eecs489 Computer Networks

Lecture 8: Content Delivery Infrastructure: Peer-to-Peer

2015 Internet Traffic Analysis

Sandvine's Global Internet Phenomena Report:
https://www.sandvine.com/trends/global-internet-phenomena/

Content Distribution

Most popular content can only be served if it is highly replicated across multiple servers

- reduce load at origin server
- improve performance for end users

Most Content Delivery Infrastructures (CDI) have a large number of servers distributed across the Internet and cache content on these servers

Content Delivery Infrastructure

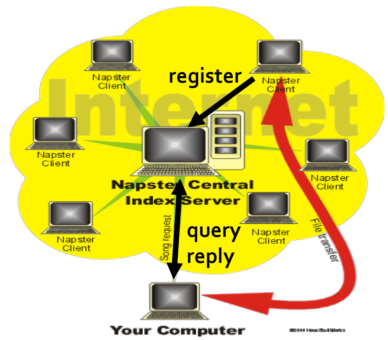
Peer-to-peer (p2p):

- hybrid p2p with a centralized server
- pure p2p
- hierarchical p2p
- end-host (p2p) multicast

Content-Distribution Network (CDN)

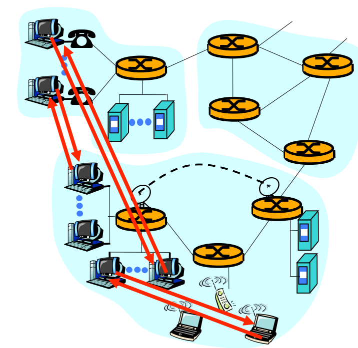
Hybrid P2P and Centralized Server

Napster:
- P2P file transfer
- centralized file search:
  - peers register IP address and content at a central index server
  - peers query central index server to locate content



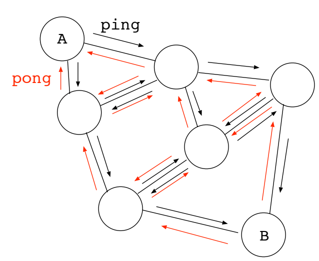
Pure P2P Architecture

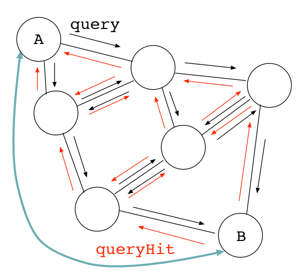
- no always-on server
- arbitrary end systems directly communicate
- peers are intermittently connected and change IP addresses
- example: Gnutella
- highly scalable (why?)
- but difficult to manage
  - how to find peer?
  - how to find content?



Gnutella

- no centralized index server
- network discovery using ping and pong messages
- file discovery using query and queryHit messages
- both ping and query messages are forwarded using the flooding algorithm: forward on all links except incoming one
- previously seen messages are not further forwarded
- new version of gnutella uses KaZaA-like supernodes

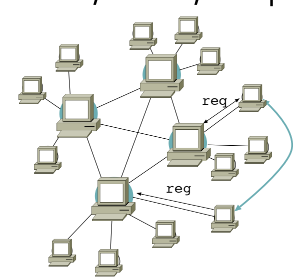




Hierarchical P2P

FastTrack used by KaZaA, Groskster, iMesh, Morpheus

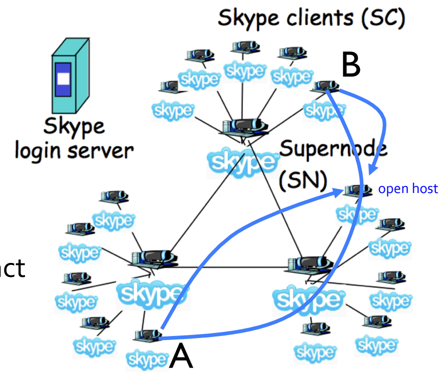
- hierarchical architecture
  - peers divided into supernodes and ordinary nodes
  - each supernode keeps an index of all its children's files
  - requests are sent to supernodes
  - supernodes query each other for files not in their local indices
- ordinary nodes are "promoted" to supernodes if they have enough resources and have stayed on network long enough
- parallel download of files



Hierarchical P2P: Skype

Skype forms a hierarchical P2P:

- index mapping usernames to IP addresses is distributed across supernodes
  - searches for Skype users are sent to supernodes
  - supernodes query each other for users not in their local index
- supernodes choose a peer to act as relay for two NATted users



eDonkey/eMule also builds a hierarchical network, but the "supernodes" are dedicated servers, not just more equal peers

Freenet: Anonymous P2P

- no index server
- requester doesn't connect directly to content provider
- instead, content is passed in a bucket-brigade fashion from provider to requester
- subsequent request for the same content is satisfied from the nearest cache
- requester cannot differentiate provider from a cache holder or a forwarding peer (allows for anonymity)

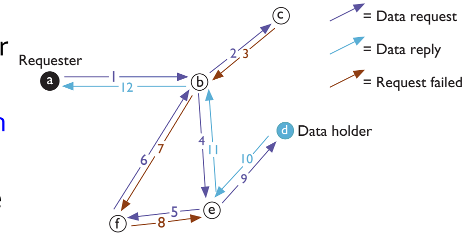
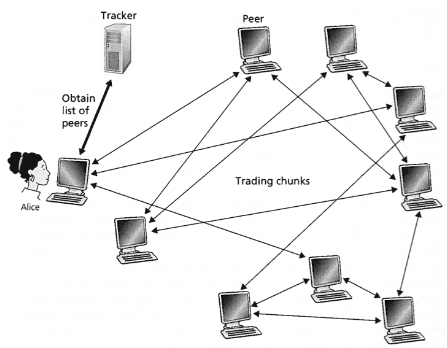


Figure 1. Typical request sequence. The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

BitTorrent



Content distribution:

- content is divided into N pieces of 16KB each and sent to N peers

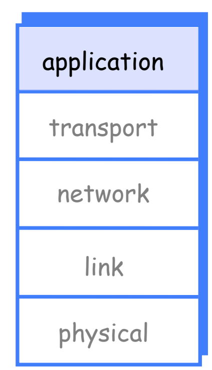
Content download:

- to download a file, a peer must first register with a Tracker
- Tracker returns a random list of peers who have the file
- peer opens about 5 TCP connections to the provided peers
- a peer will only upload to peers from whom it can also download ("tit-for-tat")

Summary: P2P Overlay Networks

P2P applications/peers need to:

- track identities and IP addresses of peers
  - there may be a large number of peers
  - peers may come and go frequently (high churn)
  - can't keep track of all peers
- route messages among peers
  - may be multi-hop



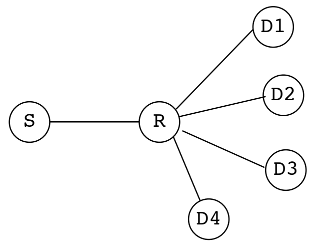
Overlay network

- peers have to do both naming and routing
- IP becomes "just" the low-level delivery substrate
  - all IP routing is opaque

Modes of Delivery

Unicast, broadcast, multicast

Assuming a video conference involving S, $D2$, and $D3$



- unicasting: two copies of packets from S are sent over the SR link
- broadcasting: one copy of packets sent from S to all destinations, but packets sent to $D1$ and $D4$ unnecessarily
- multicasting: one copy of packets from S is sent over the SR link, R then sends one copy each to $D2$ and $D3$

Multicast Delivery

Uses of multicasting:

- video conferencing, distance learning, distributed computation, p2p delivery, multi-player gaming, etc.

Multicast design goals:

- can support millions of receivers per multicast group
- receivers can join and leave any group at any time
- senders don't know all receivers
- senders don't have to be members of a group to send
- there could be more than one senders per group

Multicast Group Management

Issues in multicast group management:

1. how to advertise/discover a multicast group?
2. how to join a multicast group?
3. delivering multicast packets to the group

IP multicast:

- use multicast addresses as anonymous rendezvous point:
  - IPv4: Class-D (`224.0.0.0` to `239.255.255.255` [RPC 3171])
    - 265 M multicast groups at most
  - IPv6: multicast prefix: `FF00::/8`
- create a well-known multicast group (address) to advertise/discover multicast groups

Multicast Delivery

IP multicast:

- sender sends a single packet to the IP multicast address
- multicast data is sent best-effort, using UDP (why?)
- routers deliver packets out all interfaces that has a receiver belonging to the multicast group
- receivers join groups by informing upstream routers, e.g., by using Internet Group Management Protocol (IGMP)
- not uniformly deployed throughout the Internet

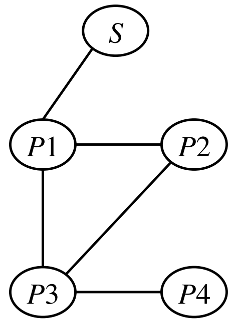
Flood and Prune

How to ensure that only one copy of packet from S is forwarded by $P3$ to $P4$?

- keep track of packet sequence number
- only forward packet that comes from shortest path from (to) source

How to ensure that only one copy of packet from S reaches $P3$?

- only forward if self is on neighbor's shortest path from (to) source
- prune ($P3$ tells $P2$ not to forward packets from S)
  - must be done per source if there are multiple sources, each source forming its own multicast group and (logically) its own multicast tree
  - must periodically flood in case of membership change



End-host Multicast

Issues in multicast group management:

1. how to advertise/discover a multicast group?
2. how to join a multicast group?
3. delivering multicast packets to the group

End-host (p2p) multicast:

- uses a well-known, centralized rendezvous server
- each peer must register with rendezvous server
- rendezvous server returns a (random) list of peers
- each peer can support only a limited number of peers
- avoid sending duplicate messages and looping:
  - if single source, constructs a shortest-path tree rooted at source
  - or uses flood-and-prune algorithm
- prefers peers in same subnet

P2P Challenges

Relative to IP networking:
- much higher function, more flexible
- much less controllable/predictable

Relative to other parallel/distributed systems:
- no administrative organizations
- few guarantees on transport, storage, etc.
- partial failure
- churn
- network bottlenecks and other resource constraints
- trust issues: security, privacy, incentives

Challenges for P2P Networks

1. NAT and firewall:
   - cannot peer with a host you can't address

   Solutions:
   - Gnutella:
     - querier sends PUSH message to responder over the p2p network
     - responder opens a TCP connection to querier and sends over the file
     - no luck if both are behind firewalls
   - KaZaA, eDonkey, Skype:
     - a supernode acts as relay if both peers are behind firewalls
   - Standards to traverse NAT (and firewall!): UPnP, STUN, TURN
2. Download/upload bandwidth asymmetry

   ⇒ needs bandwidth subsidy by content provider or CDN, or suffer long download time