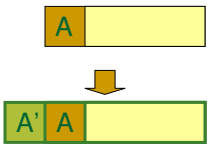
Overlay Networks

EECS 122: Lecture 18

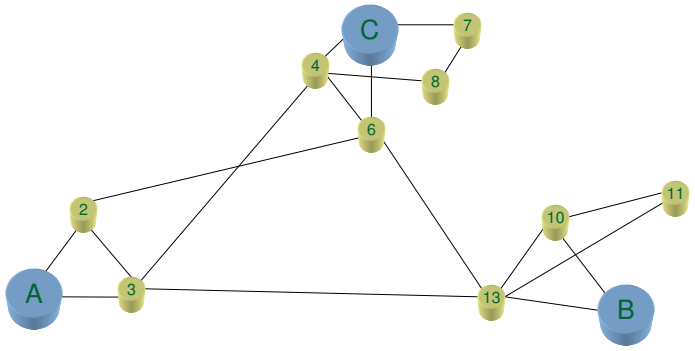
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley

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

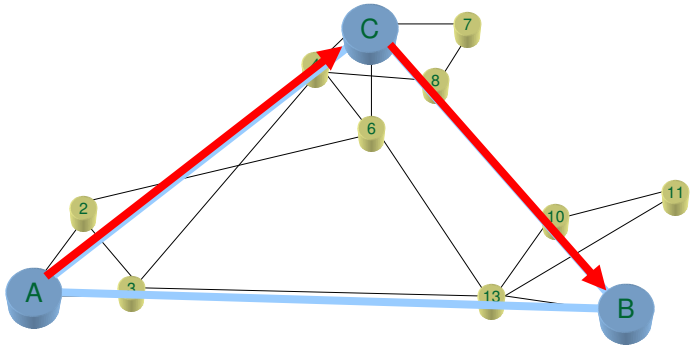


Overlay Concept



Overlay Network Nodes

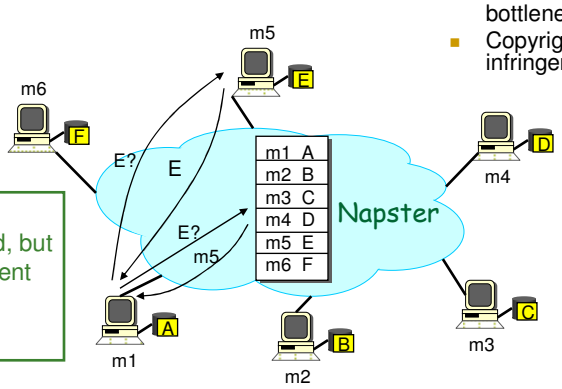
Overlay Concept



- Overlay Networks are extremely popular
- Akamai, Virtual Private Networks, Napster, Gnutella, Kazaa, Bittorrent

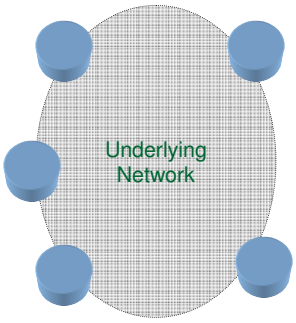
Why overlay? Filesharing Example

- Single point of failure
- Performance bottleneck
- Copyright infringement

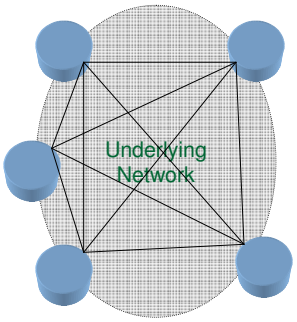


file transfer is decentralized, but locating content is highly centralized

Build an Overlay Network

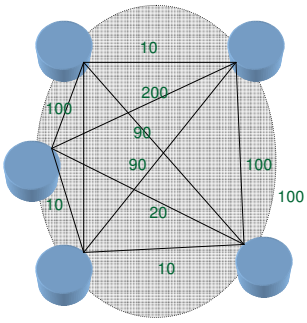


Build an Overlay Network



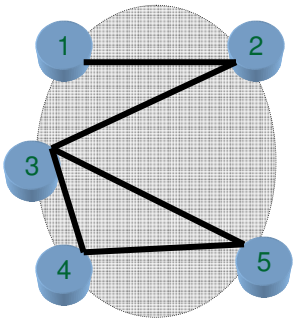
- The underlying network induces a complete graph of connectivity
  - No routing required!

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)

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)
- 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)

Kinds of Overlay Networks

- Two 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
- Overlay node structure
  - Regular: Chord, Pastry
  - Adhoc: Gnutella
- Functions
  - Route Enhancement: Better QoS, Application Level Multicast
  - Resource Discovery: P2P

Rest of the Lecture

- P2P Overlays
  - Resource Discovery in Gnutella and Kazza
  - Content Addressable Networks
- Infrastructure Overlays
- "Underlays"
- Conclusions

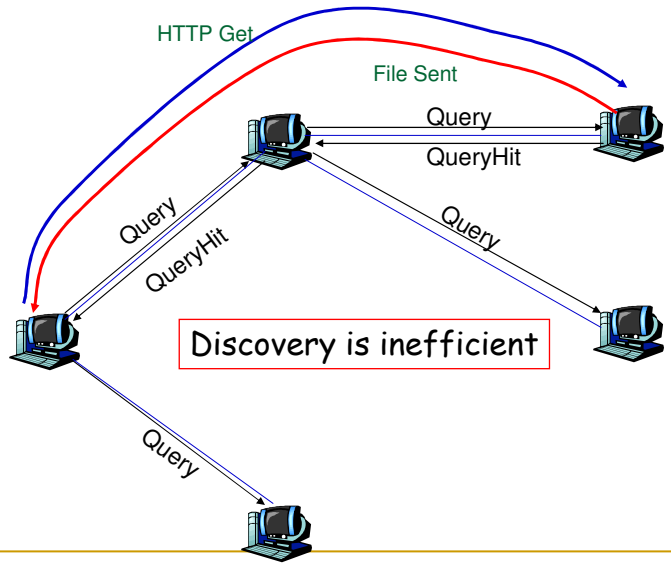
Gnutella

- Create an overlay network to share files in a fully distributed manner
- Peers run the Gnutella client
  - no central server
- public domain protocol
- Each peer connected with about 10 others

Gnutella

- Idea: broadcast the request for a file (e.g. the song "I'm Free!")
- How to find a file:
  - Flood the request
  - Eventually a machine that has the file receives the request, and it sends back a "hit" message to the requestor with its IP address
  - Requestor sends an HTTP "Get" message to the IP address
  - File sent as the response to the Get message
- 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: protocol

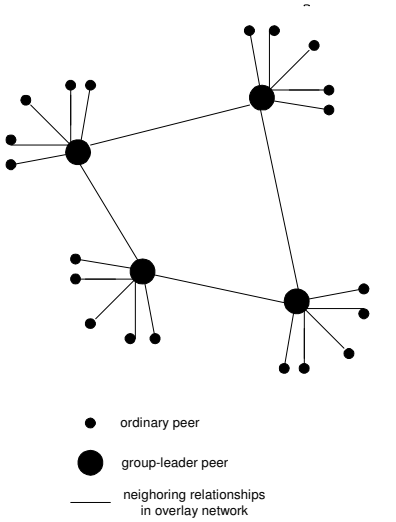


Gnutella: Peer joining

1. New peer "Alice" has a list of other peers to "bootstrap"
   - Only some may be active
2. Sends a "Join" message (called Ping in Gnutella) to first active peer
3. Join message is flooded k times
4. All peers receiving Ping message respond with a Pong message
5. Alice decides which peers to connect with on the overlay network

Exploiting heterogeneity: KaZaA

- Each peer is either a group leader or assigned to a group leader.
  - TCP connection between peer and its group leader.
  - TCP connections between some pairs of group leaders.
- Group leader tracks the content in all its children.



KaZaA: Querying

- Each file has a hash and a descriptor
- Client sends keyword query to its group leader
- Group leader responds with matches:
  - For each match: metadata, hash, IP address
- If group leader forwards query to other group leaders, they respond with matches
- Client then selects files for downloading
  - HTTP requests using hash as identifier sent to peers holding desired file

Rest of the Lecture

- P2P Overlays
  - Resource Discovery in Gnutella and Kazza
  - Content Addressable Networks
- Infrastructure Overlays
- Conclusions

Content Addressable P2P Networks (CAN)

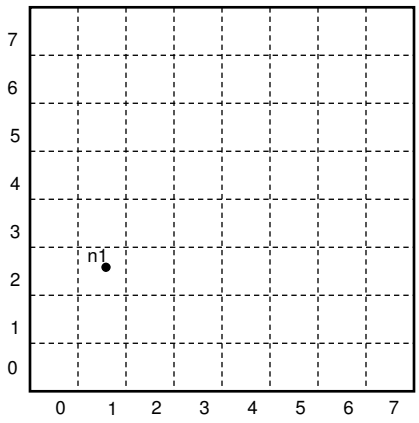
- CAN is one of several recent P2P architectures that
  - imposes a structure on the virtual topology
  - uses a distributed hash-table data structure abstraction
    - Note: item can be anything: a data object, document, file, pointer to a file…
  - routes queries through the structured overlay
  - 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 with each node and item, a unique *id* in an *d*-dimensional space
  - Example for d=3: A node might be called (1,11,5)
  - Example for d=3: A song might be called (2,3,11)
- 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

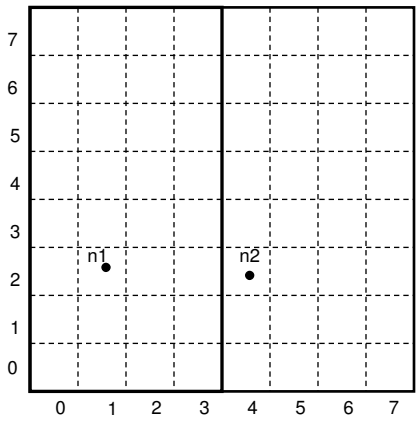
CAN Example: d=2

- Space divided between nodes
- All nodes collectively "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



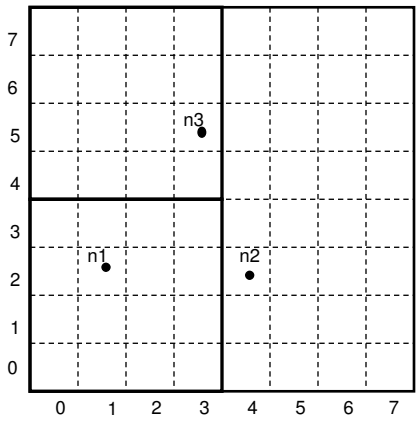
Covered space divided between nodes

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



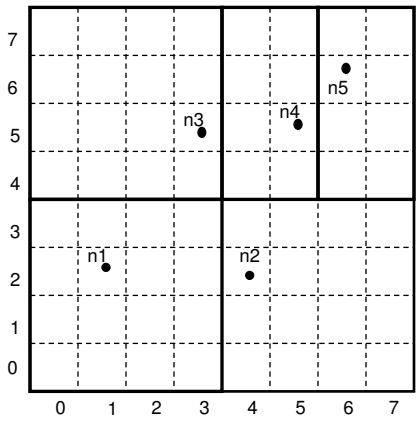
Nodes continue to join

- Node n2:(4, 2) joins → space is divided between n1 and n2
- Node n3: (3,5)



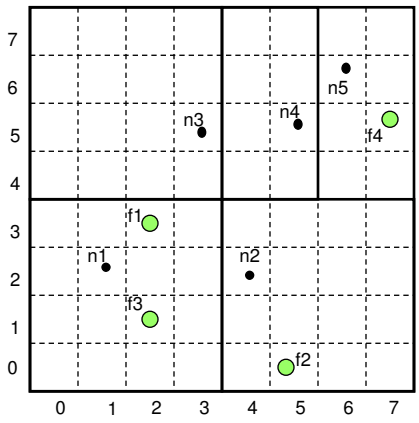
Nodes continue to join

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



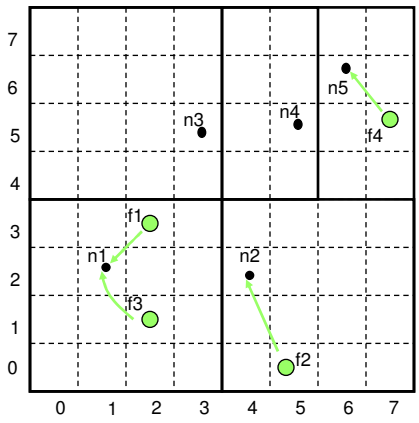
Items are also mapped in the same space

- Items: f1:(2,3); f2:(5,1); f3:(2,1); f4:(7,5);



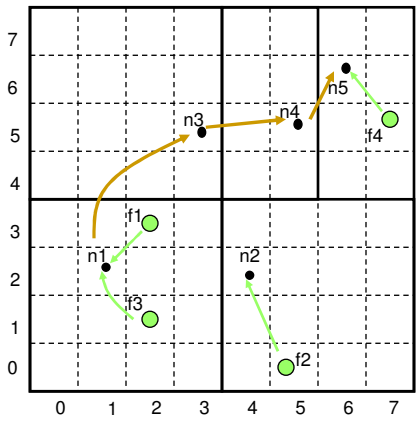
CAN Example: Two Dimensional Space

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

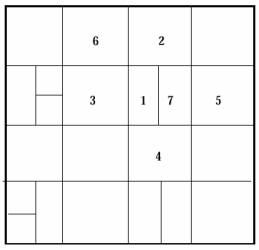


CAN: Query Example

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

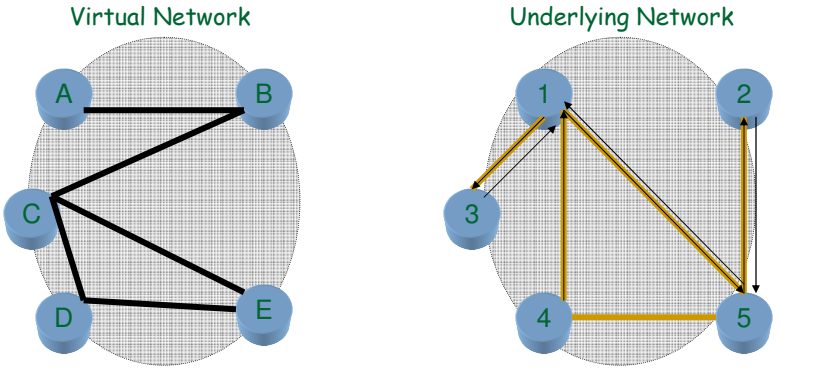


Adding/Deleting nodes



- 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 the virtual topology to the underlying network



Message from D → A
D→C→B→A

D→1→C→1→ 5 →B→ 5 →A

Extreme Inefficiencies Possible

Picking good virtual links

- Why not just have a new node send pings to all the others to figure out what it is close to?
  - Doesn't scale
- Why not utilize the underlying routing infrastructure?
  - Can't do this easily. This is an overlay network!
- What to do?

CAN: Binning

- Pick a set of well known "landmark" hosts
- Each node distributively computes its "bin"
  1. Ping each Landmark
  2. Latency is partitioned into levels
  3. Order the landmark distances in increasing order of RTT
  4. This values identify a bin
- Nodes in the same bin are "close" to each other
- When a new node comes on, pick an address such that
  - There is at least one neighbor that is in a "close by" bin

- **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 : 012)

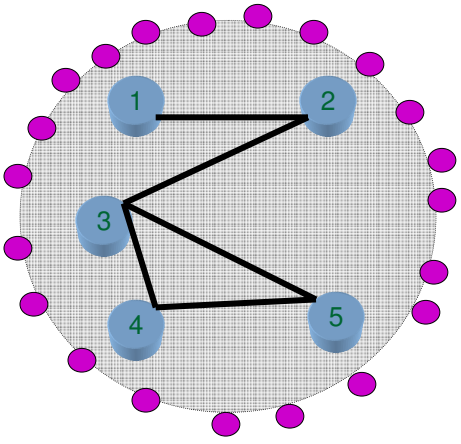
CAN Discussion

- DHT based discovery is faster than the flooding based approach of Gnutella
  - More distributed Kazaa
- Some attention paid to ensuring that the virtual topology is not too inefficient relative to the underlying network
- Disadvantage: Considerable overhead with additions and deletions

Rest of the Lecture

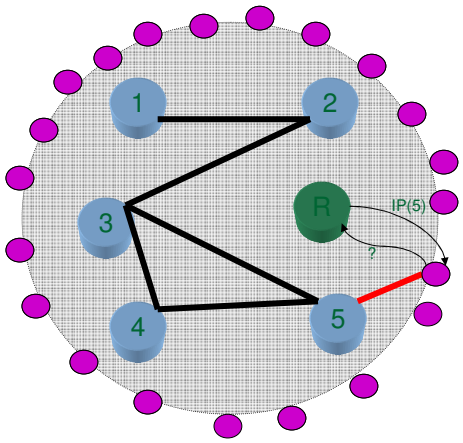
- P2P Overlays
  - Resource Discovery in Gnutella and Kazza
  - Content Addressable Networks
- Infrastructure Overlays
- Underlays
- Conclusions

Infrastructure Overlays



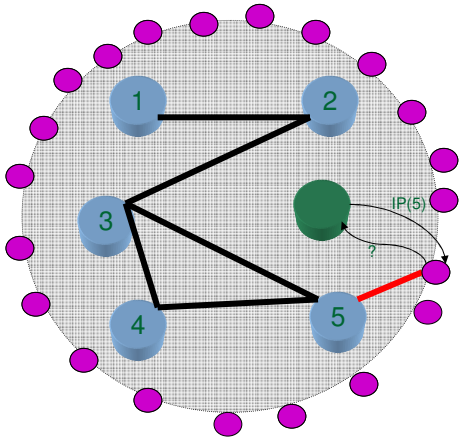
- Overlay network users are not directly connected to the overlay nodes
  - E.g. Akamai

Overlay Routing: Edge Mapping



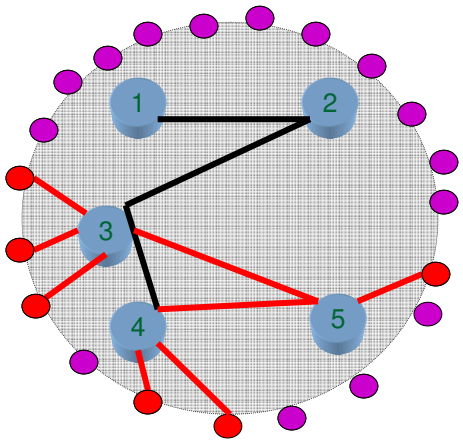
- Overlay network users are not directly connected to the overlay nodes
  - E.g. Akamai
- User must be redirected to a "close by" overlay node
- Edge-Mapping, or redirection function is hard since
  - # potential users enormous
  - User clients not under direct control

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

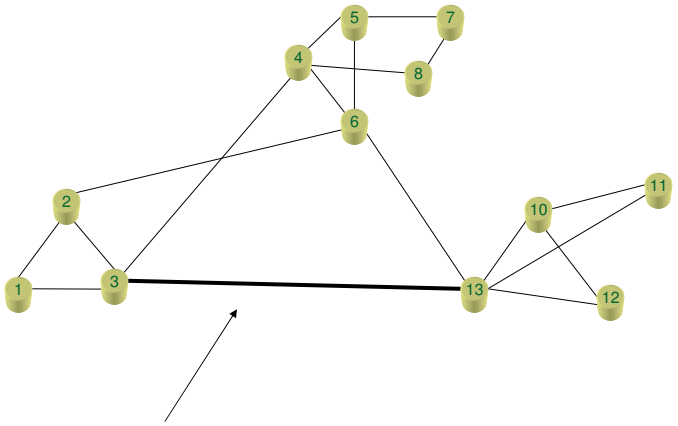


- Overlay nodes interconnect clients
- Enhance nature of connection
  - Multicast
  - Secure
  - Low Loss
- Much easier to add functionality than to integrate into a router
- Overlay nodes can become bottlenecks

Rest of the Lecture

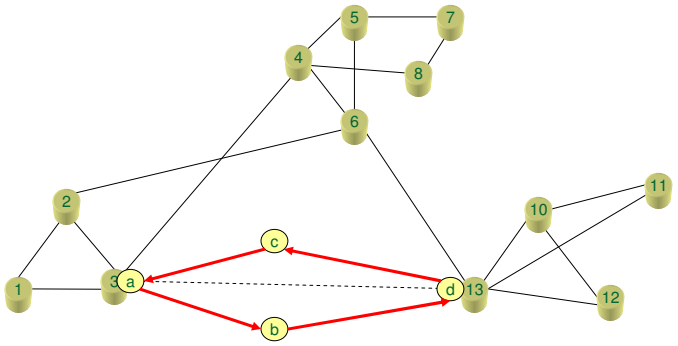
- P2P Overlays
  - Resource Discovery in Gnutella and Kazza
  - Content Addressable Networks
- Infrastructure Overlays
- Underlays
- Conclusions

Overlay Concept: Going Down



Need this link to be very reliable and fast!

IP Network is the Overlay…

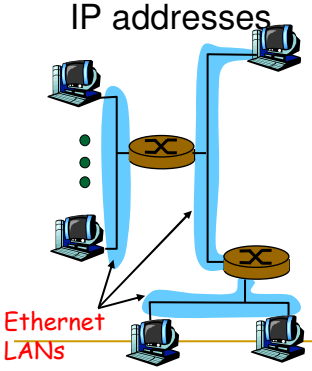


IP Routers 3 and 13 attach to a virtual circuit network
e.g. ATM
The IP network "sees" the virtual circuit network as a link
This is called "Link Virtualization" and is commonly deployed

IP-Over-ATM

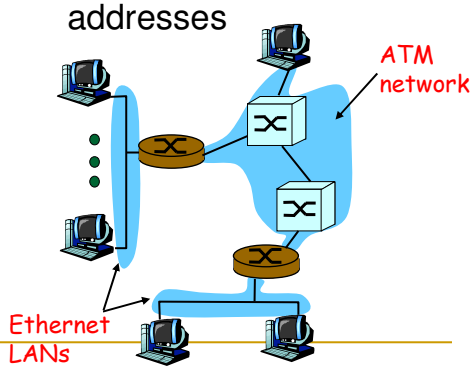
Classic IP only

- 3 “networks” (e.g., LAN segments)
- MAC (802.3) and IP addresses

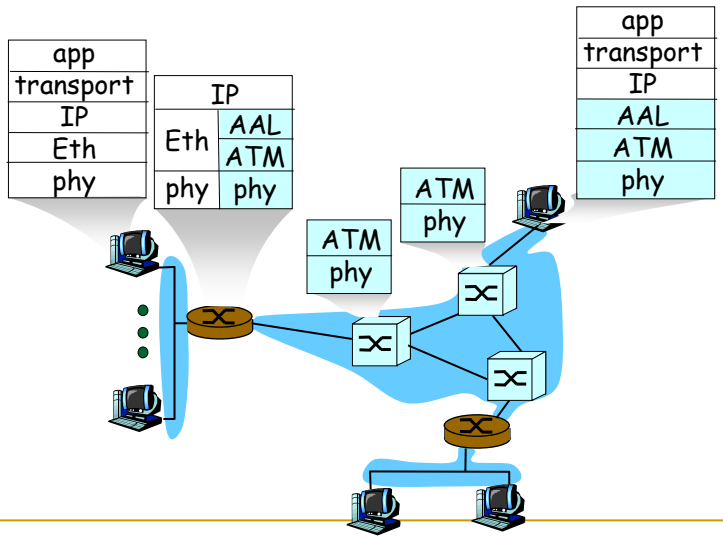


IP over ATM

- replace “network” (e.g., LAN segment) with ATM network
- ATM addresses, IP addresses



IP-Over-ATM



Summary

- Two kinds of overlays functions
  - Access to distributed resources
  - Better network performance
- 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
  - Overlay Routing Protocol
  - Edge Mapping
  - Resource Location

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