Routing on Overlay Networks

EECS 228

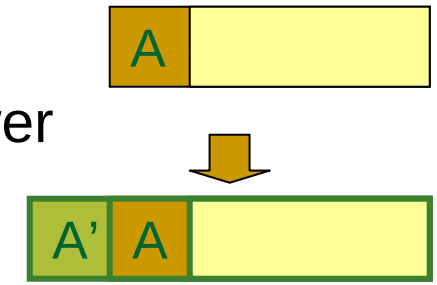
Abhay Parekh

parekh@eecs.berkeley.edu

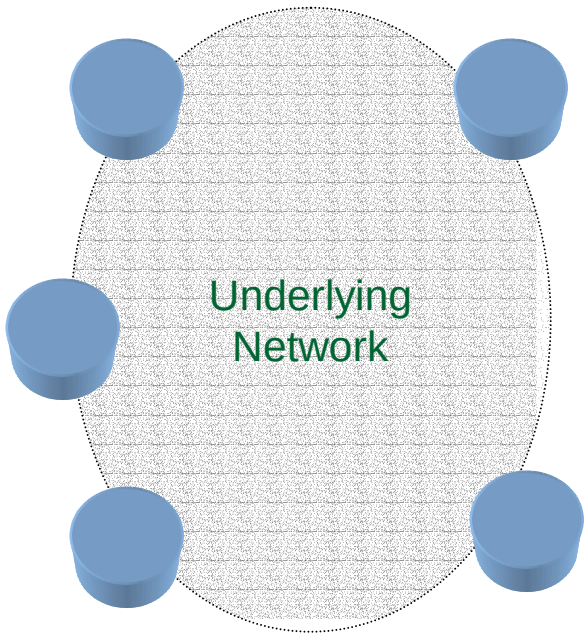
October 28, 2002

What is an overlay network?

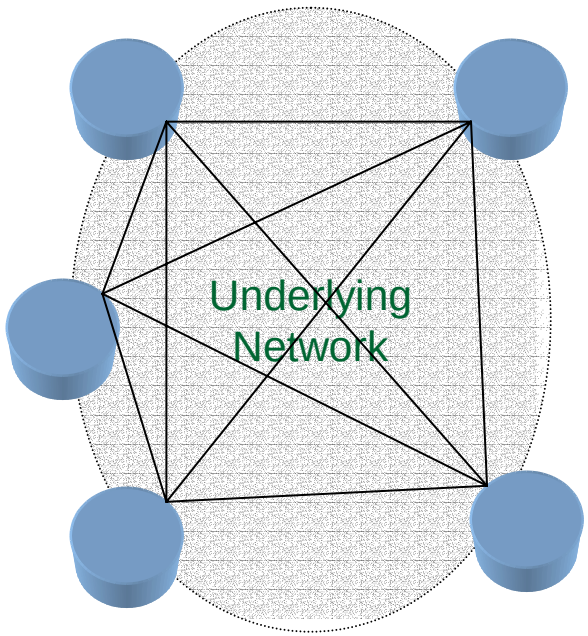
- A network defined over another set of networks
- The overlay addresses its own nodes
- Links on one layer are network segments of lower layers
  - Requires lower layer routing to be utilized
- Overlaying mechanism is called tunneling
- Example: Virtual Private Networks
  - Virtual topology defined via VPN nodes
  - Telecommuters appear as though they are on the corporate network
  - Sessions are more secure
  - Bits may traverse various kinds of underlying networks
    - PSTN, Frame Relay etc.



Routing On the overlay

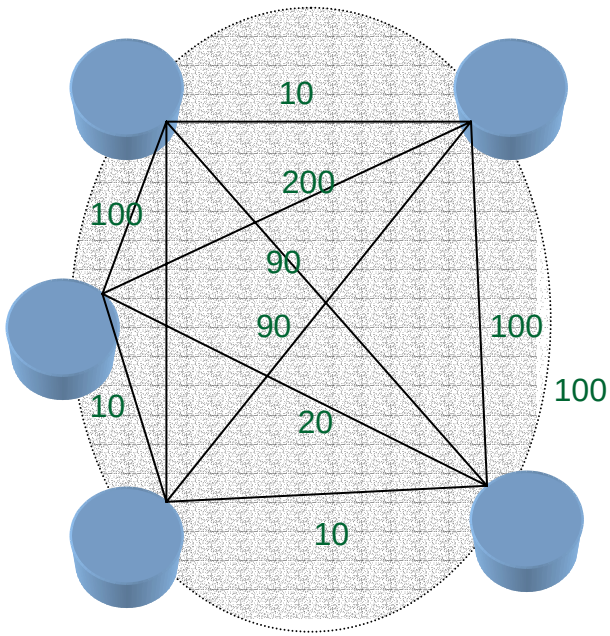


Routing on the Overlay



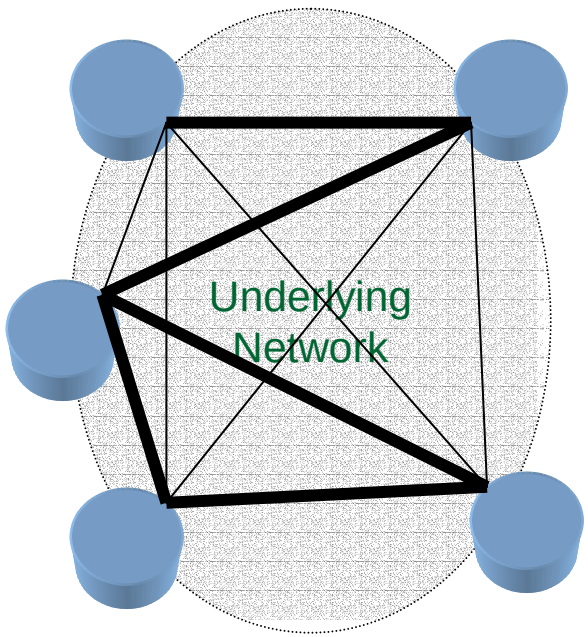
- The underlying network induces a complete graph of connectivity
  - No routing required!

Routing on the Overlay



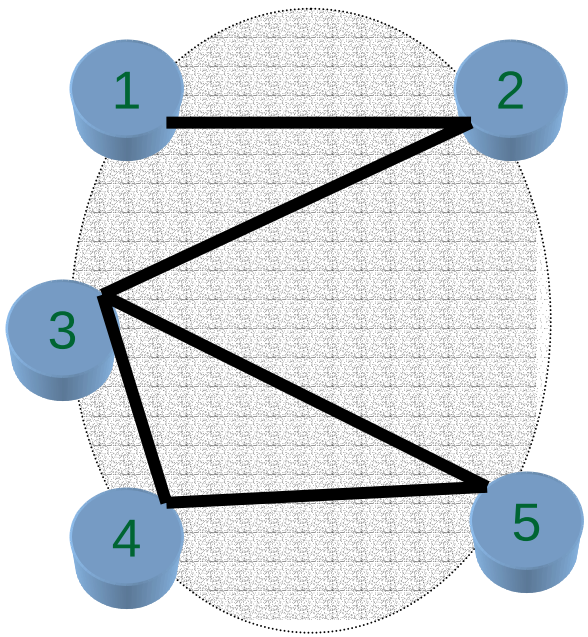
- The underlying network induces a complete graph of connectivity
  - No routing required!
- But
  - One virtual hop may be many underlying hops away.
  - Latency and cost vary significantly over the virtual links
  - State information may grow with E (n^2)

Routing Issues



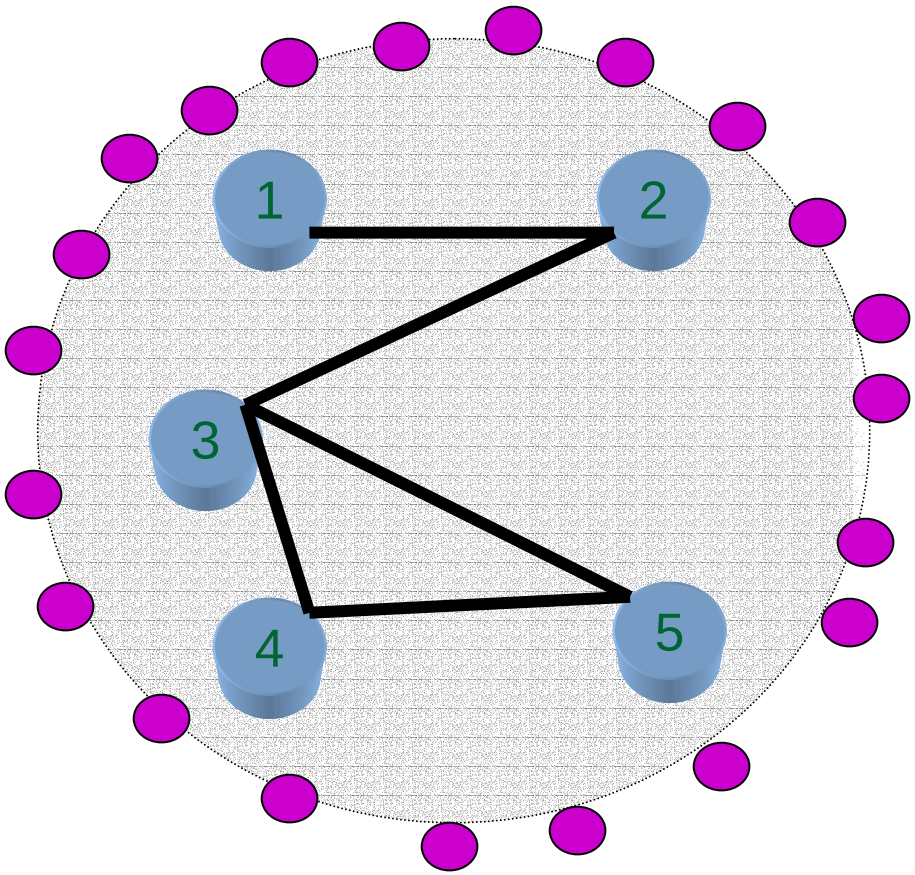
- The underlying network induces a complete graph of connectivity
  - No routing required!
- But
  - One virtual hop may be many underlying hops away.
  - Latency and cost vary significantly over the virtual links
  - State information may grow with E (n^2)
- At any given time, the overlay network picks a connected sub-graph based on nearest neighbors
  - How often can vary
  - Also, structured (Chord) v/s unstructured (Gnutella)

Routing Issues



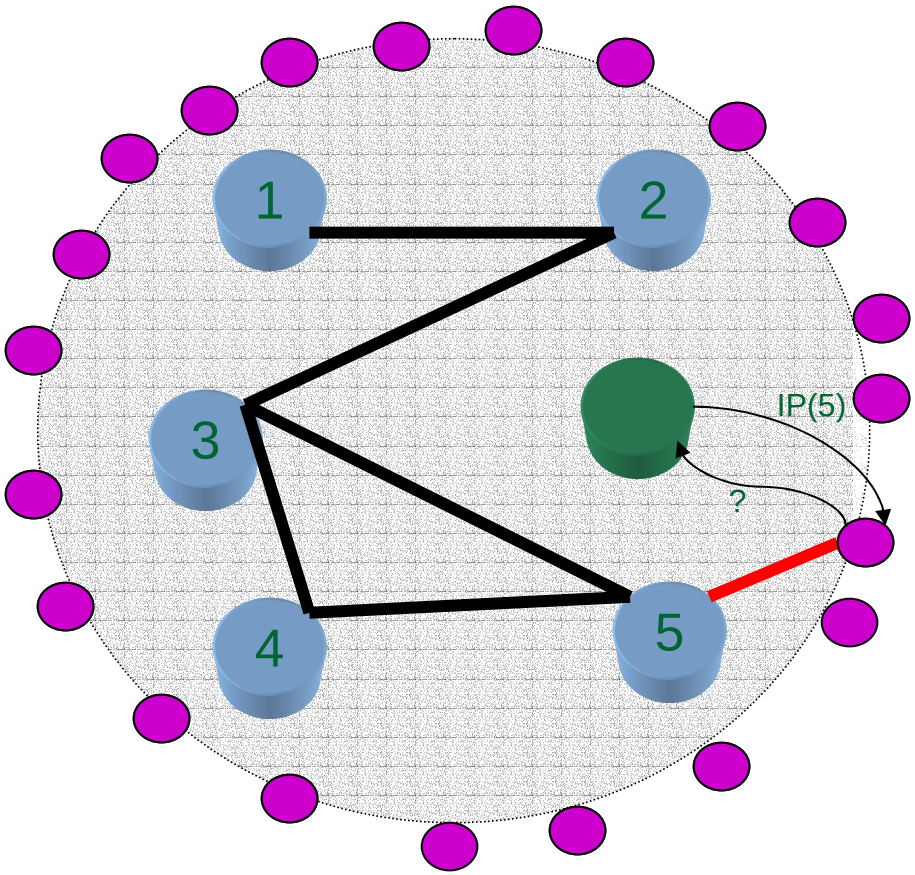
- The underlying network induces a complete graph of connectivity
  - No routing required!
- But
  - One virtual hop may be many underlying hops away.
  - Latency and cost vary significantly over the virtual links
  - State information may grow with E (n^2)
- At any given time, the overlay network picks a connected sub-graph based on nearest neighbors
  - How often can vary
  - Also, structured (Chord) v/s unstructured (Gnutella)
- The overlay network must ROUTE!

Routing Issues



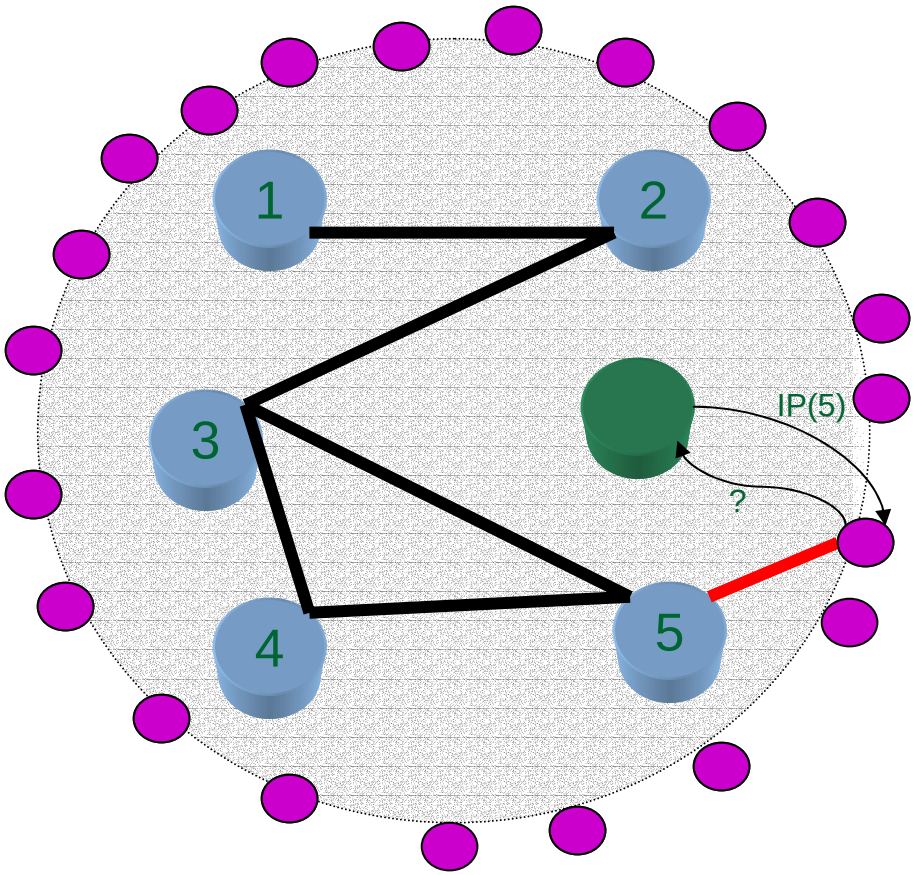
- Overlay network users may not be directly connected to the overlay nodes
  - E.g. Akamai

Overlay Routing: Edge Mapping



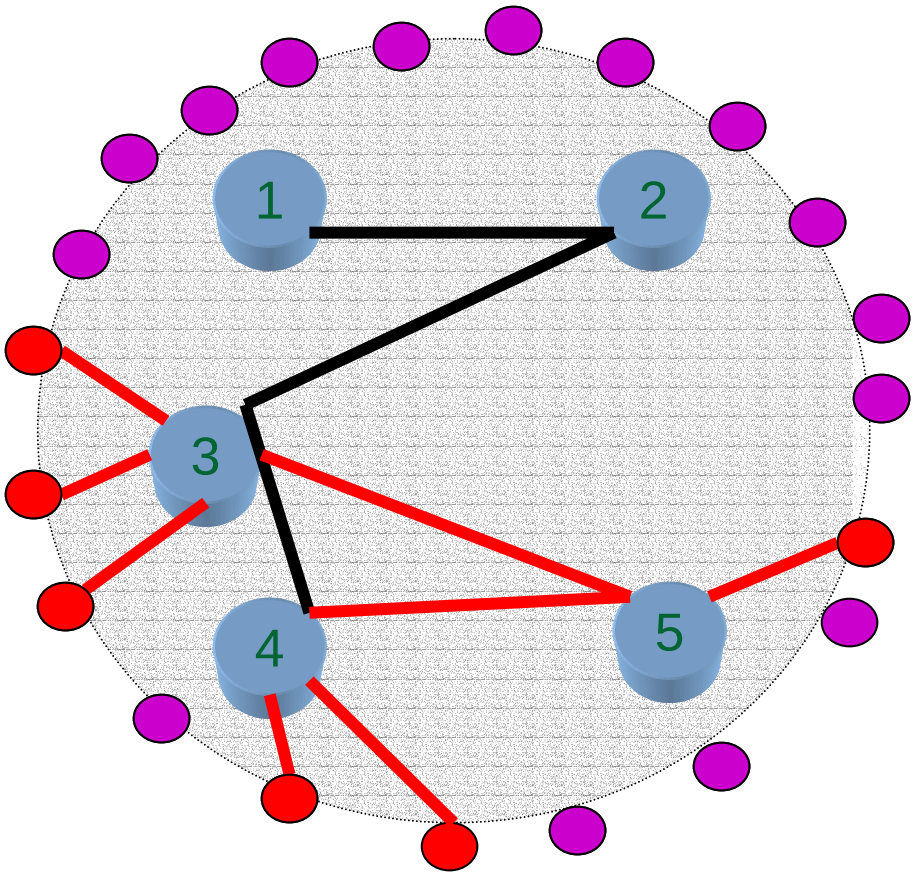
- Overlay network users may not be directly connected to the overlay nodes
  - E.g. Akamai
- User must be redirected to a "close by" overlay node
- Edge-Mapping, or redirection function is hard since
  - # potential users enormous
  - User clients not under direct control
- When overlay clients are directly connected the edge mapping function is obviated
  - E.g. P2P: users/nodes colocated

Overlay Routing: Edge Mapping



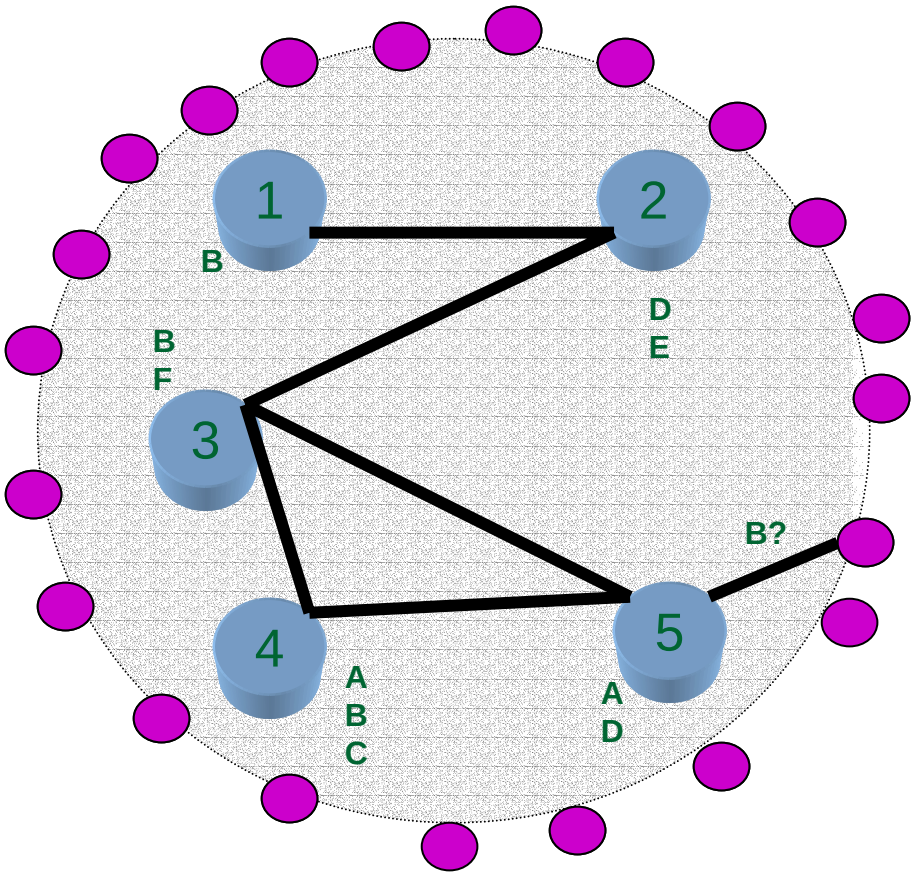
- Overlay nodes interconnect clients
- Enhance nature of connection
  - Multicast
  - Secure
  - Low Loss
- Much easier to add functionality than to integrate into a router

Overlay Routing: Adding Function to the route



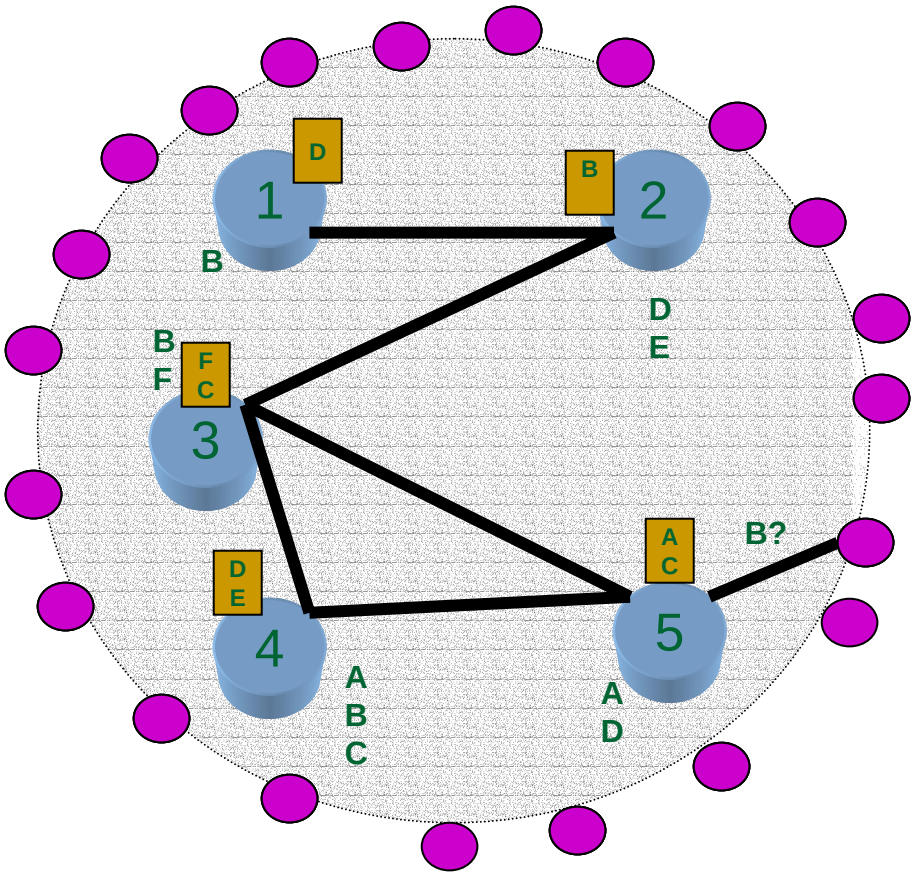
- Overlay nodes interconnect clients
- Enhance nature of connection
  - Multicast
  - Secure
  - Low Loss
- Much easier to add functionality than to integrate into a router
- Overlay nodes can become bottlenecks

Overlay Routing: Resource Location



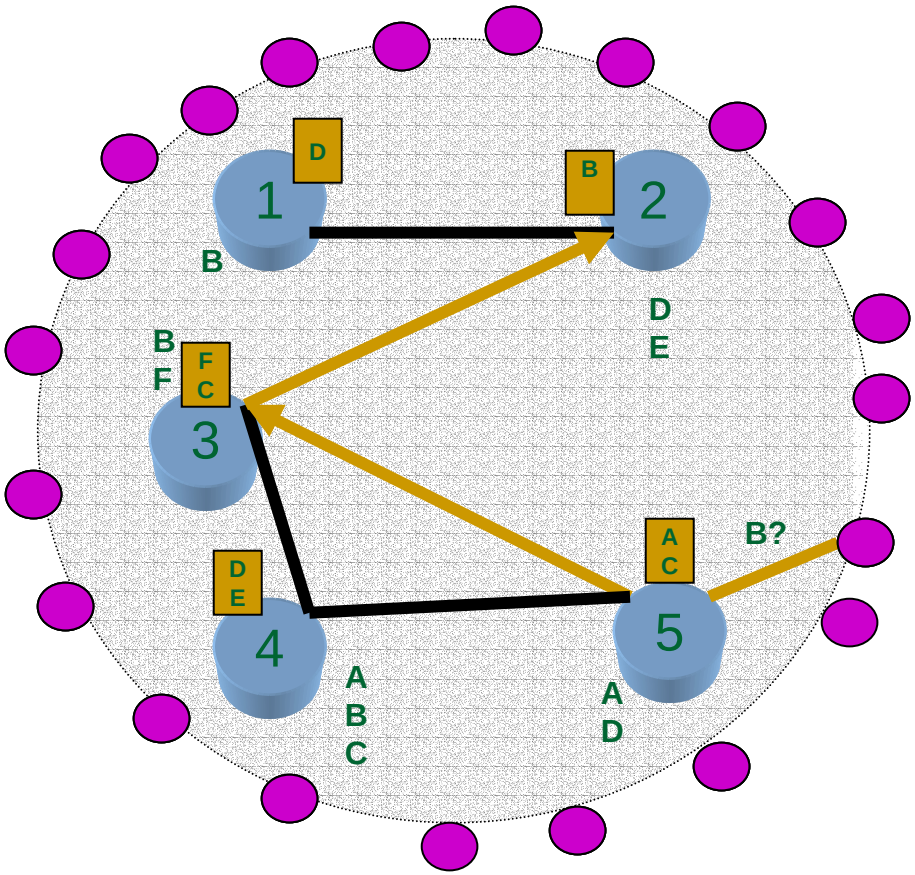
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must “search” for “closest” node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client

Overlay Routing: Resource Location



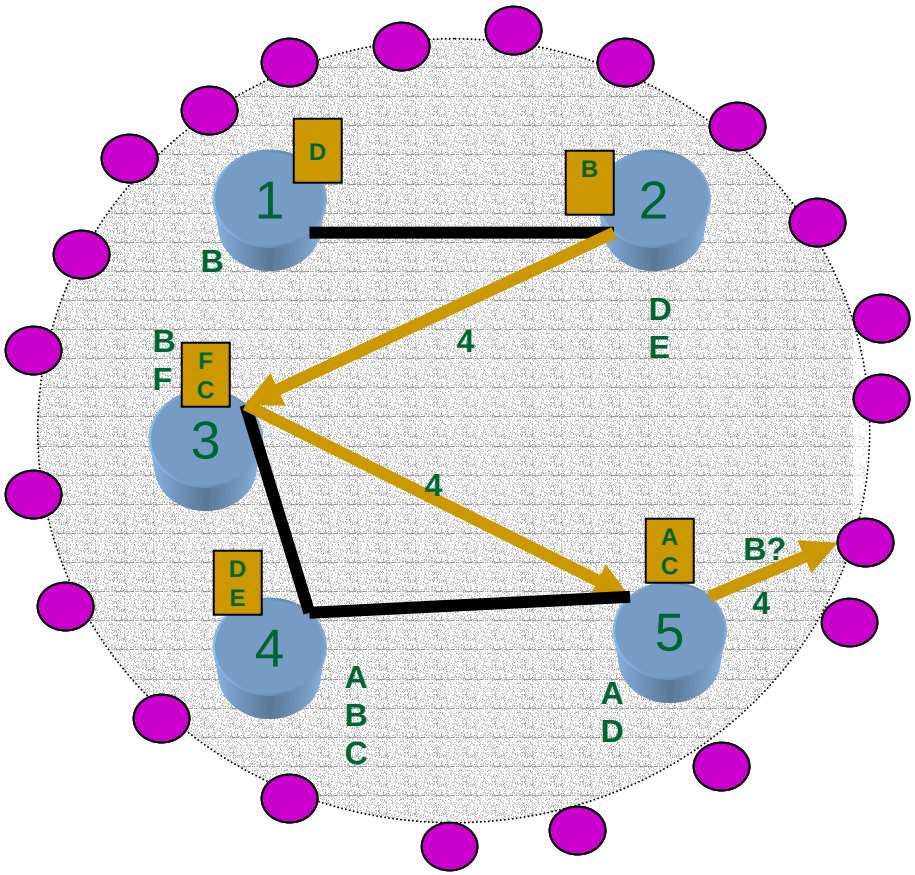
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must “search” for “closest” node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single “index” is not scalable
- Overlay launches a query to locate resource

Overlay Routing: Resource Location



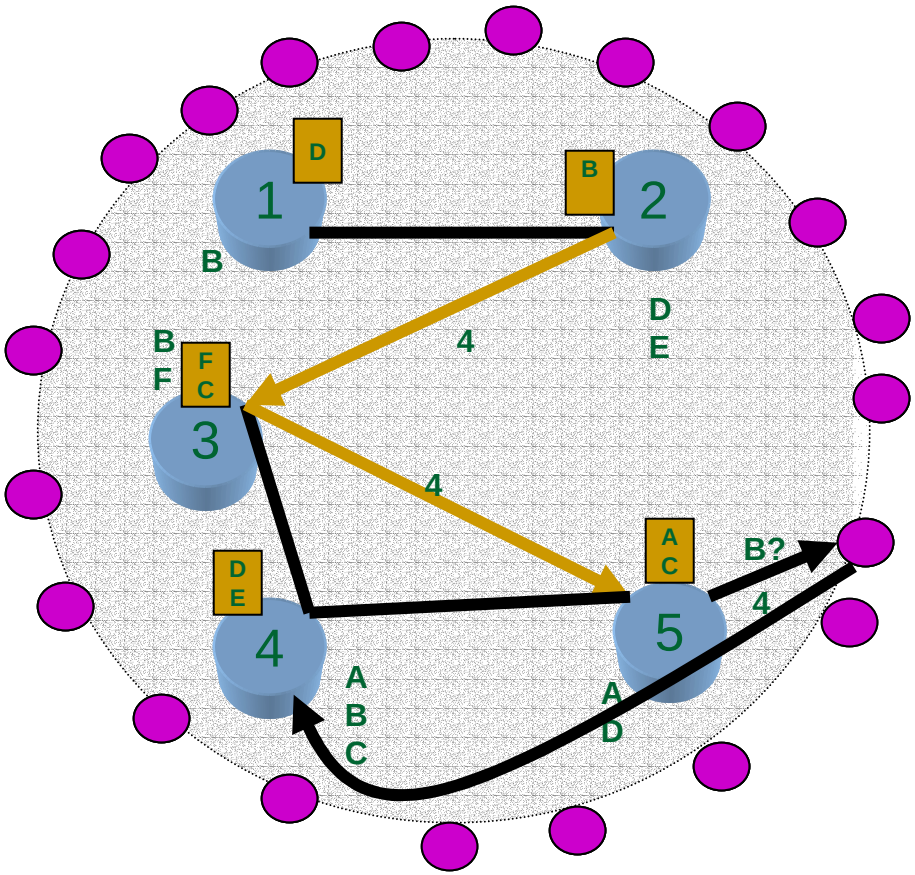
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must “search” for “closest” node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single “index” is not scalable
- Overlay launches a query to locate resource
- Query is “Routed” through the overlay until object is located

Overlay Routing: Resource Location



- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must “search” for “closest” node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single “index” is not scalable
- Overlay launches a query to locate resource
- Query is “Routed” through the overlay until object is located

Overlay Routing: Resource Location

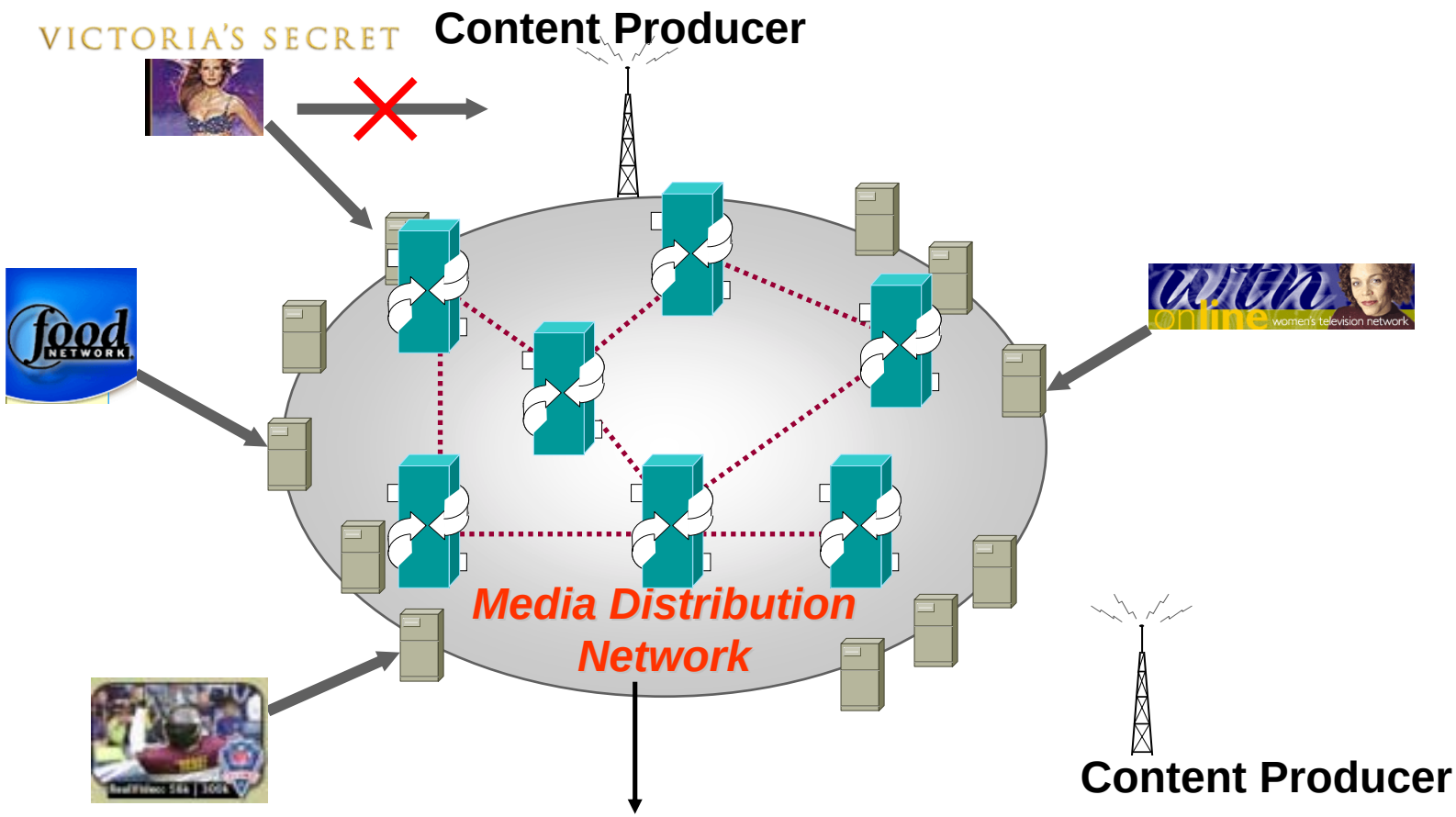


- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must “search” for “closest” node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single “index” is not scalable
- Overlay launches a query to locate resource
- Query is “Routed” through the overlay until object is located

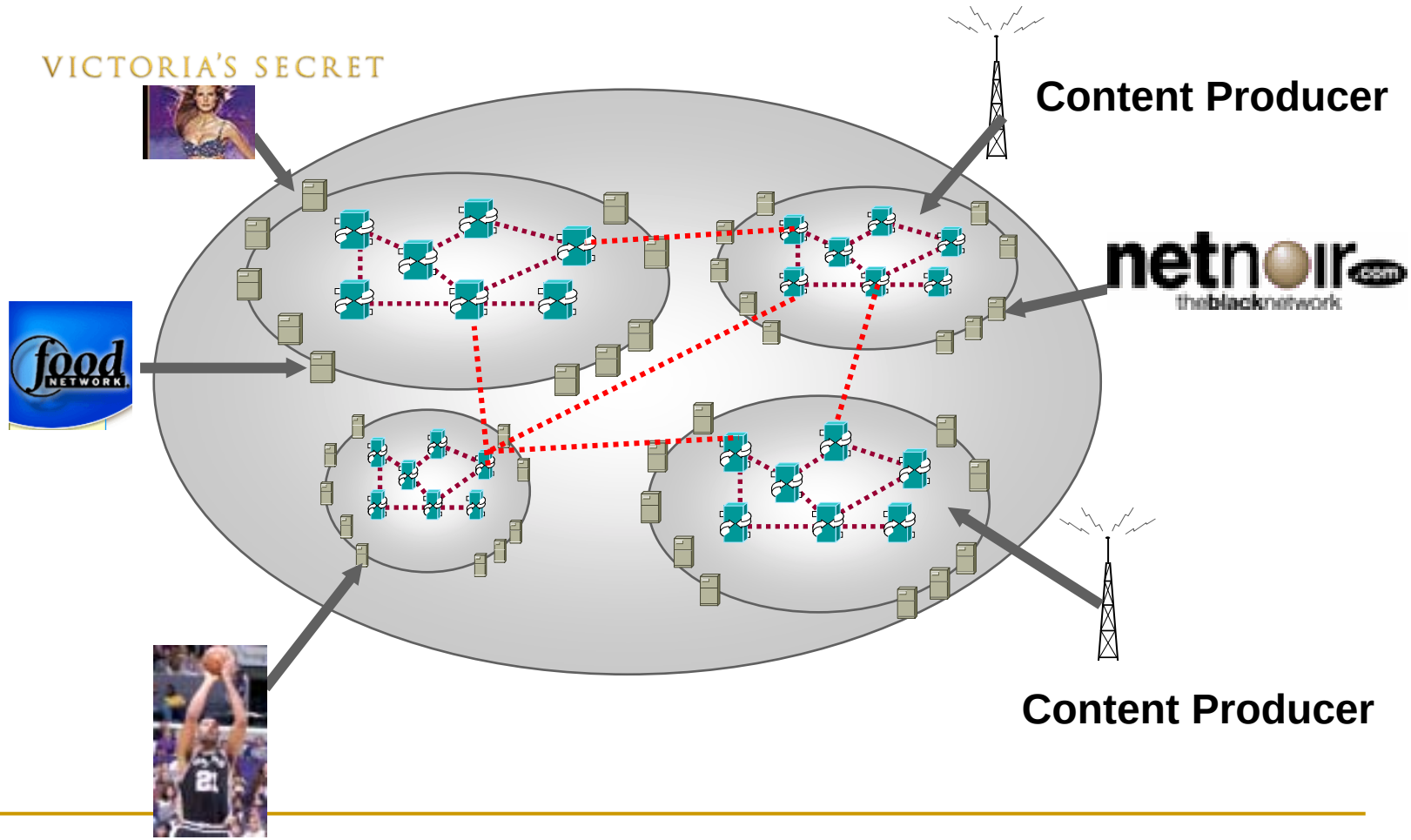
Summary

- Two kinds of overlays functions
  - Overlay contains resources
  - Overlay facilitates communication among other client applications
- Two kinds of virtual topologies
  - Structured
  - Unstructured
- Two kinds of client connectivty
  - Direct: P2P
  - Not direct: Akamai
- Overlay Network Functions
  - Select Virtual Edges (fast or slow timescales)
  - Overlay Routing Protocol
  - Edge Mapping
  - Resource Location
- Edge Mapping and Resource Location can be combined

Example: Application Level Multicast

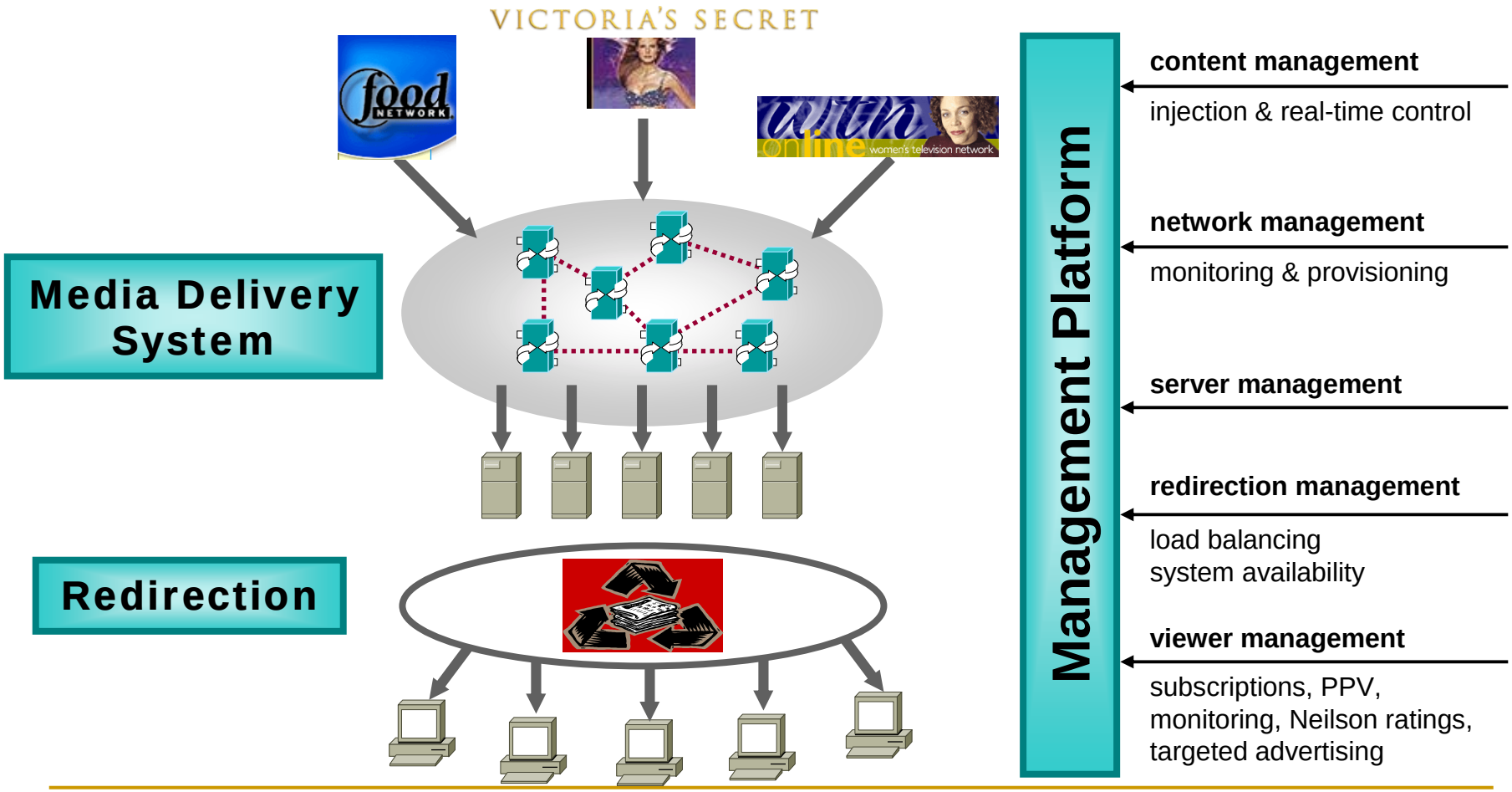


The Broadcast Internet



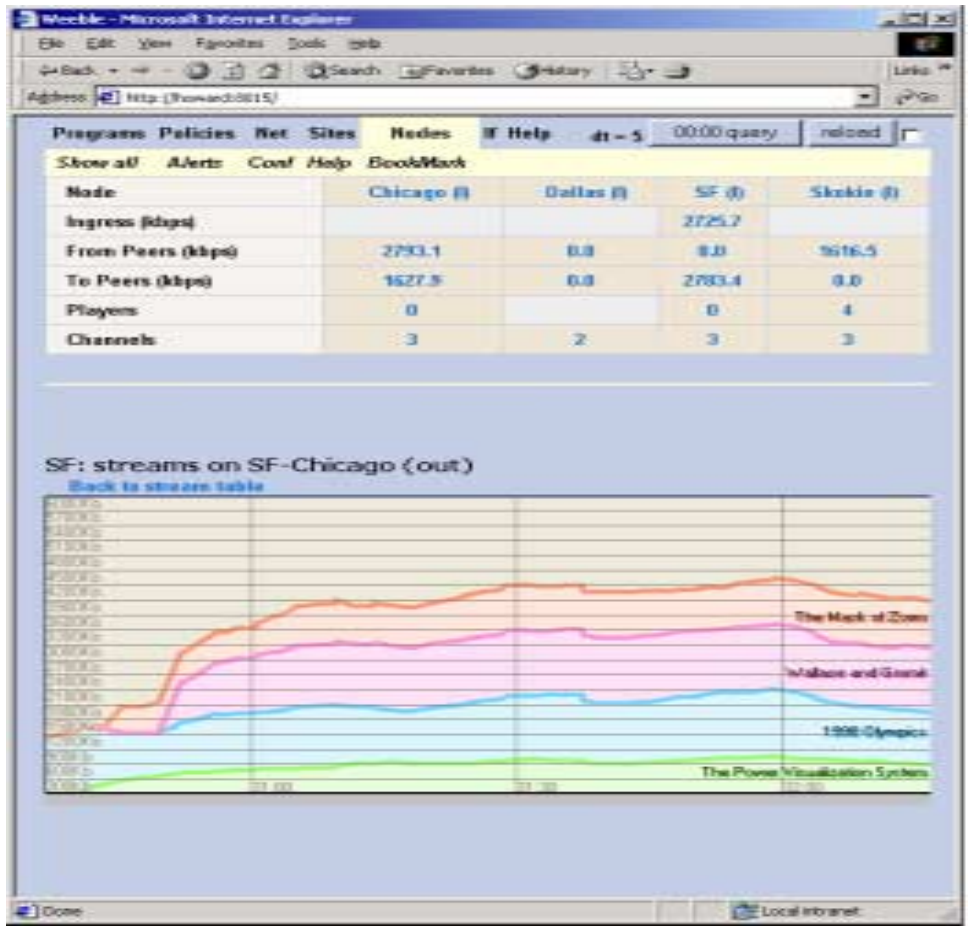
fastforward networks

Broadcast Overlay Architecture



fastforward networks

Broadcast Management

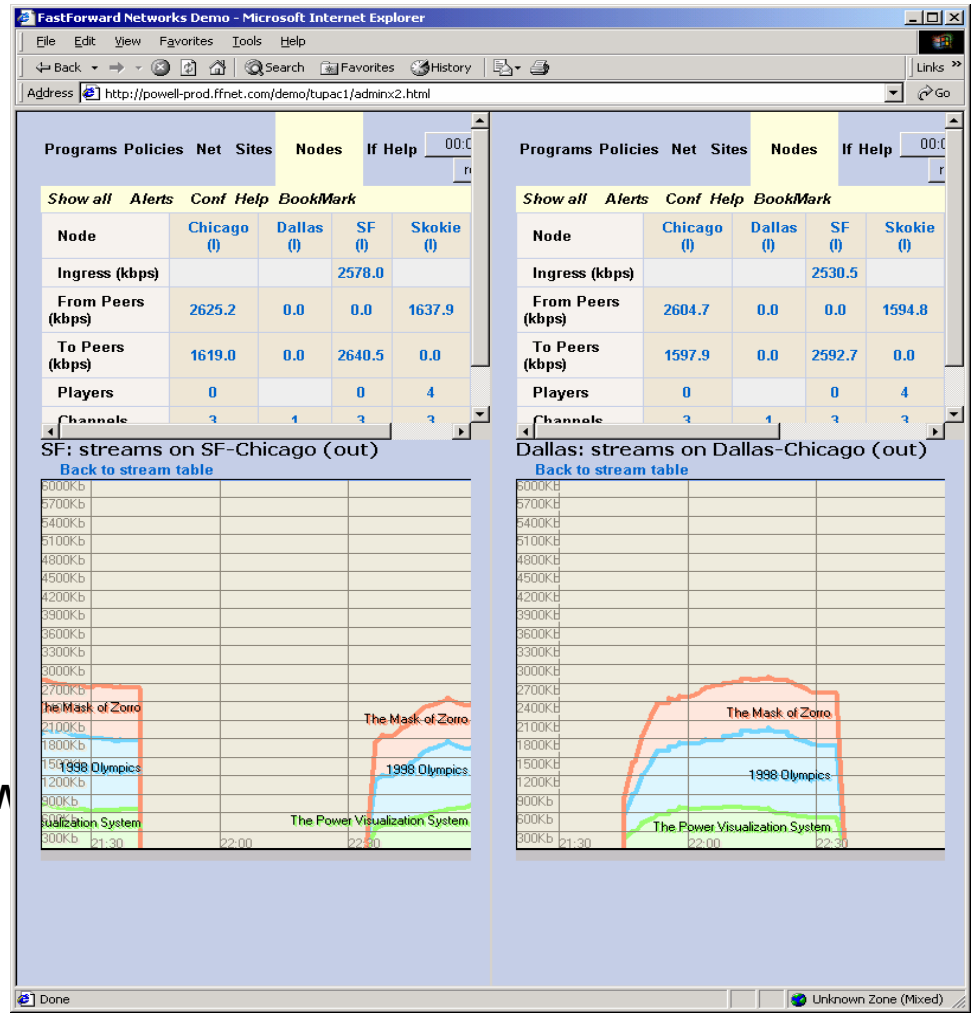


- Scales to millions
- Application-level information for management and tracking
- Works across multiple networks
- Content Producer event programming with ad-hoc query audience statistics

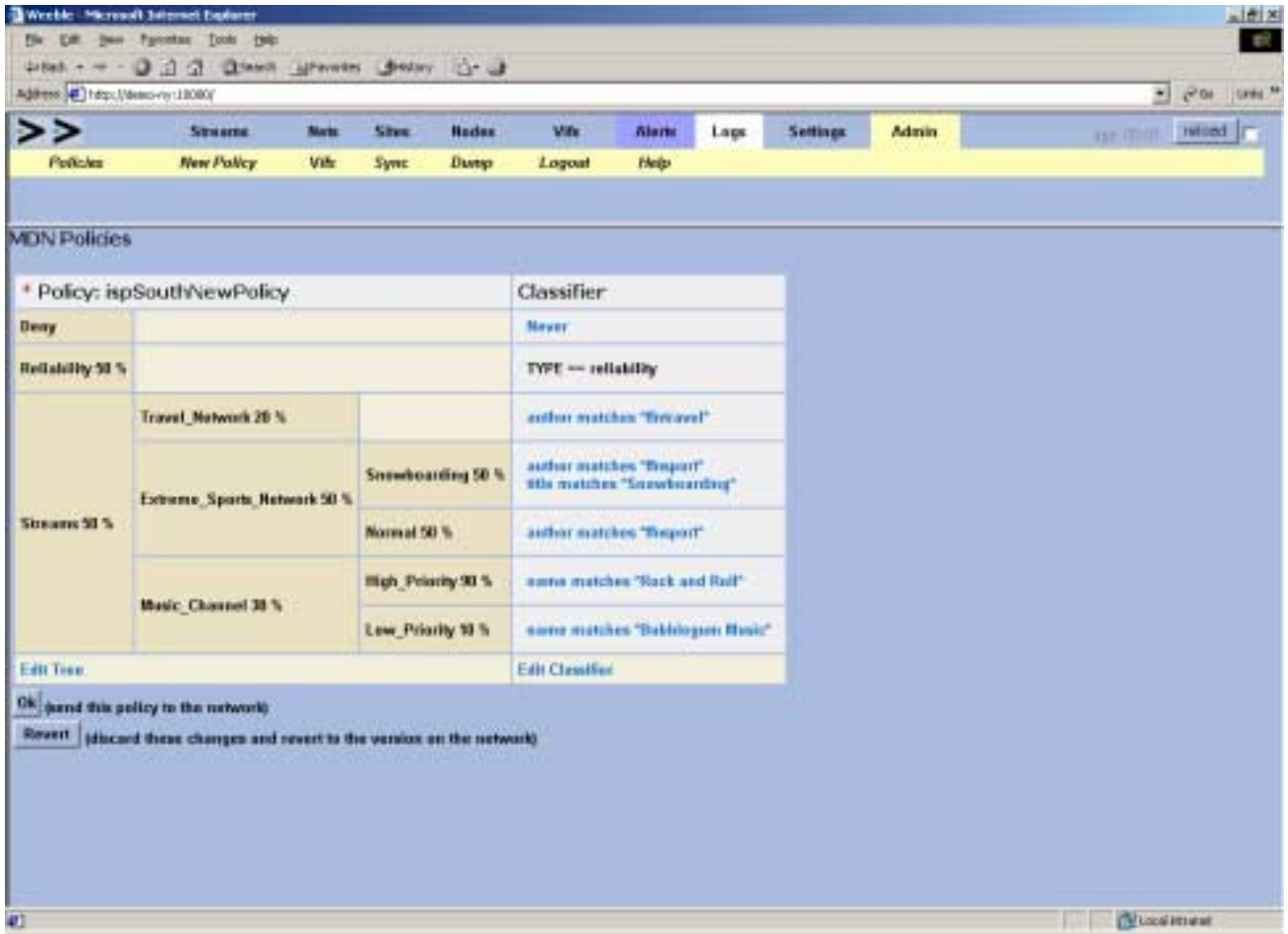
fastforward networks

Broadcast Manager

Node Information



Policy Management

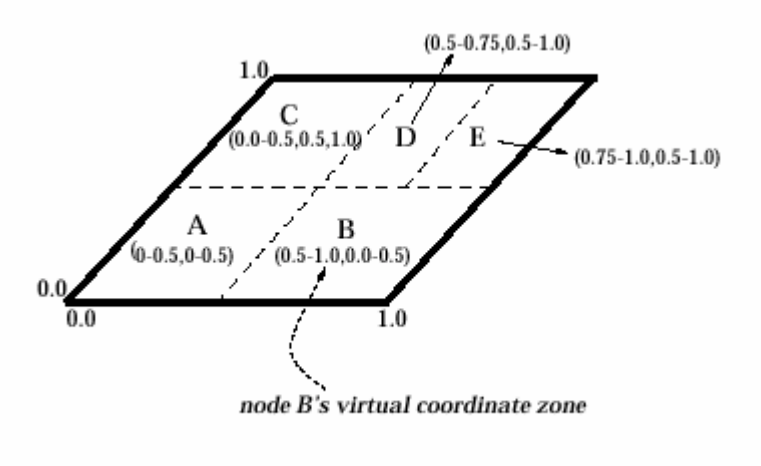


fastforward networks

Example: Content Addressable P2P Networks (CAN)

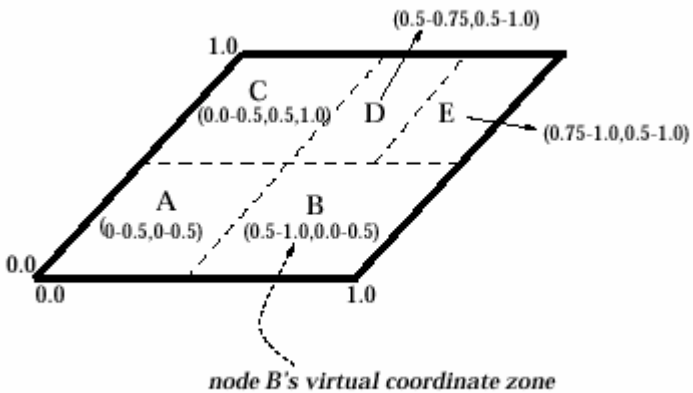
- Ratnaswamy et. al
- First generation P2P (Napster, Gnutella not scalable)
- CAN is one of several recent P2P architectures that
  - impose a structure on the virtual topology
  - locate objects via distributed hash tables DHT
    - Routes queries through the structured overlay
  - attempt to distribute (object, location) pairs uniformly throughout the network
  - support object lookup, insertion and deletion of objects efficiently.
- Others: Chord, Pastry, Tapestry

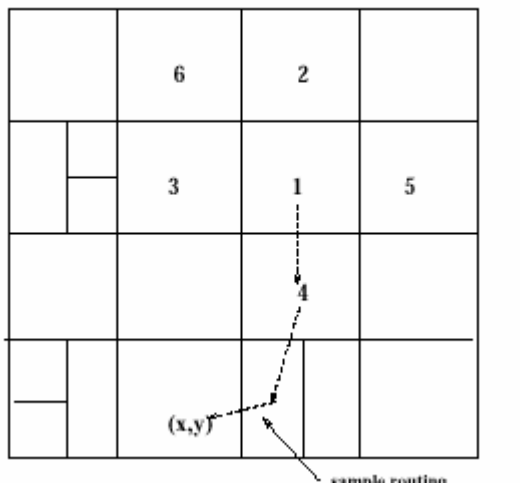
CAN Virtual Topology



- Pick a uniform hash function that maps an object to a point in $[0,1]^d$.
- Divide the d-cube into zones and assign a node to a zone
- Store (object,location) pair in the zone that owns the zone for the hashed value of the object.
- The neighbors of a zone are defined as those nodes whose zones overlap over d-1 dimensions
  - Neighbors of C are A and D
  - Each node stores the addresses of its neighbors

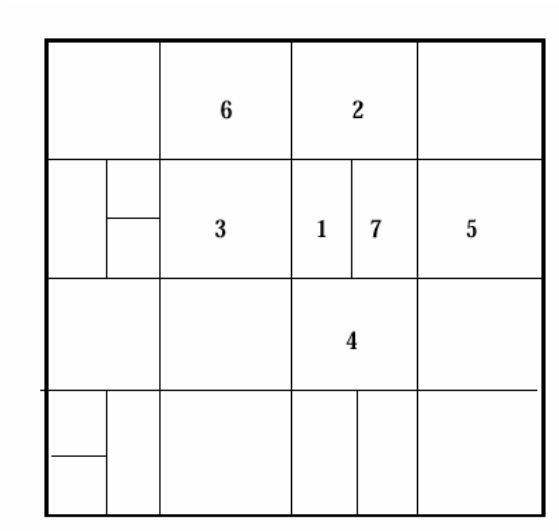
CAN Routing





- If node i wants to find out where object j is stored, it computes h(j) and then sends a lookup message to its neighbor that is closest to h(j). For efficiency assume that the space wraps around (is a torus)
- This continues until the message reaches the node that owns h(j). It then returns the location for object j
- For n equal zones in a d-CAN, each zone has about 2d neighbors
- Routing path lengths are $O(n^{-d})$.

Adding / Deleting nodes



- New node picks a point P at random
- Assuming it can find some overlay node, it sends a join message which is routed to the node that owns that point
- When the message has reached P, the node divides itself in half along one of the dimensions (first x then y etc)
- Pairs are transferred and neighbor sets updated
- Similar reasoning handles departures and failures

Relating Virtual Topology to the Underlying Network

- **Example:**
- Three landmarks
  - 0-30ms: level 0
  - 31-100ms: level 1
  - 101-300ms: level 2
- Node j measures latencies of 10ms, 110ms, 40ms to the three landmarks.
- The bin of node j is
  - $(l_1, l_3, l_2 : 021)$

- Neighbors should be close to each other in terms of latency on the underlying network
- Pick a set of well known landmark hosts
- Each node distributively computes its “bin”
  - Orders the landmark set in increasing order of RTT from it.
  - Latency is partitioned into levels
  - Thus, associated with each landmark, at each node is a rank and a level.
  - These values identify the bin

Constructing the CAN overlay from Bins

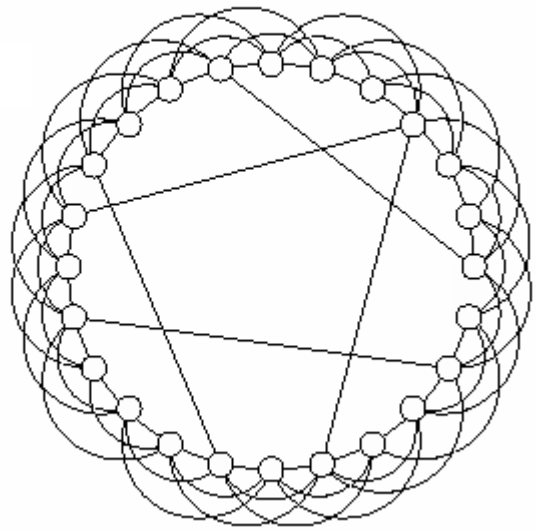
- Let us say that there are B bins possible
  - E.g. ignoring levels and assuming m landmarks there are B=m! bins
- Partition the CAN space into B equal parts
- Allocate nodes in bin b to points at random in the space corresponding to b.
- Popular values of b will have more neighbors
  - Space not uniformly populated – State requirements could be higher
  - But the overlay routing paths will be more comensurate to the underlying network latencies

Constructing Overlays with the Small World Property

- Suppose you were given the name of a person randomly chosen from anywhere in the world.
- How long is the “acquaintance chain” between you and this person?
- Stanley Miligram (kind of) showed empirically that on average the answer is SIX, even though
  - most people don’t know that many people (small # of neighbors)
  - there is considerable overlap between the people two aquaintences know (neighborhoods overlap)
- The small world property is great for routing!
- Need
  - The right topology
  - A distributed routing algorithm to exploit the topology

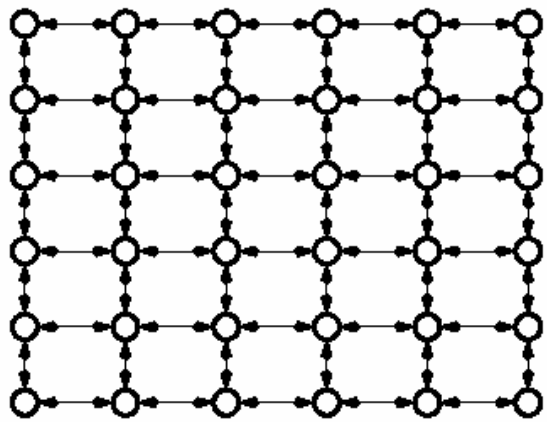
What are the requirements on the Topology?

- Must have
  - Low average degree
  - Small diameter
  - Neighborhood overlap (Clustering)
- Two extreme topologies
  - Random graphs have the small world effect (small avg distance) but no clustering
  - D-Lattice has all three properties but does not have the small world effect
- Watts and Strogatz
  - Rewired lattice. Replace some lattice links with random edges (long range neighbors)
  - Possess the small world property
  - Path is polynomial in log n

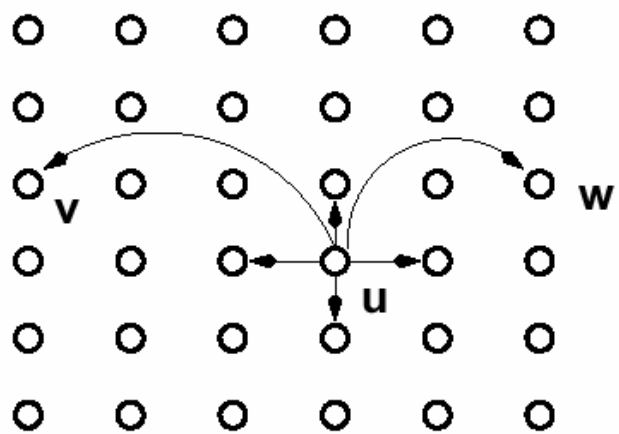


General Model Model… J. Kleinberg

- 2x2 grid, and p,q,r
  - Each node has an edge to all of its neighbors p hops away
  - Each node has q edges picked such that prob that (j,k) is an edge is inversely proportional to d(j,k)^r
  - r=0 yields the Watts Strogatz Model
- Question: If a message is to be routed between two nodes that have local information how many hops would it take on average?
  - Should be at least polynomial in log n
- When routing table size constraints exist this is a critical question!
- Suppose the originating node knows its neighbors and the grid
  - Best routing algorithm is exponential in minimum distance UNLESS r=2.
  - Result holds even if the each node also knows the set of q edges thus far traversed by the message!!



p=1, q=0



p=1, q=2

Intuition for negative result for extreme values of r

- Picking the “right” long contacts is crucial
- Chances are, need to pick several long contacts
- Distributed algorithm is “blind” in this respect
- If r<<2, the long contacts have very little to do with the grid, and can lead message on a “random” walk
- If r>>2, the long contacts are too clustered and are actually “short” themselves

Somewhat finer grained reasoning

- Level sets from node u:
  - A_j is the set of nodes whose dist from dest is between 2^{j+1} and 2^j
- When r=2, u's long contacts are equally likely to be in any A^j
  - A greedy algorithm that tries to route towards the destination has a good chance of being able to pick a useful long contact
- When r>2, bias is towards closer A_j's so once far message tends to stay far for a while
- When r<2 bias is towards further A_j's so even after the message is close, it may have to move away further (jump away) from the destination

Recall Routing Subfunctions…

- **Addressing:** Uniquely identify the nodes
  - host IP address, group address, attributes
  - set is dynamic!
- **Topology Update:** Characterize and maintain connectivity
  - Discover topology
  - Measure “distance” metric(s)
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** Find node identifiers of the destination set
- **Route Computation:** Pick the tree (path)
  - Kind of path: Multicast, Unicast
  - Global or Distributed Algorithm
  - Policy
  - Hierarchy
- **Switching:** Forward the packets at each node

Routing Subfunctions…

- **Addressing:** Uniquely identify the nodes
  - h ddress, attributes
  - se
- **Topology Update:** Characterize and maintain connectivity
  - Discover topology
  - Me
  - Dynamically provision (on slower timescale)
- **Destination Discovery:** F of t set
- **Route Computation:** Pick the tree (path)
  - Kin
  - Glo
  - Pol
  - Hierarchy
- **Switching:** Forward the packets at each node

Structured Topology

Add/Insert Nodes, Binning

Resource Location

Edge Mapping

Application Level Routing. E..g streaming broadcast

Structured Topology

Overlay Effectiveness: Performance v/s New Function

- Overlays add new functions to the network infrastructure much faster than by trying to integrate them in the router
- But when they are in the path (process every packet)
  - Overlay nodes can create performance bottlenecks
  - New end-to-end protocols may not work
  - Peer-to-Peer networks in which the overlay nodes are end hosts of the underlying network are not in the path
- Generally better to improve performance by building an “underlay” and add functionality by building an overlay

Routing Lectures Summary

- Routing is the most critical function of a network
- For it to scale it must be distributed and resilient to topological changes
- Routing in large systems is hierarchical for reasons of performance, administration and governance
- It is currently unknown how to design a robust routing protocol that yields low latency paths and that takes into account the dynamic nature of a large network
- It is currently unknown how to design a policy routing protocol for the internet that is resilient to router misconfiguration
- Overlay networks are quick and effective ways to add new functionality to the network
- The central problem in designing P2P networks involves solving a distributed search problem that involves routing.
- Routing is a wide open research area