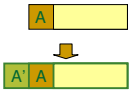
Overlay Networks

EECS 122: Lecture 18

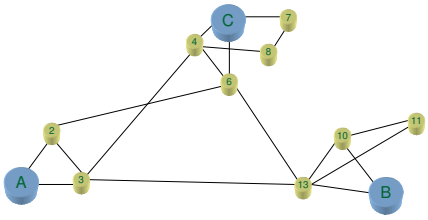
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley

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

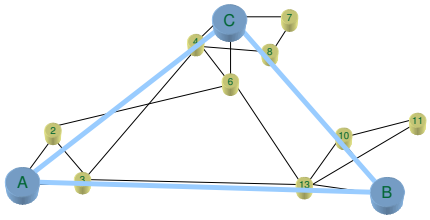


Overlay Concept: Going Up



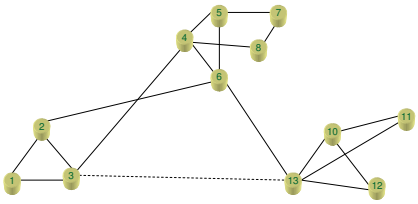
Overlay Network Nodes

Overlay Concept: Going Up

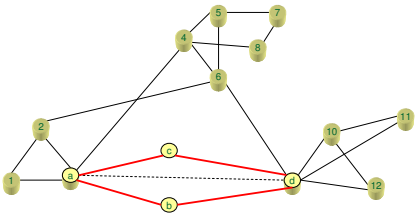


- Overlay Networks are extremely popular
- MBONE, Akamai, Virtual Private Networks, Napster, Gnutella
- Overlay Networks may even peer!

Overlay Concept: Going Down

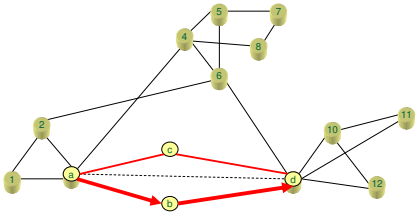


IP Network is the Overlay…



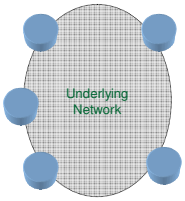
ATM links can be the "physical layer" for IP

IP Network is the overlay

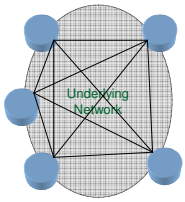


Virtual Circuit under Datagram!

Routing On the overlay

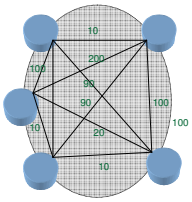


Routing on the Overlay



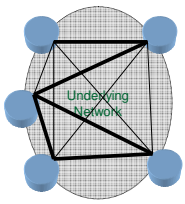
- The underlying network induces a complete graph of connectivity
  - No routing required!

Routing on the Overlay



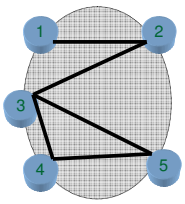
- The underlying network induces a complete graph of connectivity
  - No routing required!
- But
  - One virtual hop may be many underlying hops away.
  - Latency and cost vary significantly over the virtual links
  - State information may grow with E (n^2)

Routing Issues



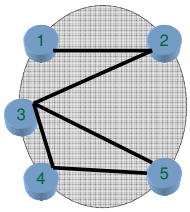
- The underlying network induces a complete graph of connectivity
  - No routing required!
- But
  - One virtual hop may be many underlying hops away.
  - Latency and cost vary significantly over the virtual links
  - State information may grow with E (n^2)

Routing Issues



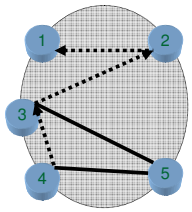
- The underlying network induces a complete graph of connectivity
  - No routing required!
- But
  - One virtual hop may be many underlying hops away.
  - Latency and cost vary significantly over the virtual links
  - State information may grow with E (n^2)

Relating the virtual topology to the underlying network

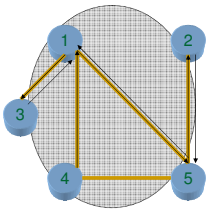


Message from 4 → 1

Relating the virtual topology to the underlying network



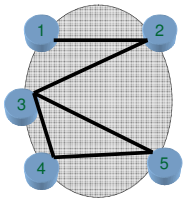
Message from 4 → 1
4→3→2→1



4→**1**→3→**1**→ **5** →2→ **5** →1

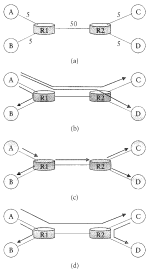
Extreme Inefficiencies Possible

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)

End System Multicast Example



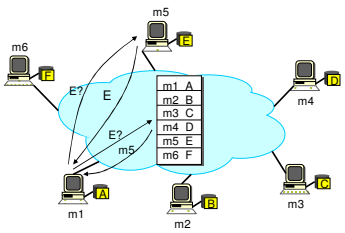
a) Physical Topology
b) Naive unicast transmission
c) IP multicast
d) Application level multicast

From "Computer Networks", by Peterson & Davie

Kinds of Overlay Networks

- Three kinds of Overlays
  1. Only Hosts: Peer to Peer Networks (P2P)
     - Example: Gnutella, Napster
  2. Only Gateway nodes: Infrastructure Overlays
     - Content Distribution Networks (CDNs)
       - Example: Akamai
  3. Host and Gateway Nodes:
     - Virtual Private Networks
- Overlay node structure
  - Regular: Chord, Pastry
  - Adhoc: Gnutella
- Functions
  - Route Enhancement: Better QoS, Application Level Multicast
  - Resource Discovery: P2P

Example: Napster

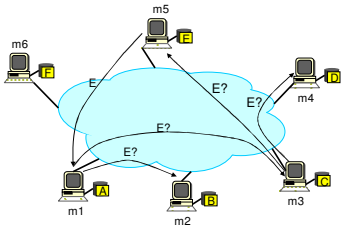


Gnutella

- Distribute file location
- Idea: multicast the request
- Hot to find a file:
  - Send request to all neighbors
  - Neighbors recursively multicast the request
  - Eventually a machine that has the file receives the request, and it sends back the answer
- Advantages:
  - Totally decentralized, highly robust
- Disadvantages:
  - Not scalable; the entire network can be swamped with request (to alleviate this problem, each request has a TTL)

Gnutella: Example

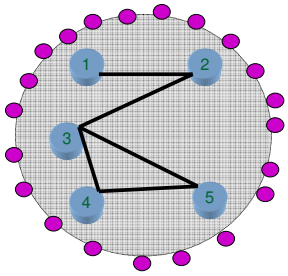
- Assume: m1's neighbors are m2 and m3; m3's neighbors are m4 and m5;…



Outline

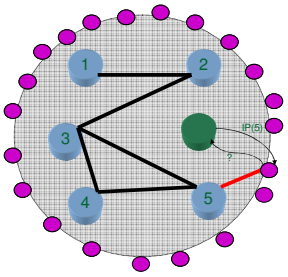
- Infrastructure Overlays
  - Adding performance and route functionality
  - Resource Discovery
- P2P Overlays
  - Resource Discovery in Gnutella
- Example of an Infrastructure Overlay
  - Application Level Multicast
- Example of a P2P Overlay
  - Content Addressable Networks
- Conclusions

Infrastructure Overlays



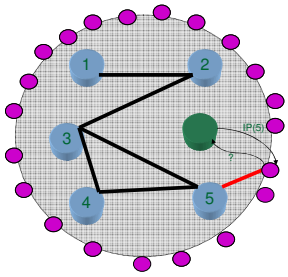
- Overlay network users are not directly connected to the overlay nodes
  - E.g. Akamai

Overlay Routing: Edge Mapping



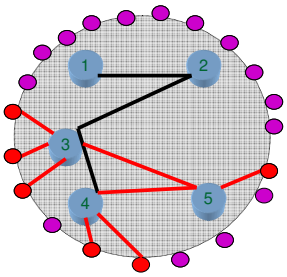
- Overlay network users are not directly connected to the overlay nodes
  - E.g. Akamai
- User must be redirected to a "close by" overlay node
- Edge-Mapping, or redirection function is hard since
  - # potential users enormous
  - User clients not under direct control
- When overlay clients are directly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

Overlay Routing: Edge Mapping



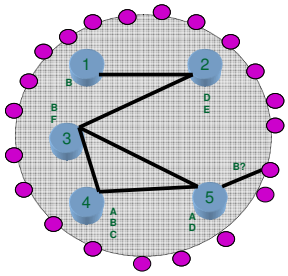
- Overlay nodes interconnect clients
- Enhance nature of connection
  - Multicast
  - Secure
  - Low Loss
- Much easier to add functionality than to integrate into a router

Overlay Routing: Adding Function to the route



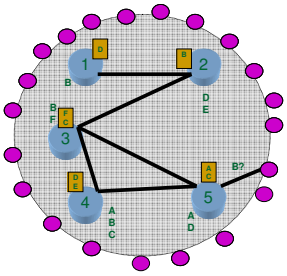
- Overlay nodes interconnect clients
- Enhance nature of connection
  - Multicast
  - Secure
  - Low Loss
- Much easier to add functionality than to integrate into a router
- Overlay nodes can become bottlenecks

Overlay Routing: Resource Location



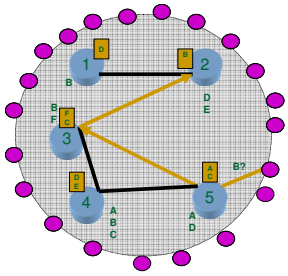
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must "search" for "closest" node that has the resource
  - E.g. find the least loaded server that has a piece of content and that has low network latency to client

Overlay Routing: Resource Location



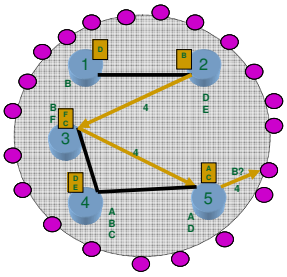
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must "search" for "closest" node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single "index" is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

Overlay Routing: Resource Location



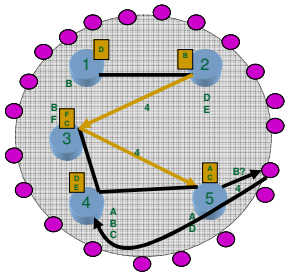
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must "search" for "closest" node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single "index" is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

Overlay Routing: Resource Location



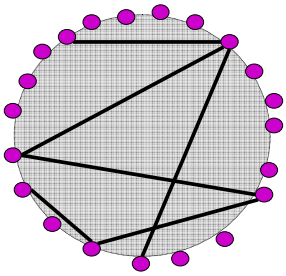
- Overlay network may contain resources. Eg.
  - Servers
  - Files
- Client makes request for resource
- Overlay must "search" for "closest" node that has the resource
  - E.g. find the least loaded server that has a piece of content and that is has low network latency to client
- A single "index" is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

P2P Overlays



- Overlay network users are not directly connected to the overlay nodes
  - E.g. Napster, Gnutella
- No edge mapping problem
- No gateways to maintain

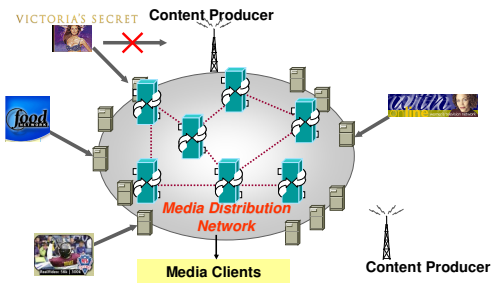
But

- Nodes have limited resources
  - storage,
  - connectivity
  - computational power

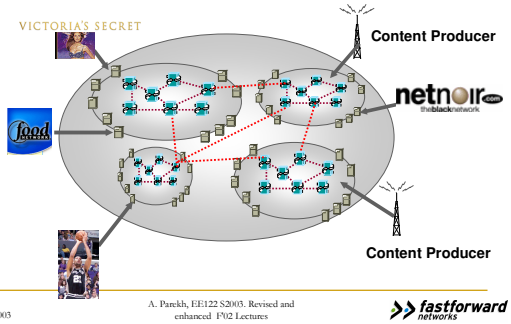
Summary

- Two kinds of overlays functions
  - Overlay provides access to distributed resources
  - Overlay facilitates communication among other client applications
- Two kinds of virtual topologies
  - Structured: mesh, ring etc.
  - 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

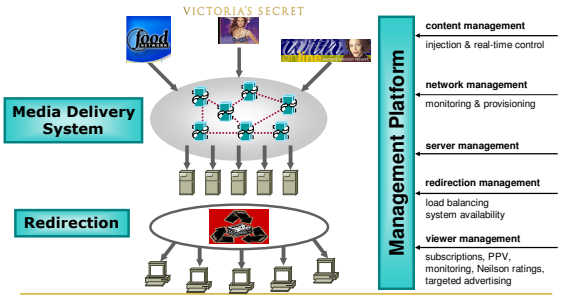
Example: Application Level Multicast



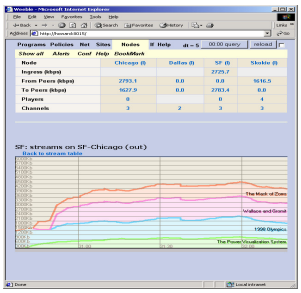
The Broadcast Internet



Broadcast Overlay Architecture

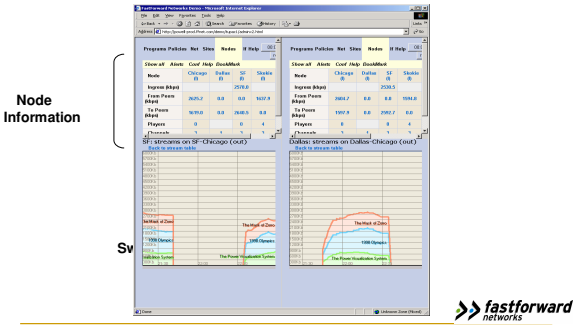


Broadcast Management

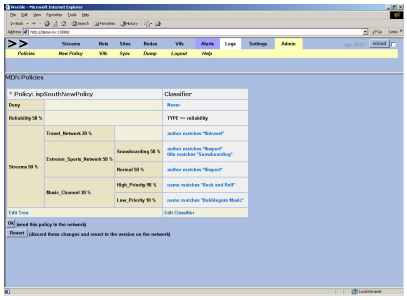


- Application-level information for management and tracking
- Works across multiple networks
- Content Producer event programming with ad-hoc query audience statistics

Broadcast Manager



Policy Management



Example: Content Addressable P2P Networks (CAN)

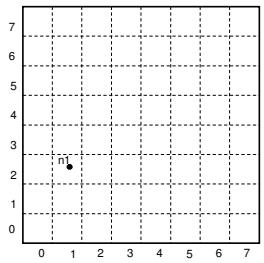
- CAN is one of several recent P2P architectures that
  - imposes a structure on the virtual topology
  - uses a distributed hash-table data structure abstraction
    - Note: item can be anything: a data object, document, file, pointer to a file…
  - routes queries through the structured overlay
  - attempts to distribute (object, location) pairs uniformly throughout the network
  - supports object lookup, insertion and deletion of objects efficiently.
- Others: Chord, Pastry, Tapestry

Content Addressable Network (CAN)

- Associate to each node and item a unique *id* in an *d*-dimensional space
- Properties
  - Routing table size O(*d*)
  - Guarantee that a file is found in at most $d * n^{1/d}$ steps, where *n* is the total number of nodes

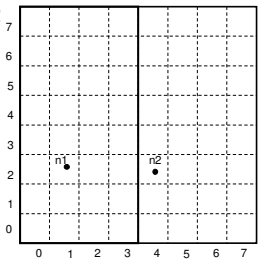
CAN Example: Two Dimensional Space

- Space divided between nodes
- All nodes cover the entire space
- Each node covers either a square or a rectangular area of ratios 1:2 or 2:1
- Example:
  - Assume space size (8 x 8)
  - Node n1:(1, 2) first node that joins → cover the entire space



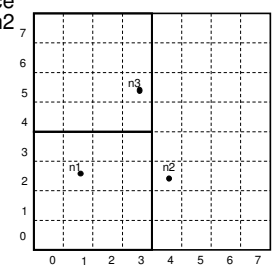
CAN Example: Two Dimensional Space

- Node n2:(4, 2) joins → space is divided between n1 and n2



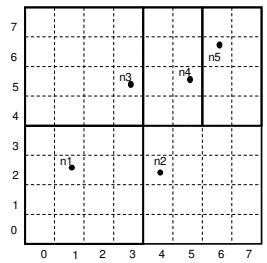
CAN Example: Two Dimensional Space

- Node n2:(4, 2) joins → space is divided between n1 and n2



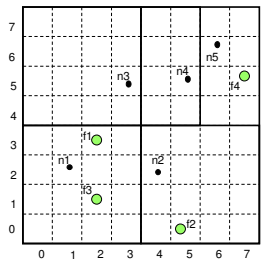
CAN Example: Two Dimensional Space

- Nodes n4:(5, 5) and n5:(6,6) join



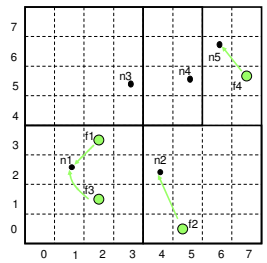
CAN Example: Two Dimensional Space

- Nodes: n1:(1, 2); n2:(4,2); n3:(3, 5); n4:(5,5);n5:(6,6)
- Items: f1:(2,3); f2:(5,1); f3:(2,1); f4:(7,5);



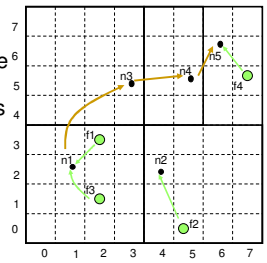
CAN Example: Two Dimensional Space

- Each item is stored by the node who owns its mapping in the space

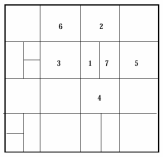


CAN: Query Example

- Each node knows its neighbors in the *d*-space
- Forward query to the neighbor that is closest to the query *id*
- Example: assume n1 queries f4



Adding/Deleting nodes



- New node picks a point P at random
- Assuming it can find any overlay node, it sends a join message to the node which 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"
  - Nodes in the same bin are "close" to each other
  - 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

Your standard Networking Functions…

- **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

And Their Overlay Analogs

- **Addressing:** Uniquely identify the nodes
  - Structured Topology
- **Topology Update:** Characterize and maintain connectivity
  - Add/Insert Nodes, Binning
- **Destination Discovery:** Resource Location Edge Mapping
- **Route Computation:** Pick the tree (path)
  - Application Level Routing. E..g streaming broadcast
  - Structured Topology
- **Switching:** Forward the packets at each node

Conclusions

- Overlays are an irreversible trend in network
- Overlays add new functions to the network infrastructure much faster than
  - by trying to integrate them in the router
  - relying on a infrastructure service provider on deploy the function
- Disadvantages
  - Overlay nodes can create performance bottlenecks
  - New end-to-end protocols may not work since the overlay nodes don't understand them
- Generally better to improve performance by building an "underlay" and add functionality by building an overlay