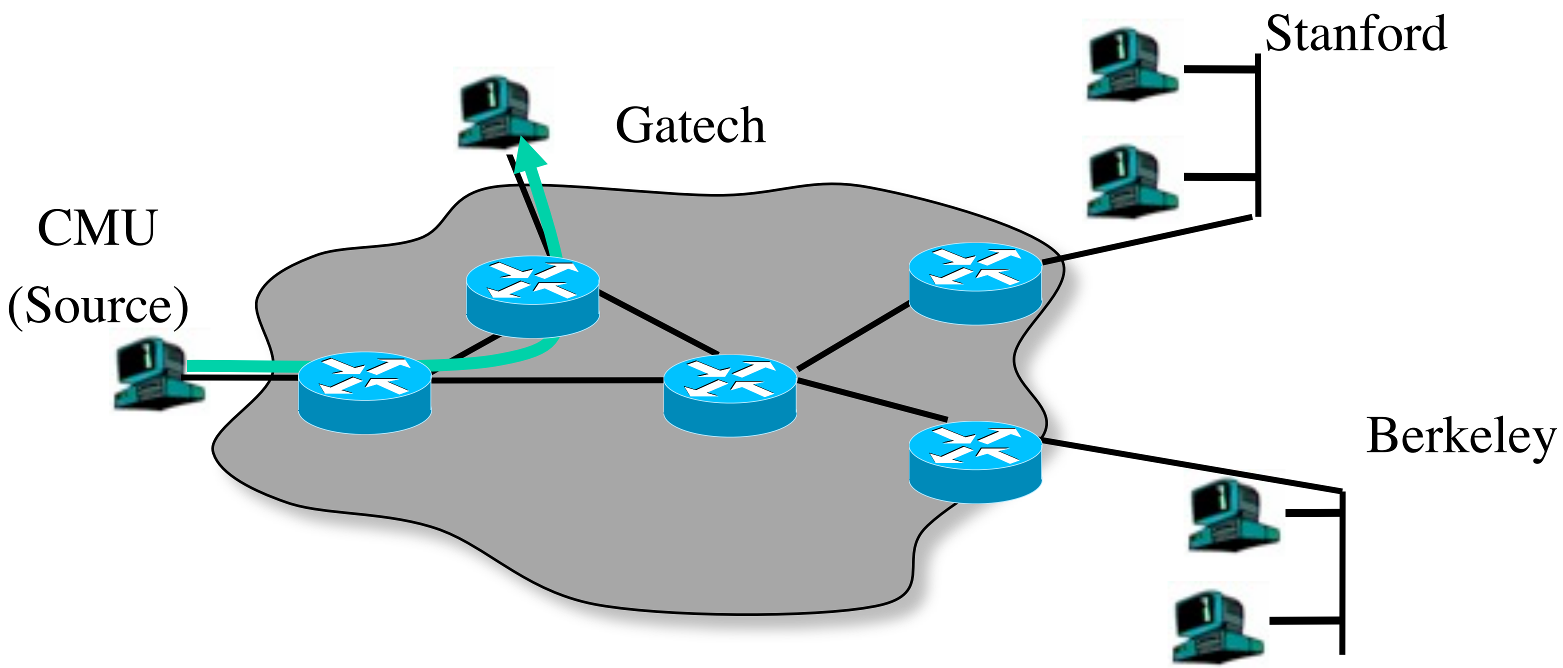
Motivating Example: Conference Attendance



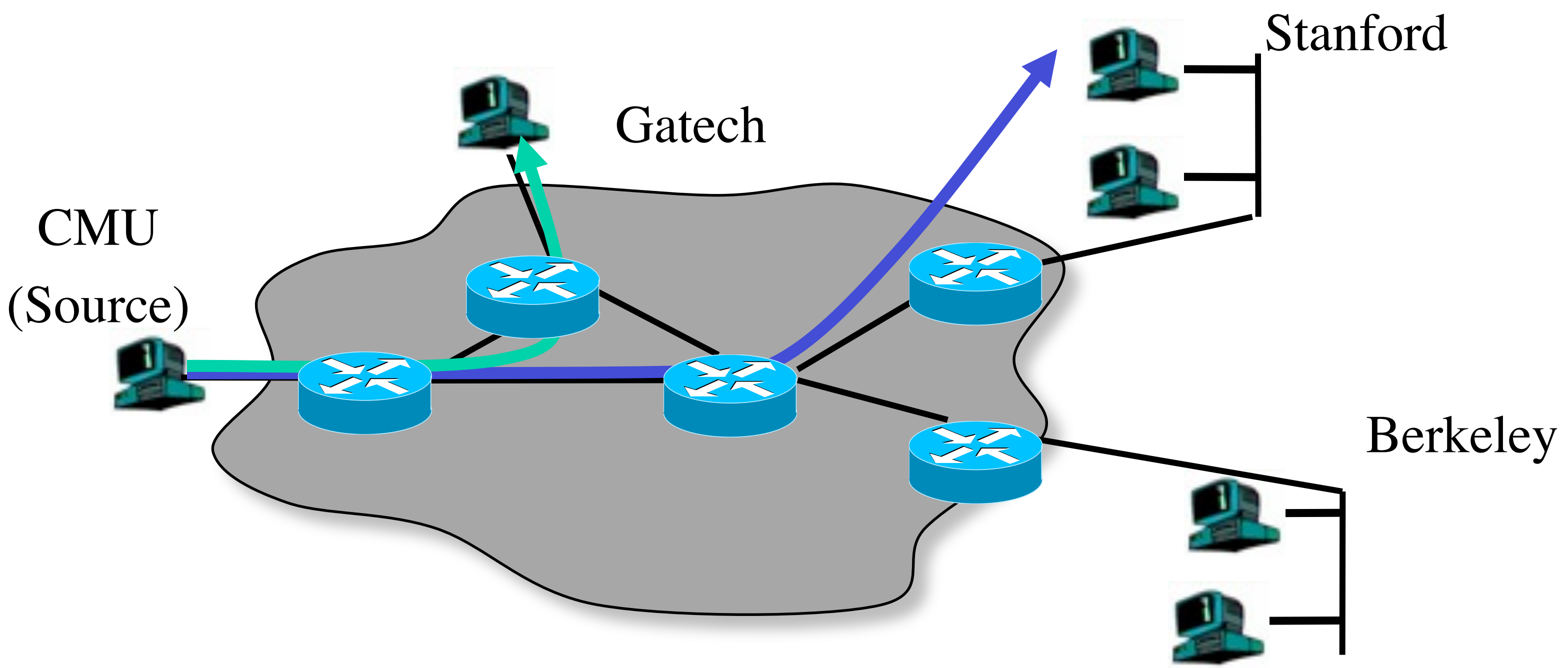
Solution based on Unicast



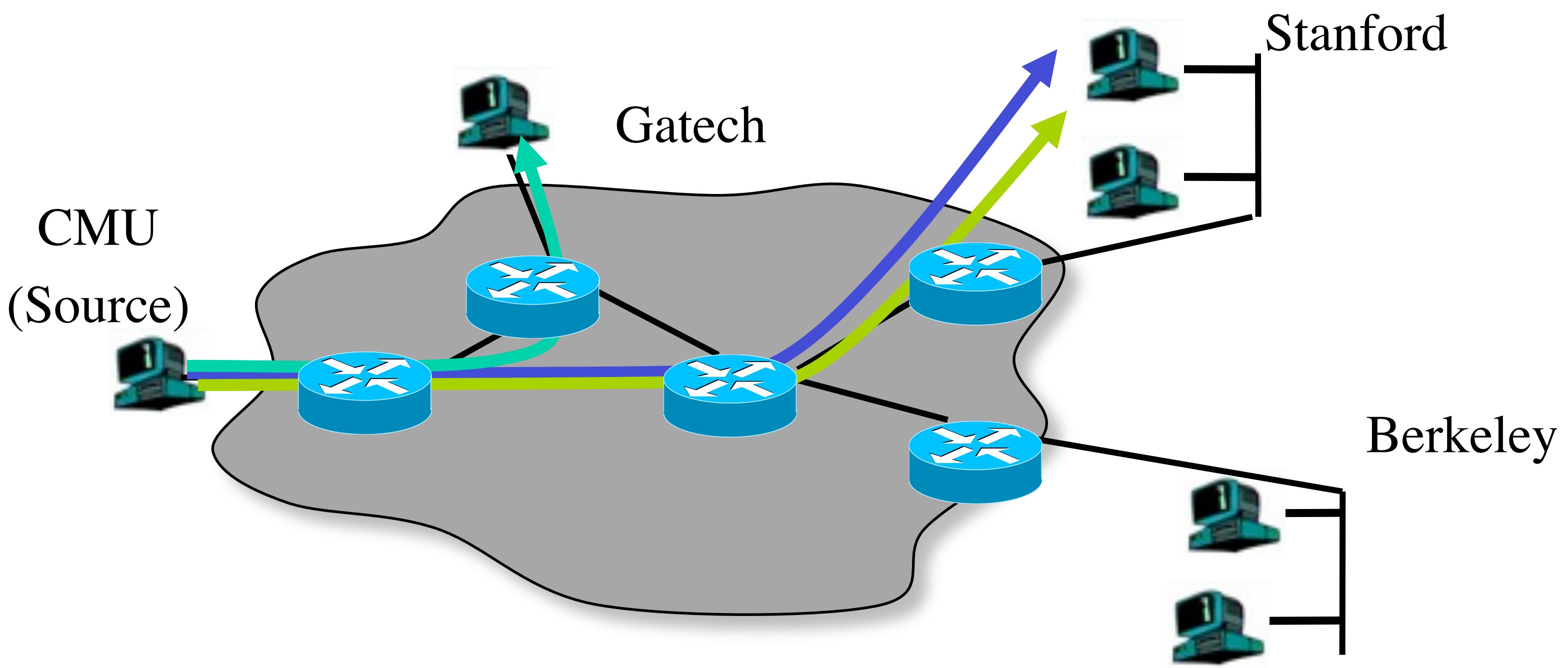
Solution based on Unicast



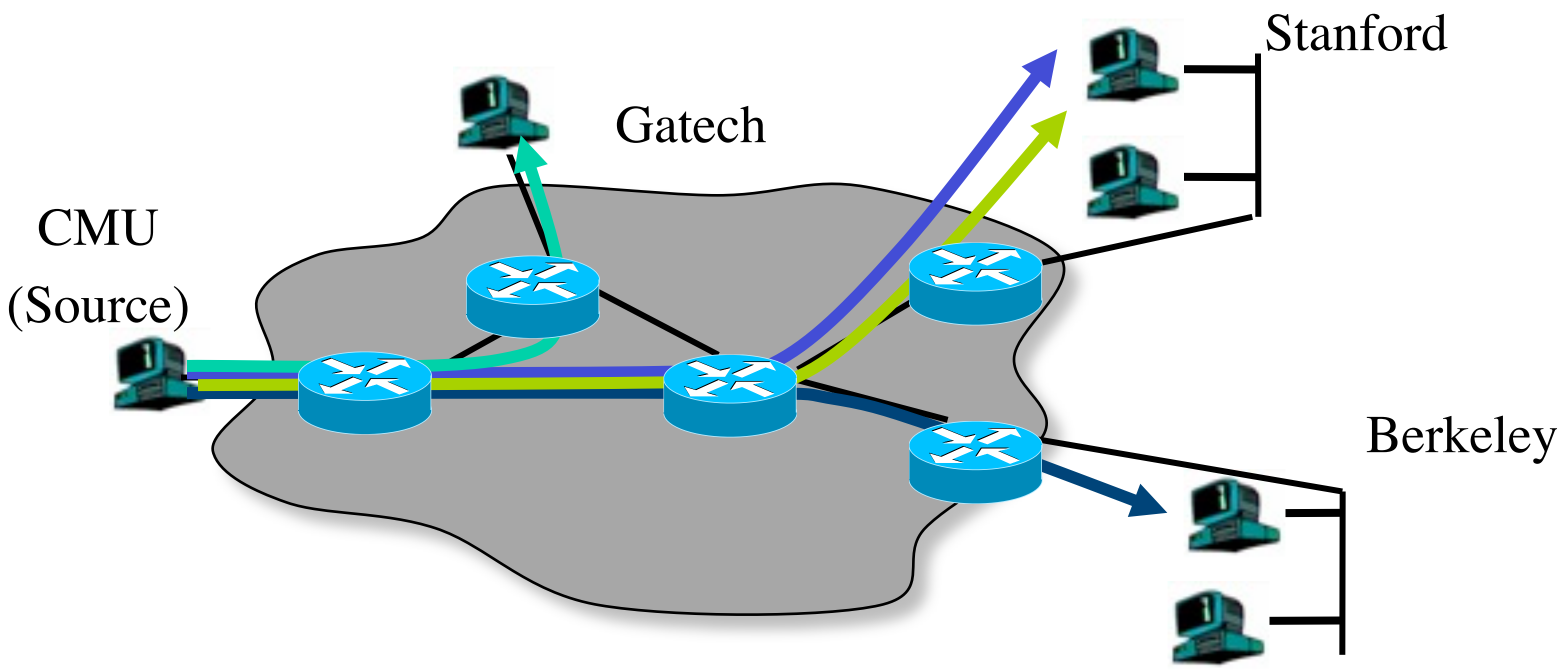
Solution based on Unicast



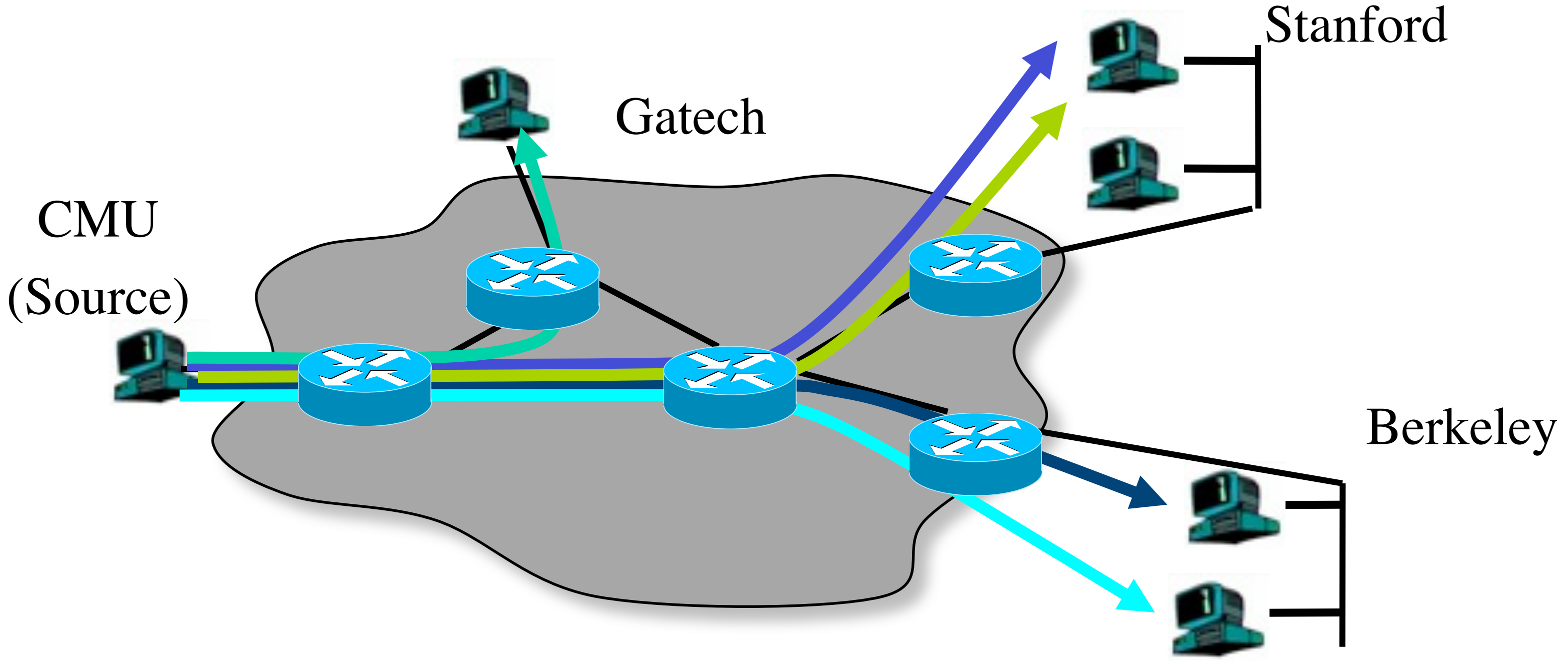
Solution based on Unicast



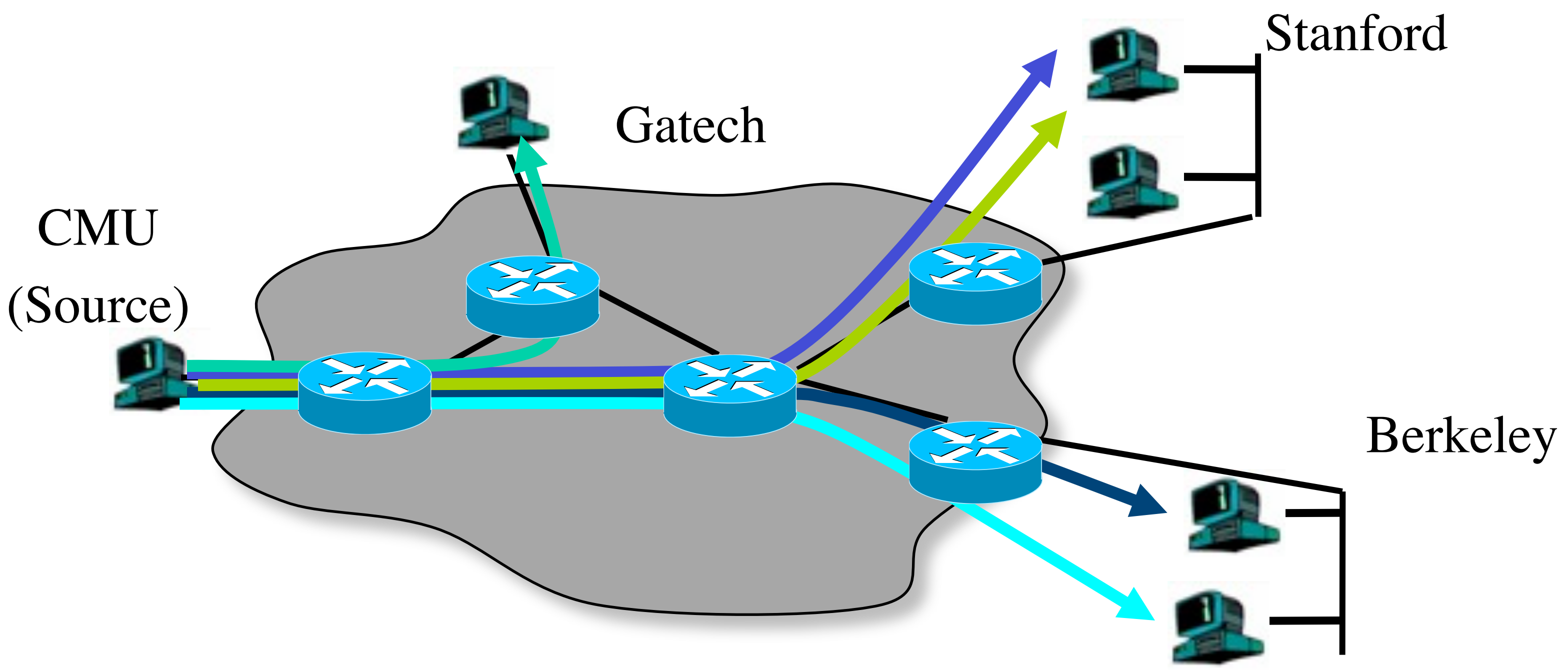
Solution based on Unicast



Solution based on Unicast

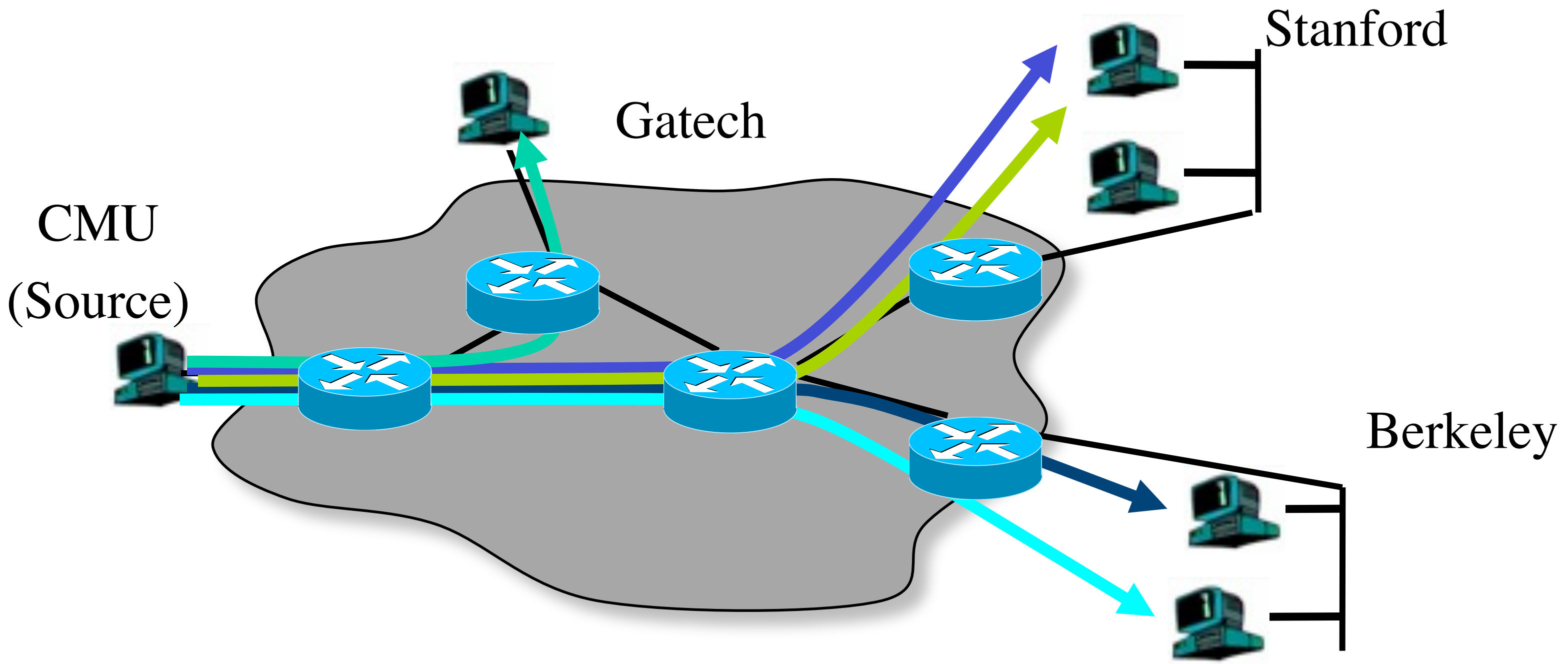


Solution based on Unicast



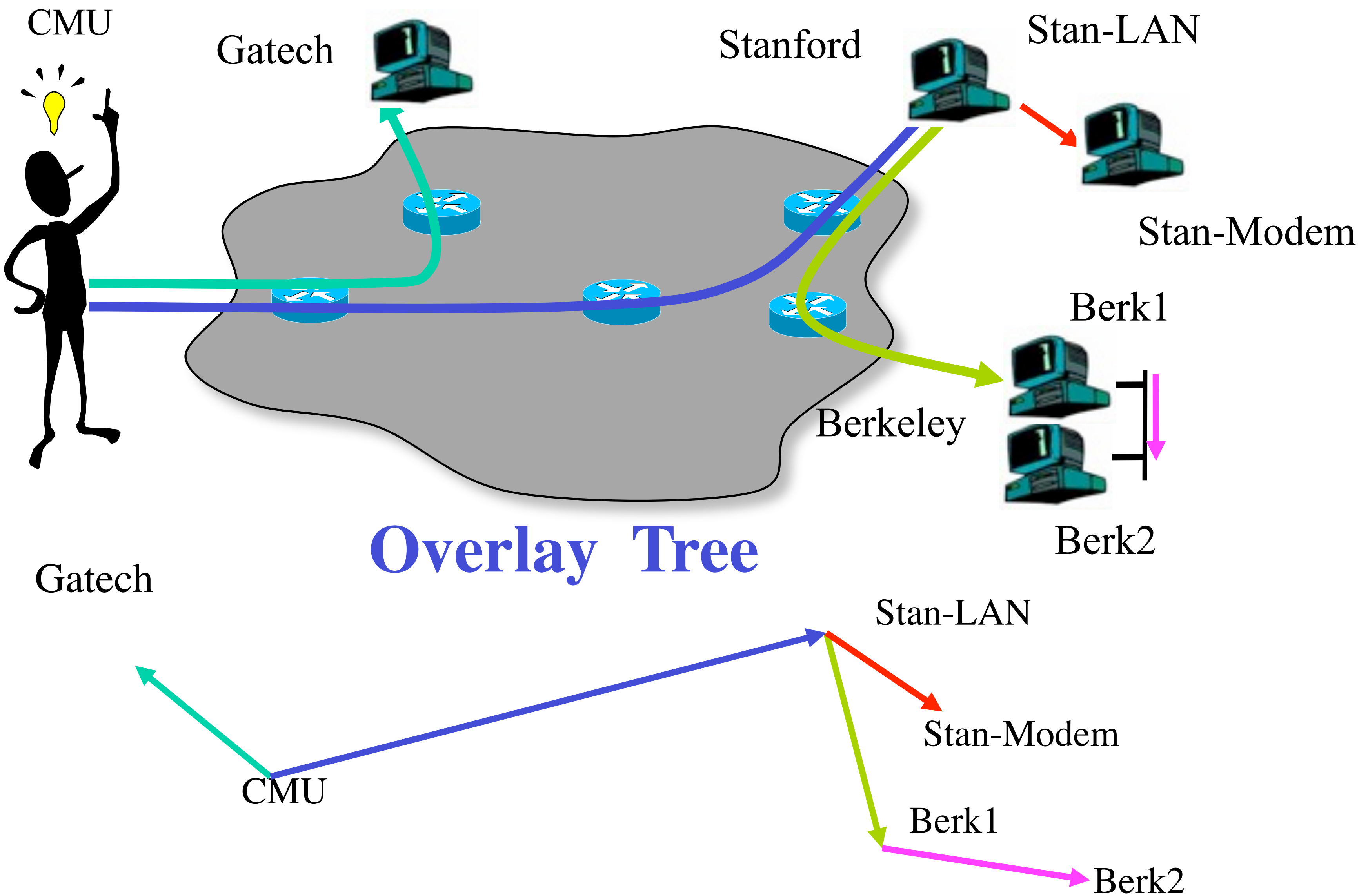
- Client-server architecture (the Web)

Solution based on Unicast



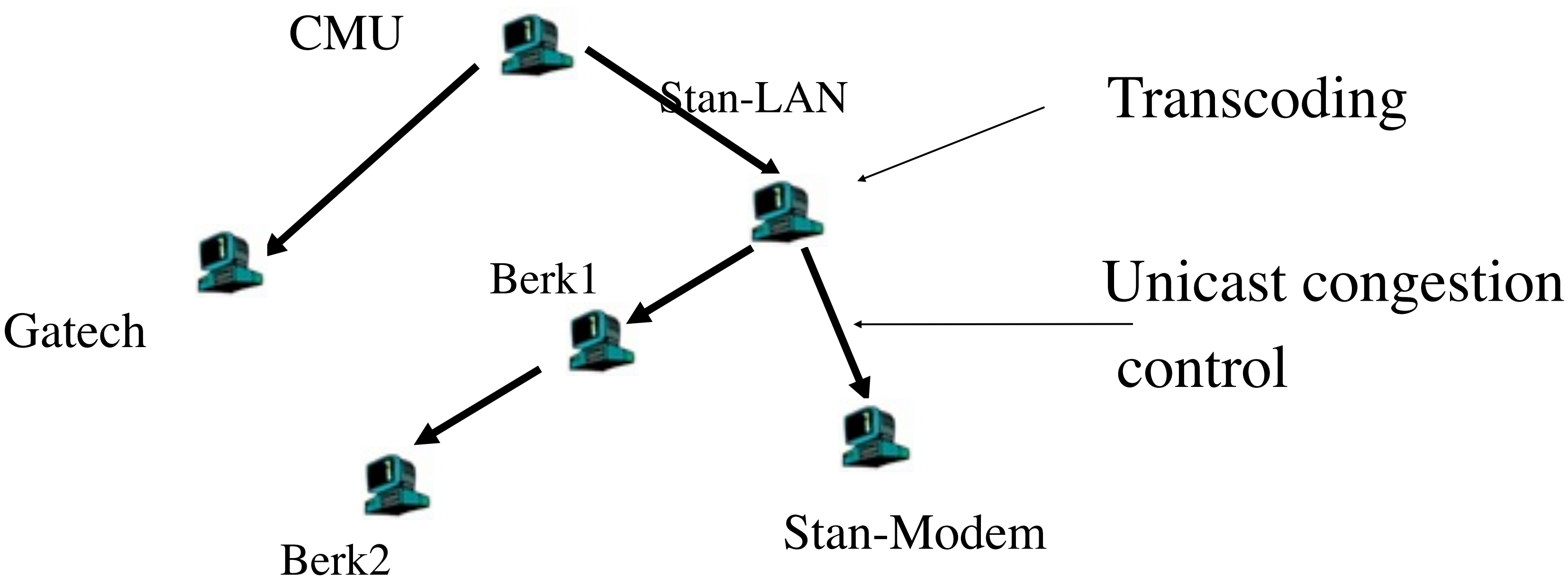
- Client-server architecture (the Web)
- Does not scale well with group size
  - Source host is the bottleneck

End System Multicast



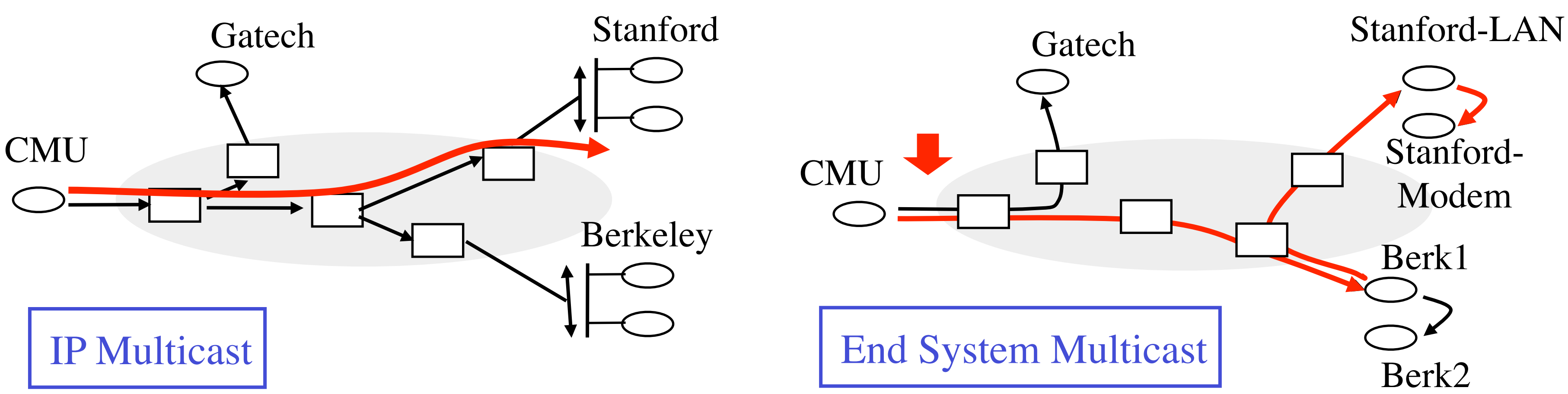
End System Multicast: Benefits

- Scalability
  - Routers do not maintain per-group state
- Easy to deploy
  - Works over the existing IP infrastructure
- Can simplify support for higher level functionality

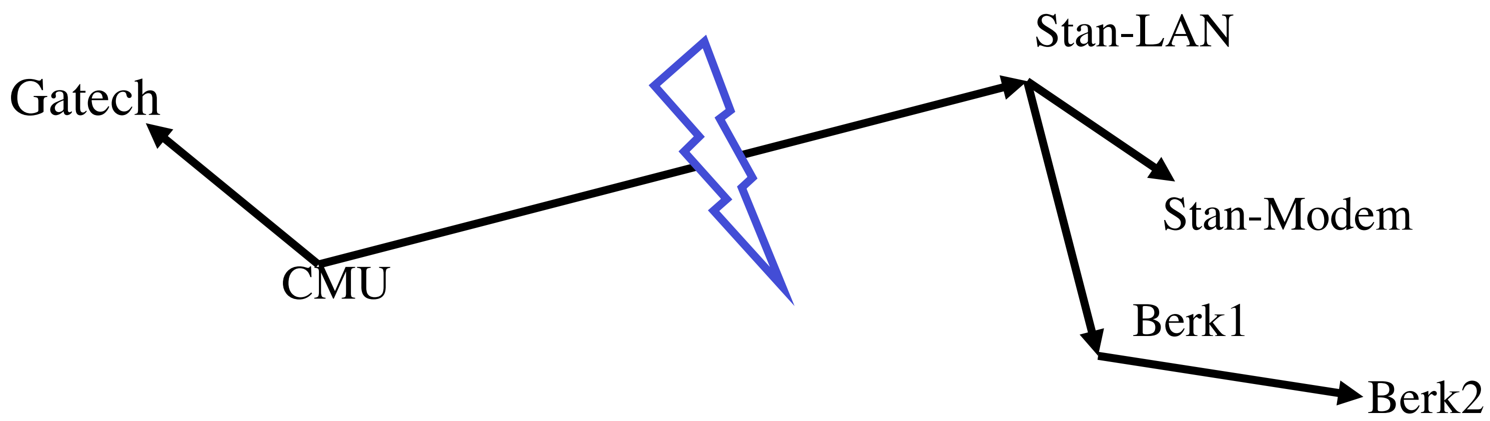


Concerns with End System Multicast

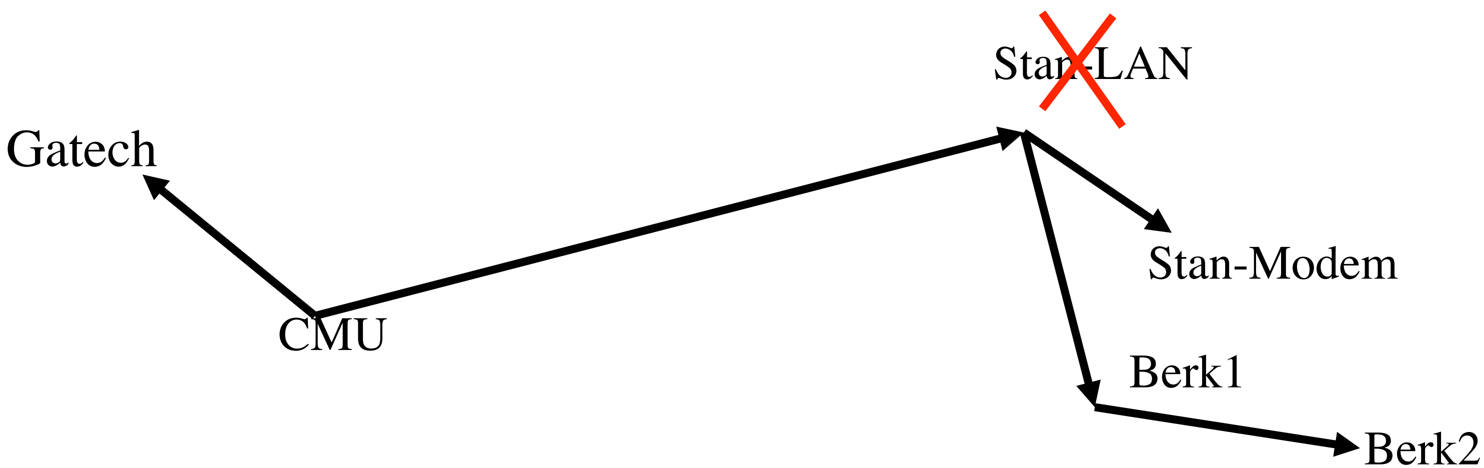
- Challenge to construct efficient overlay trees
- Performance concerns compared to IP Multicast
  - Increase in delay
  - Bandwidth waste (packet duplication)



More Challenges

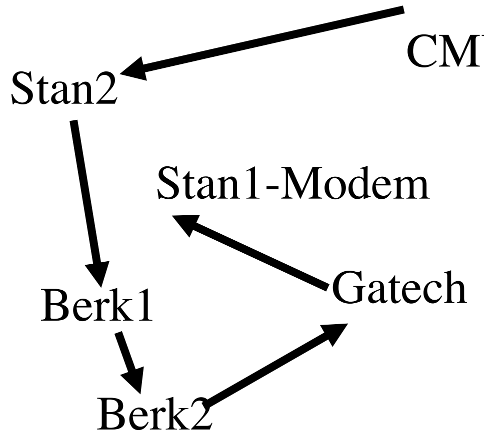


Overlays must adapt to network dynamics and congestion

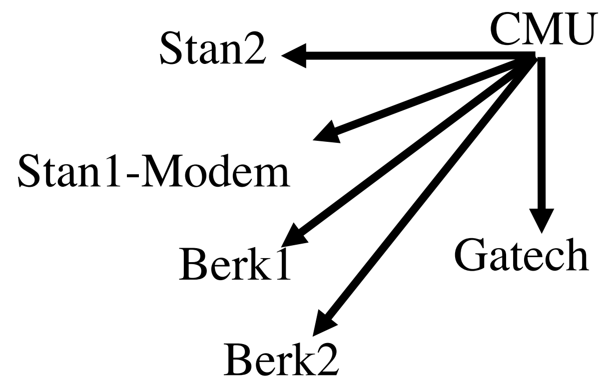


Group membership is dynamic: members can join and leave

Inefficient Overlay Trees

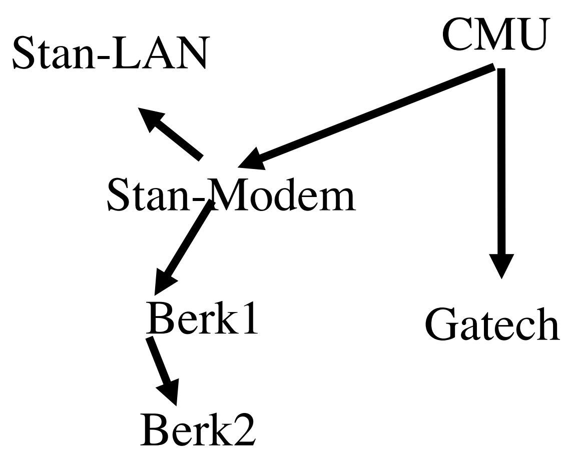


High latency



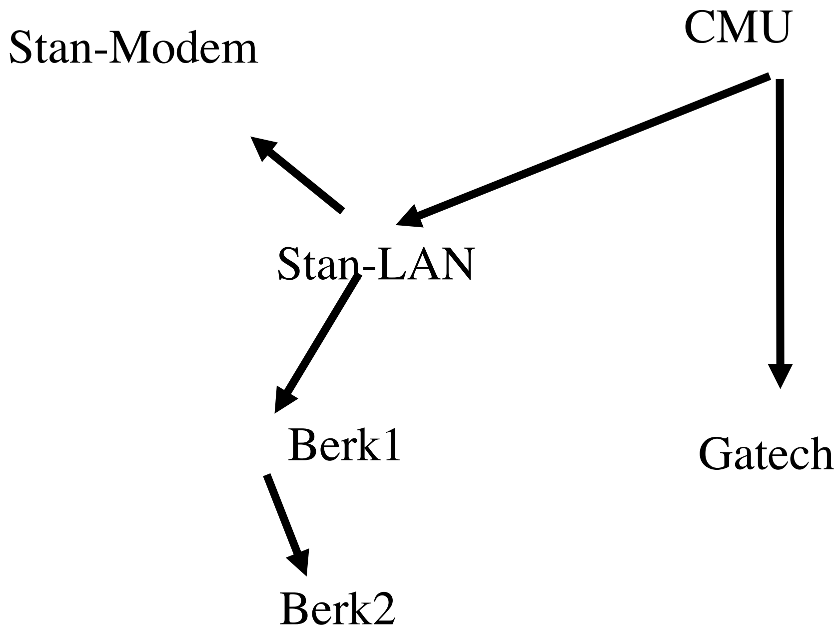
-Poor network usage

-Potential congestion near CMU



Poor bandwidth to members

An Efficient Overlay Tree



End System Multicast System

- Focus on video broadcast applications
- Implementation
  - Integrate with Apple QuickTime
  - Support for receiver heterogeneity
  - Support peers behind NAT and firewall
  - Run on Windows and Linux platforms
- Showcase
  - SIGCOMM (max 60 simultaneous users)
  - Several CMU Distinguished Lectures
  - Slashdot (max 180 simultaneous users)

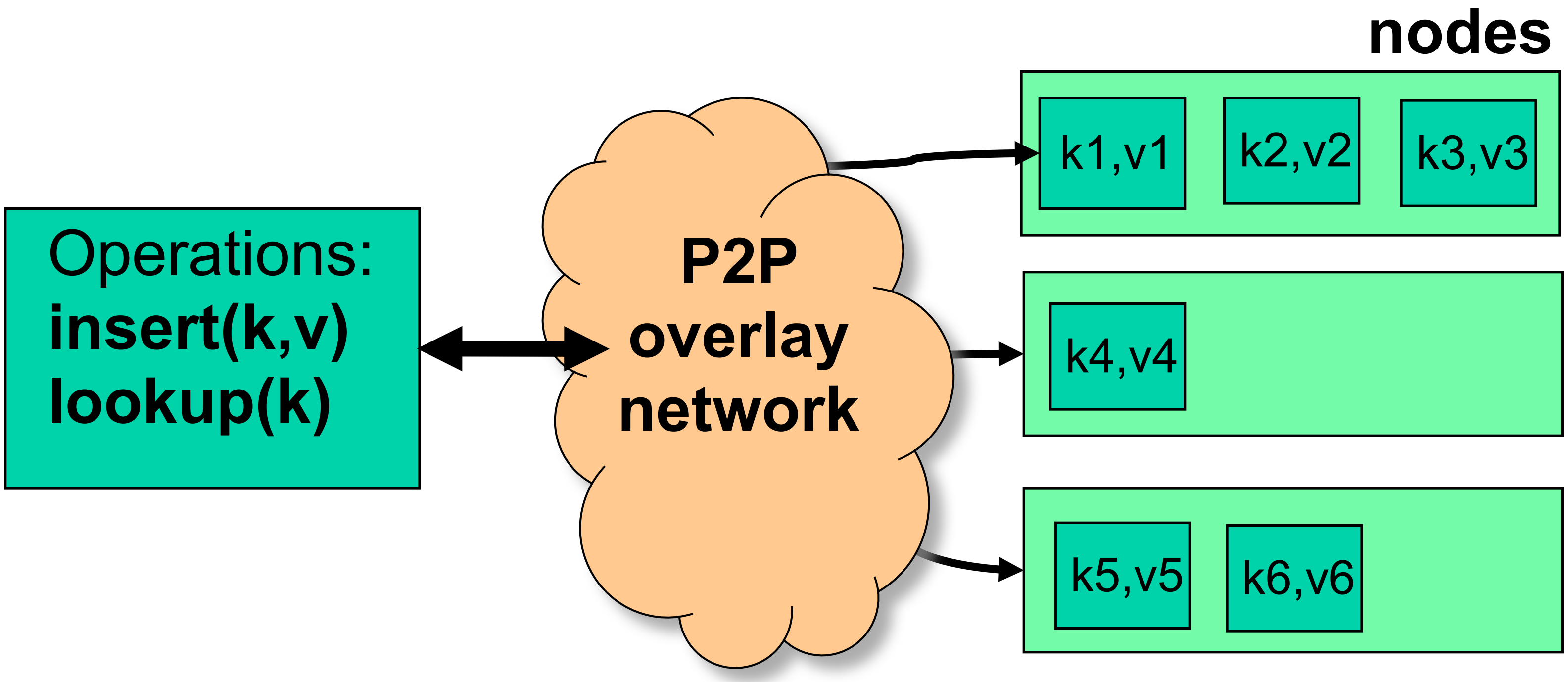
Structured p2p overlays

One primitive:

route(M, X): route message *M* to the live node with *nodeId* closest to key *X*

- nodeIds and keys are from a large, sparse id space

Distributed Hash Tables (DHT)

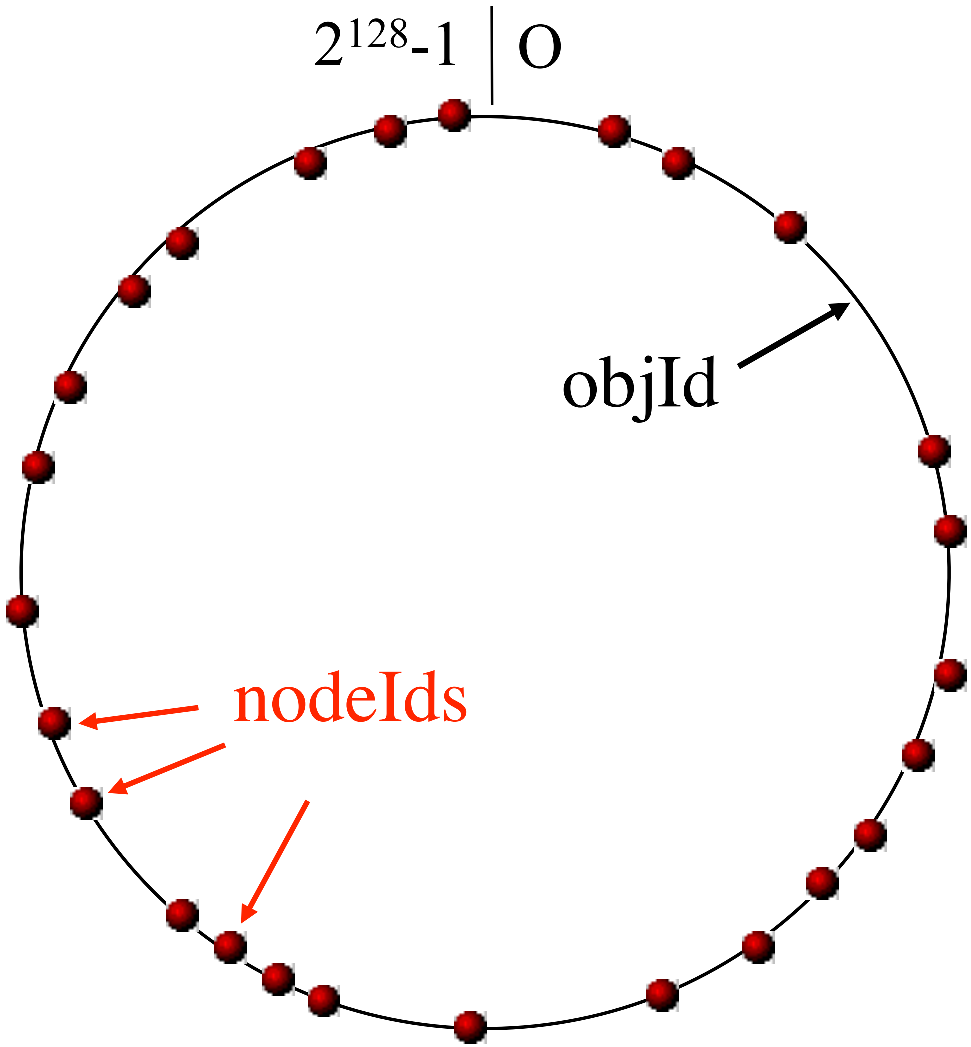


- p2p overlay maps keys to nodes
- completely decentralized and self-organizing
- robust, scalable

Why structured p2p overlays?

- Leverage pooled resources (storage, bandwidth, CPU)
- Leverage resource diversity (geographic, ownership)
- Leverage existing shared infrastructure
- Scalability
- Robustness
- Self-organization

Pastry: Object distribution



Consistent hashing [*Karger et al. '97*]

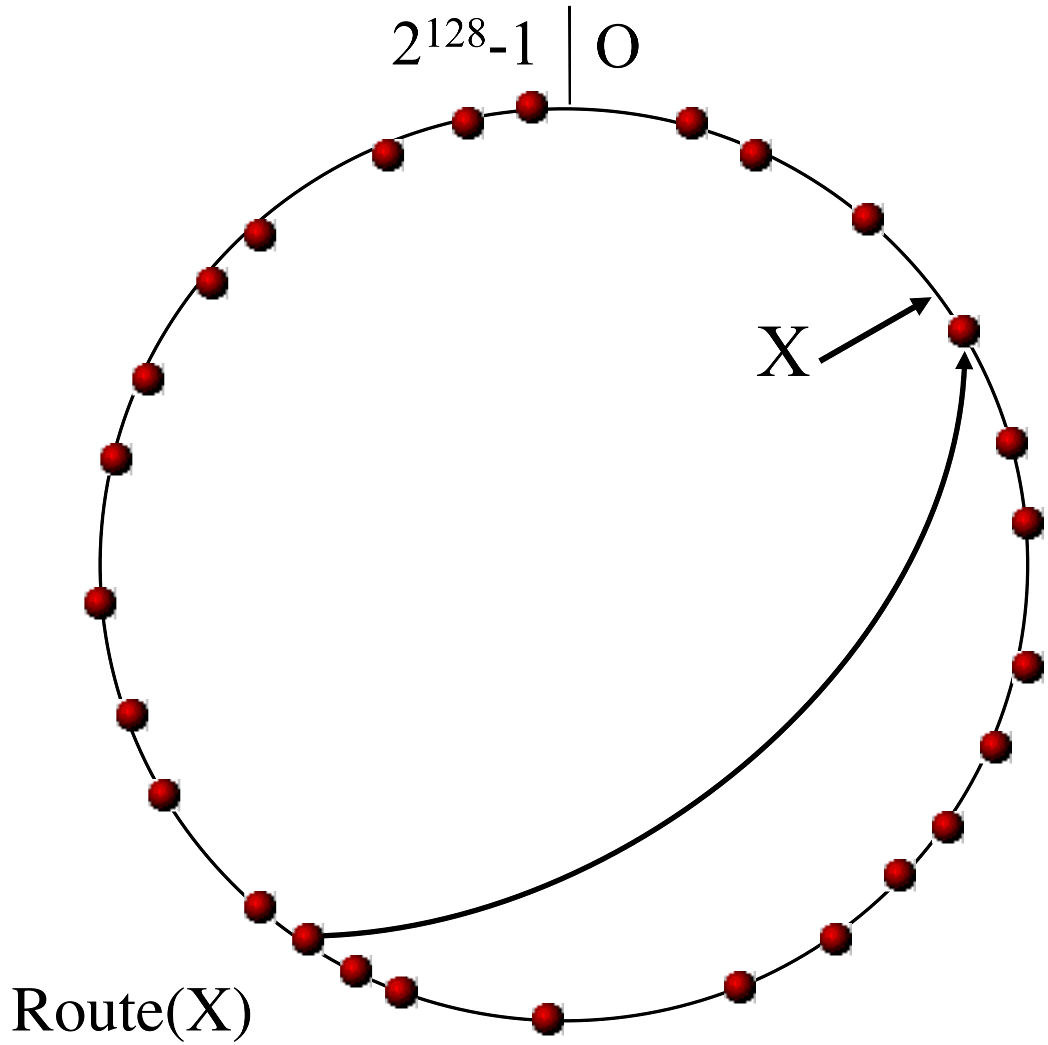
128 bit circular id space

nodeIds (uniform random)

objIds (uniform random)

Invariant: node with numerically closest nodeId maintains object

Pastry: Object insertion/lookup



Msg with key *X* is routed to live node with nodeId closest to X

Problem:
complete routing table not feasible

Pastry: Routing

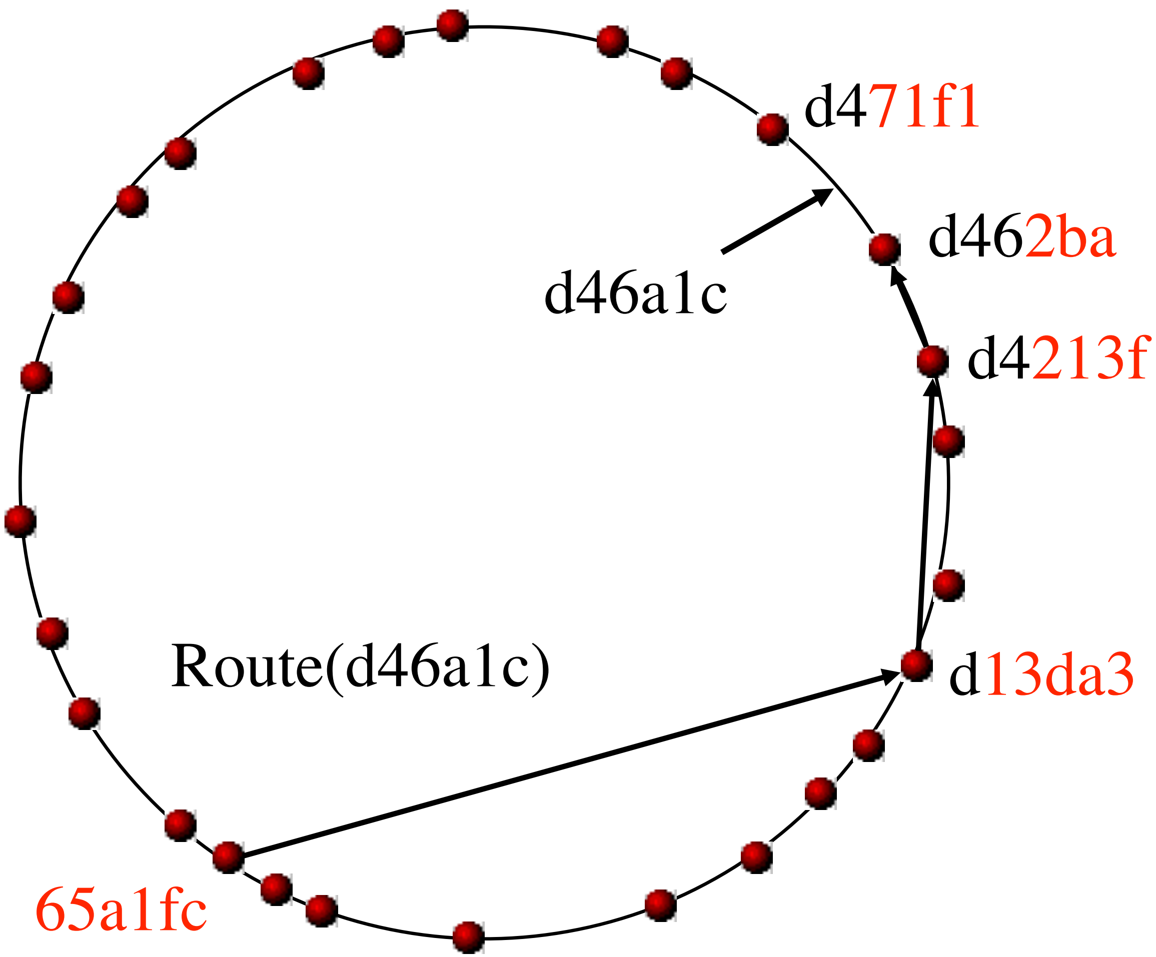
Tradeoff

- O(*log N*) routing table size
- O(*log N*) message forwarding steps

Pastry: Routing table (# 65a1fc*x*)

Row 0	*0x*	*1x*	*2x*	*3x*	*4x*	*5x*		*7x*	*8x*	*9x*	*ax*	*bx*	*cx*	*dx*	*ex*	*fx*
Row 1	*60x*	*61x*	*62x*	*63x*	*64x*		*66x*	*67x*	*68x*	*69x*	*6ax*	*6bx*	*6cx*	*6dx*	*6ex*	*6fx*
Row 2	*650x*	*651x*	*652x*	*653x*	*654x*	*655x*	*656x*	*657x*	*658x*	*659x*		*65bx*	*65cx*	*65dx*	*65ex*	*65fx*
Row 3 $\log_{16} N$ rows	*65a0x*		*65a2x*	*65a3x*	*65a4x*	*65a5x*	*65a6x*	*65a7x*	*65a8x*	*65a9x*	*65aax*	*65abx*	*65acx*	*65adx*	*65aex*	*65afx*

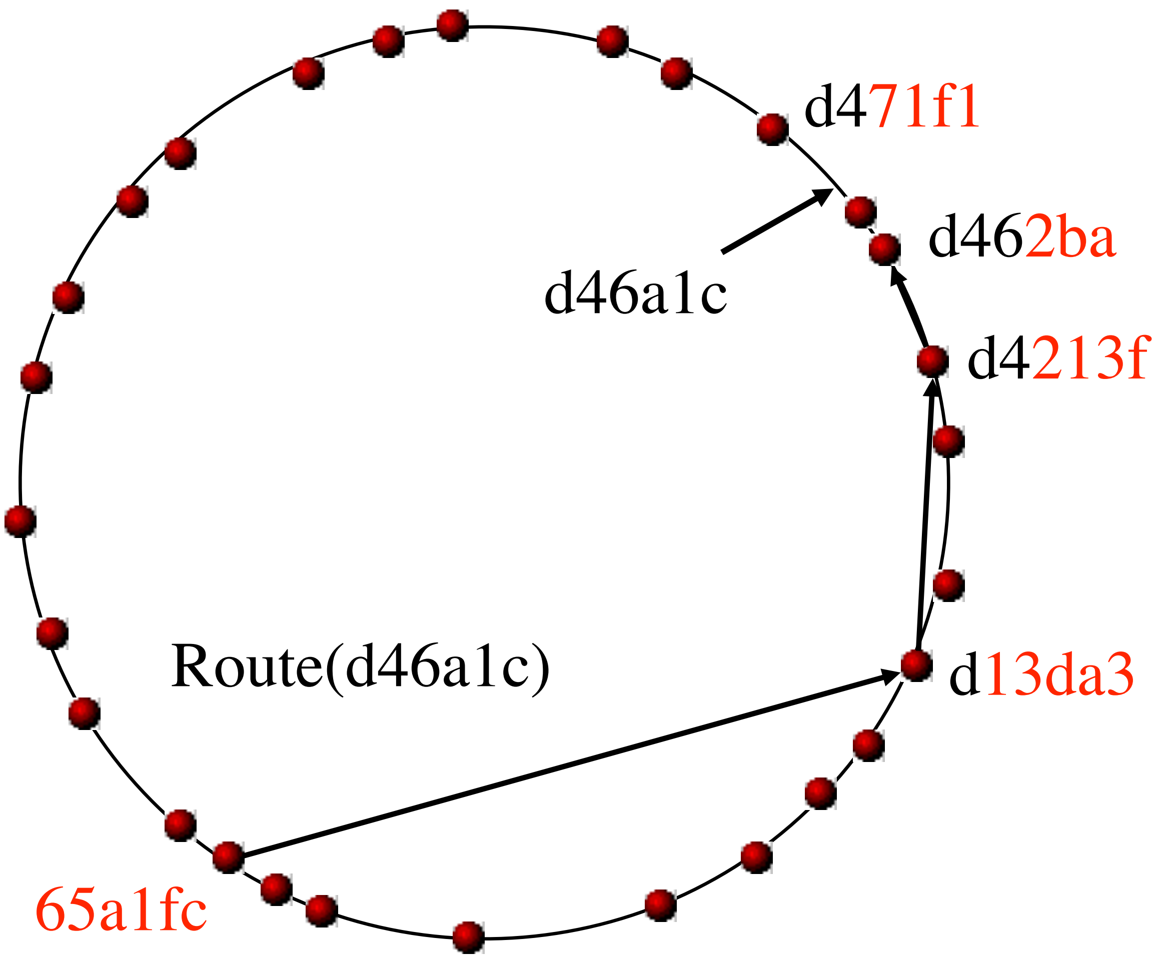
Pastry: Routing



Properties

- $\log_{16} N$ steps
- $O(\log N)$ state

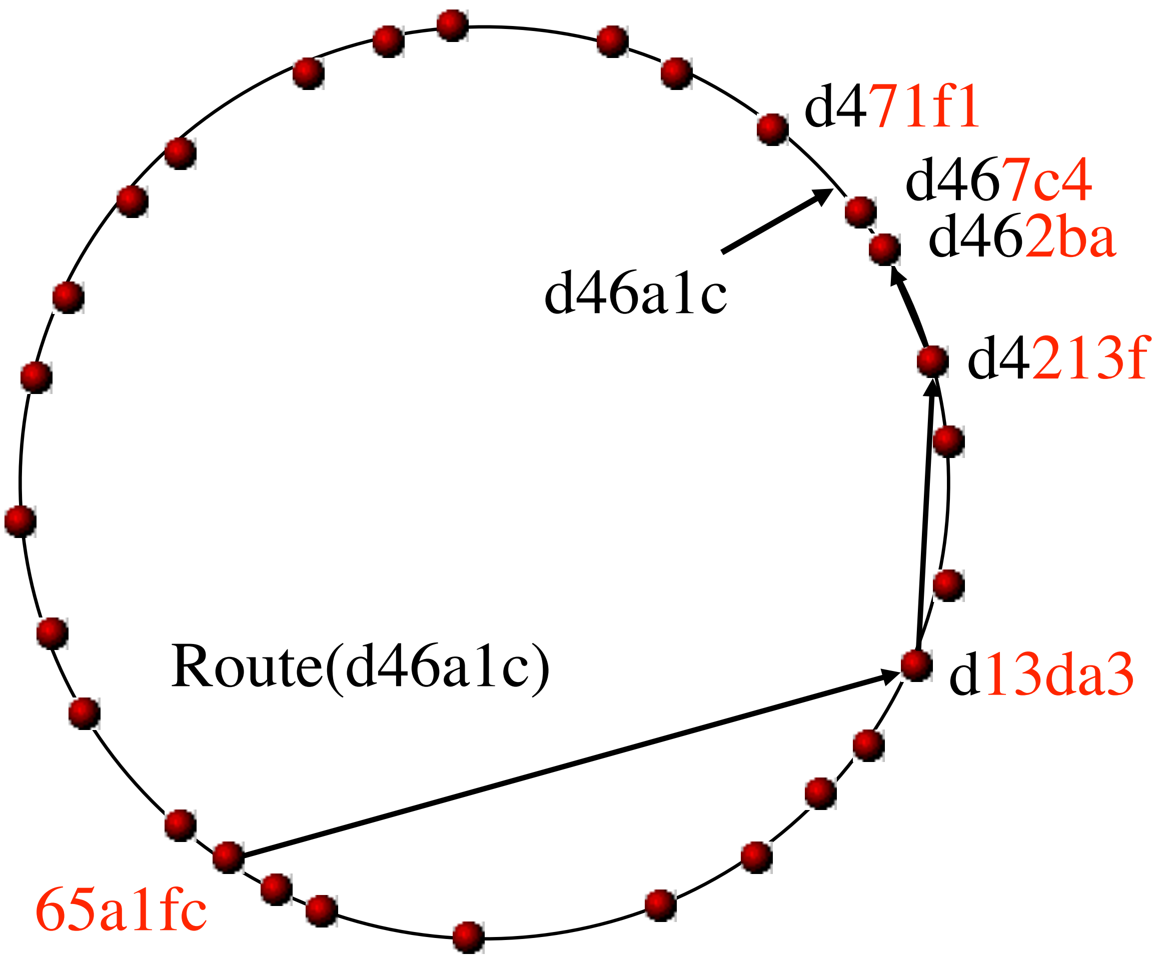
Pastry: Routing



Properties

- $\log_{16} N$ steps
- $O(\log N)$ state

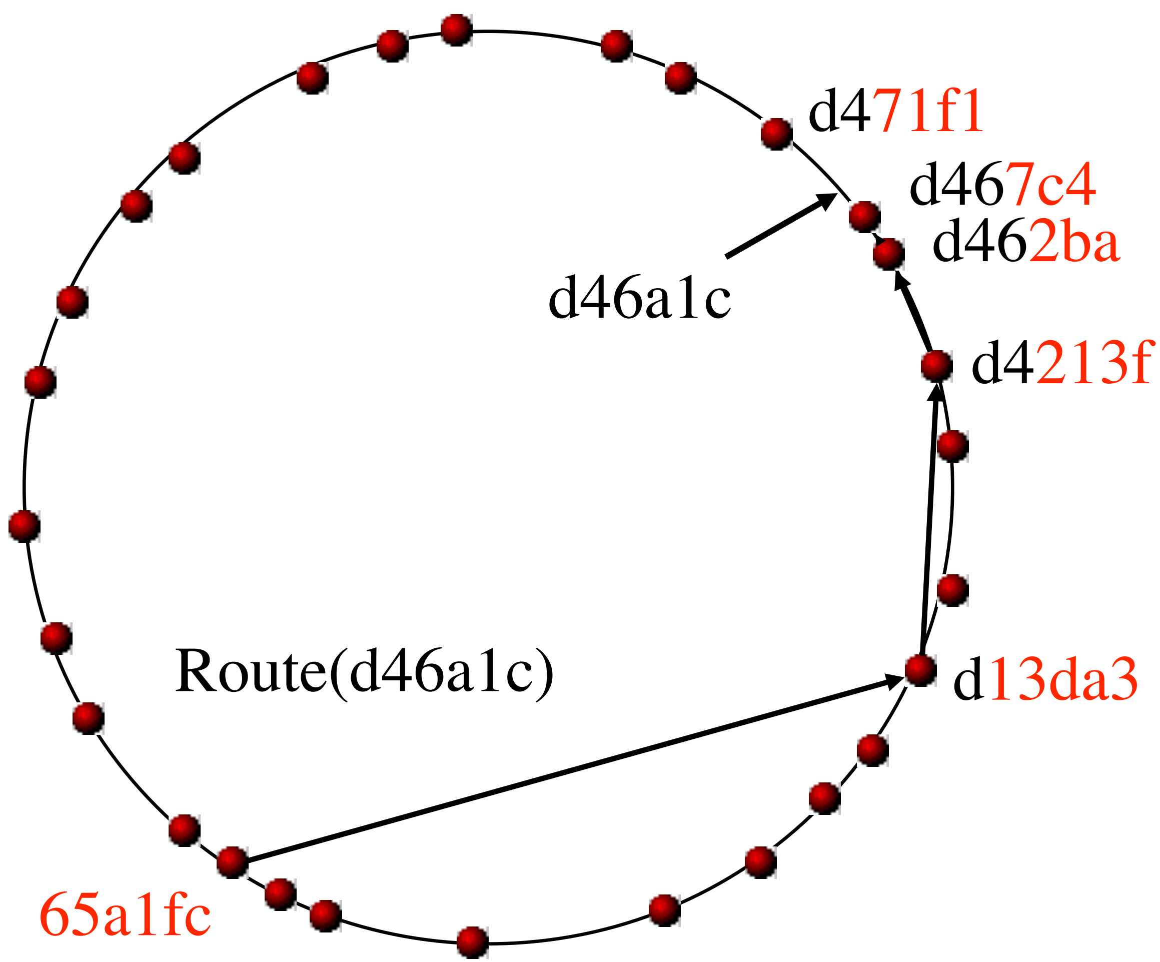
Pastry: Routing



Properties

- $\log_{16} N$ steps
- $O(\log N)$ state

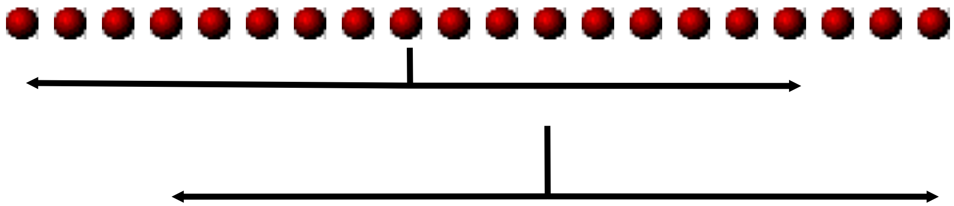
Pastry: Routing



Properties

- $\log_{16}$ N steps
- O(*log N*) state

Pastry: Leaf sets



Each node maintains IP addresses of the nodes with the L/2 numerically closest larger and smaller nodeIds, respectively.

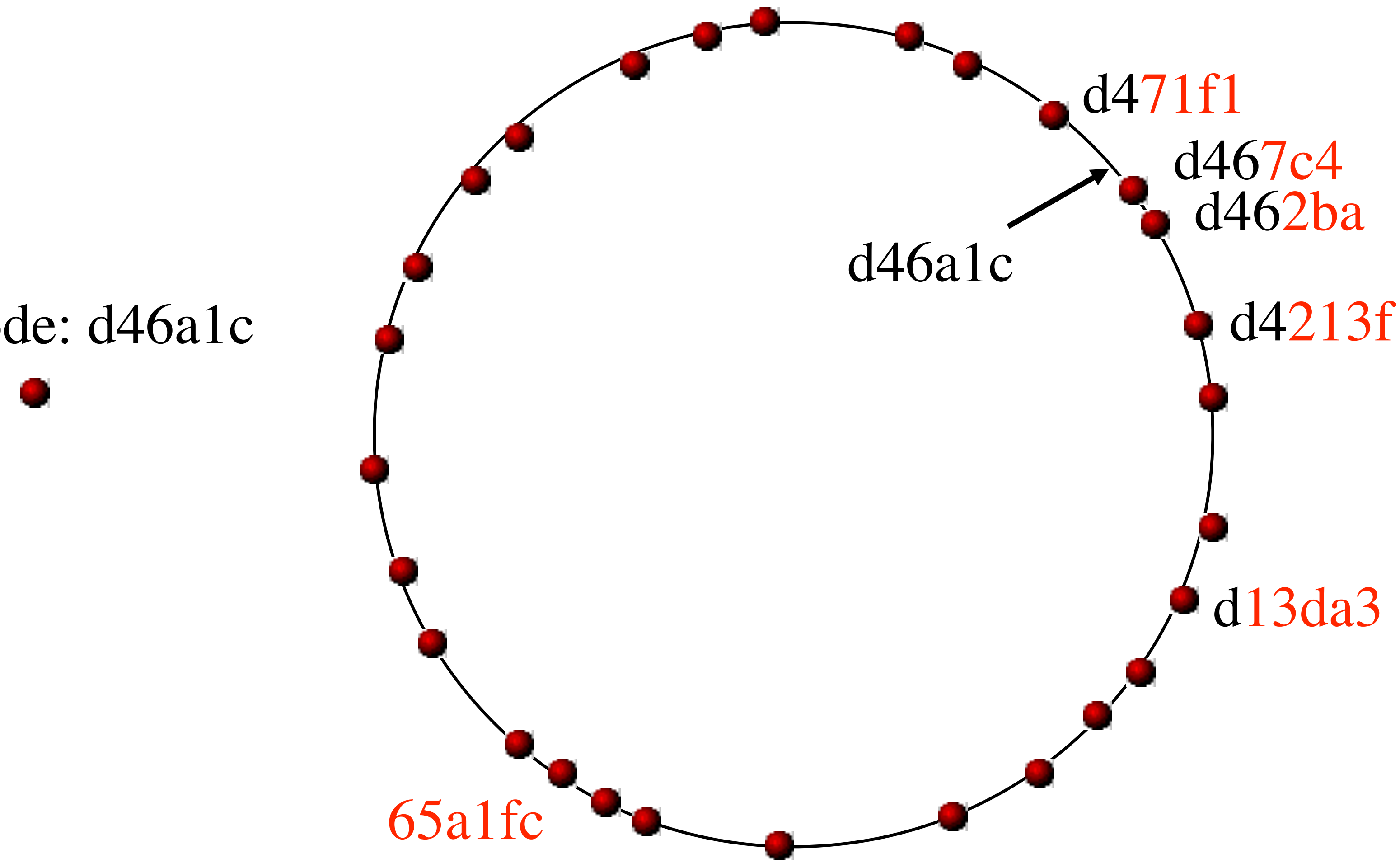
- routing efficiency/robustness
- fault detection (keep-alive)
- application-specific local coordination

Pastry: Routing procedure

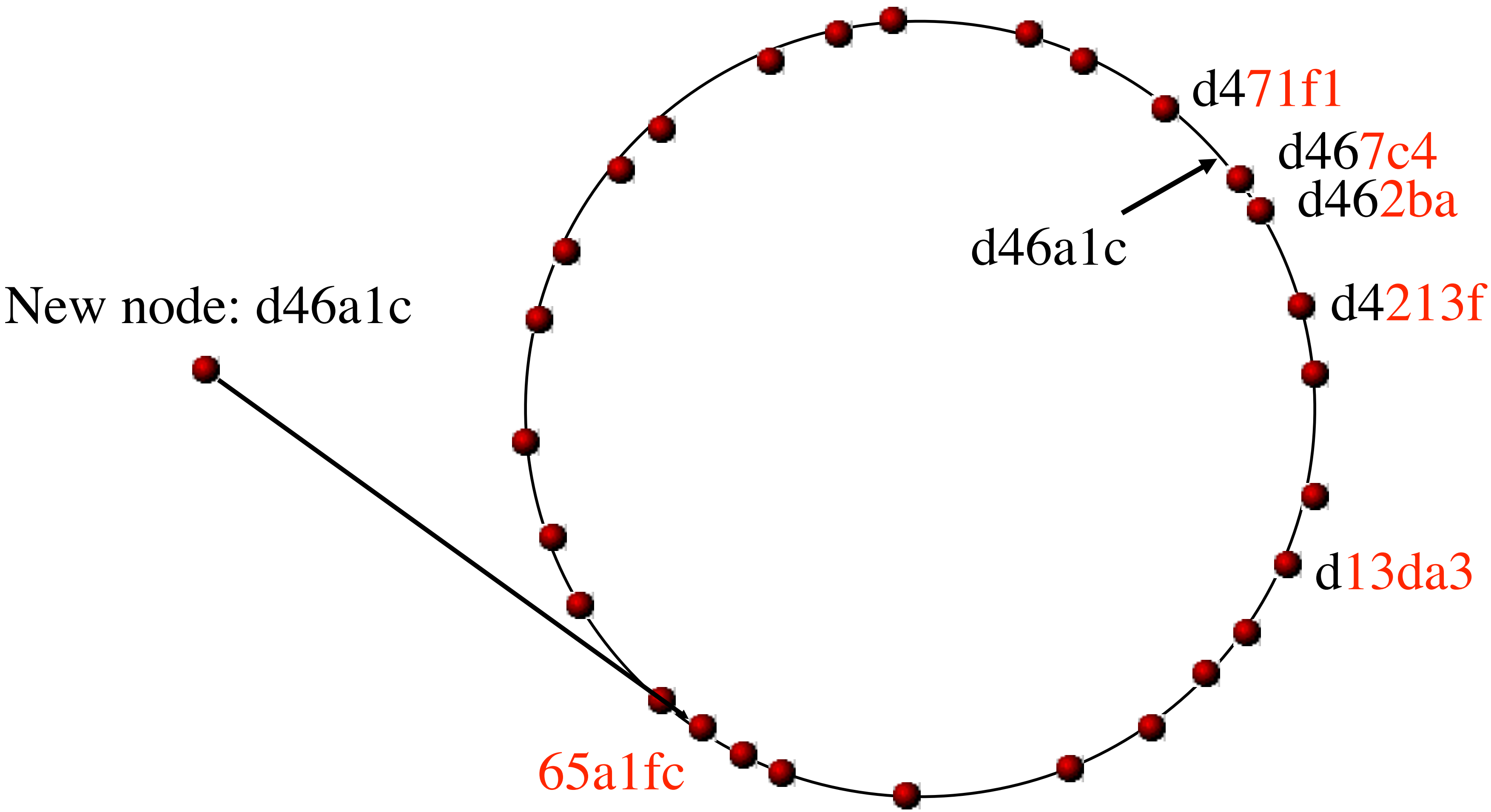
if (destination is within range of our leaf set)
    forward to numerically closest member
else
    let l = length of shared prefix
    let d = value of l-th digit in D's address
    if (R_l^d exists)
        forward to R_l^d
    else
        forward to a known node that
        (a) shares at least as long a prefix
        (b) is numerically closer than this node

Pastry: Node addition

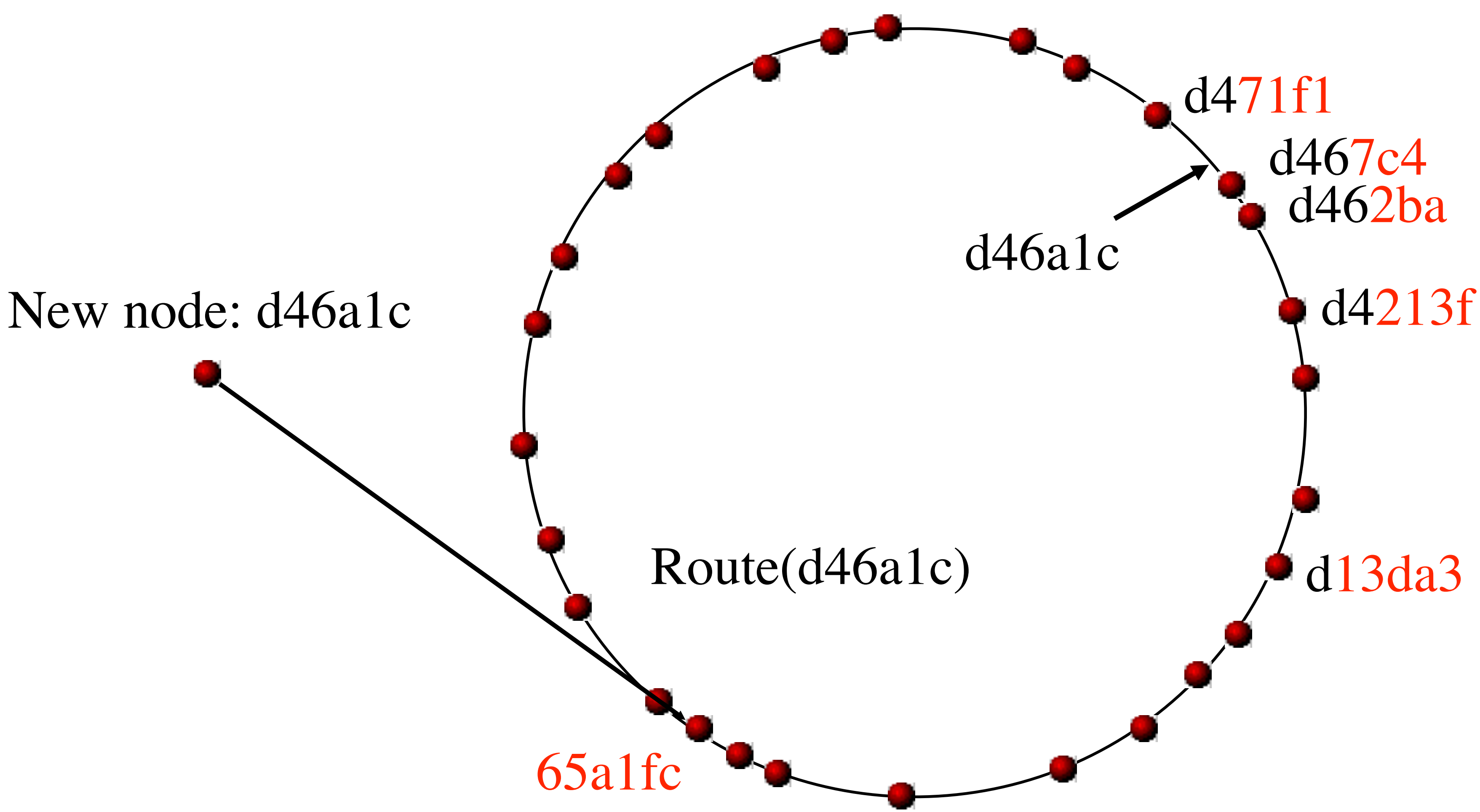
New node: d46a1c



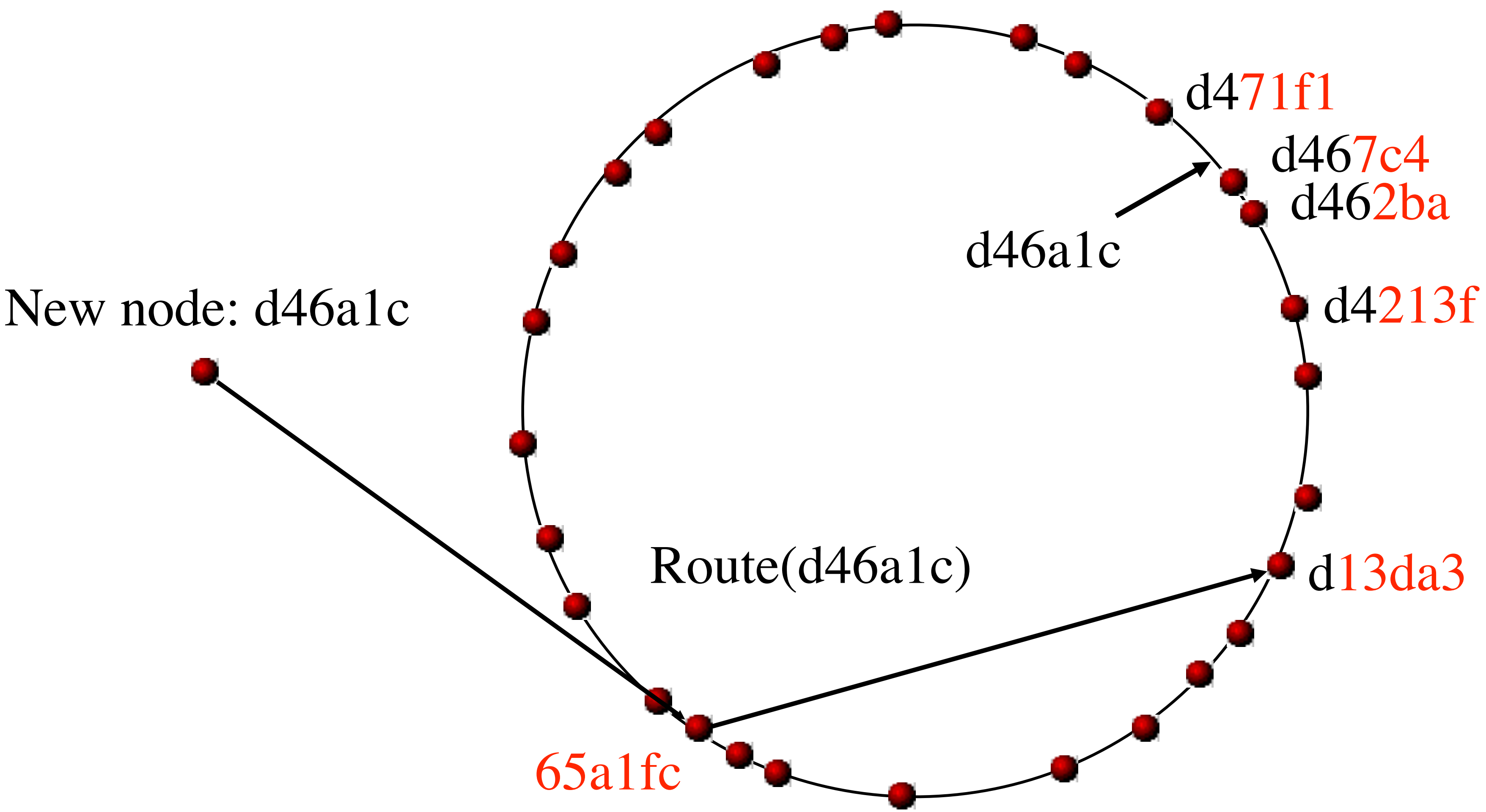
Pastry: Node addition



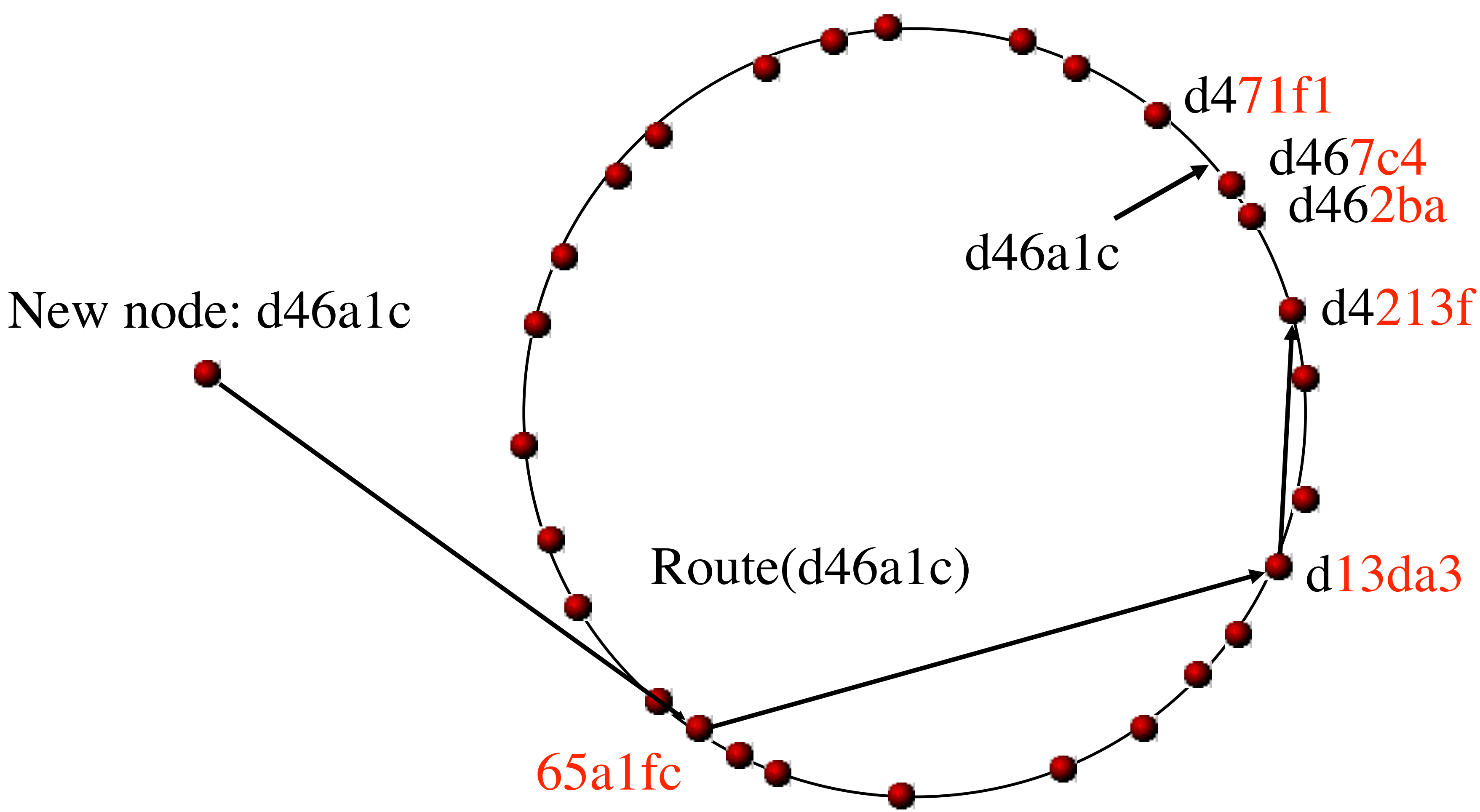
Pastry: Node addition



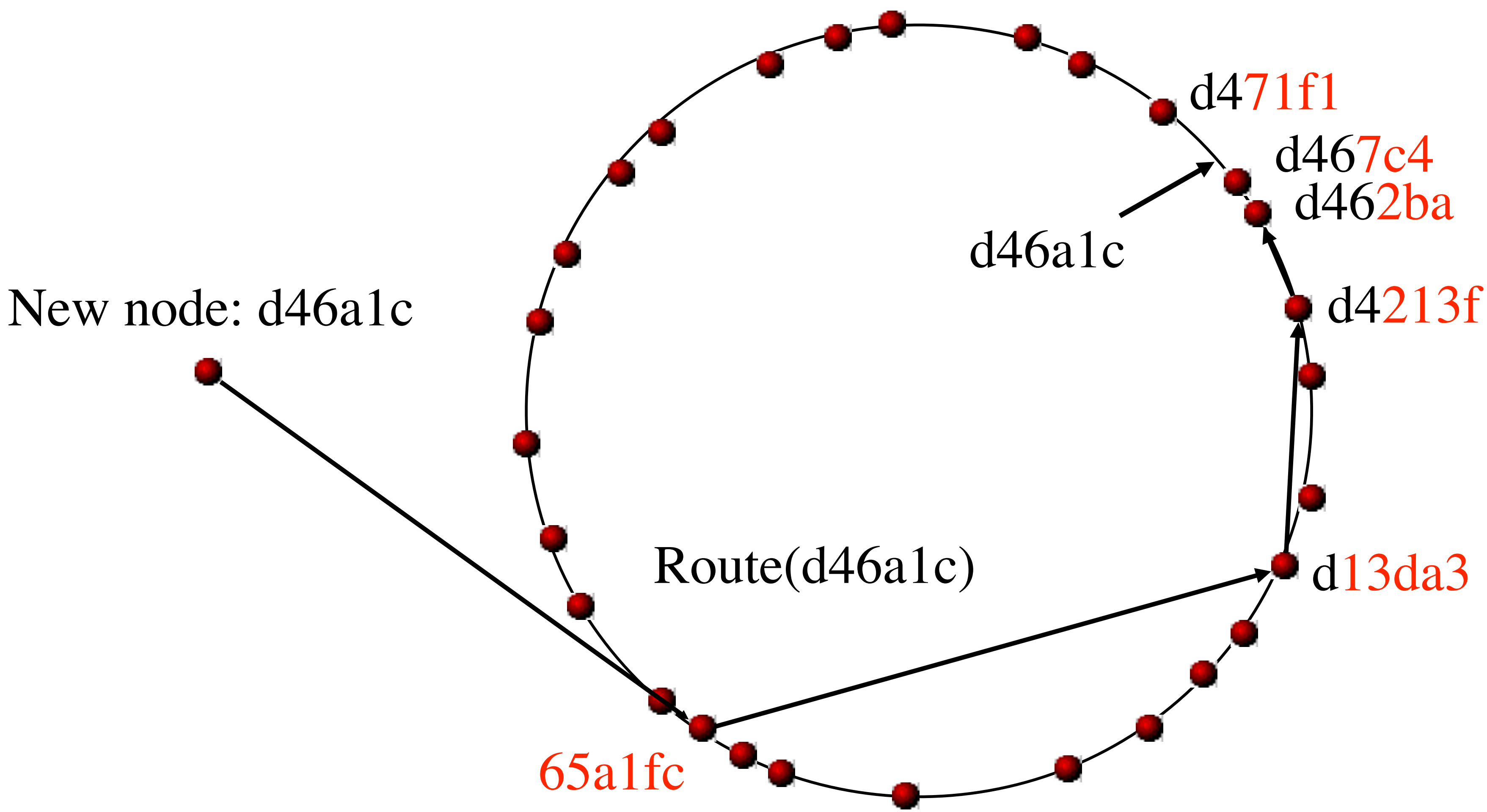
Pastry: Node addition



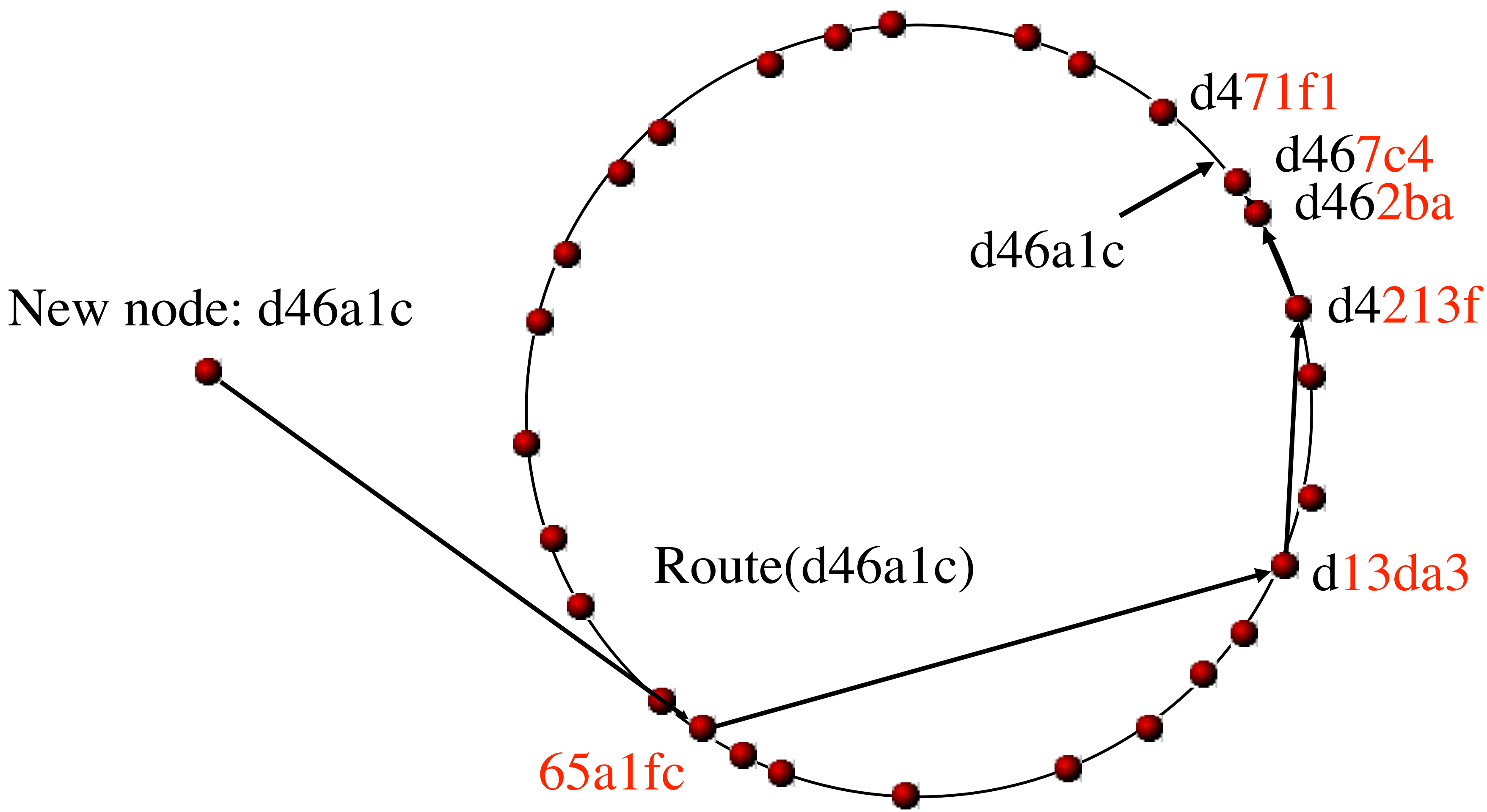
Pastry: Node addition



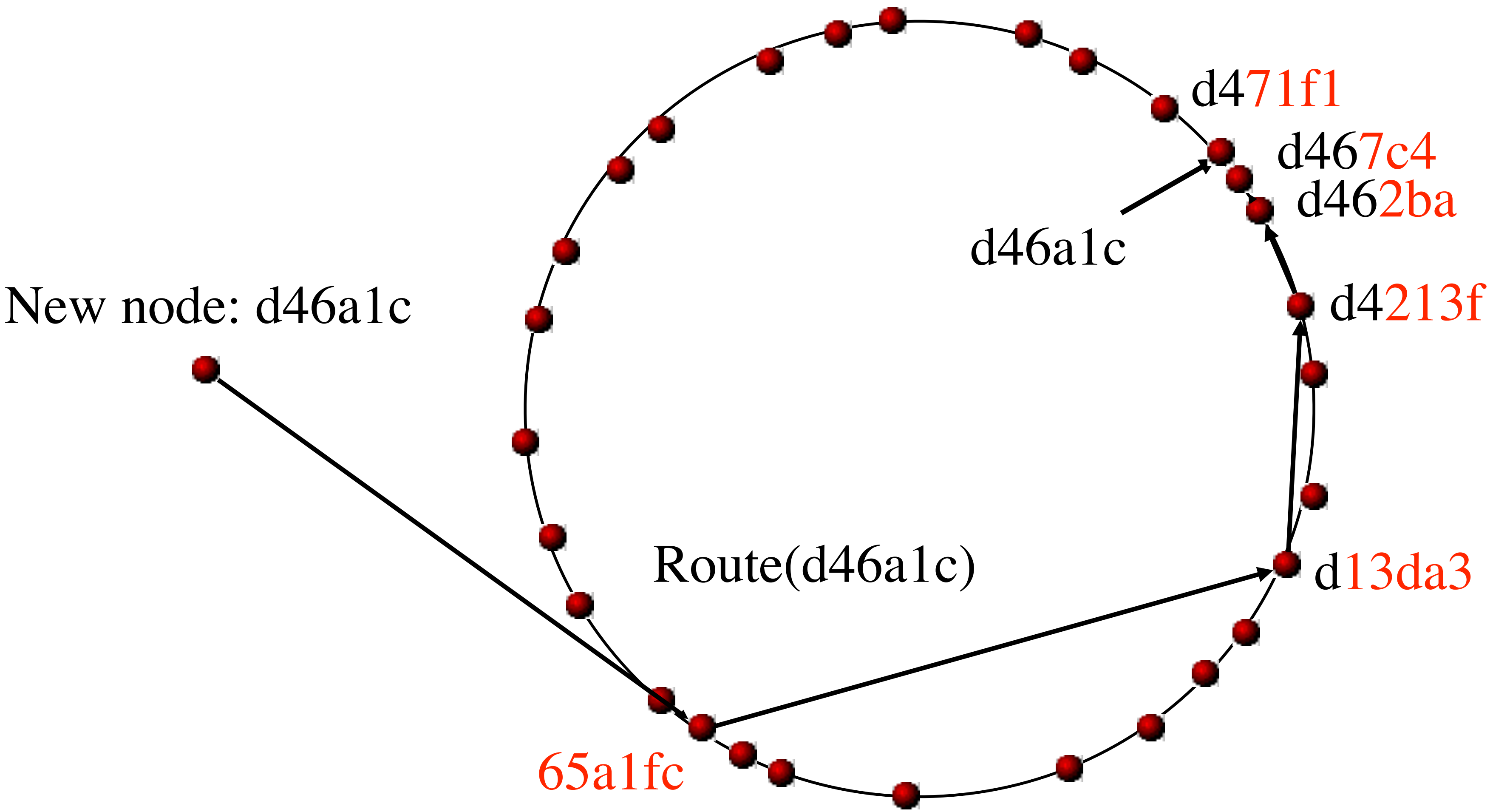
Pastry: Node addition



Pastry: Node addition



Pastry: Node addition

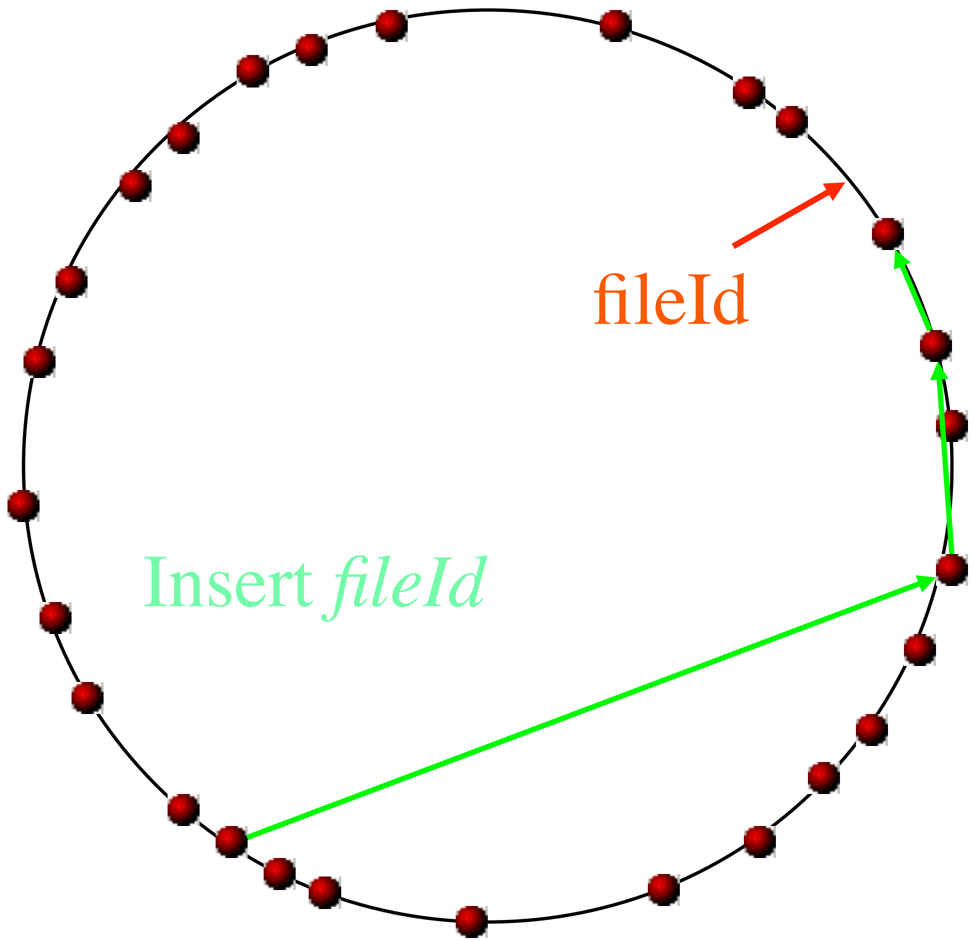


Node departure (failure)

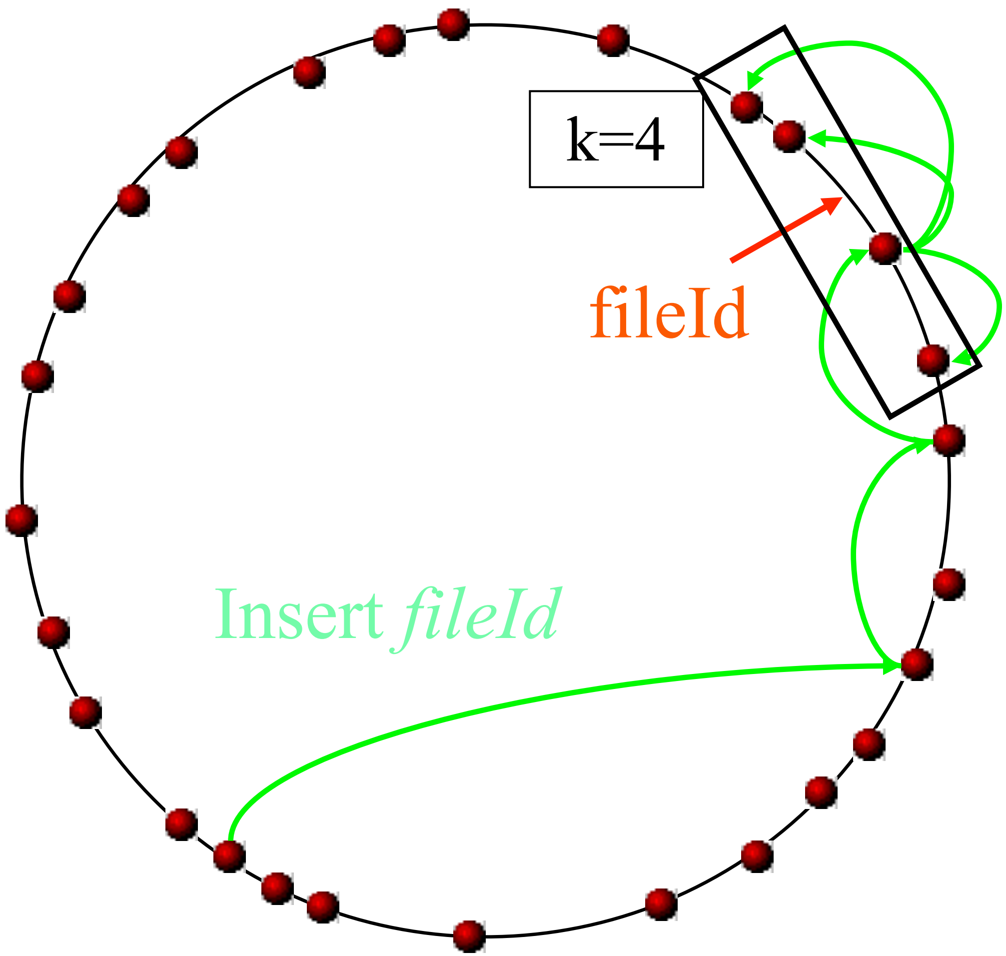
Leaf set members exchange keep-alive messages

- **Leaf set repair (eager):** request set from farthest live node in set
- **Routing table repair (lazy):** get table from peers in the same row, then higher rows

PAST: File storage



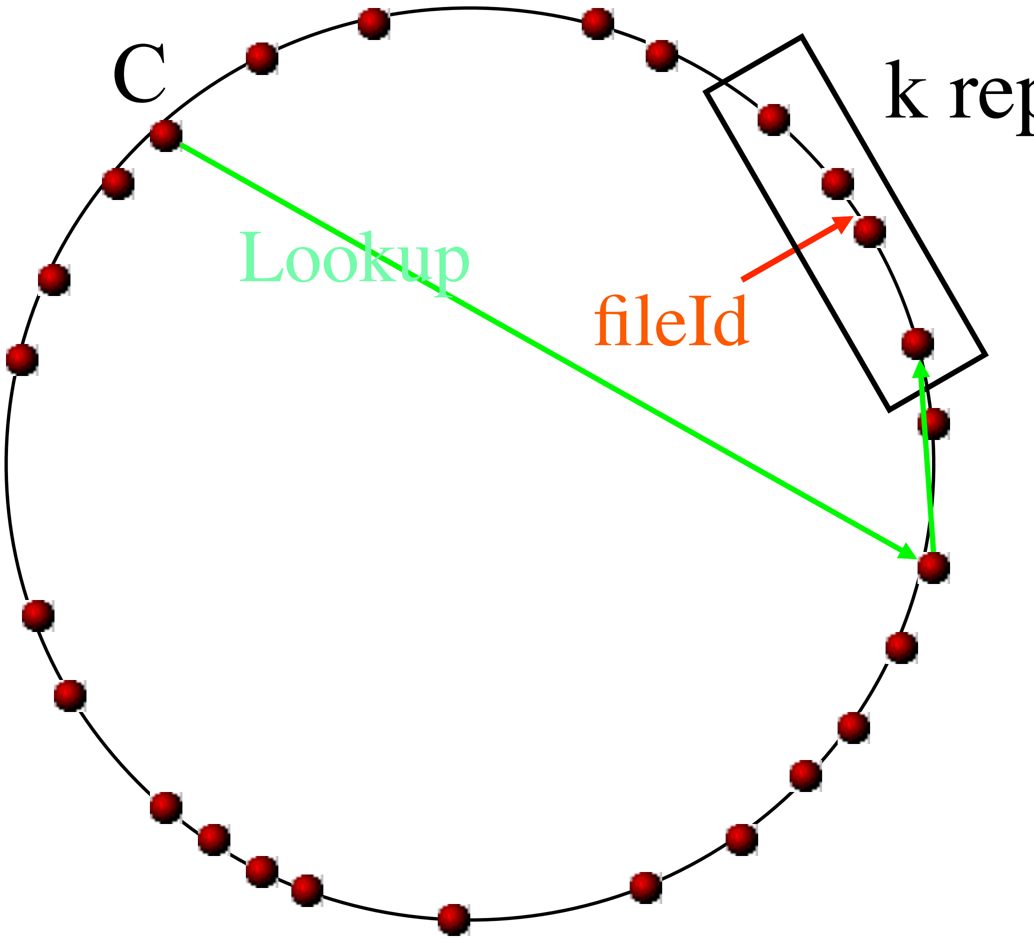
PAST: File storage



Storage Invariant: File "replicas" are stored on k nodes with nodeIds closest to fileId

(k is bounded by the leaf set size)

PAST: File Retrieval



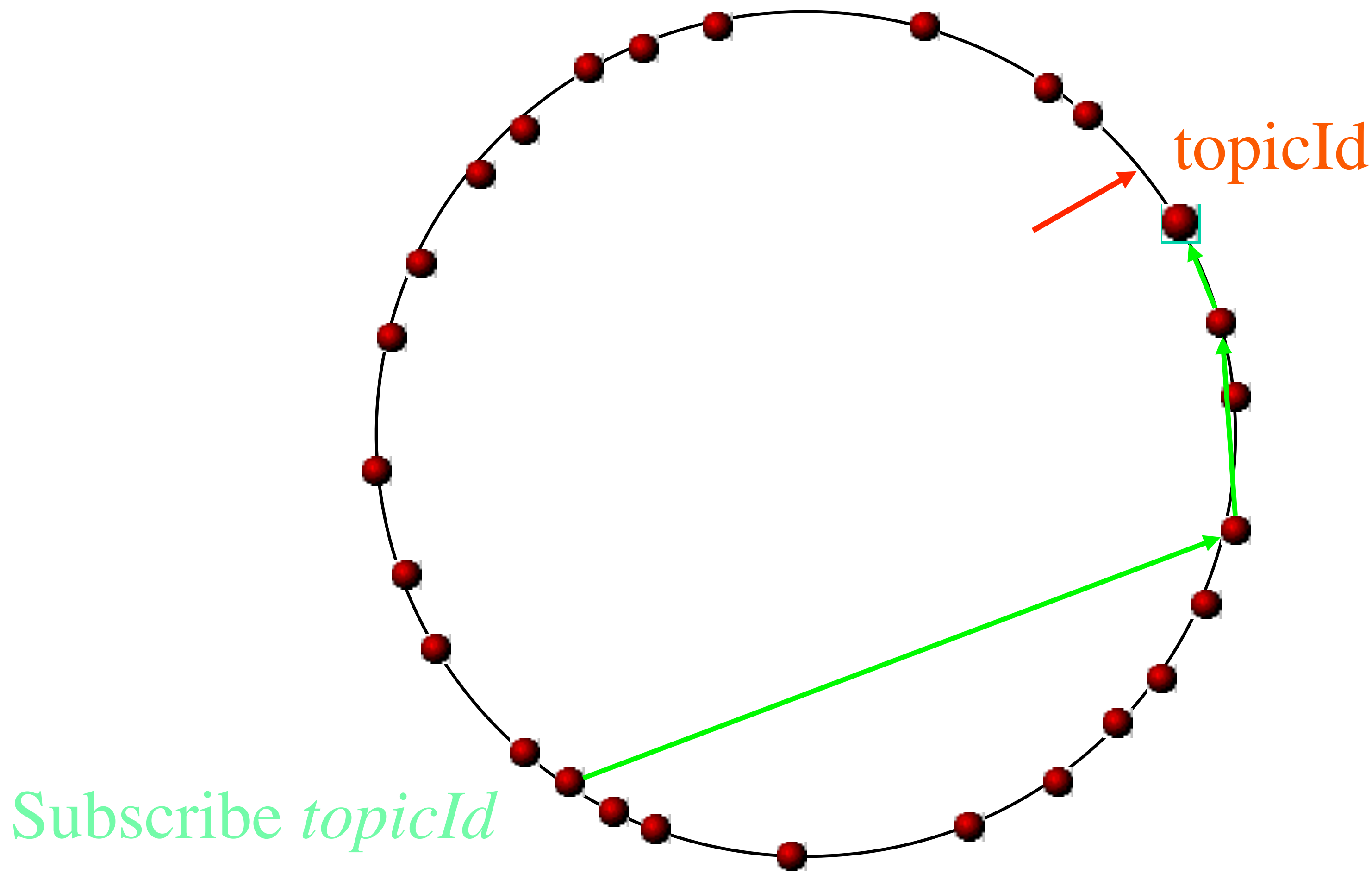
file located in $\log_{16} N$ steps (expected)

usually locates replica nearest client C

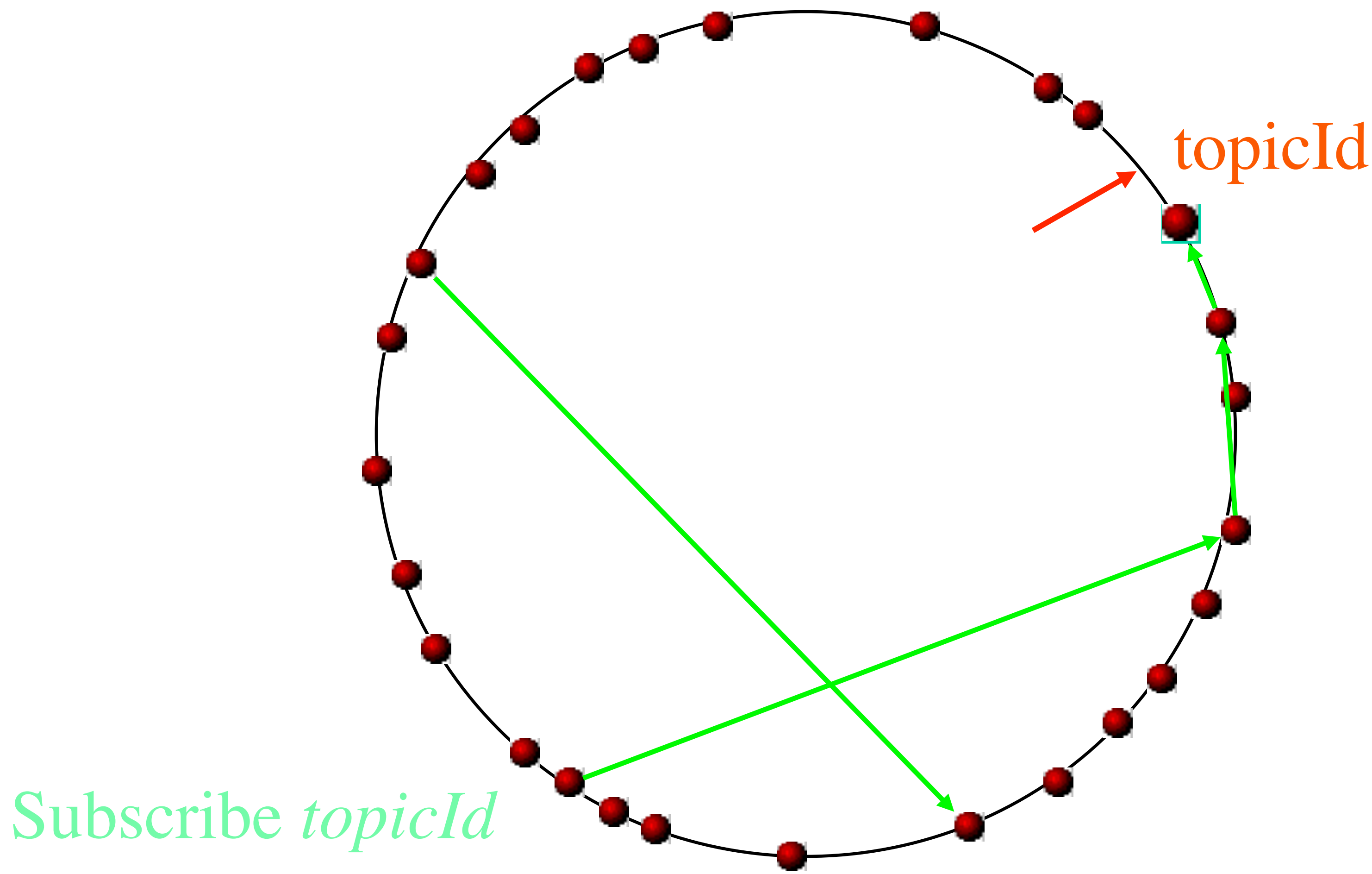
SCRIBE: Large-scale, decentralized multicast

- Infrastructure to support topic-based publish-subscribe applications
- Scalable: large numbers of topics, subscribers, wide range of subscribers/topic
- Efficient: low delay, low link stress, low node overhead

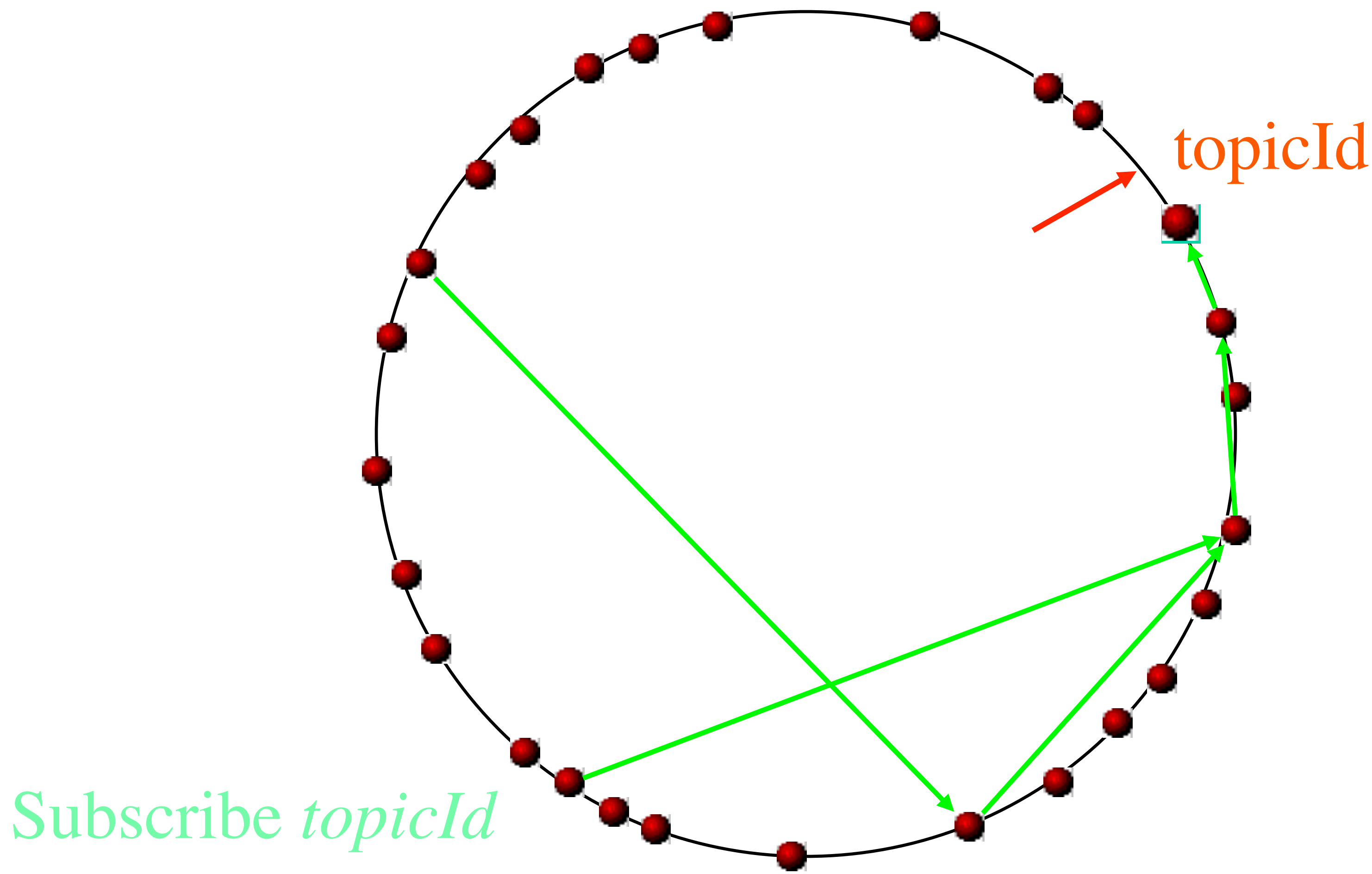
SCRIBE: Large scale multicast



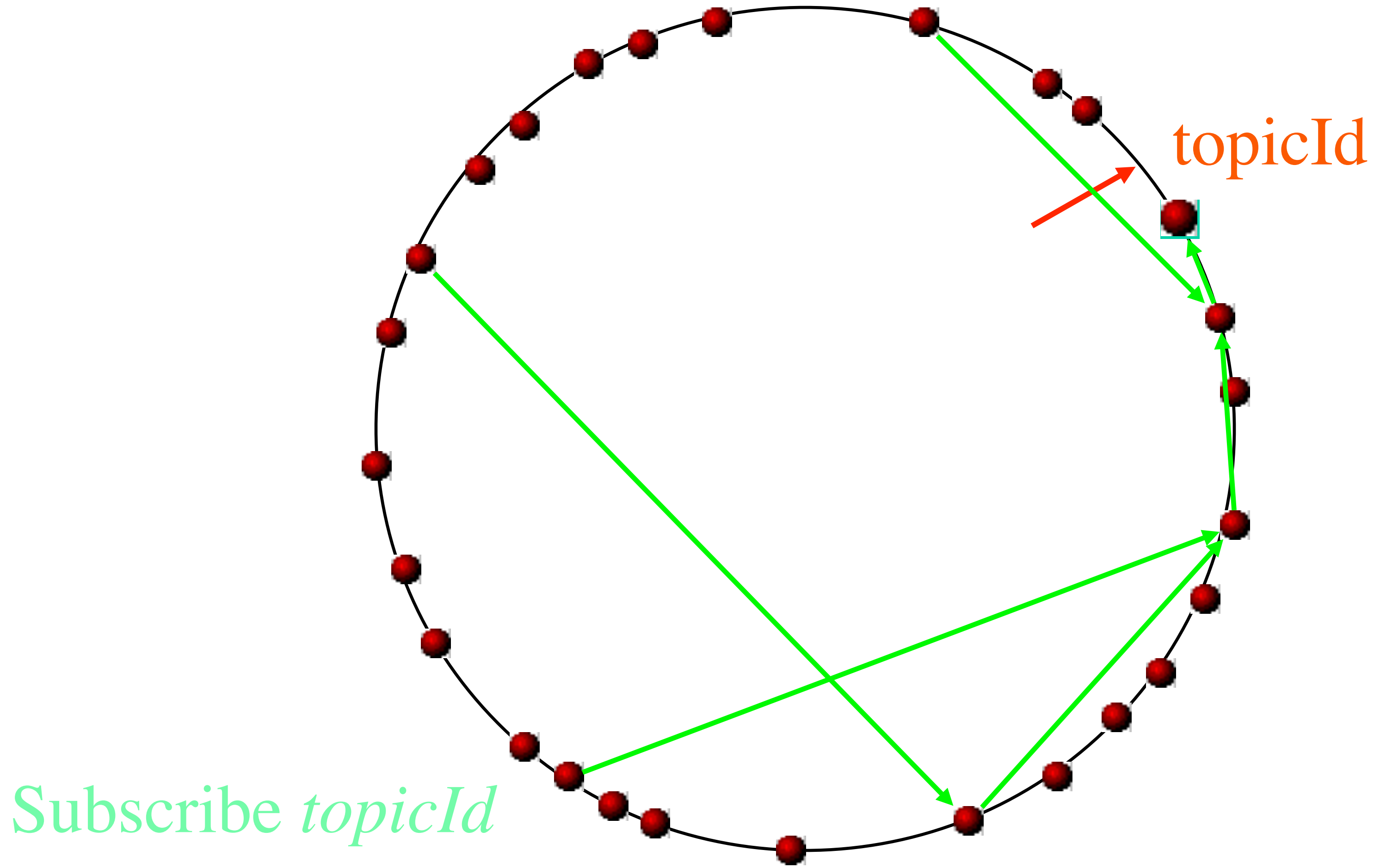
SCRIBE: Large scale multicast



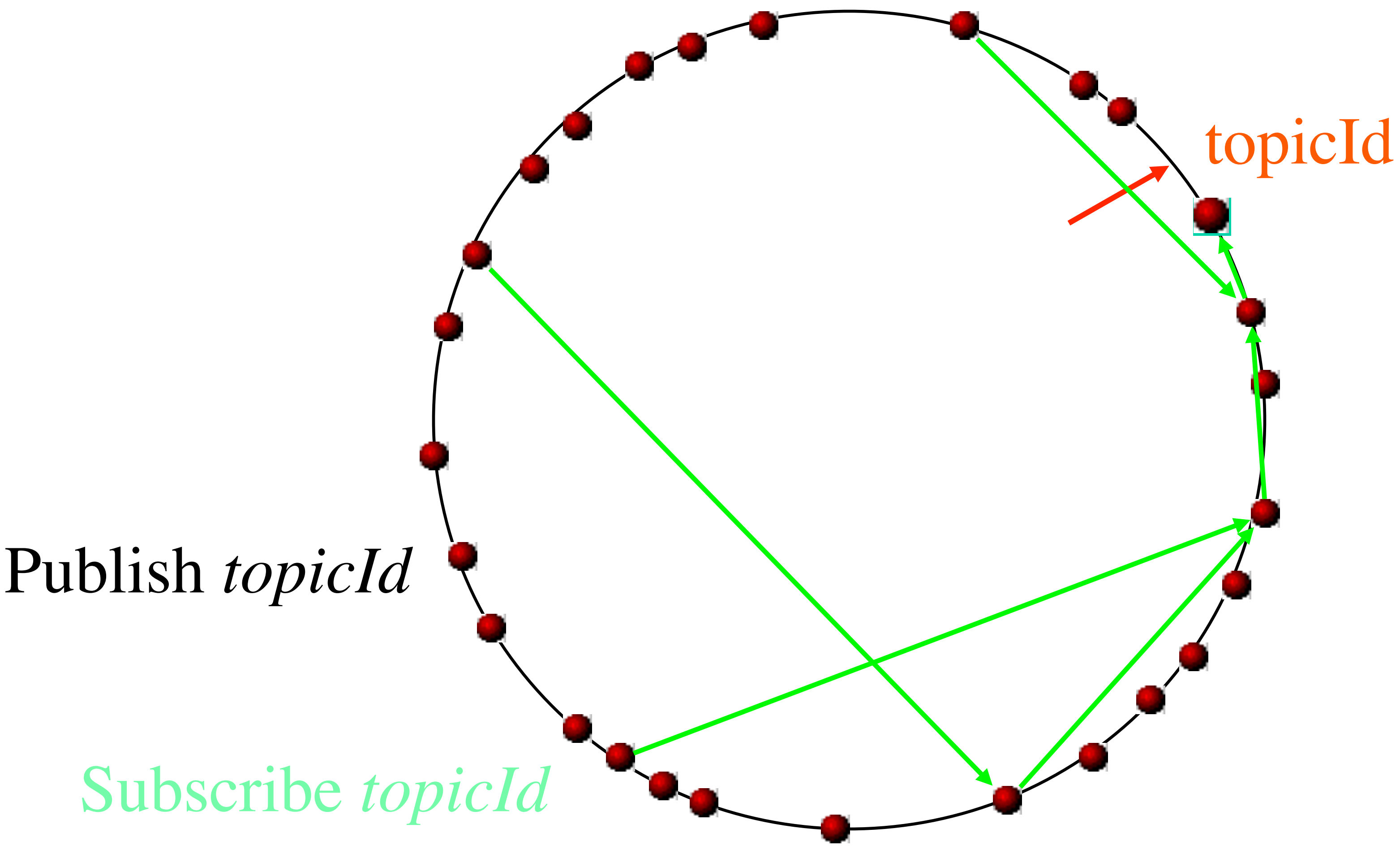
SCRIBE: Large scale multicast



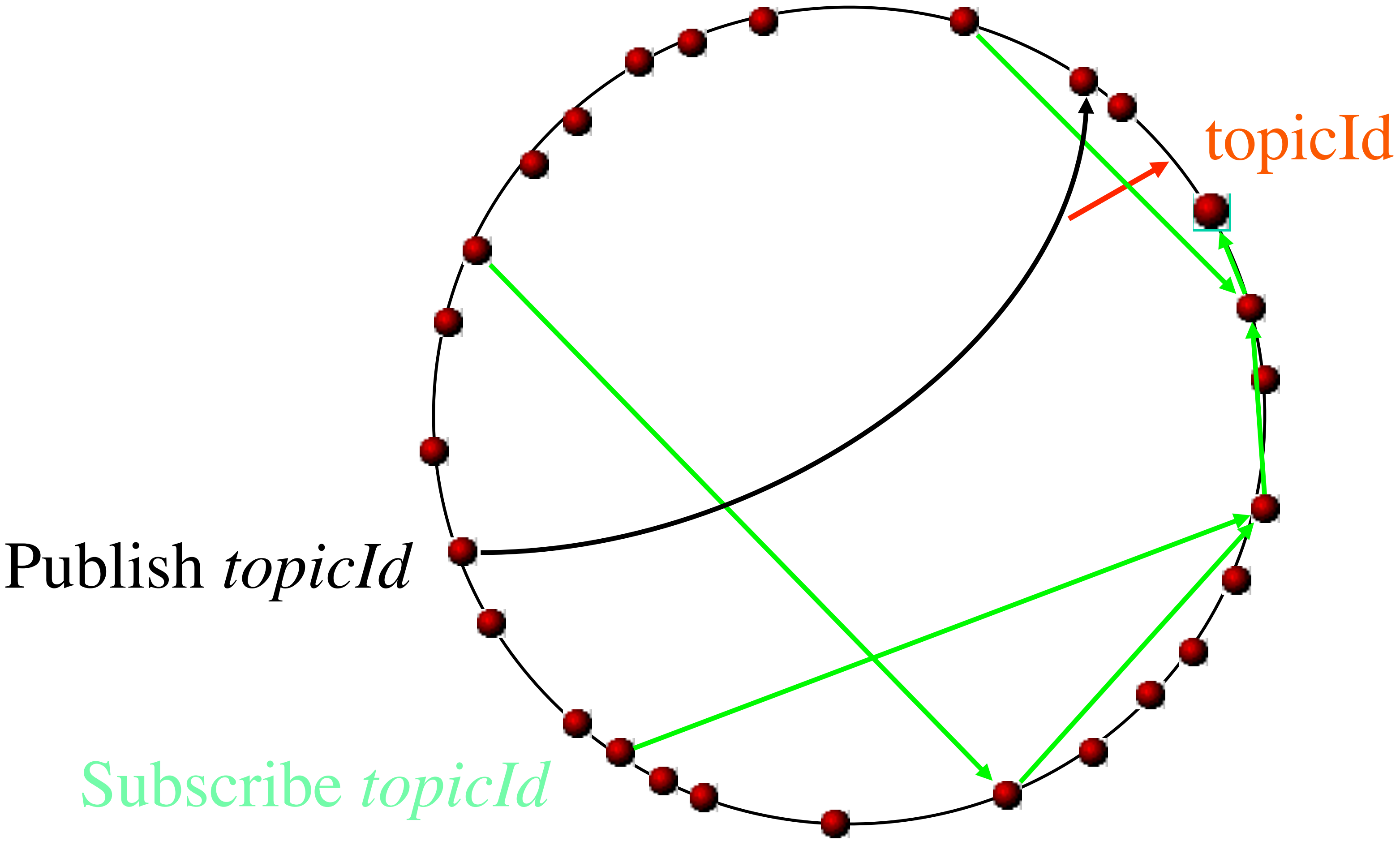
SCRIBE: Large scale multicast



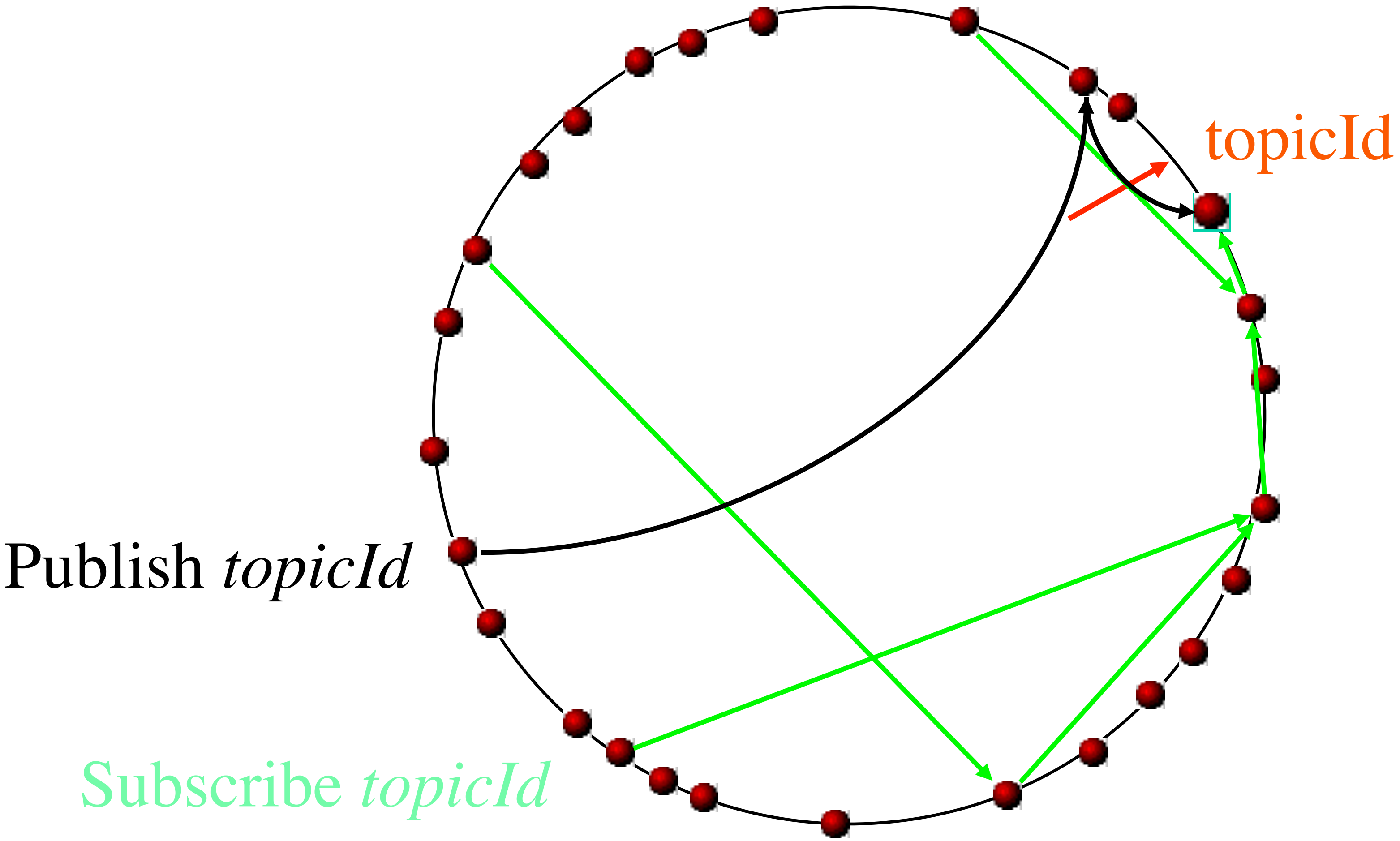
SCRIBE: Large scale multicast



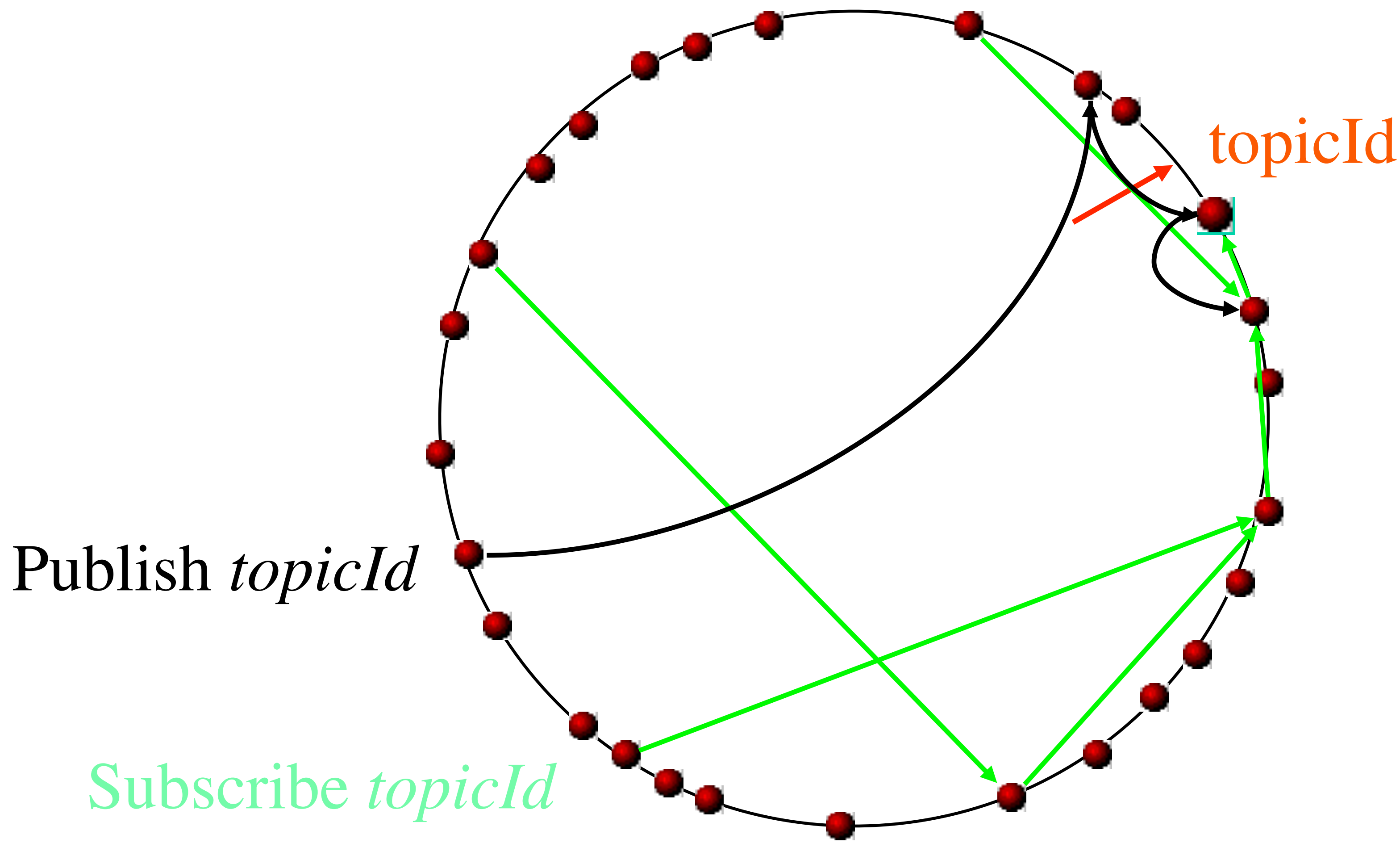
SCRIBE: Large scale multicast



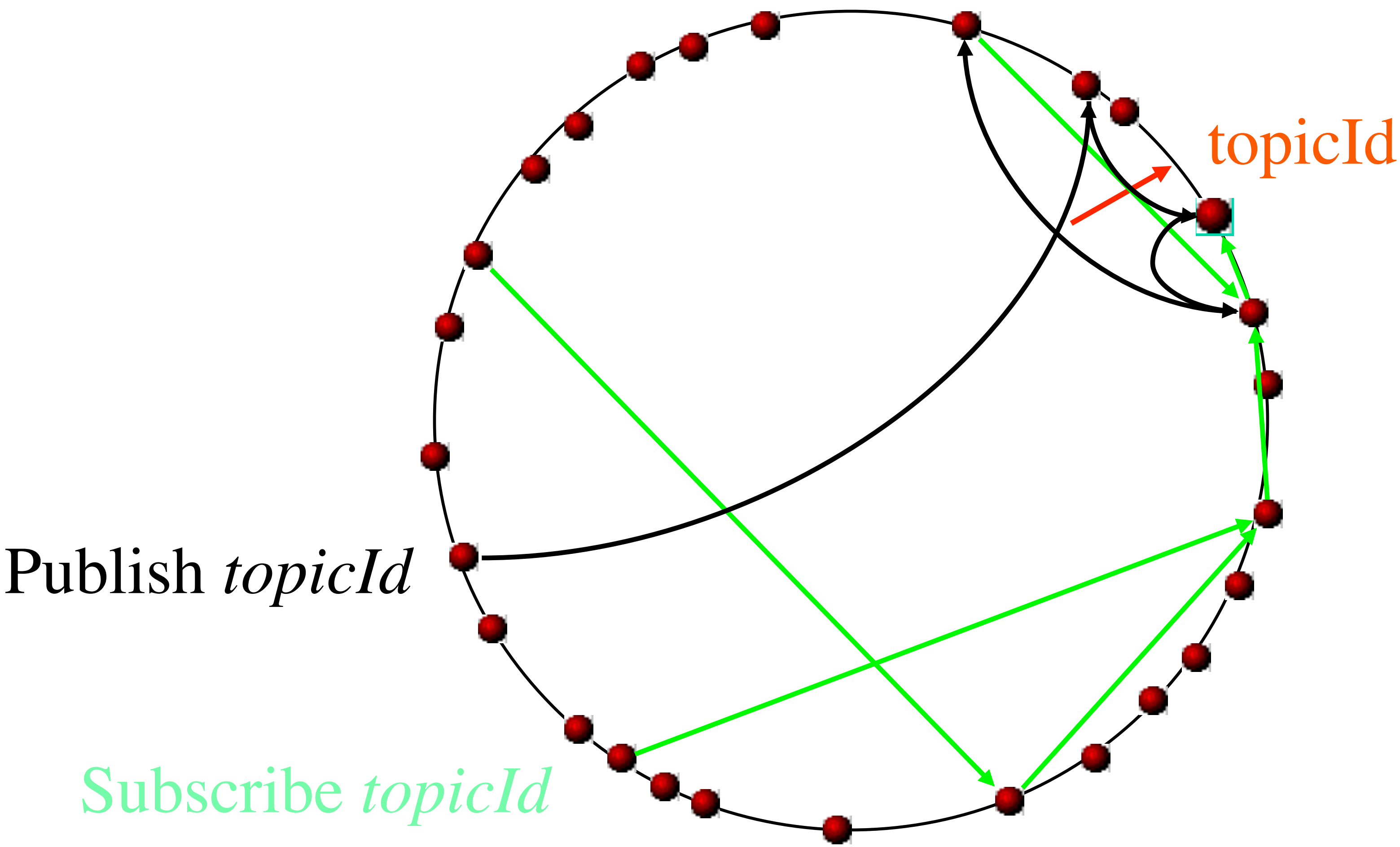
SCRIBE: Large scale multicast



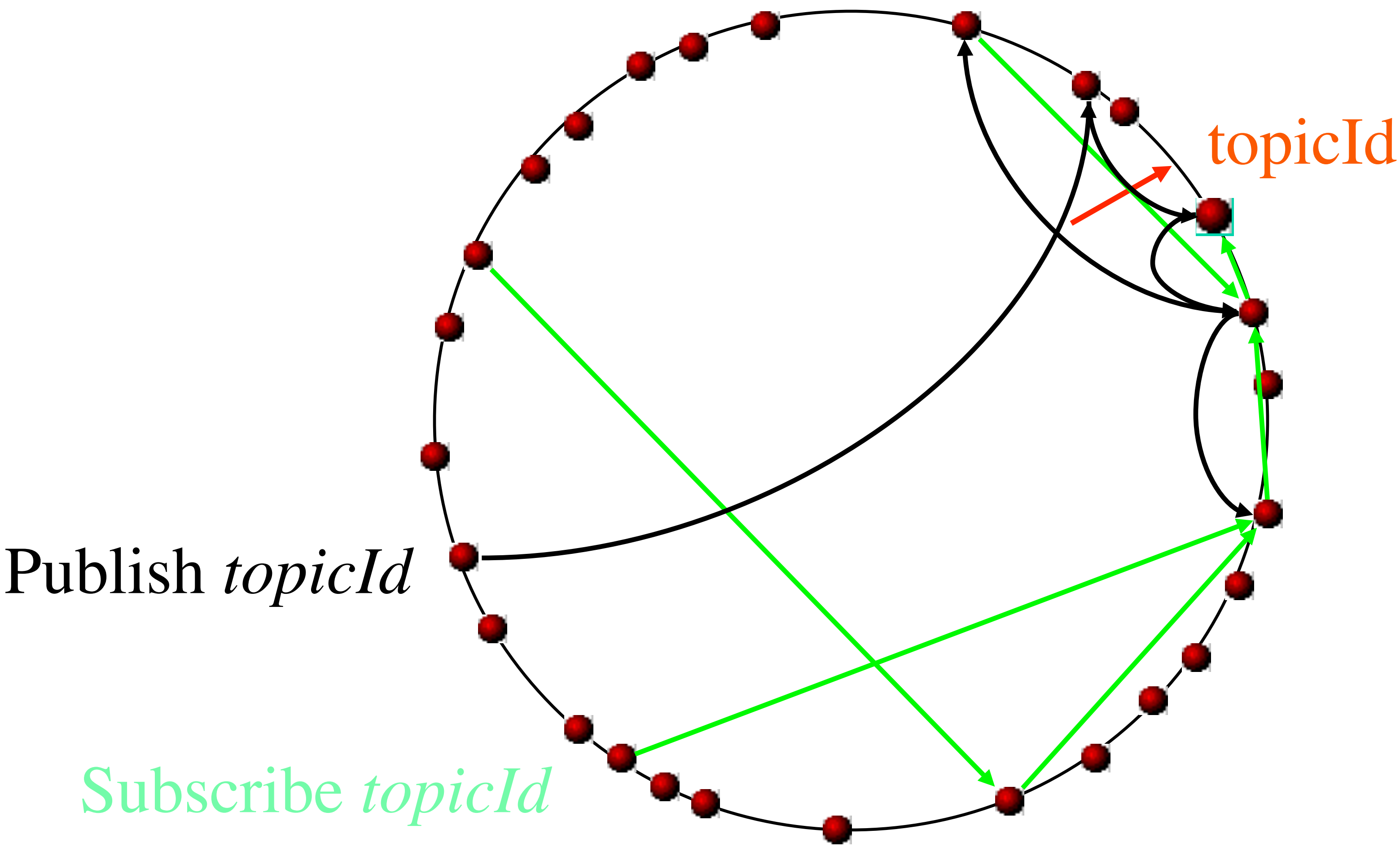
SCRIBE: Large scale multicast



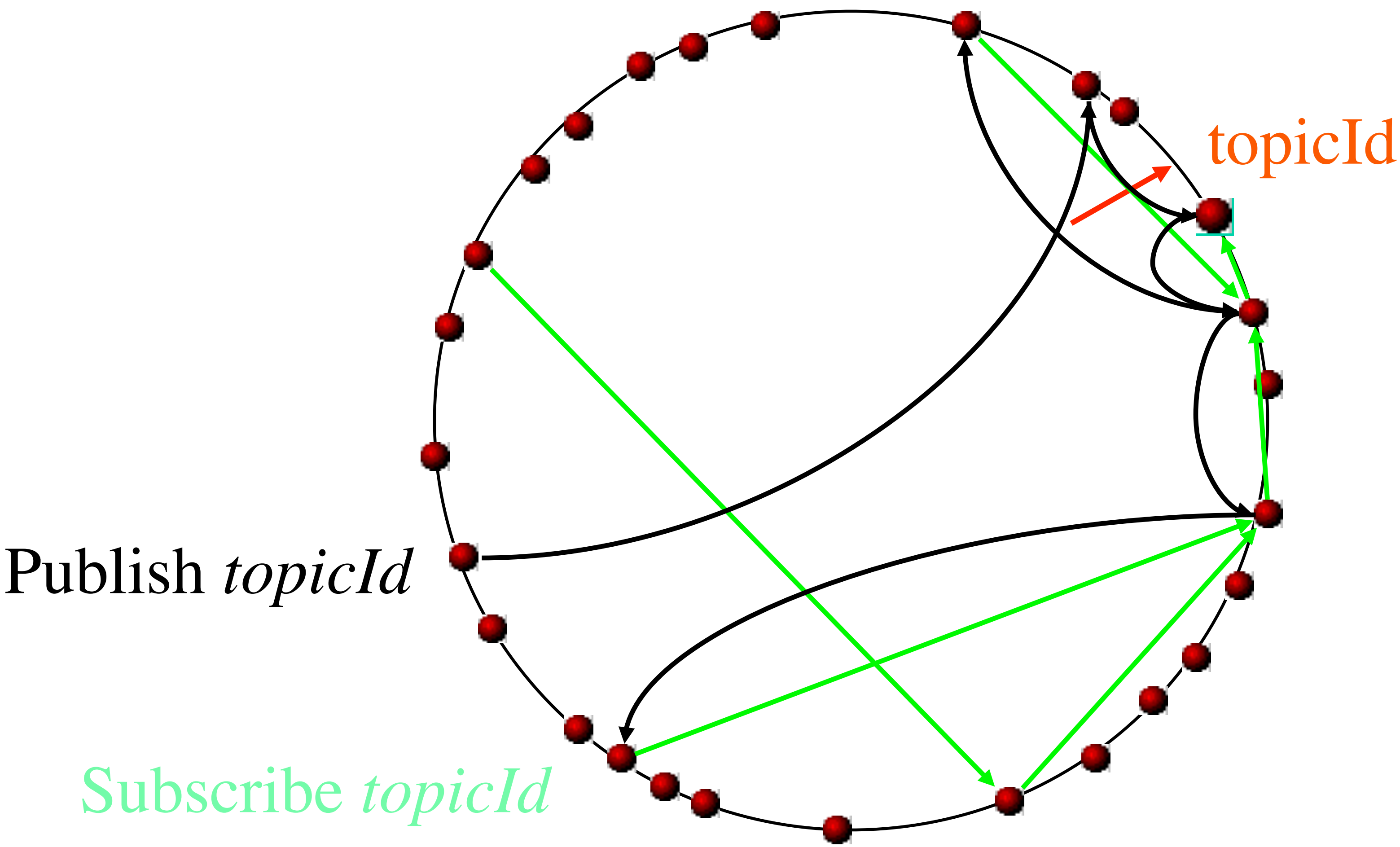
SCRIBE: Large scale multicast



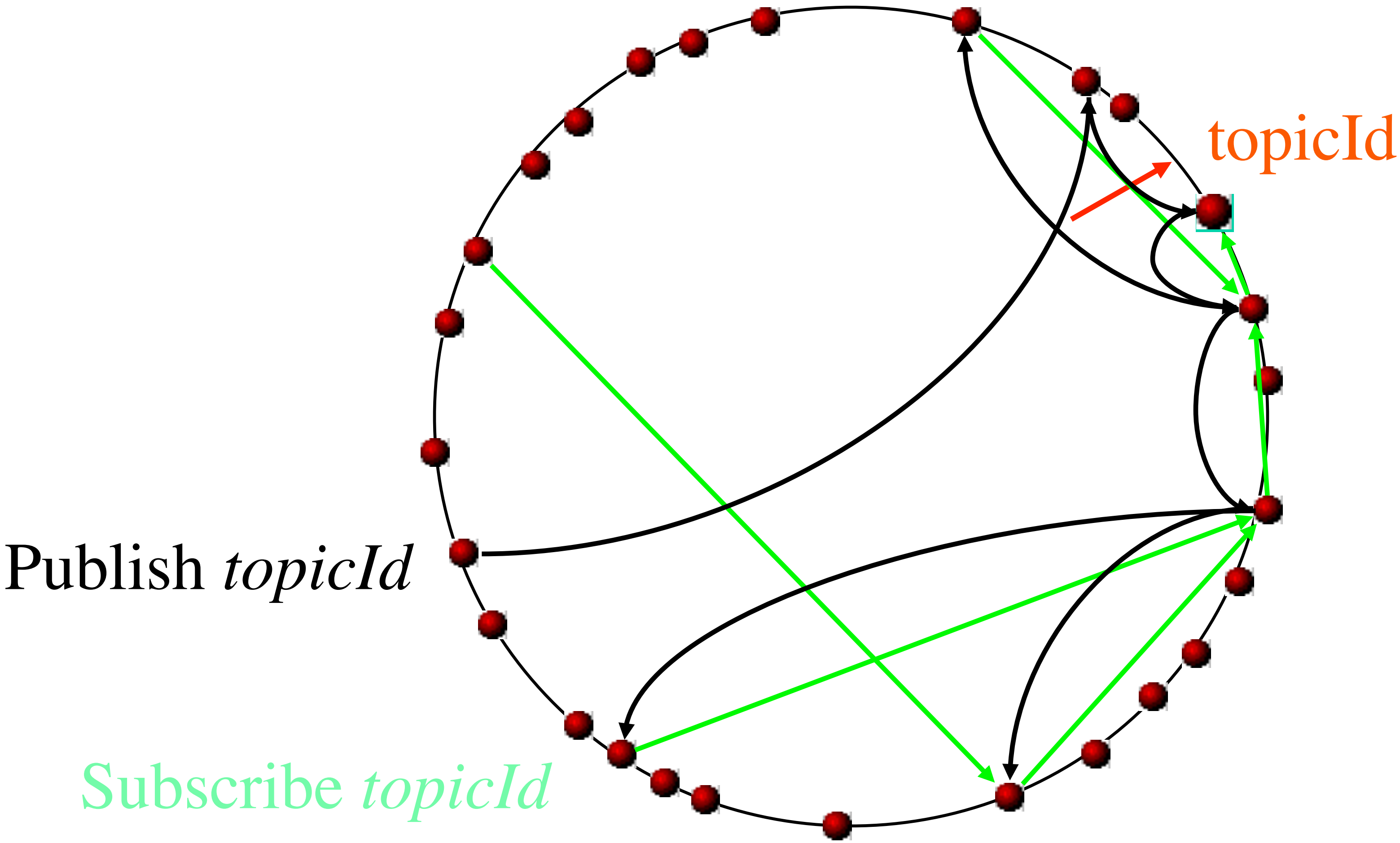
SCRIBE: Large scale multicast



SCRIBE: Large scale multicast



SCRIBE: Large scale multicast



SCRIBE: Large scale multicast