



Content Distribution Network (CDN)

Amir H. Payberah

(amir@sics.se)

Fatemeh Rahimian

(fatemeh@sics.se)





GOAL

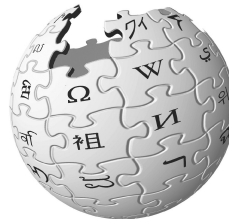
- What is Content Distribution Network (CDN)?
- The solutions for CDN.
- CDN applications
 - File Sharing
 - Media Streaming





Content Distribution Network

CDN is a system of computers, networked together that cooperate **transparently** to deliver content to end users.



Definition 😊



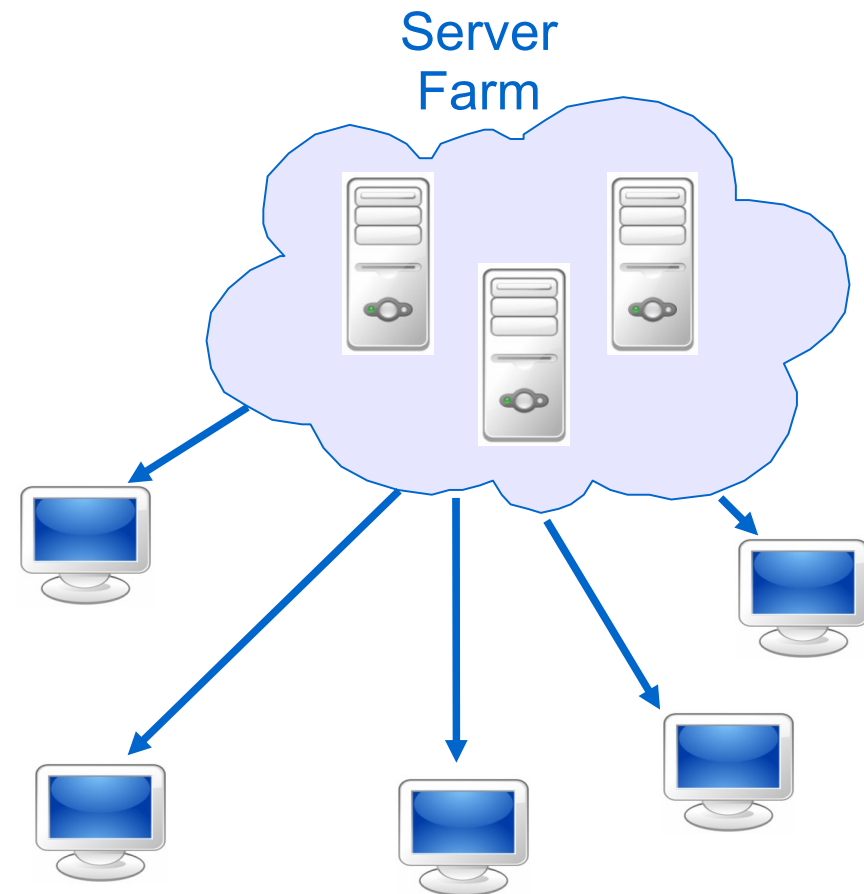


Traditional Solution (Client-Server)

- Akamai



- Youtube





So What Is The Problem?

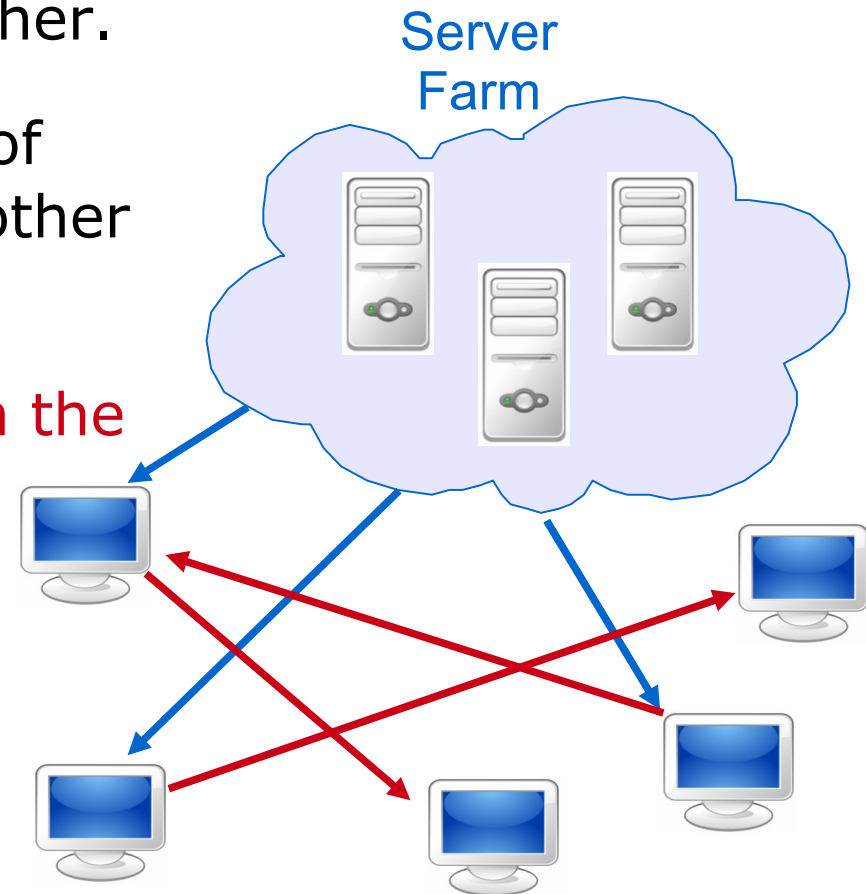






The Smarter Solution (P2P)

- The peers can help each other.
- The peers who have parts of the data can forward it to other requesting peers.
- The capacity increases with the number of peers.





Let's Continue With P2P Solutions






Two Main Questions

- Node discovery
- Data delivery





Two Main Questions

- Node discovery 
- Data delivery





Node Discovery

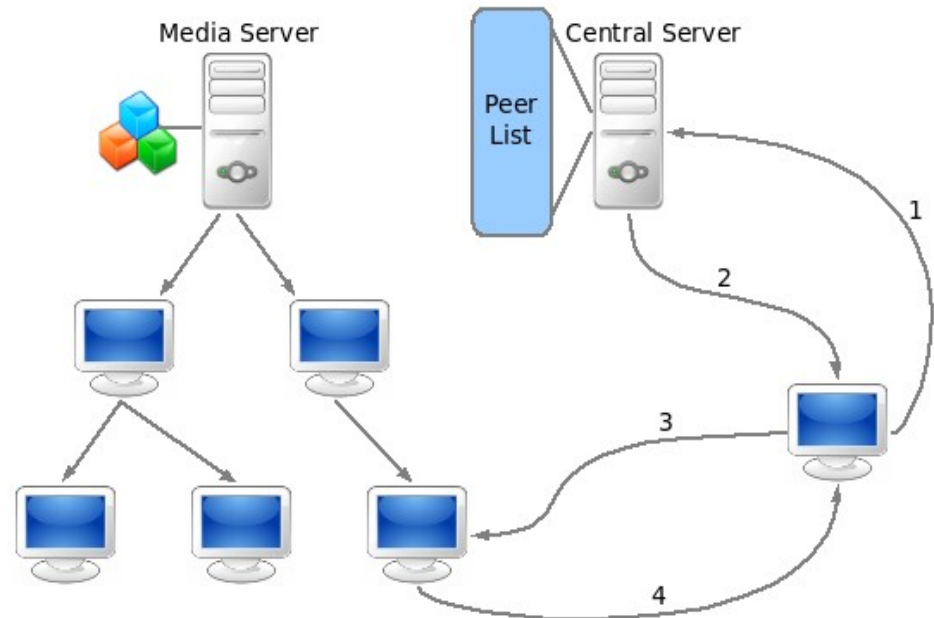
- Centralized method
- Controlled flooding method
- Hierarchical method
- DHT-based method
- Gossip-based method





Node Discovery

- Centralized method
- Controlled flooding method
- Hierarchical method
- DHT-based method
- Gossip-based method

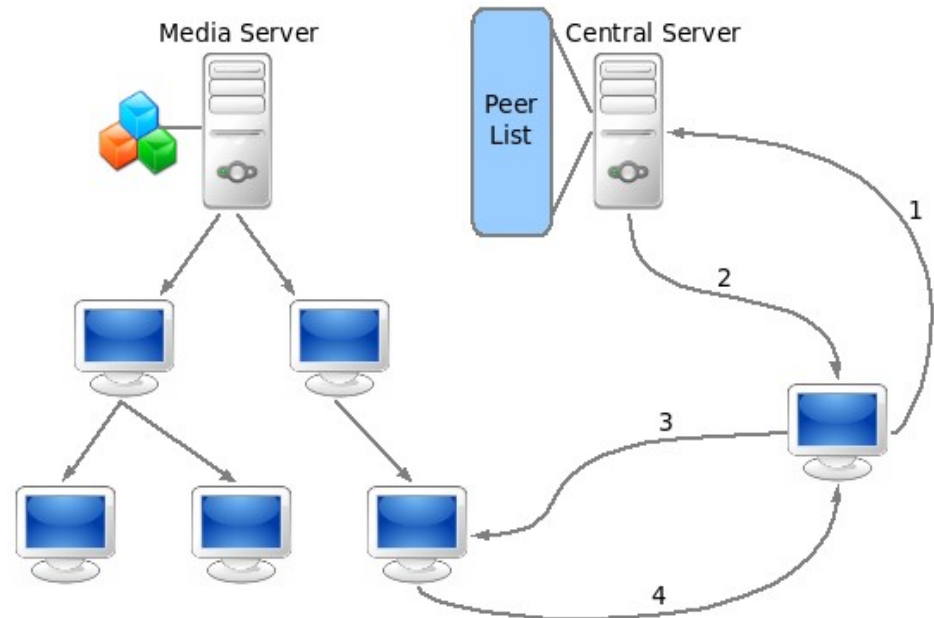




Node Discovery

- Centralized method
- Controlled flooding method
- Hierarchical method
- DHT-based method
- Gossip-based method

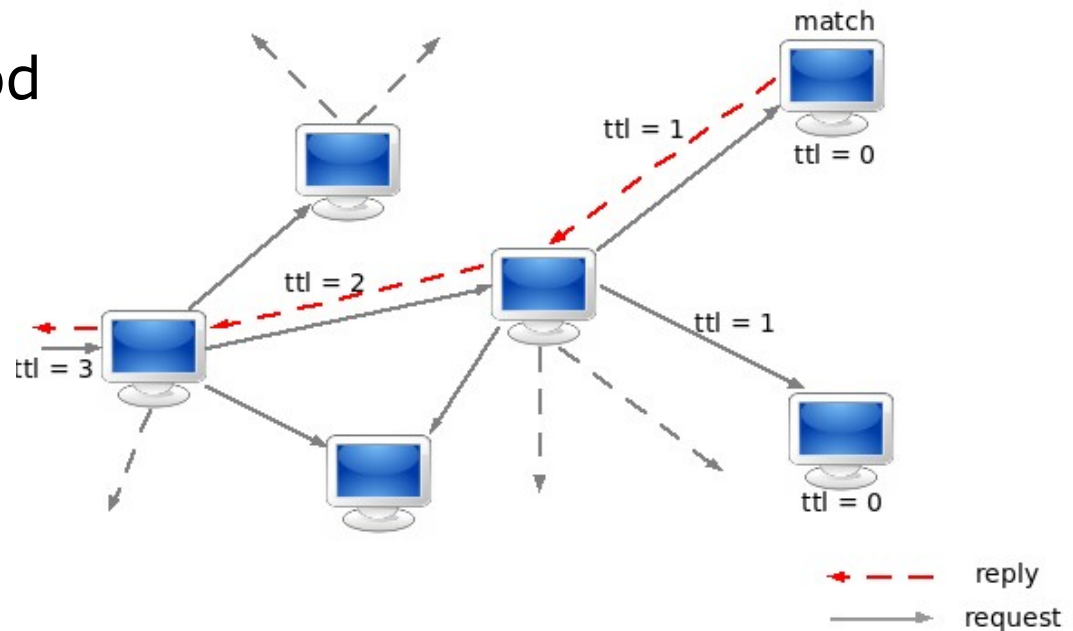
ForestCast





Node Discovery

- Centralized method
- **Controlled flooding method**
- Hierarchical method
- DHT-based method
- Gossip-based method

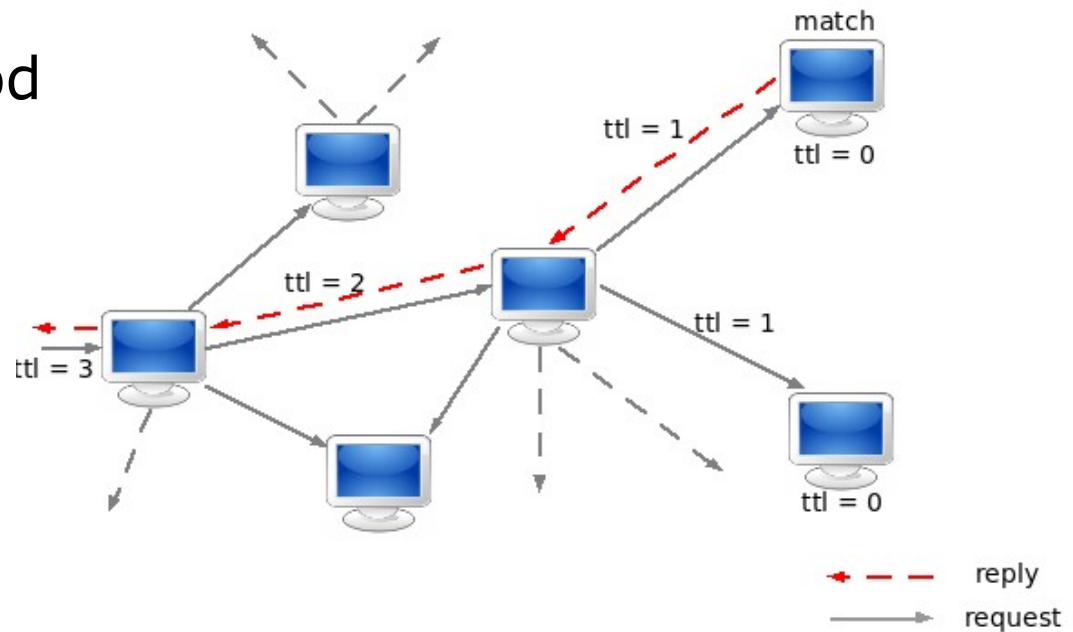




Node Discovery

- Centralized method
- **Controlled flooding method**
- Hierarchical method
- DHT-based method
- Gossip-based method

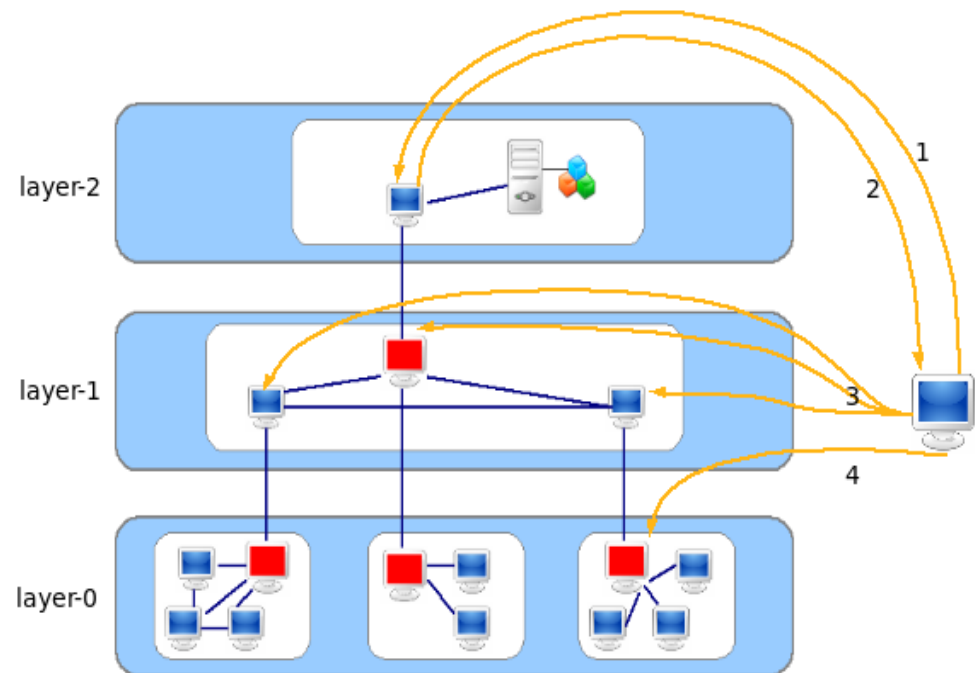
GnuStream
(using Gnutella)





Node Discovery

- Centralized method
- Controlled flooding method
- **Hierarchical method**
- DHT-based method
- Gossip-based method

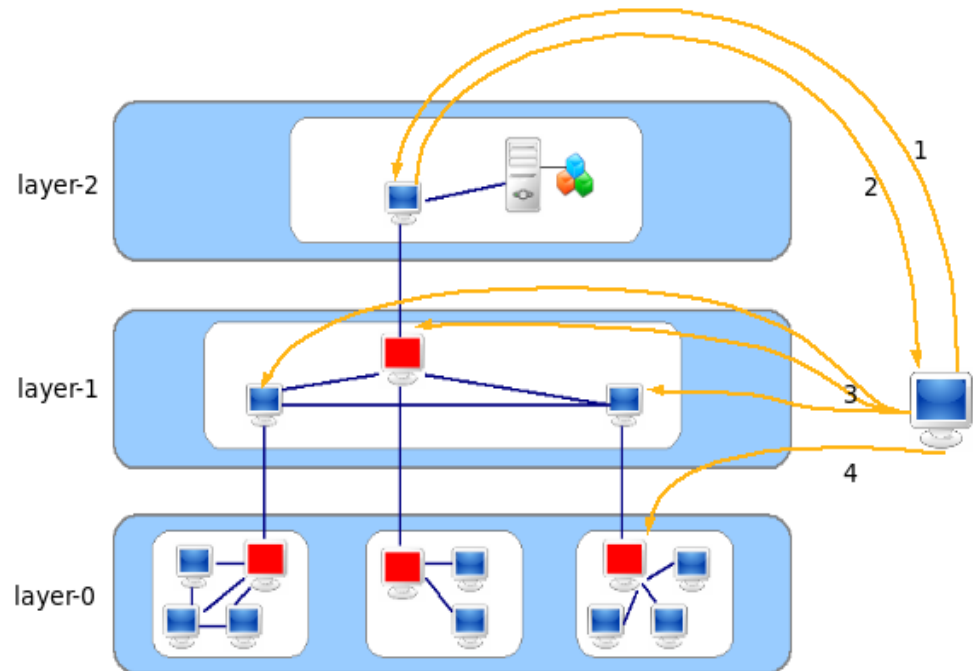




Node Discovery

- Centralized method
- Controlled flooding method
- **Hierarchical method**
- DHT-based method
- Gossip-based method

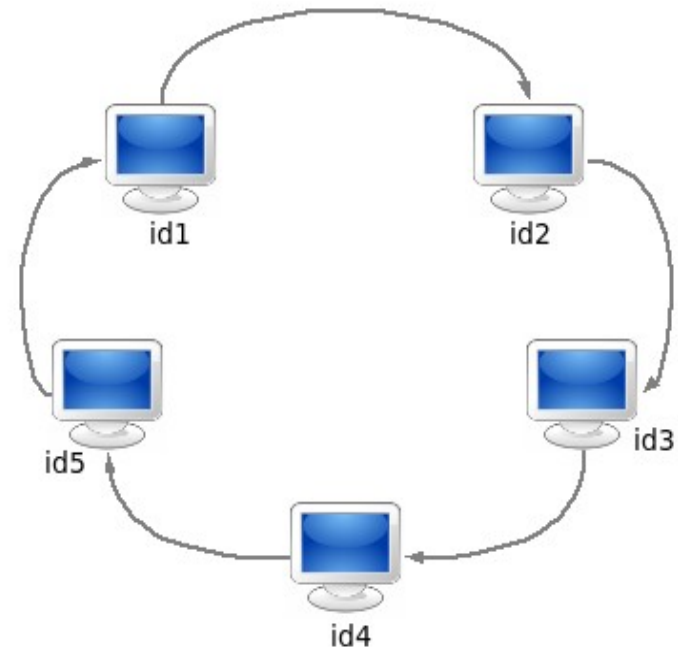
ZigZag





Node Discovery

- Centralized method
- Controlled flooding method
- Hierarchical method
- **DHT-based method**
- Gossip-based method

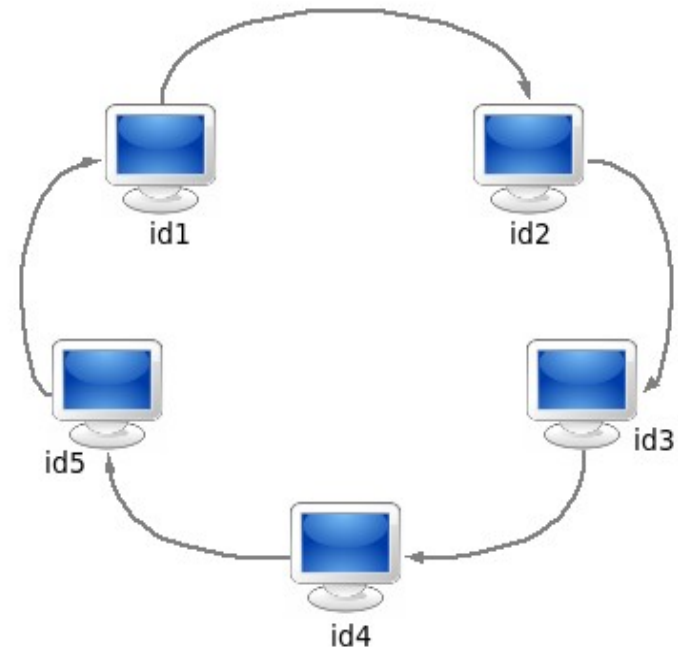




Node Discovery

- Centralized method
- Controlled flooding method
- Hierarchical method
- **DHT-based method**
- Gossip-based method

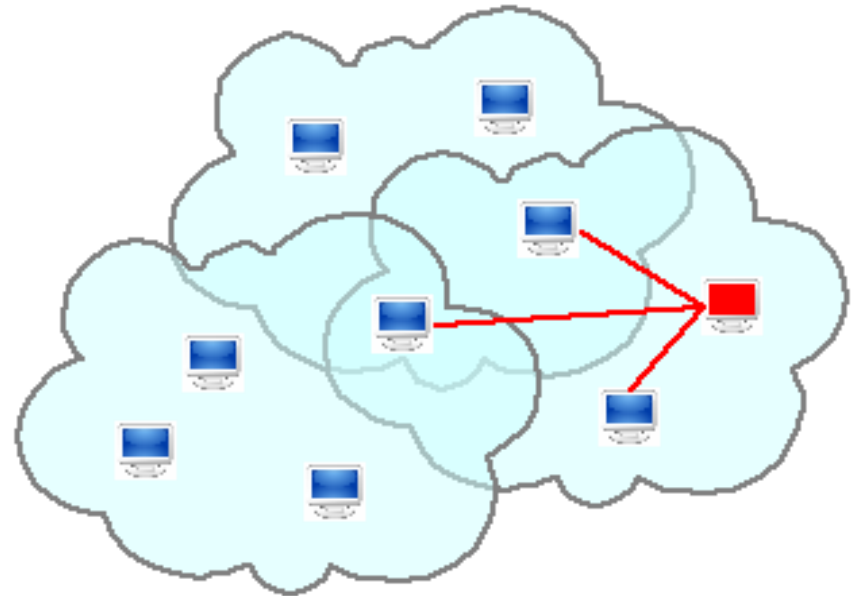
SplitStream





Node Discovery

- Centralized method
- Controlled flooding method
- Hierarchical method
- DHT-based method
- Gossip-based method

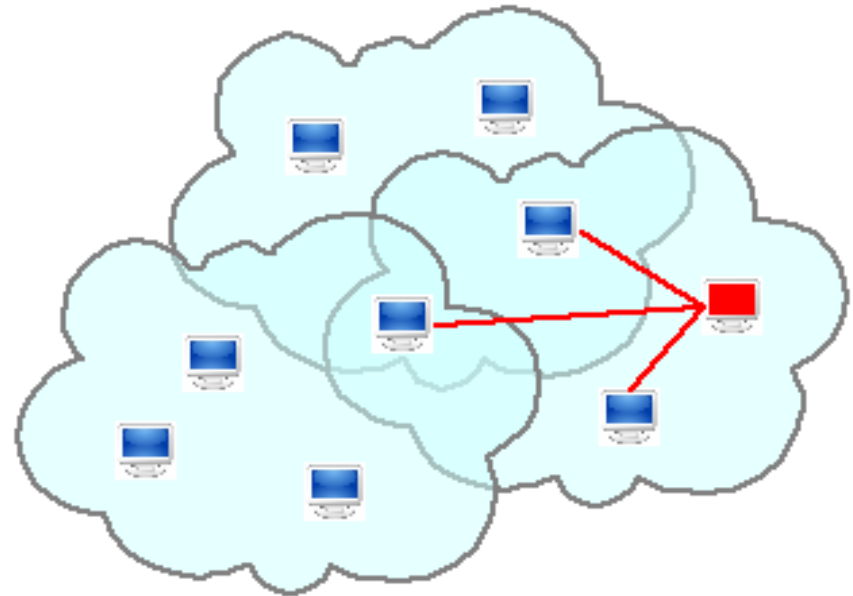




Node Discovery


- Centralized method
- Controlled flooding method
- Hierarchical method
- DHT-based method
- Gossip-based method

PULSE





Two Main Questions

- Node discovery
- Data delivery 





Data Delivery

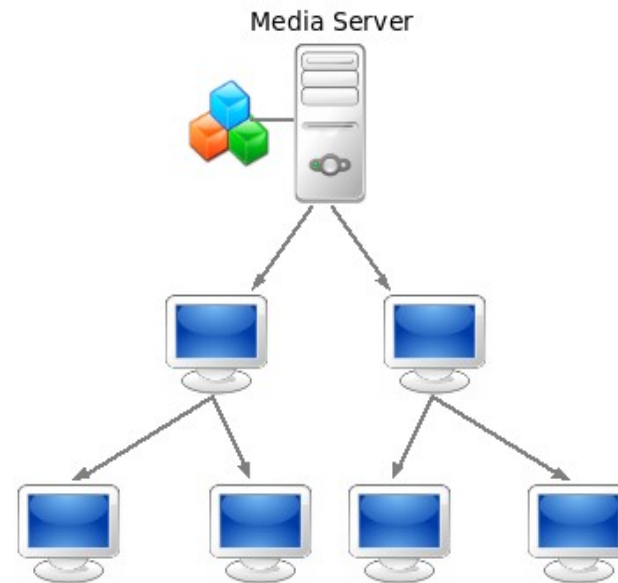
- Push method
 - Single tree
 - Multiple trees
- Pull method
- Push-Pull method





Data Delivery

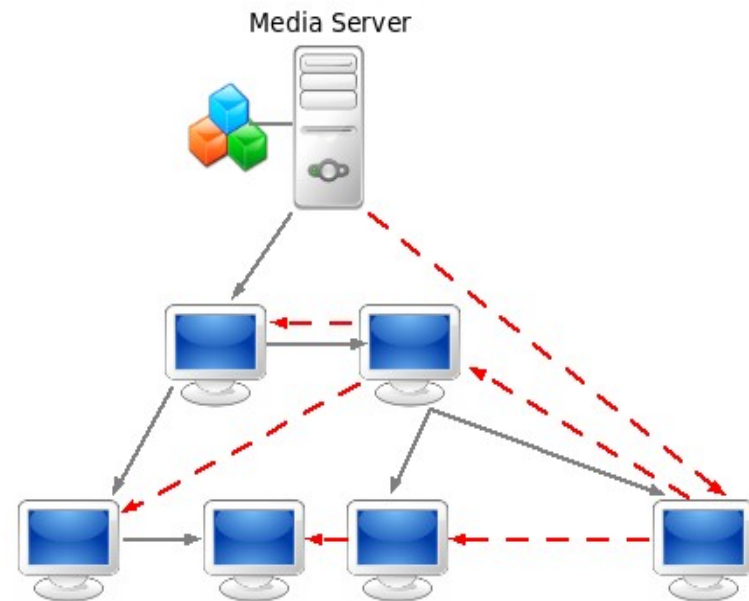
- Push method
 - Single tree
 - Multiple trees
- Pull method
- Push-Pull method





Data Delivery

- Push method
 - Single tree
 - Multiple trees
- Pull method
- Push-Pull method



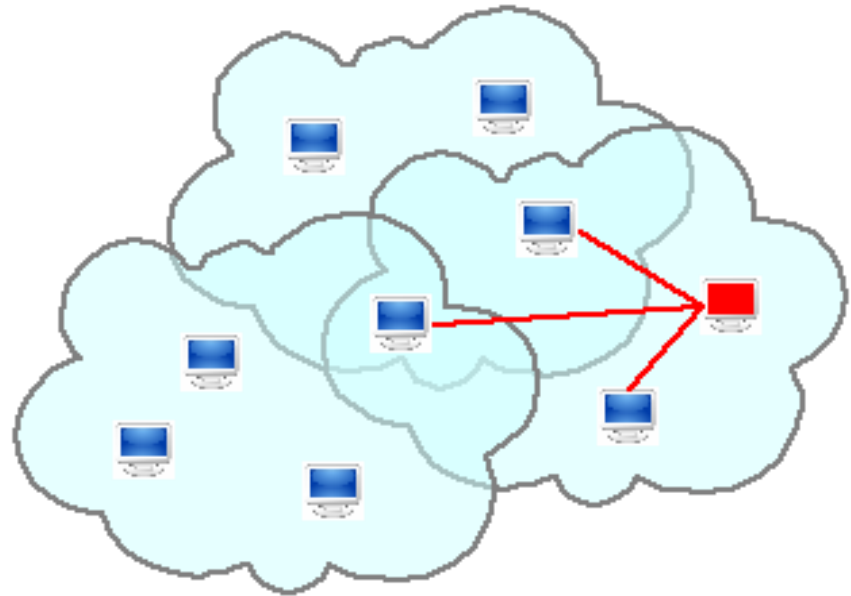
Split the data into segments and send each one through a separate tree





Data Delivery

- Push method
 - Single tree
 - Multiple trees
- Pull method
- Push-Pull method





All Together

Data Delivery Finding supplying peers	Push method (Single tree)	Push method (Multiple trees)	Pull method	Push-Pull method
Centralized method	DirectStream (2006)			Prime (2007) mTreeBone (2007)
Hierarchical method	ZigZag (2003)			mTreeBone (2007)
DHT-based method	SAAR (2007)	SAAR (2007) SplitStream (2003)	SAAR (2007)	Pulsar (2007) mTreeBone (2007)
Controlled flooding method			GnuStream (2003)	
Gossip-based method		Orchard (2006) ChunkySpread (2006)	CoolStreaming (2005) PULSE (2006) ChainSaw (2005) PPLive (2004)	Bullet (2003)





What Is Next?





P2P CDN Applications

- File sharing



- Media streaming





File Sharing (BitTorrent)





BitTorrent

- BitTorrent is a system for **efficient** and **scalable** replication of large amounts of **static** data.
- **Scalable**: the throughput increases with the number of peers.
- **Efficient**: it utilises a large amount of available network bandwidth.





Peer Roles

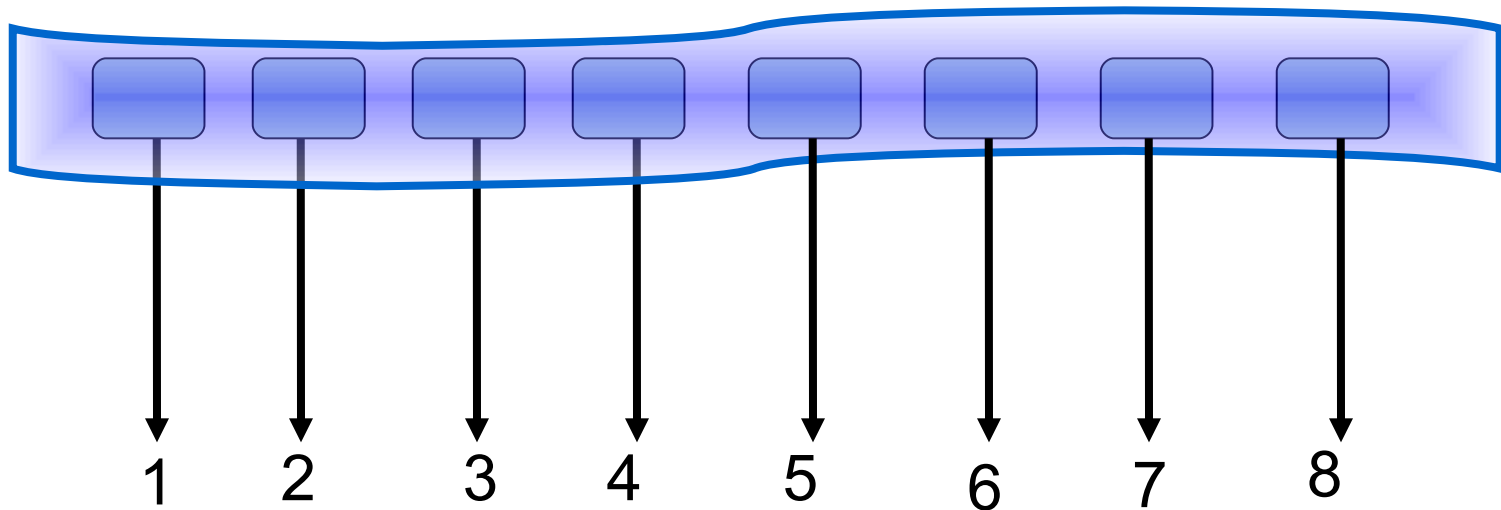
- Tracker
 - A central server helping peers find each other
- Seed
 - Have entire file
- Leecher
 - Still downloading





The Files ...

- Large files are broken into **pieces** of size between 64 KB and 1MB.





Metadata

- .torrent file
- Contains:
 - URL of tracker
 - Information about file
 - Filename
 - Length
 - Hashing information
 - ...

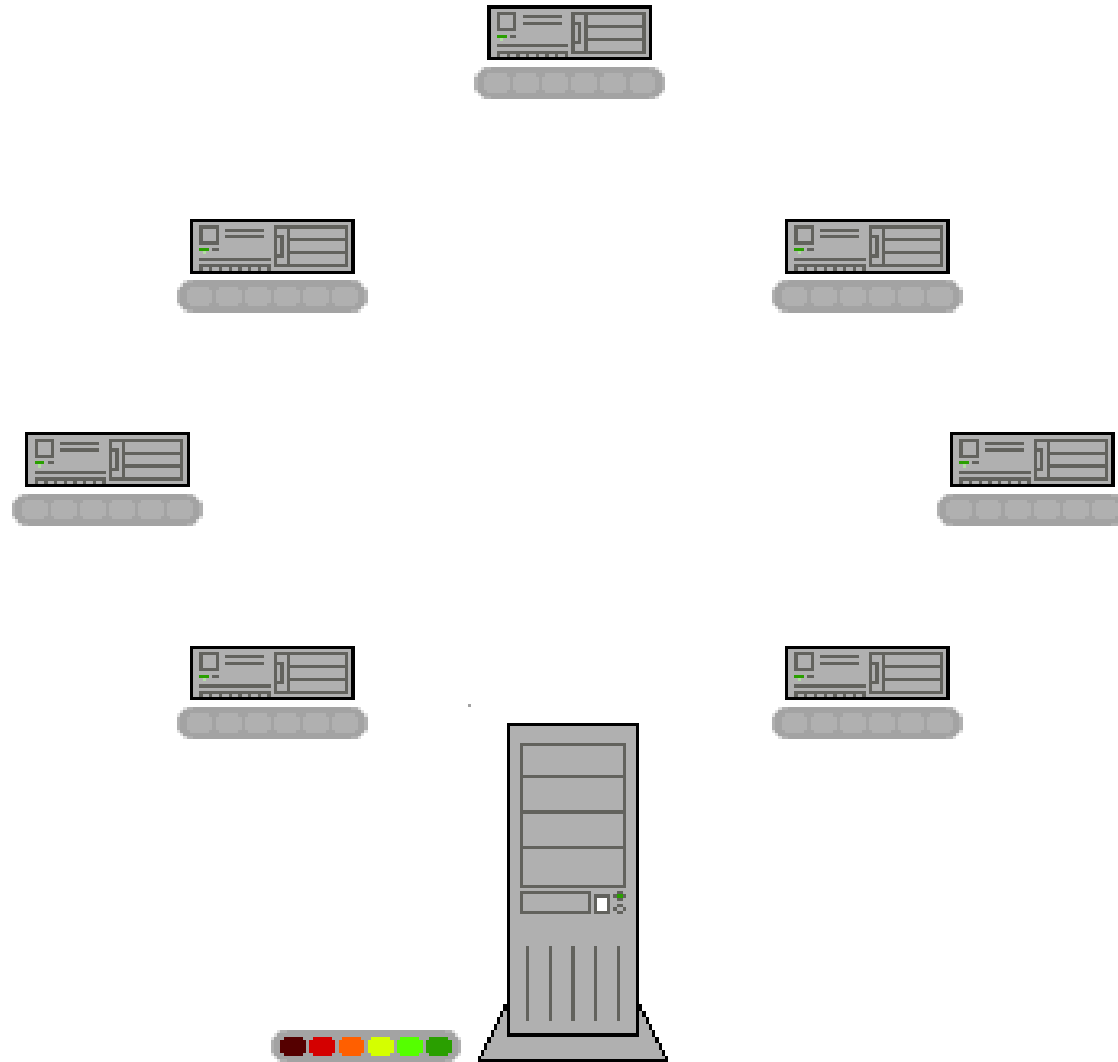


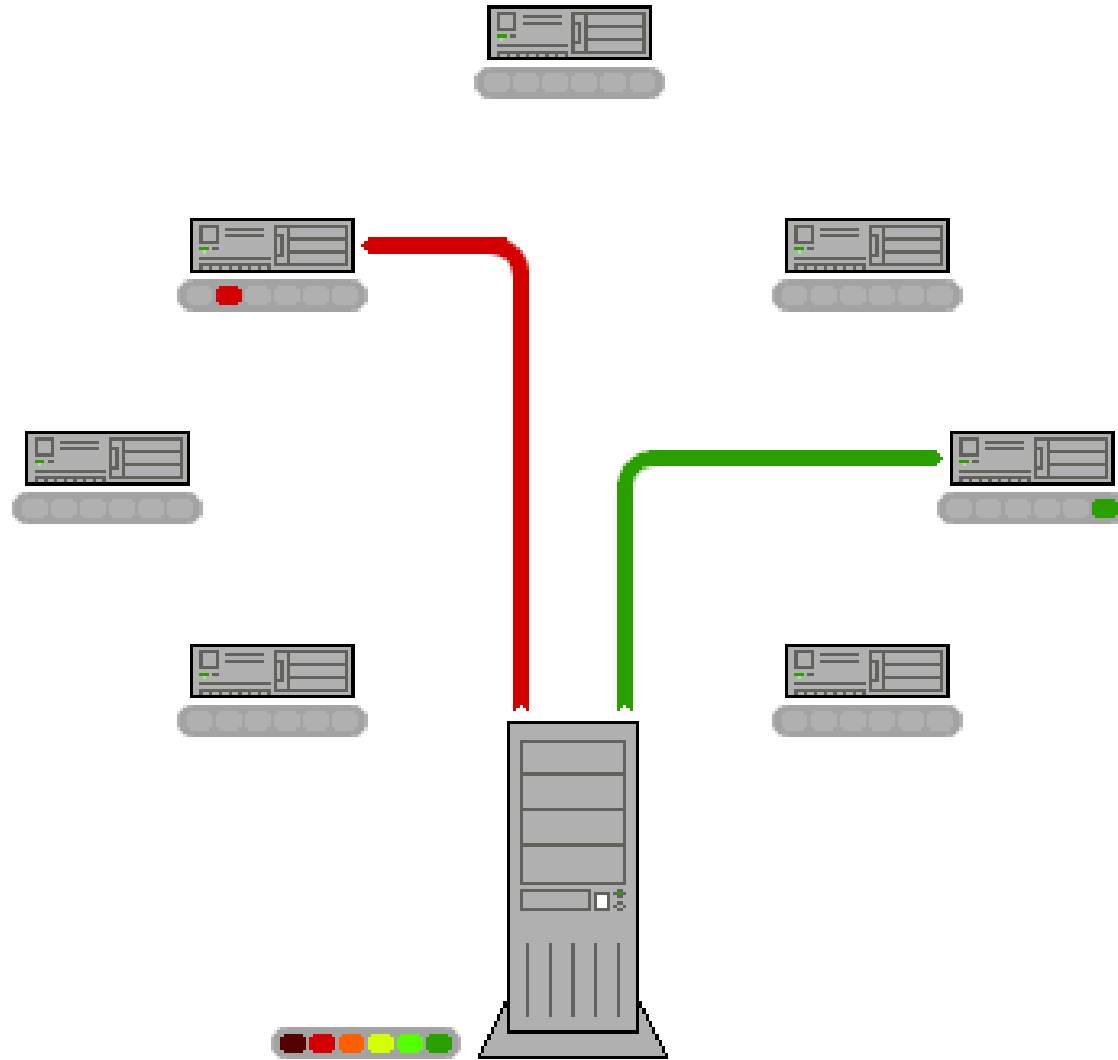


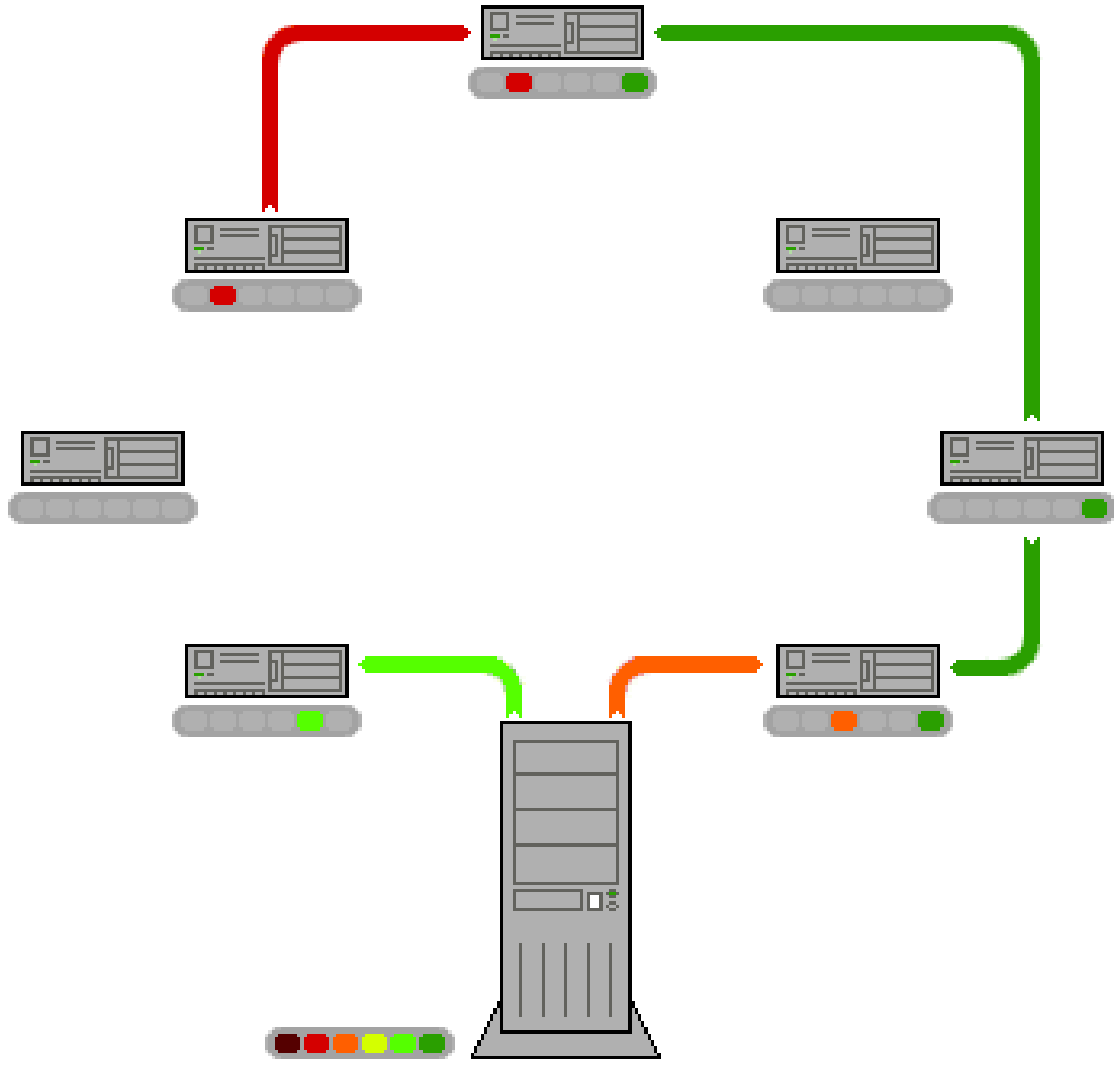
Core Idea Of BitTorrent

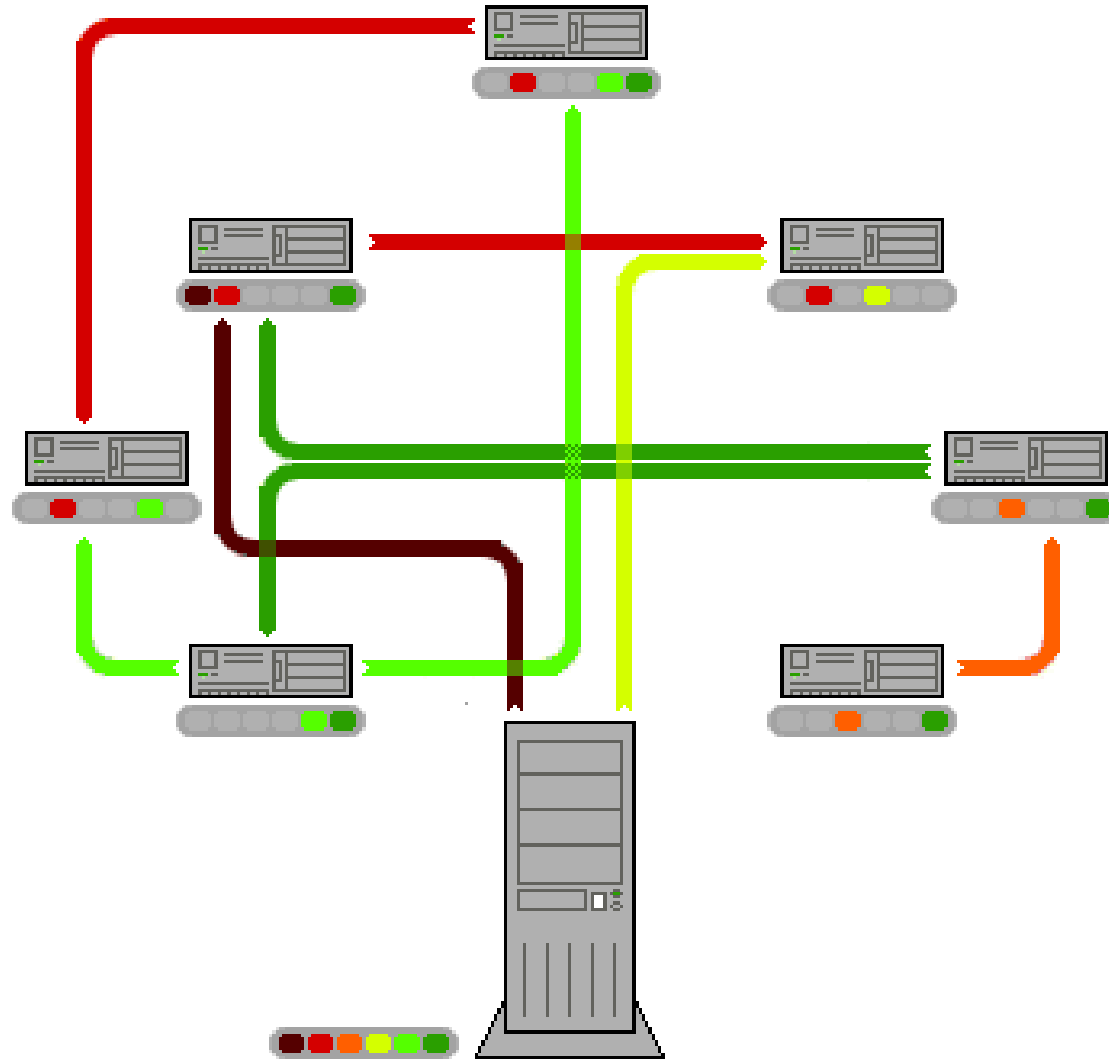
- A peer obtains .torrent file.
- Then it connects to the tracker.
- The tracker tells the peers from which other peers to download the pieces of the file.
- Peers use this information to communicate to each other.
- The peers send information about the file and themselves to tracker.

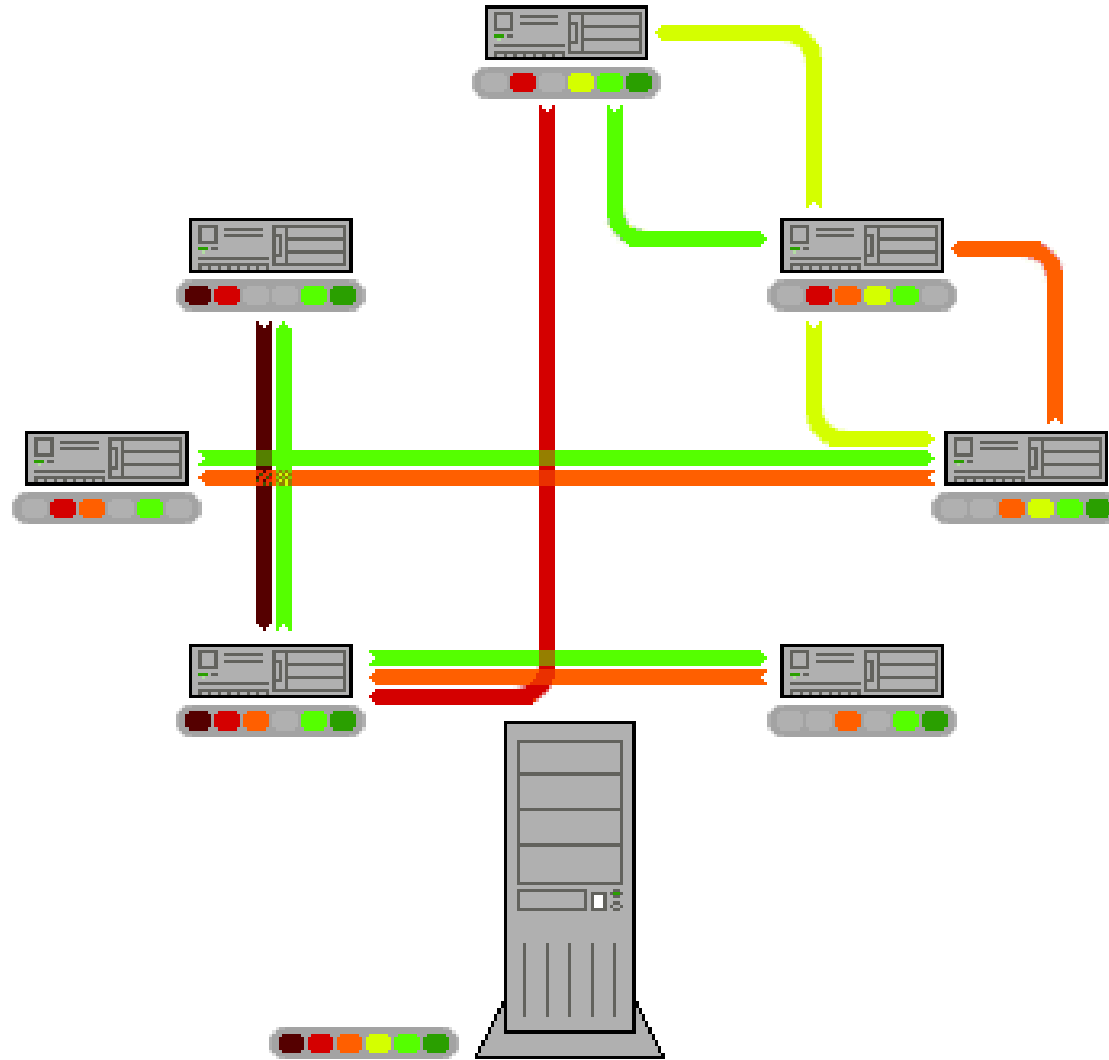


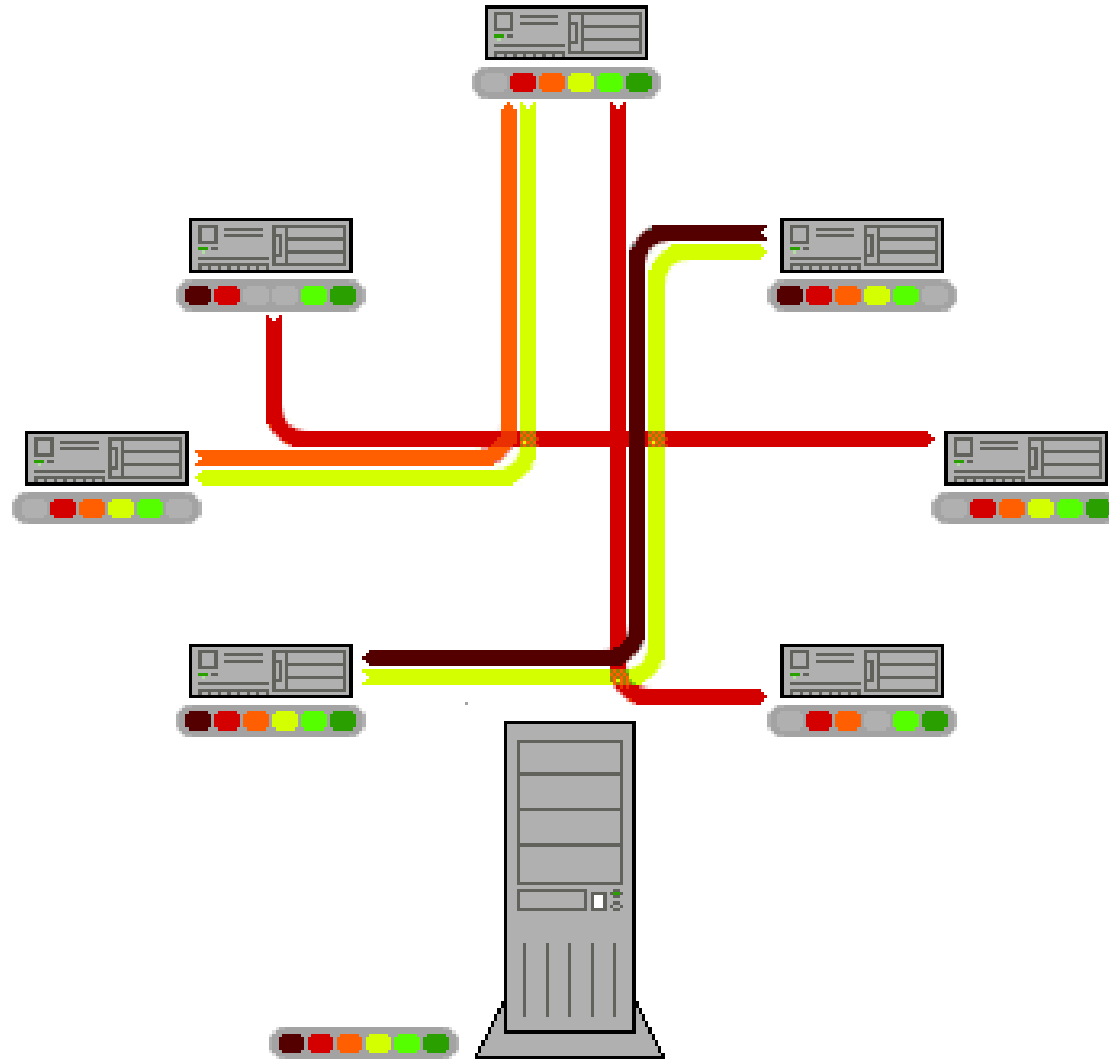


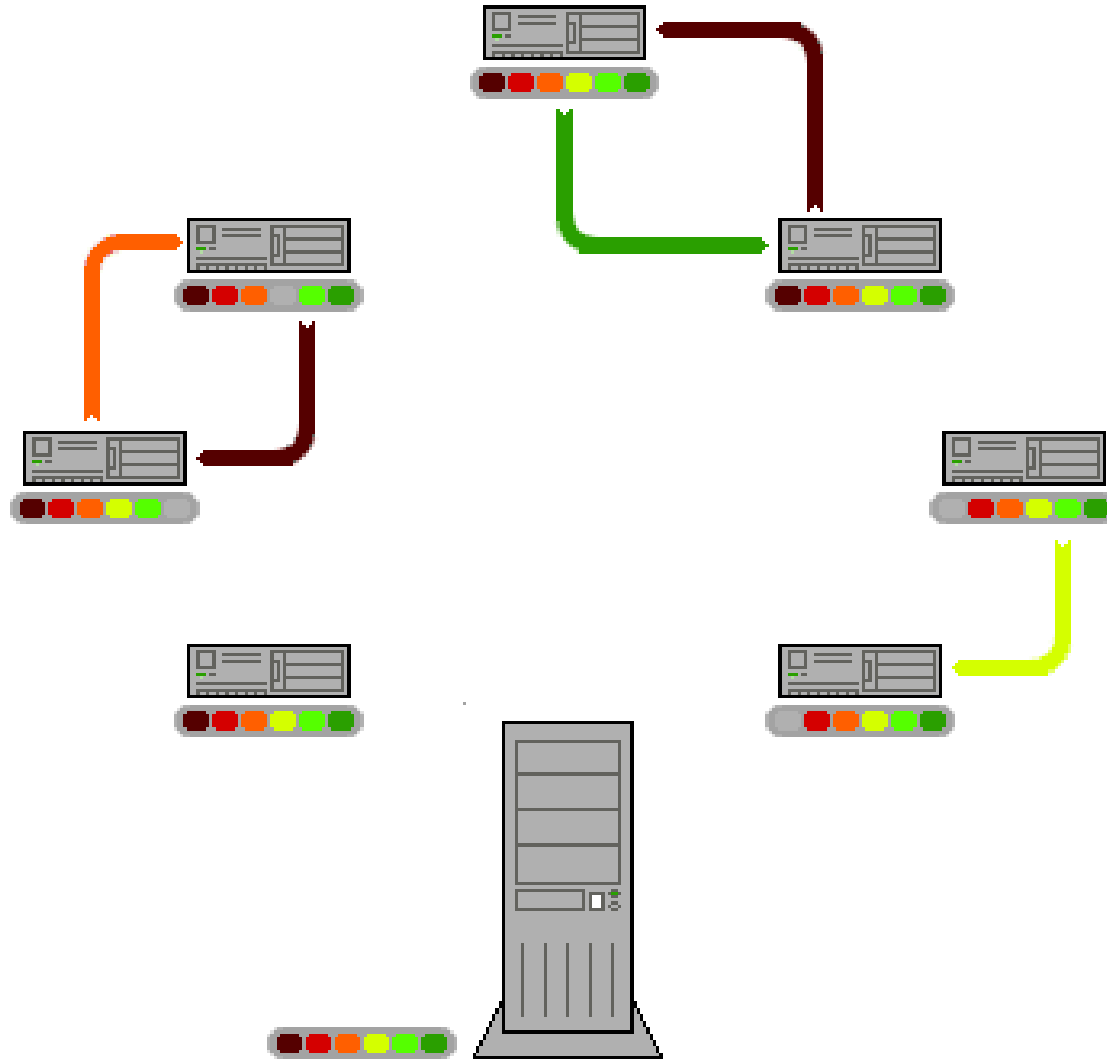


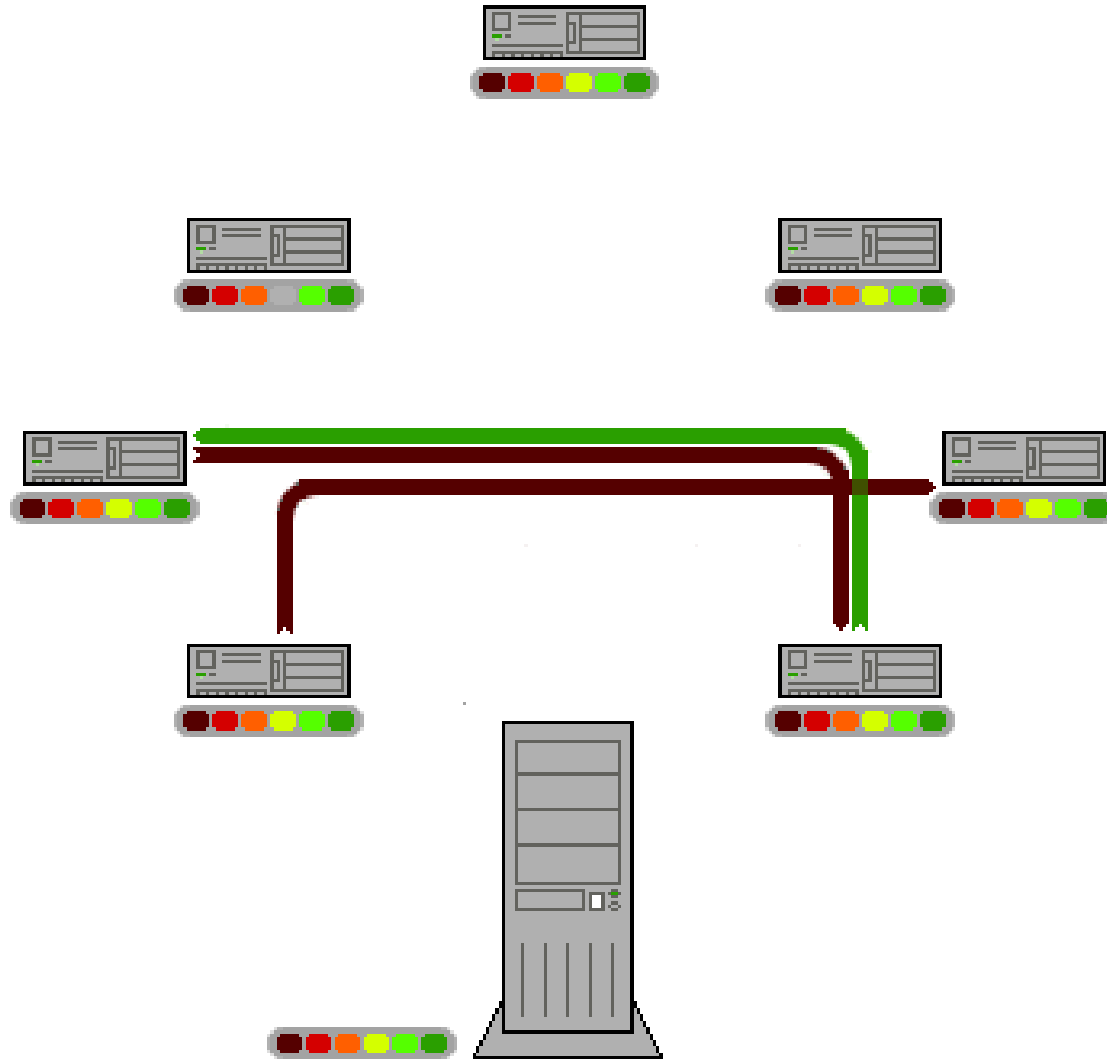


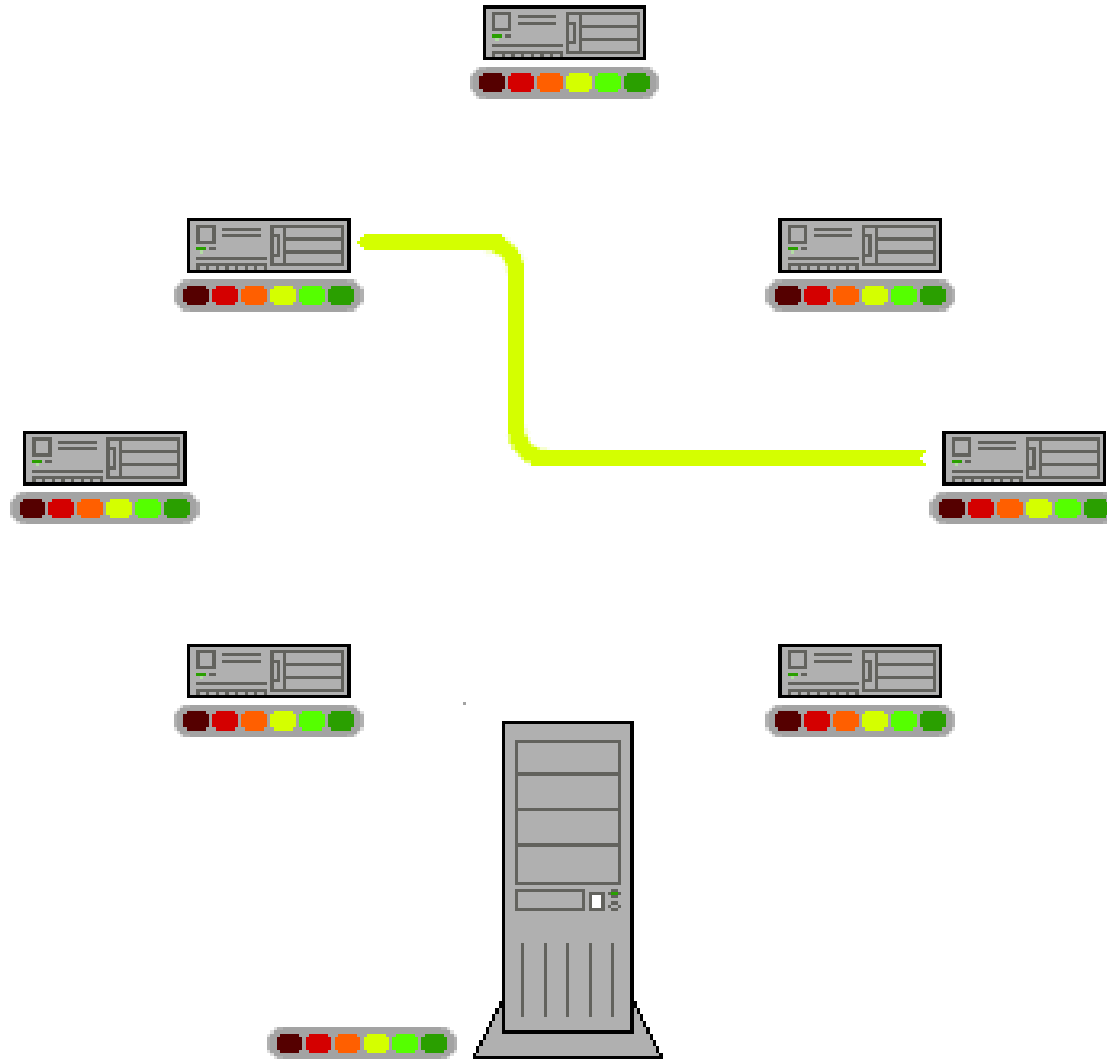


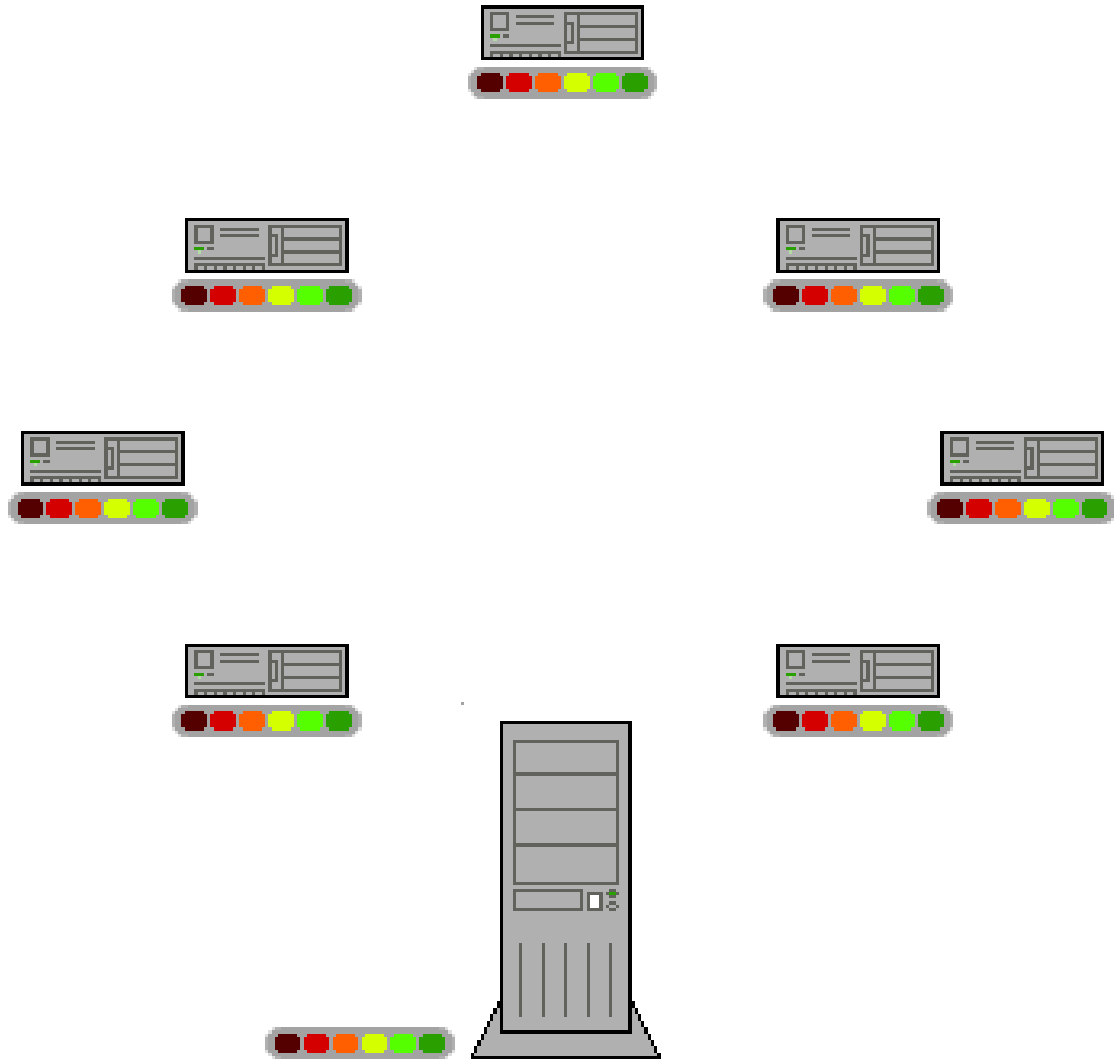














Two Things Are important in BitTorrent

- Peer selection to upload
- Piece selection





Peer Selection To Upload

- BitTorrent does **no central** resource allocation.
- Each peer is responsible for attempting to maximize its own download rate.
- Peers do this by:
 - Downloading from **whoever** they can.
 - Deciding which peers to upload to via a variant of **tit-for-tat**.





Peer Selection To Upload

- Chocking algorithm
- Optimistic unchoking
- Anti-snubbing





Peer Selection To Upload (Choking Algorithm)

- Choking algorithm is a temporary refusal to upload.
- A peer always unchokes a fixed number of its peers.
 - Default of 4.
- Decision to choke/unchoke done based on **current download rates**.
- Upload to peers who have uploaded to you recently (**tit-for-tat**).
- It ensures that nodes cooperate and eliminates the **free-rider** problem.





Peer Selection To Upload (Optimistic Unchoke)

- Upload regardless of the current download rate from the peer.
- To discover currently unused connections are better than the ones being used.
- Optimistic unchoke is rotated periodically.





Peer Selection To Upload (Anti Snubbing)

- When a peer received no data over a minute from a particular peer, does not upload to it except as an optimistic unchoke.
- If choked by everyone, increase the number of simultaneous optimistic unchokes to more than one.





Piece Selection

- Random First Piece
- Rarest First
- Endgame Mode





Piece Selection

(Random First Piece)

- **Policy:** Select a random piece of the file and download it.
- Initially, a peer has nothing to trade.
- Important to get a complete piece ASAP.





Piece Selection (Rarest First)

- **Policy:** Determine the pieces that are most rare among your peers and download those first.
- This ensures that the most common pieces are left till the end to download.
- Rarest first also ensures that a large variety of pieces are downloaded from the seed.





Piece Selection

(Endgame Mode)

- **Policy:** Near the end, missing pieces are requested from every peer containing them.
- When the piece arrives, the pending requests for that piece are canceled.
- This ensures that a download is not prevented from completion due to a single peer with a slow transfer rate.
- Some bandwidth is wasted, but in practice, this is not too much.





Two Main Questions

- Node discovery
- Data delivery





Two Main Questions

- Node discovery
 - Centralized method
 - Tracker
 - DHT-based method
 - Kademlia (Trackerless)
- Data delivery
 - Pull method





Media Streaming (CoolStreaming/DONet)





Media Streaming

- Media Streaming over Internet is getting more popular everyday.



- Media streaming
 - Video on Demand (VoD)
 - Live media streaming





Media Streaming

- Bandwidth intensive
- Time sensitive
 - A negligible startup delay
 - Smooth playback
 - A negligible playback latency
 - only for live streaming
- P2P Challenges:
 - Nodes join, leave and fail continuously.
 - Called **churn**
 - Network capacity changes.





CoolStreaming/DONet

- DONet is an overlay network for live media streaming.
- CoolStreaming is an **Internet-based** DONet implementation.





Core Idea of DONet

- Every node periodically exchanges data availability information with a set of partners, and retrieves unavailable data from one or more partners, or supplies available data to partners.





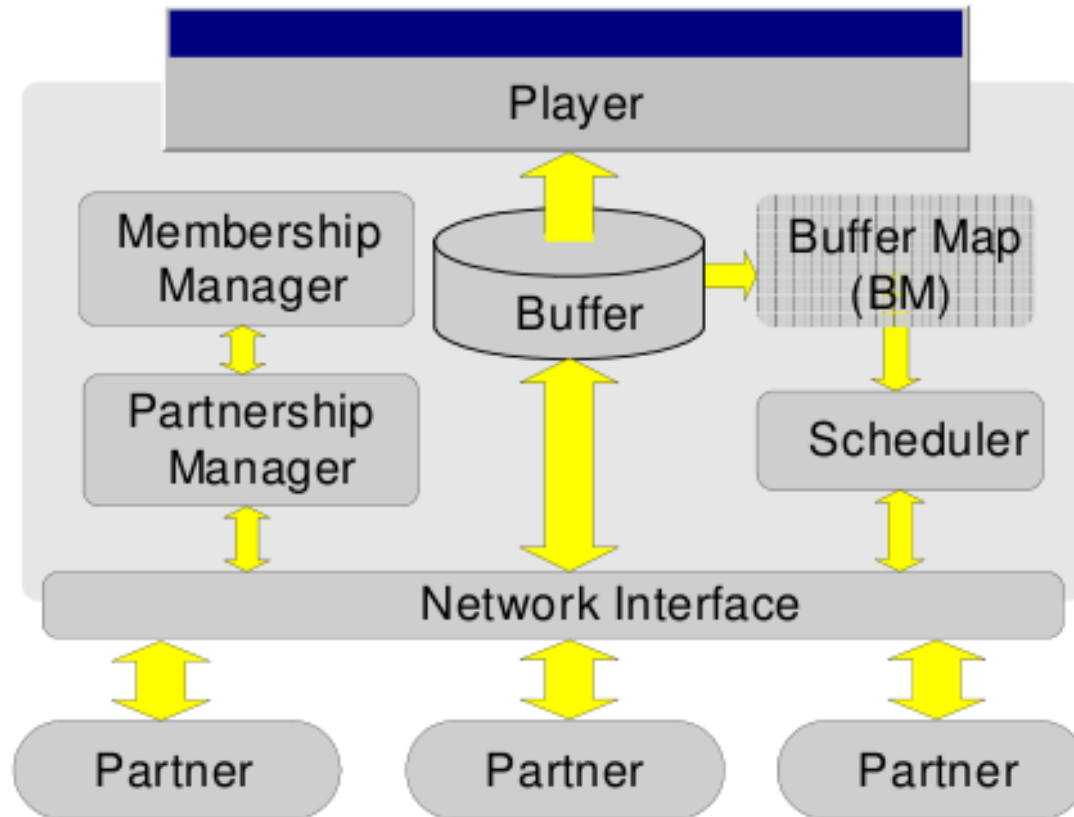
Assumption

- The media stream is divided into **segments**.
- For each segment, a node can be receiver or supplier.
- The source node is always supplier.
 - Origin node
- Each node has a unique ID.
 - IP address



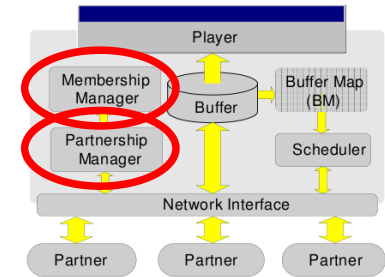


DONet Node System Diagram





Membership Management

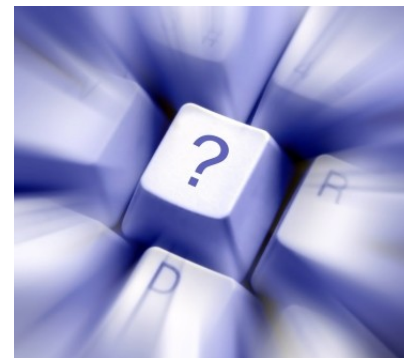


- Each node has a partial list of the ID for the active nodes.
 - mCache
- A newly joined node first contacts the origin node.
- The origin node randomly selects a deputy node from its mCache and redirects the new node to the deputy.
 - To have more uniform partner selection
- The new node obtains a list of partner candidates from the deputy.
- It then contacts these candidates to establish its partners in the overlay.



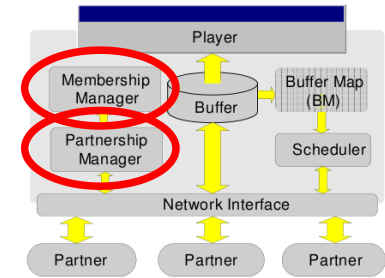


How To Create And Update mCache?





Create and Update mCache

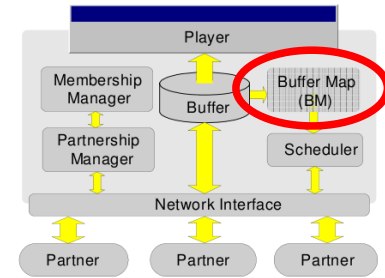


- Each node periodically generates a **membership message** and distributes it among the nodes.
- Upon receiving the message, the node updates its mCache entry for node id.
- Each node **periodically** establishes new partnerships with nodes **randomly selected** from its mCache.





Buffer Map (BM)



- Shows the availability of the segments in the buffer of a node.
- Each node continuously exchange its BM with its partners.



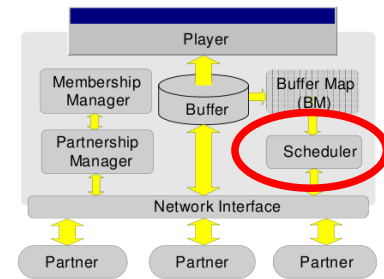


From Which Partner Fetch Which Segment?





Scheduling



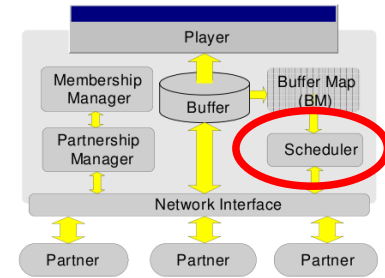
- For a homogeneous and static network a simple round-robin scheduler may work well.
- But what about for a dynamic and heterogeneous network?





Scheduling

(Dynamic and Heterogeneous)

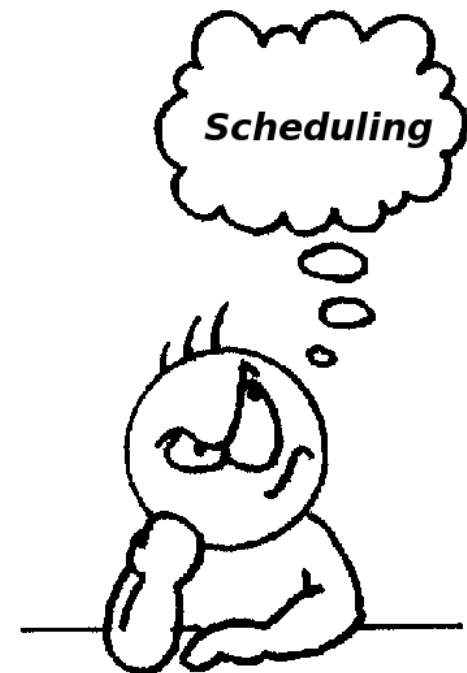


- Two constraints:
 - The playback deadline for each segment.
 - The heterogeneous streaming bandwidth from the partners.
- If the first constraint cannot be satisfied, then the number of segments missing deadlines should be kept minimum.





But Finding An Optimal Solution Is Not Easy.





Simple Heuristic

- First calculates the number of potential suppliers for each segment.
- A segment with less potential suppliers is more difficult to meet the deadline constraints.
 - Starting from those with only one potential supplier, then those with two, and so forth.
- Among the multiple potential suppliers, the one with the highest bandwidth.





Failure Recovery

- Graceful departure
 - The departing node should issue a departure message, which has the same format as the membership message.
- Node failure
 - A partner that detects the failure will issue the departure message on behalf the failed node.





Again Our Two Main Questions

- Node discovery
- Data delivery





Two Main Questions

- Node discovery
 - Gossip-based method
- Data delivery
 - Pull method





Wake Up!





The Only Page To Remember

- Content Distribution Network

- Client-Server solution

- Expensive



- P2P solution

- The peers can help each other and the capacity increases with the number of peers.

- Two main questions

- Node discovery
- Data delivery





Reading Assignments

- V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr. "*Chainsaw: Eliminating Trees from Overlay Multicast*", In The Fourth International Workshop on Peer-to-Peer Systems, February 2005.
- J. C. V. Venkataraman, P. Francis, "*Chunkyspread: Multi-tree Unstructured Peer-to-Peer Multicast*", in Proc. 5th International Workshop on Peer-to-Peer Systems, February 2006.
- Kunwoo Park, Sangheon Pack, and Taekyoung Kwon, "*Climber: An Incentive-based Resilient Peer-to-Peer System for Live Streaming Services*", in The 7th International Workshop on Peer-to-Peer Systems (IPTPS 2008), Tampa, USA, February 2008.
- Yang Guo, Kyoungwon Suh, Jim Kurose, Don Towsley, "*DirectStream: A directory-based peer-to-peer video streaming service*", Journal of Computer communications (COMCOM), Elsevier, in-print, 2007
- C. Tang, R. N. Chang, and C. Ward, "*Gossip enhanced overlay multicast for fast and dependable group communication*", in DSN, 2005, pp. 140–149.





Reading Assignments

- Animesh Nandi, Aditya Ganjam, Peter Druschel, T. S. Eugene Ng, Ion Stoica, Hui Zhang, and Bobby Bhattacharjee. "*Saar: A shared control plane for overlay multicast*", In NSDI. USENIX, 2007.
- J. J. D. Mol, D. H. J. Epema, and H. J. Sips. "*The orchard algorithm: P2P multicasting without free-riding*", In P2P '06: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing, pages 275-282, Washington, DC, USA, 2006. IEEE Computer Society.
- Fabio Pianese, Joaquin Keller, and Ernst W Biersack. "*PULSE, a exible P2P live streaming system*", In 9th IEEE Global Internet Symposium 2006 in conjunction with IEEE Infocom 2006, 28-29 April 2006, Barcelona, Spain, Apr 2006.
- Feng Wang, Yongqiang Xiong, and Jiangchuan Liu. "*mtreebone: A hybrid tree/mesh overlay for application-layer live video multicast*", In ICDCS 07: Proceedings of the 27th International Conference on Distributed Computing Systems, page 49, Washington, DC, USA, 2007. IEEE Computer Society.

