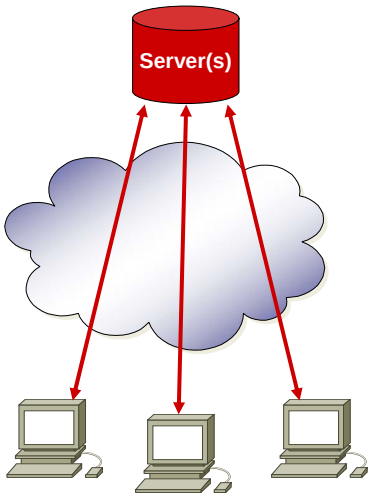
Peer-to-Peer Media Streaming

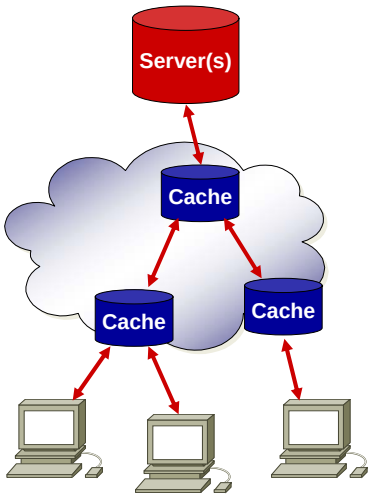
Service Provisioning in the Internet

- Two important questions
  - Where the services or the data reside?
  - How are they found?
- Traditional approach: Client-Server Architectures
  - Centralized or hierarchical structure
    - Within the system organization and/or
    - Within the naming structure
  - Passive caching
  - Active replication
- Distributed systems
  - May be centralized or hierarchical with different roles assigned to nodes
  - May also use equal nodes (“peers”) or
  - Dynamically changing role assignments (e.g., “superpeers”)

Storing Data (Services)

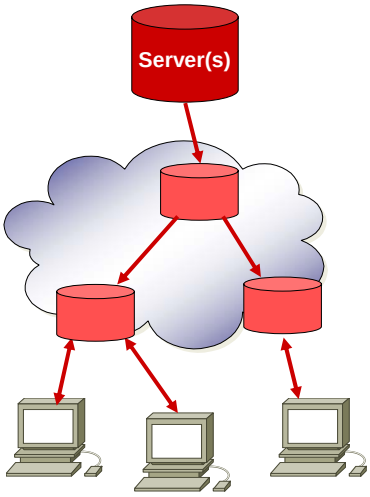


Plain Client-Server

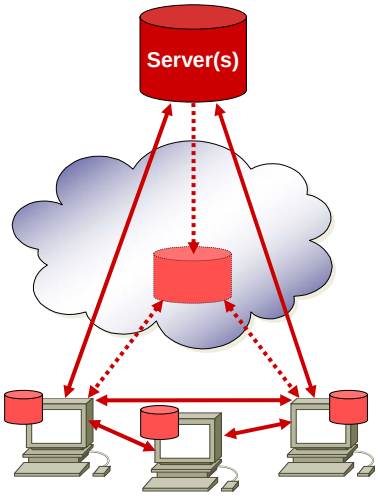


Passive Caching

Storing Data (2)

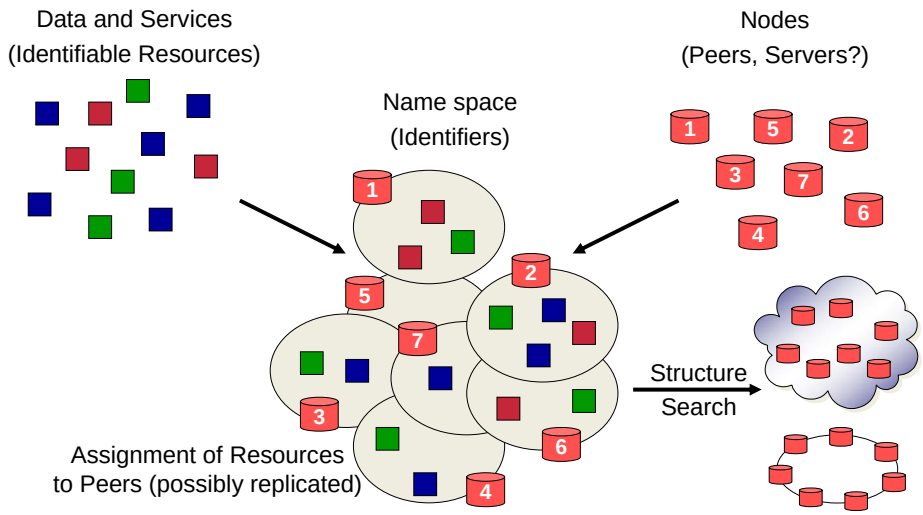


Active Replication:
Content Distribution Network (CDN)



Adding Peer-to-Peer
Elements

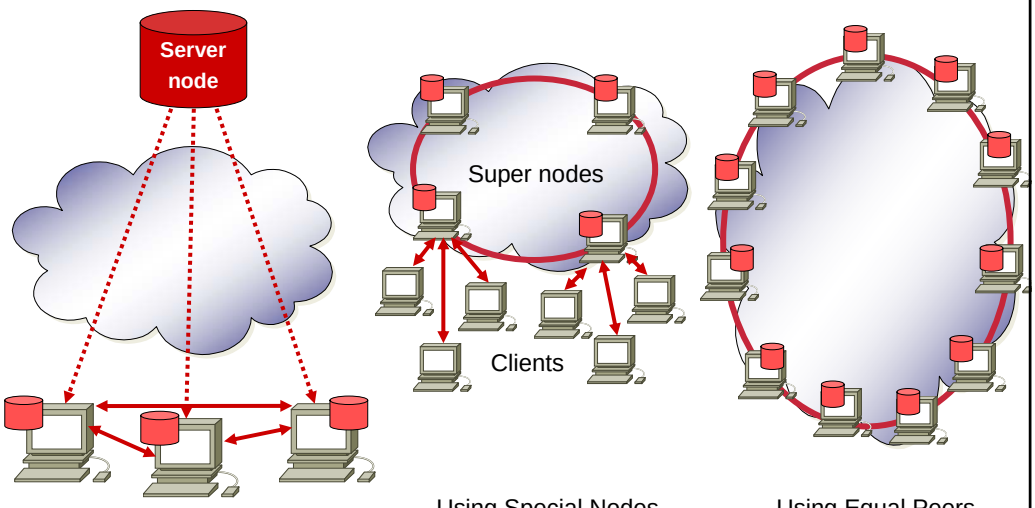
Creating a Peer-to-Peer Overlay



Locating Data (and Services)

- Hierarchical naming
  - Using DNS: http://www.cnn.com/
  - Intercepting requests and distributing/redirecting these to caches/replicas
- (Centralized/replicated) servers
  - Per data resource or service
  - Example: Torrent files and Trackers in BitTorrent
    - BitTorrent can also operate using a DHT (see below)
- Decentralized: spread across (super)peers
  - Structured overlays (e.g., Distributed Hash Tables, DHTs)
    - Chord: ring structure
    - Content-address network (CAN): n-dimensional space
    - Pastry: tree structure
    - Search carried out, e.g., in an iterative or recursive fashion
  - Unstructured overlays
    - Search carried out, e.g., using flooding, random walk

Locating Data (and Services) (2)

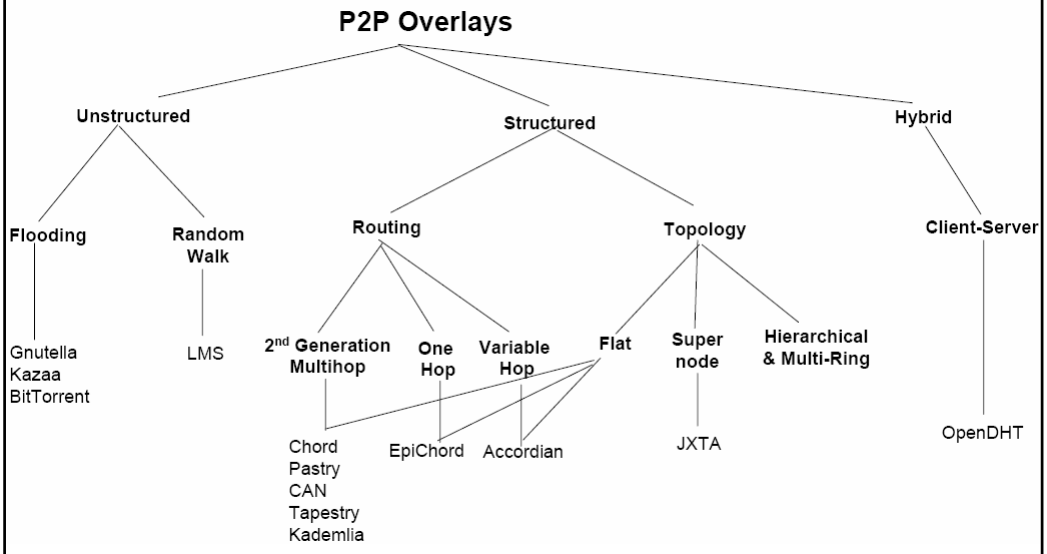


Server-based

Using Special Nodes ("Super nodes/peers")

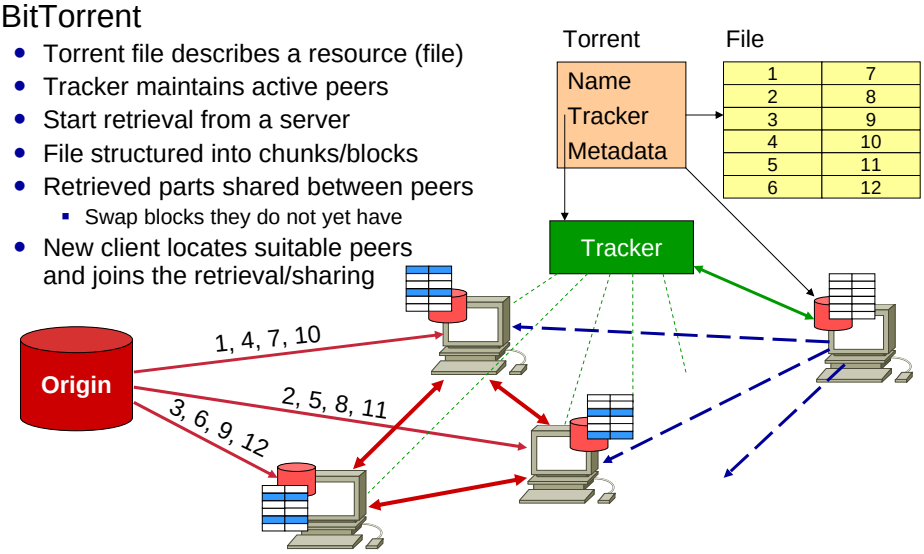
Using Equal Peers

Taxonomy and Examples of P2P Overlays



Example Data Retrieval: File Sharing

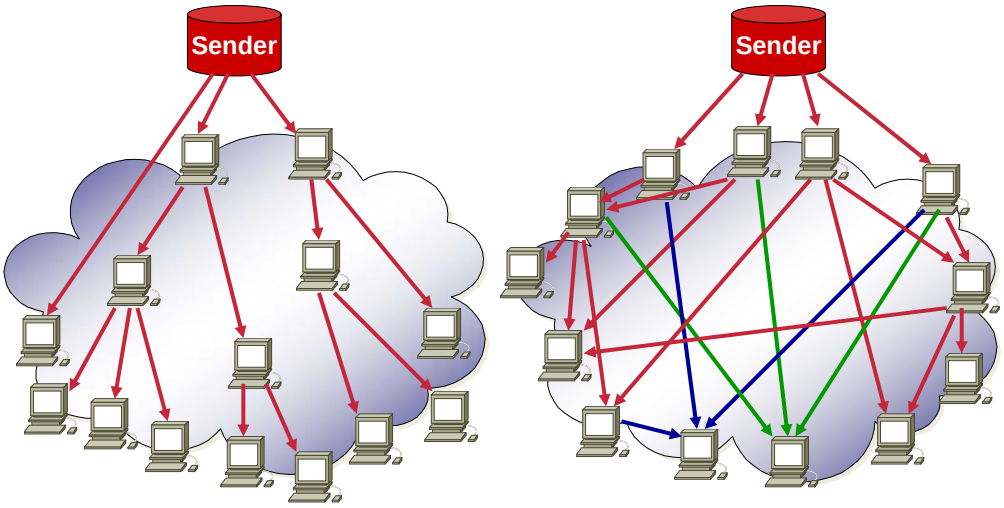
- BitTorrent
  - Torrent file describes a resource (file)
  - Tracker maintains active peers
  - Start retrieval from a server
  - File structured into chunks/blocks
  - Retrieved parts shared between peers
    - Swap blocks they do not yet have
  - New client locates suitable peers and joins the retrieval/sharing



From File Sharing to Streaming

- File sharing is patient
  - No real-time constraints: downloads may take a while
  - Poses less stringent requirements on overlay maintenance and repair
  - Retrieval may occur in arbitrary order
  - Pieces from all over the file (or the entire file) may be available for a long time
  - Elastic application: download rate may vary
    - In case of congestion or during restructuring
- Video-on-demand style retrieval
  - Preferably low start-up delay (limited by user patience)
  - Non-elastic traffic: needs to sustain the stream bitrate towards each client (continuity!)
  - File retrieval from the beginning to the end
    - But “pre-fetching” of later pieces allowed if the available bandwidth permits
  - Old pieces of a file (those already played back) may not be kept
- Live streaming
  - Adds real-time constraint from source to receiver (end-to-end and startup delay)
  - Rough synchronization of playout (not too large differences)
  - Limits the useful buffer sharable between receivers
  - No future data available

Streaming Topologies



Tree (possibly multiple trees)

Mesh

Streaming Topologies: Tree

- Tree construction
  - Based upon minimal delays (P2P RTT measurements)
  - Single-tree or multiple trees
  - Multiple trees (=multiple root nodes) may
    - Provide redundant delivery paths against failures (churn)
    - Provide complementary data flows to each receiver
    - Help dealing with asymmetric upload/download data rates
- Permanent peer monitoring for tree maintenance and repair
  - In case individual links/path fail or become congested
  - In case receivers join and leave (churn)
- Push-based delivery from the source(s) to the receivers
  - Data forwarded along the tree with minimal delay
- Issues
  - Maintaining an “optimal” tree structure incur a lot of overhead
    - Particularly in conjunction with churn
  - Churn may also cause disruptions to downstream receivers

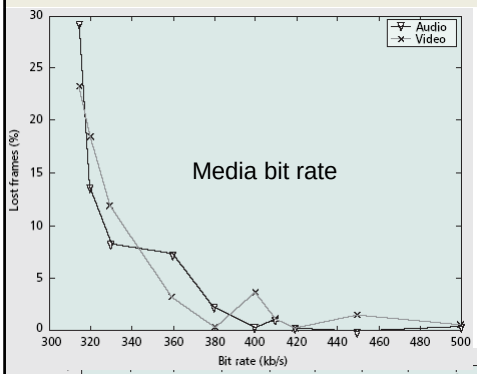
Streaming Topologies: Mesh

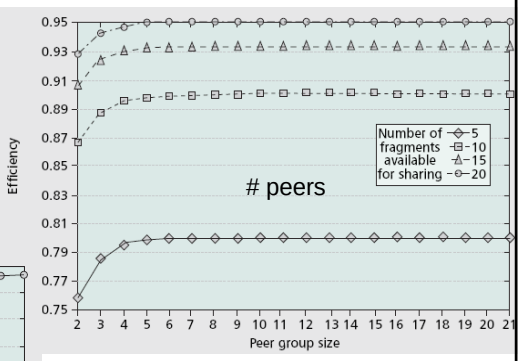
- Mesh construction
  - Newcomers learn, e.g., from a control node about potential peers
  - Find set of peers with minimal delays (P2P RTT measurements)
  - Choosing a subset of the peers initially provided
  - Gossiping protocols to learn about further peers
- Permanent peer monitoring
  - RTT and data rate (to deal with congestion or individual overload)
  - Also used to deal with churn
  - Possibly reconfiguration (substituting peers)
- Active pulling of media segments from peers
  - Exchange of buffer maps (who has which data)
  - Explicit requests for missing chunks (receiver-controlled)
  - Kept locally available for forwarding to other peers
  - Tradeoff between fragment size (delay) and control traffic (overhead)
- Issues
  - Lots of control traffic (explicit pull)
  - Higher end-to-end delay (due to buffering and pull-based forwarding)

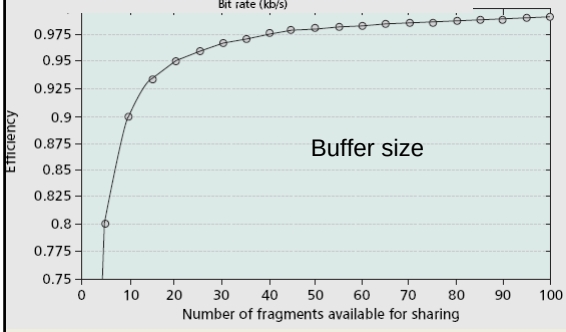
Comparison: Tree vs. Mesh

	Push	Pull
Overlay	Maintains multiple transmission trees	Defines partnership mesh and for the whole streaming session Schedules block of packets
Sign of infeasibility	Reconnection failure	Infeasible transmission schedule
Delay control	Tree structure	Parent selection and scheduling
Loss control	Redundancy and retransmission	Redundancy, scheduling, including retransmission and network coding
Bandwidth utilization	Tree construction and maintenance	Scheduling
Performance optimization	Tree maintenance	Scheduling and parent reselection
Resilience to churn	Tree construction and loss control	Mesh maintenance, scheduling, and loss control
Control cost	Tree maintenance	Mesh maintenance and packet pulling
Trades resilience for	Redundancy and control	Delay and control

Three Data Points







Peer-to-Peer Streaming Systems

	Push/ pull	Tree/ Mesh	Buffer	Playout Delay	Startup Delay	Quality
PPLive	Pull	Mesh	2 min	1 min	20 s-2 min	Rate: 300 – 350 kbit/s Res. 320 x 240 pixel
Coolstreaming	Pull	Mesh	2 min	1 min	1 min	
Anysee	Push	Hybrid	40 s	20-30 s	20 s	
SopCast	Pull	Mesh	1 min	1 min	1-5 min	

- More
  - CoopNet
  - PALS
  - PROMISE
  - SPLIT Stream
  - Bullet

Peer-to-Peer Streaming Scenarios

- Works for user-provisioned content in self-organized groups
- Can also support service providers
  - To reduce the required server capacity (and thus infrastructure cost)
    - # servers, access bit rate
  - Video-on-Demand: To help offering contents from the “long tail”
    - But videos need to be buffered beyond playout
  - Example: set-top boxes offering assistance to other clients
- Issue: Asymmetric DSL bit rates
  - E.g., 8 Mbit/s downstream, 1 Mbit/s upstream gross data rates
  - Supporting higher media bit rates will require support from the service provider
    - To make up the missing data rate per user
  - But: user with symmetric access rates may make up for this (fairness?)
    - Universities, enterprises, fibre/Ethernet-based access
- Meta issue: Digital Rights Management (DRM)?