

Internet content distribution

IERG5090 April 10, 2017

Outline

This week's topics:

- Alternative technology and research on Internet content distribution
- IP multicast and related protocols
- Client-Server video streaming
- Content Distribution Networks (CDN)
- Peer-to-Peer approach (P2P)

Content



Tencent 腾讯



YOUKU 优酷



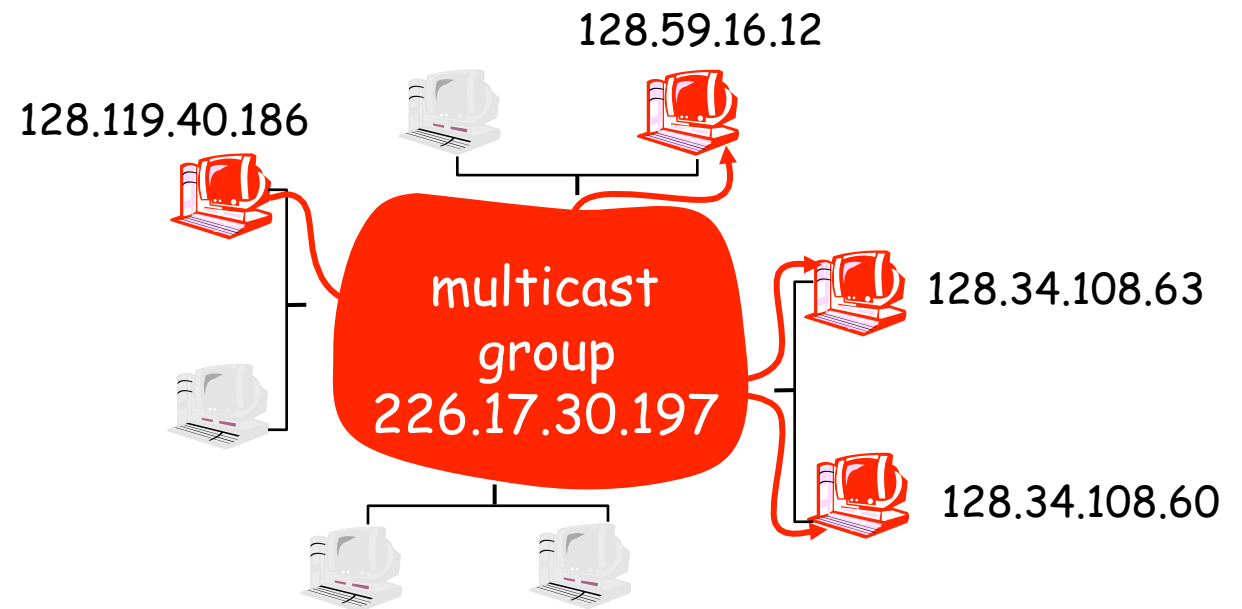
- Besides audio, video, VR, also
 - Software
 - Games
 - Broadcast
 - Conferencing
 - Stock quotes
 - News
 - etc

Content distribution mechanisms

1. IP Multicast
 2. Client – Server
 3. Content Distribution Network (CDN)
 4. Peer-to-Peer network
- Methods 2,3,4 can be thought of variations/extensions of the same basic (Client-Server) mechanism
 - They are all considered Over-The-Top (OTT) of network solutions
 - IP Multicast is a network layer mechanism of network **broadcast** to a **subset** of nodes
 - A lot of research and standardization efforts went into this
 - It is only used in limited local cases

Multicast as a service

- The service semantics:
 - Receivers join multicast group with a given multicast address
 - Sender sends datagrams with the multicast address as destination
 - All receivers in the group will receive the datagrams
 - Anyone can send
 - Sender does not know who are receivers
 - There is no reliability guarantee
- What are the benefits, and drawbacks/limitations of this service?



- How is the service implemented?

Limitations

These are some key challenges and limitations:

1. Need to allocate some IP addresses for this purpose, for the whole world to use
2. Need the receivers to know the multicast address and the start time, and receive together
3. Hard to add reliability, if there are many receivers

- Short-term solutions for 1&2:

- Any IP address starting with prefix 1110 reserved for multicast:



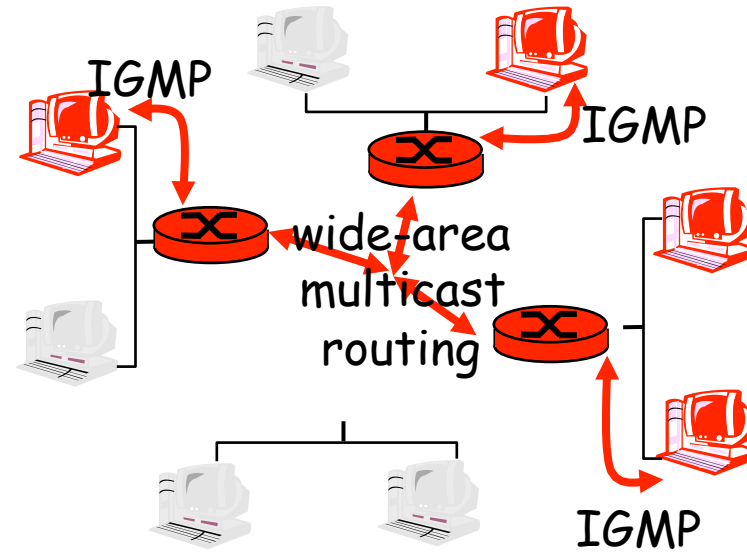
← 28 bits →

- Use a **Session Advertisement Protocol** (SAP) and **Session Description Protocol** (SDP) to announce the multicast address, starting time, transport, payload format for multicast of a given content
- Ignore 3 for now.

Setting up a multicast group

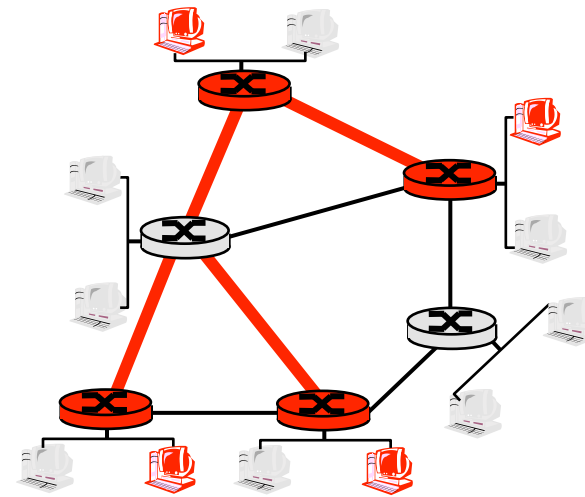
Two steps:

- Locally, hosts tell local router if they want to join a multicast group by using: **IGMP (Internet Group Management Protocol)**
- Across WAN, routers that need to help a particular multicast session talk to each other and set up their forwarding tables
 - There are several different protocols to do this
 - They are called **multicast routing protocols**
 - For multicast, routers may need to forward a packet to multiple neighbors

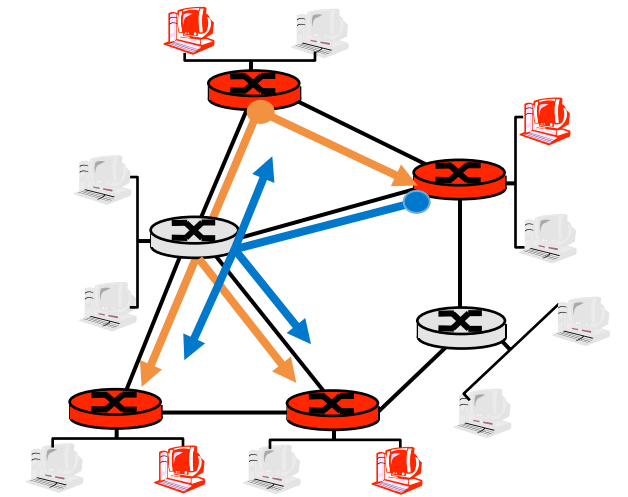


Multicast routing – two approaches

- **Goal:** find a tree (or trees) connecting routers having local multicast group members
- **tree:** not all paths between routers used
 - **source-based:** different tree from each sender to receivers
 - **shared-tree:** same tree used by all group members



Shared tree



Source-based trees

Different approaches

Approaches:

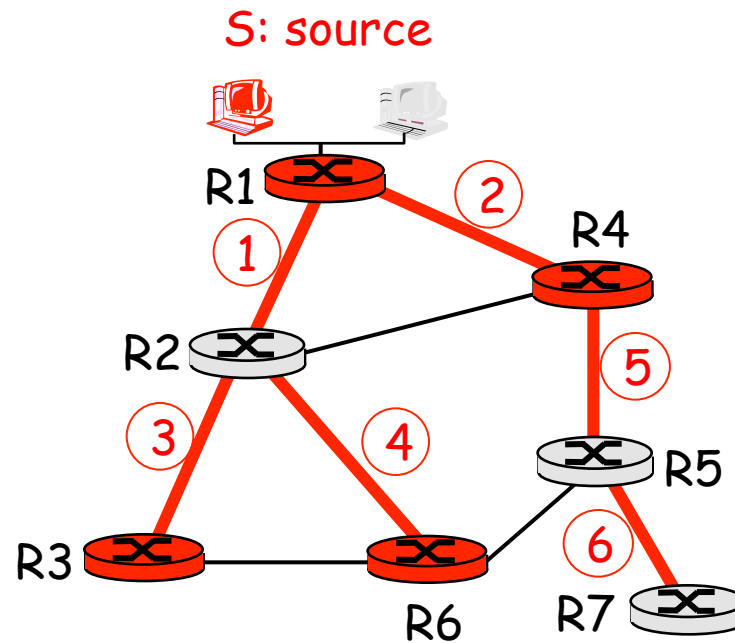
- **source-based tree:** one tree per source
 - shortest path trees
 - **reverse path forwarding**
- **group-shared tree:** group uses one tree
 - minimal spanning trees (**Steiner trees**)
 - center-based trees

Shortest path tree




Tree of shortest path routes from source to all receivers

- Dijkstra's algorithm

If you are already using link state routing, you can use the same information for this purpose

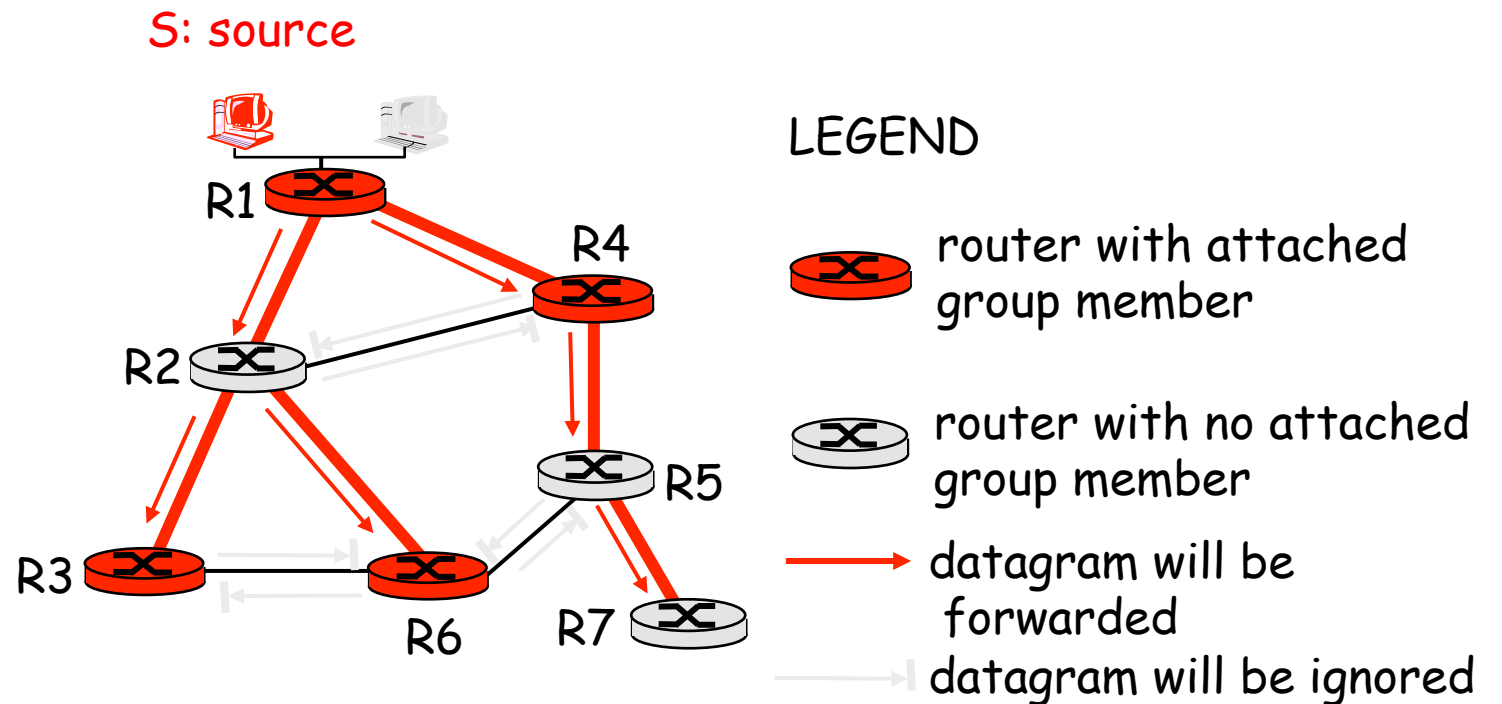


LEGEND

-  router with attached group member
-  router with no attached group member
-  link used for forwarding, i indicates order link added by algorithm

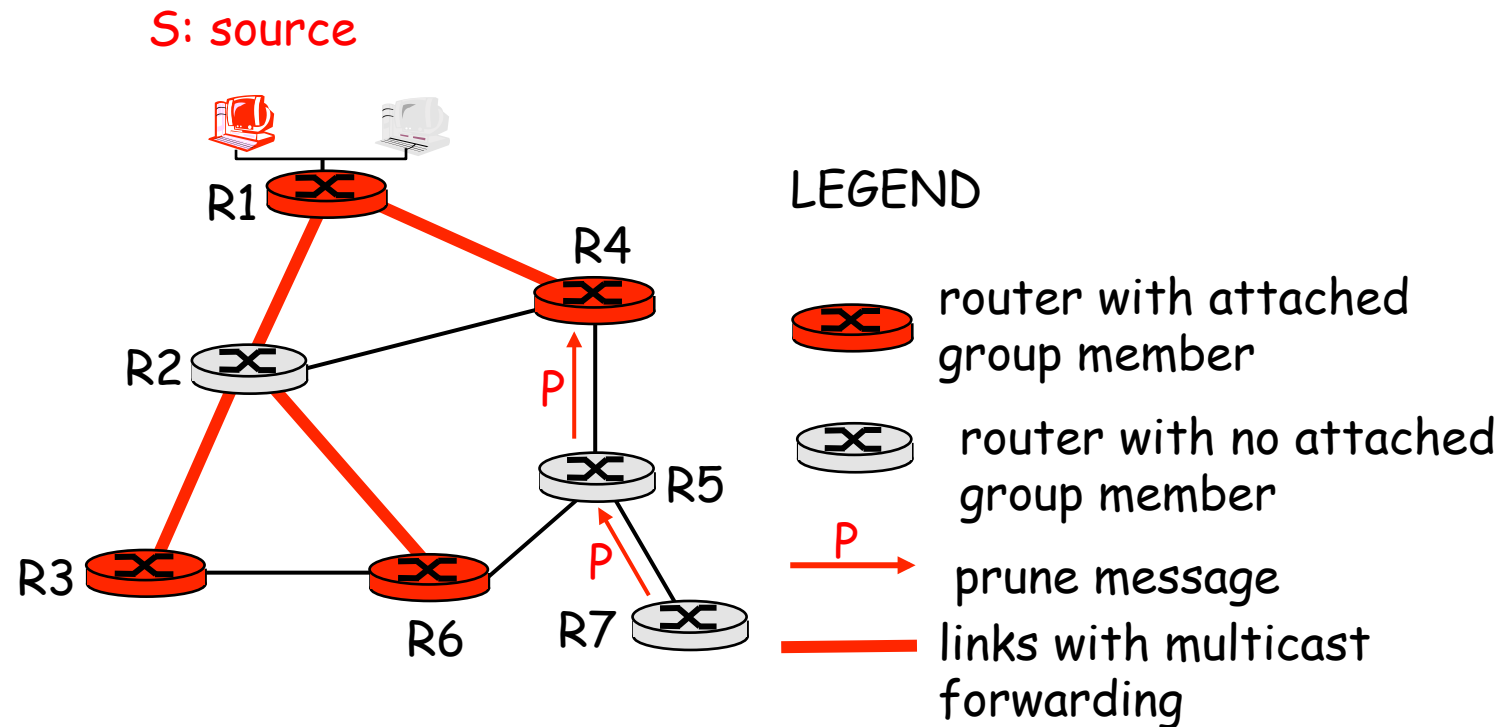
Reverse Path Forwarding (RPF)

- A method to find a tree from source to all receivers, without loop
- When a receiver receives a multicast datagram, if the incoming link is on the shortest path to reach source, then flood the packet; else drop the packet.
 - Is the result the shortest path tree from source?



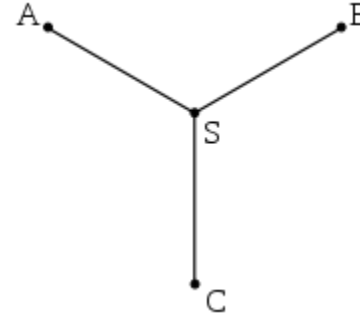
Pruning

- RPF is a flooding process
 - Loops avoided
- But resulting forwarding tree contains subtrees with no multicast receivers
 - no need to forward datagrams down subtree
 - “prune” messages sent upstream by router with no downstream group members

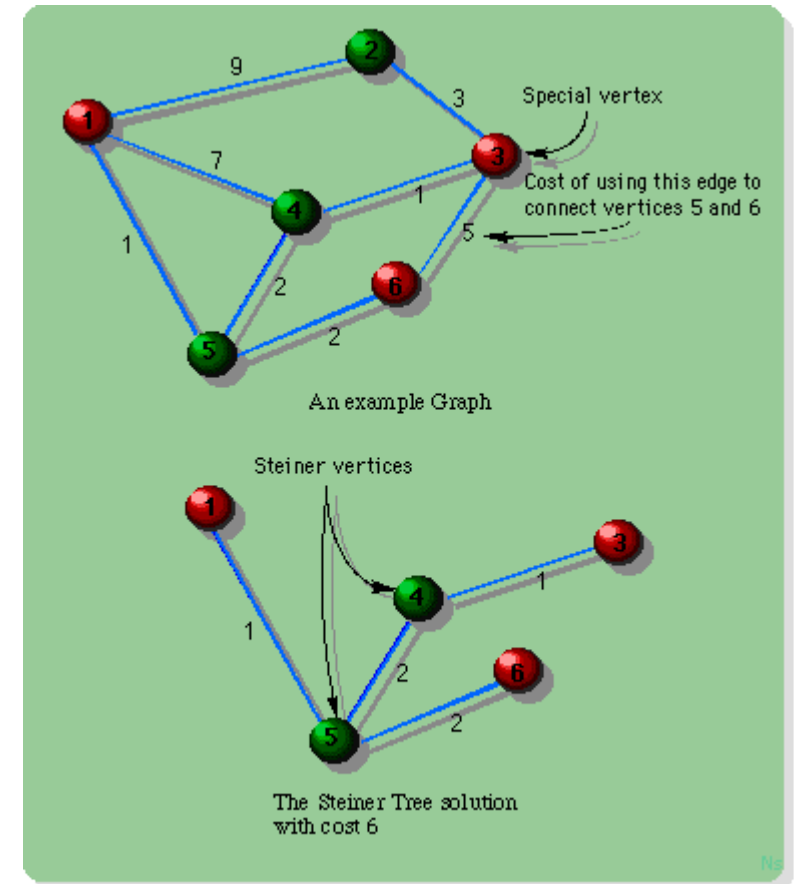


Shared tree

- Given a graph with weighted edges $G(V,E,w)$, and a subset of nodes M , finding a minimum cost tree connecting M is known as the Steiner tree problem
 - The solution is allowed to contain nodes not in M
 - The problem is **NP-complete**, meaning hard to compute in the worst case
 - Good heuristics exist



Examples of
Euclidean Steiner
tree and Graph
Steiner tree problems

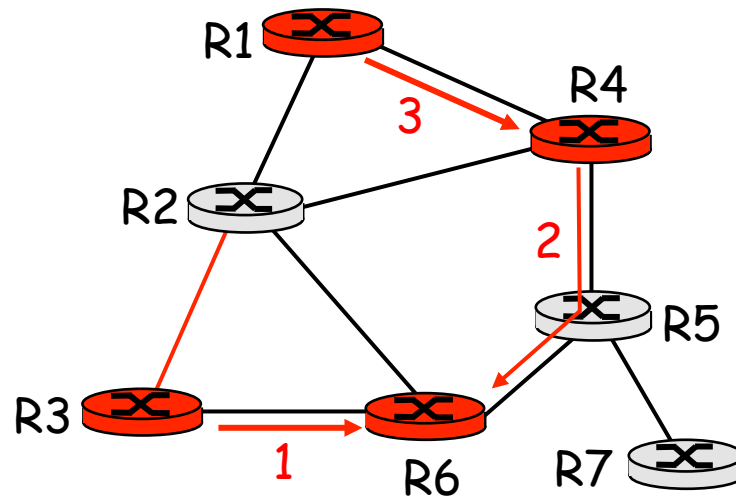


Shared tree construction in practice




- Steiner tree is not used in practice: why?
 - Complexity
 - Multicast group can change over time (some members leave, others join); not practical to keep changing to forwarding tree
- Center-based tree
 - One router identified as “center” of tree
 - To join, edge router sends unicast *join-msg* addressed to center router
 - *join-msg* “processed” by intermediate routers and forwarded towards center
 - *join-msg* either hits existing tree branch for this center, or arrives at center
 - path taken by *join-msg* becomes new branch of tree for this router

Example of center-based tree formation

- Assume R6 picked as center router
 - Subsequent steps as shown



LEGEND

-  router with attached group member
-  router with no attached group member
-  path order in which join messages generated

Some multicast routing protocols

Multicast routing builds a tree
(unicast needs a path)

- Source-based tree
 - Reverse path forwarding (RPF)
 - Distance Vector Multicast Routing Protocol (DVMRP)
 - Protocol-Independent Multicast (PIM): Dense-Mode (PIM-DM)
 - Shortest path tree
 - Multicast-OSPF
 - PIM Sparse Mode (PIM-SM)

- Shared tree
 - Steiner tree – optimal but not practical in layer 3
 - Center-based tree – PIM-SM, Core-Based-Tree

We will skip the protocol details.

Building IP multicast applications

- Use UDP socket
- Build transport mechanisms (reliability, congestion control) with application
- Pick multicast address and hope no collision
- Multicast toolkit
 - Transport
 - Session announcement
 - Group member management, and reporting
- But in practice, IP multicast did not live up to its original promise
 - Using multicast address not convenient: need allocation, announcement...
 - In WAN, economic issue – who pays for it? Complicated to control if it is a network service
 - Security problems; some multicast routing mechanisms prone to DoS attack
 - How to do resource allocation relative to unicast traffic, when congestion?

IPTV

Used in local, private network, isolated from other traffic

- Avoids most of the challenges and limitations;
- Essentially, only IGMP is used to switch between channels (a different multicast address used for a different channel)
- The router/switch is doing the multi-destination forwarding, more efficient than a streaming server

- Some ISPs use IPTV to provide TV service to the home, including AT&T, Swiss Telecom, PCCW (HK)
 - The receiver is a set-top box (STB)
 - Content/service provider set up special infrastructure to deliver content to buildings
 - The service is charged

How about Video-on-Demand (VoD) service?

Video-on-Demand

- In reality, there is more demand for Video-on-Demand (VoD)
 - User viewing of the same video is not synchronized
 - There are lots of videos
 - IP multicast is not practical
 - Youtube, Netflix, Youku etc are all VoD services
 - The VoD service is usually delivered via Content Distribution Network (CDN), or Peer-to-Peer service (P2P)
- IPTV versus Internet TV
 - IPTV refers to TV broadcast via IP multicast
 - Internet TV usually refers to TV videos delivered by VoD service
- Since VoD is delivered over Internet, there is no service guarantee
- VoD service provider usually earn income by advertisement

Application layer multicast

- Towards end of 1990s and early 2000s, several notable proposals for “application layer multicast”
- This led to multiple notable projects on “overlay networks”:
https://en.wikipedia.org/wiki/Overlay_network
- They serve various distributed applications, such as VoIP, storage, distributed computing; for video content distribution:
 - Content Distribution Networks (CDN)
 - Peer-to-Peer networks (P2P)

Client-Server VoD

- From file downloading to streaming:
 - Copy the video file over, and play from a local file
 - Progressive downloading, start playing when enough of file buffered
 - Content treated as stream rather than file; try to minimize delay, may have to skip content if not delivered on time
- Video streaming protocols:
 - **RTSP** (Real Time Streaming Protocol), together with RTP/RTCP as transport protocol, supports VCR operations (skip, pause etc), used since late 1990s
 - **RTMP** (Real Time Message Protocol), also known as **Flash**
 - **DASH** (Dynamic Adaptive Streaming over HTTP), uses multiple copies for each chunk with different resolution
- One reason for using HTTP-based protocols: firewalls like HTTP

Content Distribution Network (CDN)

- As number of users increases, a single server is not enough
- **Server farms** with **load balancing**:
 - Use DNS to redirect request to different servers
 - Random redirection, or according to load
- Caching:
 - First by ISPs, using proxy servers
 - Content provider provisioned caching: can better handle dynamic content
 - DNS redirecting allocates edge servers to users according to locality and load
- This becomes the CDN:
 - A set of servers spread out in many areas (close to users)
 - A redirection service (by DNS and other proxy servers) to assign right server for each user request
 - A network (can be public or private network) that connects the edge servers
 - A content management service that places / replicates content to the edge servers according to need

CDN business

- There are many CDN services around the world
 - The large ones include: **Akamai**, **Amazon CloudFront**, **ChinaCache**...

<http://www.cdnplanet.com/cdns/>
- CDN providers need to peer with access ISPs
 - Their presence is good for access ISP, to reduce transit traffic
 - CDN is also good for content providers, offloading the streaming service
- Major content providers many use multiple CDN services at the same time, to cover different areas
- In China, CDN service helps solve another problem: weak ISP peering between major ISPs
- Different CDN strategies:
 - Use more servers, placed closer to users – require more work in peering arrangements
 - Use large servers or data centers – easier to expand

Peer-to-Peer Networks

- Can be viewed as a specialized CDN, the peers are both users (viewers) as well as servers!
- In order to server, the peers need content, where / how do they get such content?
 - While they get content to consume, they serve with the same content (for downloading and streaming)
 - They use a buffer to store content consumed already (for VoD)
- Interesting distributed algorithms invented to make this work
 - Lots of academic research
- P2P technology is also easily used to distribute copyrighted content
 - For this reason, major content providers tend to not use it

Tree-based vs data-driven

- IP Multicast is tree-based
- For P2P, multiple trees can be used at the same time for speed-up
 - Needs to effort of building the trees
- Most successful P2P protocols are data-driven:
 - Each chunk of data follows its own tree, dynamically built
 - It is more robust, even if peers join and leave at different times

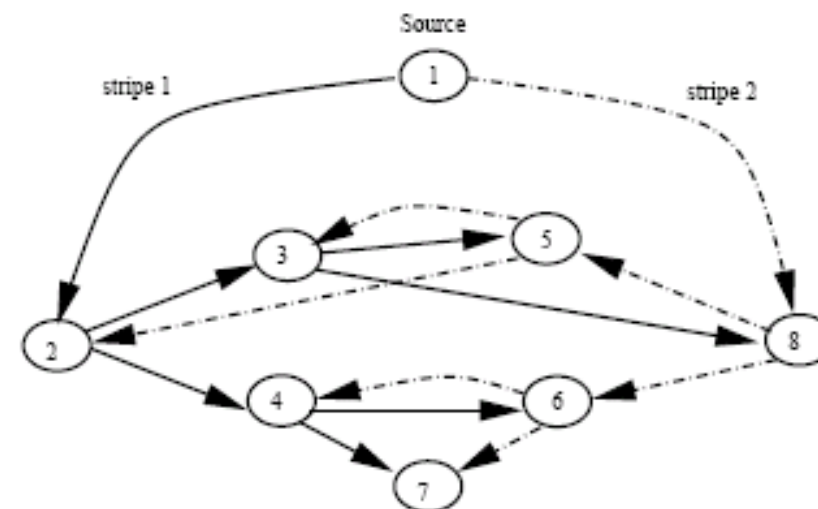


Figure 1: A simple example illustrating the basic approach of SplitStream. The original content is split into two stripes. An independent multicast tree is constructed for each stripe such that a peer is an interior node in one tree and a leaf in the other.

BitTorrent

Popular, and well-studied P2P

Key components:

- Web server
- Torrent file:
 - Name
 - Size
 - checksums (one for each chunk)
 - tracker address
- Tracker: keeps track of users, their address etc

• Peers

- Users who want to use BitTorrent must install client software or plug-in for web browsers.
- *Downloader (leecher)* : A peer who has only **a part (or none)** of the file.
- *Seeder*: A peer who has the **complete** file, and chooses to stay in the system to allow other peers to download

BitTorrent algorithm

- A file is split into pieces of fixed size, typically 256Kb
- Each downloader reports to all of its peers what pieces it has.
- To verify data, Hash codes are used for all the pieces, included in .torrent files.
- Peer selection algorithm
 - This is used to provide incentives
- Piece selection algorithm
 - This helps build a separate tree (implicitly) to distribute each chunk, to maximize uploading capacity
 - The algorithm also helps to minimize losing a piece

Piece selection algorithm

- Strict Priority
 - Once you selected a piece, try to get all segments of that piece
- **Rarest First**
 - During download, always try to get the rarest piece
- Random First Piece
 - At the beginning, just try to get a piece, randomly
- Endgame Mode
 - Near the end, try hard to finish, ask all peers for pieces

These are all engineering heuristics, but they work exceptionally well, in simulation as well as real system

Peer selection algorithm

Used as incentive mechanisms:

- *Neighbors*

- Always maintain several “neighbors” to exchange pieces

- Keep track their downloading rates

- Default = 4 neighbors

- *Optimistic Unchoking*

- Periodically pick a new neighbor randomly and see if it is good;

- If good, replace a regular neighbor

- That regular neighbors is “choked”

- This is used to discover new neighbors

- *Anti-snubbing*

- If all neighbors not responding, try to select more than one neighbor randomly

- Simultaneously unchoke multiple new neighbors

This strategy is sometimes referred to as “tit-for-tat”

Why does BitTorrent scale?

- When multicast is implemented in the network layer, a single multicast tree is used to reach all receivers
- In BT, you can think of the pieces of a file following **different** (often non-overlapping) multicast trees
- Each peer can be thought of as a server, helping the source in distributing the file
- So, as long as the peers are reasonably powerful, they can help server to achieve throughput equal to its uplink capacity.

Bram Cohen, "Incentives Build Robustness in BitTorrent",

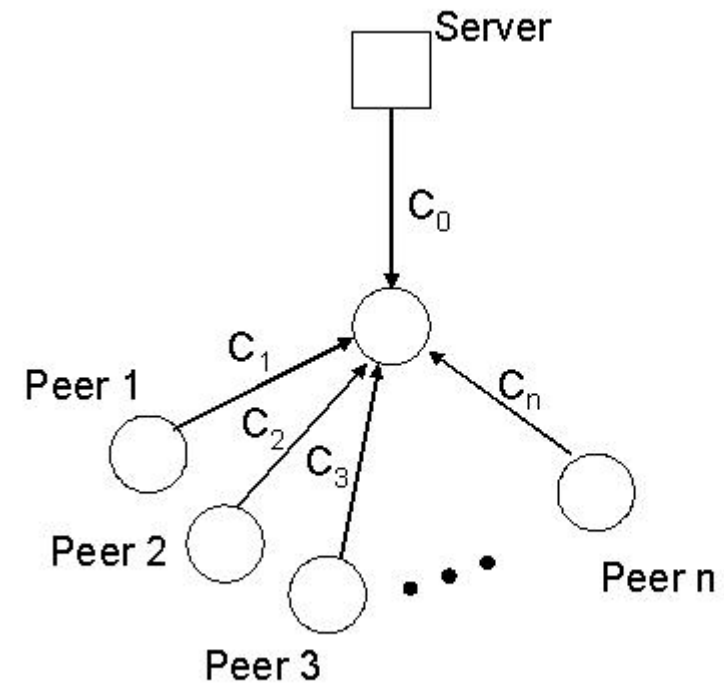
<http://bittorrent.com/bittorrentecon.pdf>

3475 citations according to Google Scholar

Uplink sharing model

- An abstract model to understand why P2P scales well
- Server's uplink capacity is C_0
- Each peer's uplink capacity is C_i
- The rest of the network not a bottleneck
- The downlinks also not a bottleneck

What is the maximum rate the server can download content to **all peers**?



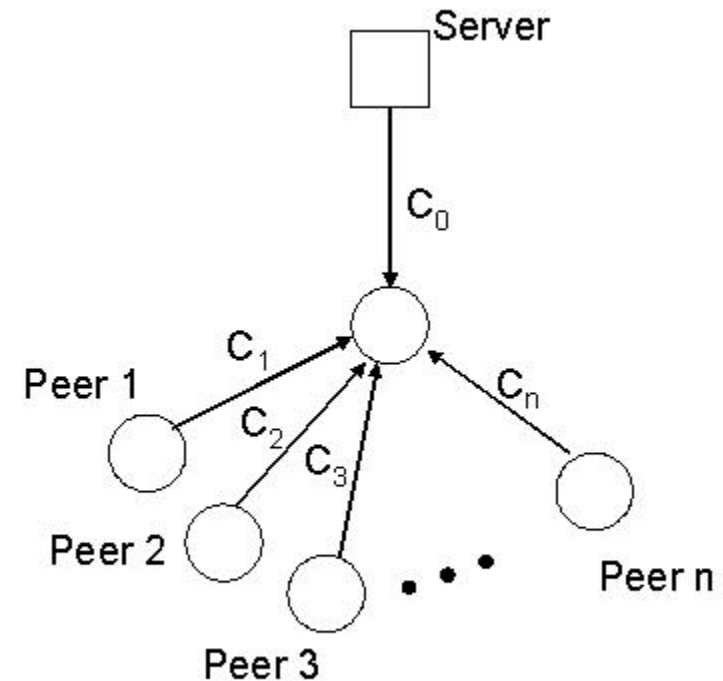
Upper bound of throughput

What is the maximum unicast throughput?

From server to one peer = C_0

What is the maximum throughput if there is network multicast? Also C_0

P2P throughput cannot be more than C_0 , as all content comes out of server

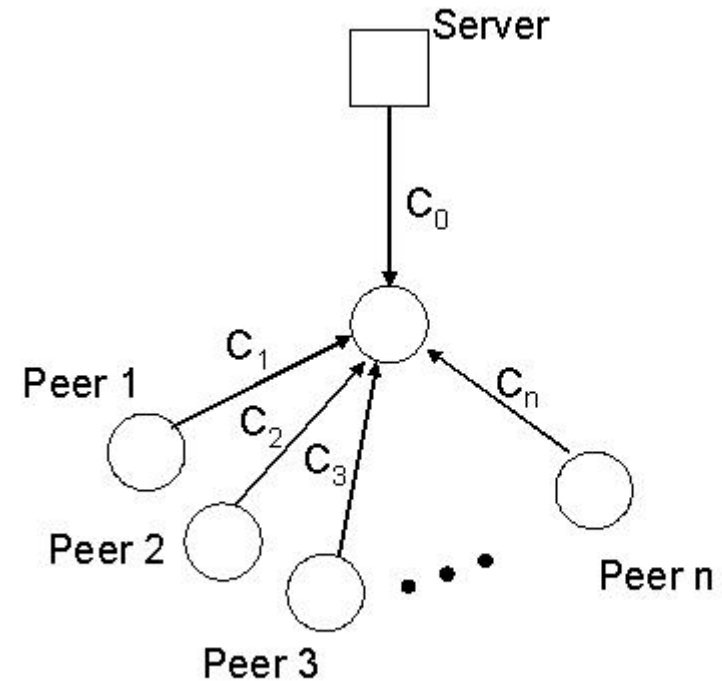


Another upper bound for throughput

- Assume each peer already has the whole file
- Each peer wants to receive another copy from any peer
- What is the minimum time (hence maximum rate) for all peers to get another copy?

$$= (C+C_0)/n$$

$$\text{where } C = \sum_{j=1,n} C_j$$



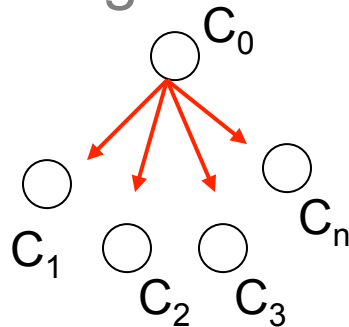
Can the upper bound be achieved?

- Yes, through very careful scheduling and coordination
- The upper bound is

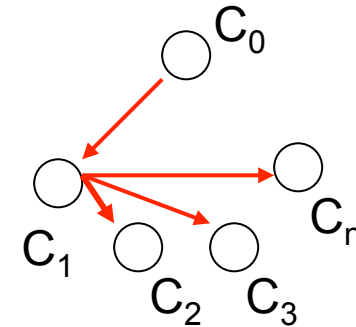
$$R = \min\left\{C_0, \frac{C_0 + \sum_j C_j}{n}\right\}.$$

- Consider using the following spanning trees:

- “1-hop tree”
(from server to all peers)



- and n “2-hop” spanning trees (server sends to 1 peer, which then forward the content to rest of the peers)



- How to assign rates to each tree, subject to uplink constraints, to maximize total downloading rate?

How upper bound is achieved

- Case 1: $C_0 > C/(n-1)$ where $C = \sum C_i$
 - This means $(C_0+C)/n < C_0$
 - Assign rate $C_i/(n-1)$ to the i^{th} 2-hop spanning tree
 - The i^{th} spanning tree can deliver rate $C_i/(n-1)$ to all peers
 - Assign rate $C_0 - C/(n-1)$ to the 1-hop spanning tree
 - Each peer receives $C/(n-1)$ from 2-hop trees, and $(C_0 - C/(n-1))/n$ from 1-hop tree
 - The total each peer receives is $(C_0+C)/n$
- Case 2: $C_0 \leq C/(n-1)$
 - This means $C_0 < (C_0+C)/n$
 - Assign rate $C_0 * C_i/C$ to the i^{th} 2-hop spanning tree, which can then deliver to all other peers at that rate
 - Each peer receives $\sum(C_0 * C_i/C) = C_0$

Assumptions, and examples

- There are “infinitely” number of pieces to distribute
 - Or alternatively, each piece is very small -> “fluid assumption”
 - We consider the throughput in “steady state”
- Perfect scheduling so there is no wastage on any uplink capacity

Uplink sharing model (Mundinger)

<http://www.statslab.cam.ac.uk/~jm288/publications.html>

Our paper explaining this:

<http://personal.ie.cuhk.edu.hk/~dmchiu/p2pnetcoding.pdf>

- Example 1:
 - Two peers, with uplink 1Mb/sec and 2Mb/s
 - Server’s uplink is 1Mb/sec
 - What is the max throughput?
 - What is server’s uplink is 10Mb/sec?
- Example 2:
 - Suppose in a p2p session, there are 1000 home users (ADSL access links with uplink 100Kb/sec)
 - The content provider would like to provide video at 400Kb/sec
 - What server uplink is needed?
 - Suppose server uplink is only 1Mb/sec, but there are some additional peers from Ethernet, how many such high BW uplink peers are needed?

P2P streaming

- BitTorrent-like P2P protocol can be used to achieve P2P streaming
- In streaming, it is natural to download content sequentially
 - If all peers downloading the same (few) chunks, then the degree of parallelism is limited
 - “Rarest first” helps maximize parallelism
- Use modified piece selection: mixed strategy

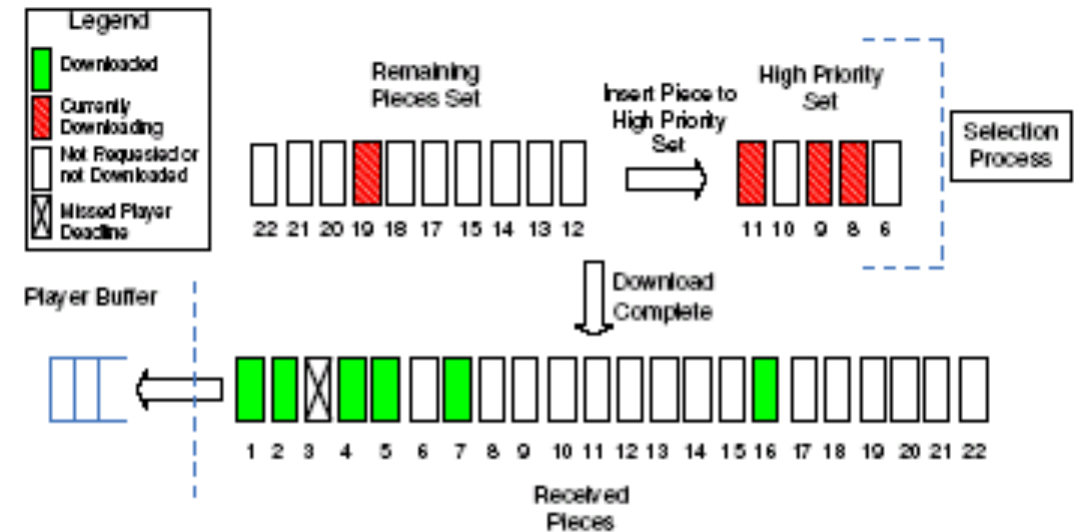


Fig. 1. Our Approach for Supporting Streaming in BT

P2P streaming first demonstrated by CUHK student

- Coolstreaming

Xinyan Zhang et al, “CoolStreaming/
DONet: A Data-driven Overlay
Network for Peer-to-Peer Live Media
Streaming”

[http://citeseer.ist.psu.edu/
zhang05coolstreamingdonet.html](http://citeseer.ist.psu.edu/zhang05coolstreamingdonet.html)

XY Zhang was a CUHK M.Phil student

“Deployed” in May 30, 2004 during
European soccer finals;

Total 30000 downloads, 4000
simultaneous users

- Another experimental system:
BiTos

[http://home.ie.cuhk.edu.hk/~dmchiu/
p2pstreaming_ton.pdf](http://home.ie.cuhk.edu.hk/~dmchiu/p2pstreaming_ton.pdf)

- We created a theoretical model to
explain it works:

[http://home.ie.cuhk.edu.hk/~dmchiu/
p2pstreaming_ton.pdf](http://home.ie.cuhk.edu.hk/~dmchiu/p2pstreaming_ton.pdf)

System and performance

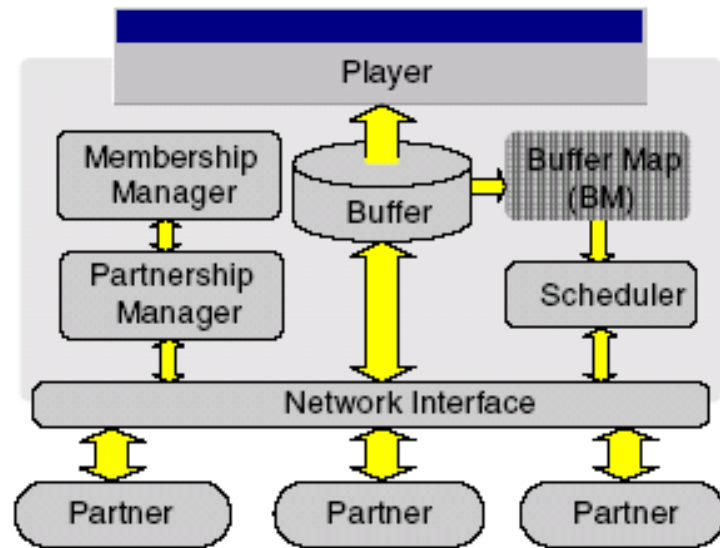


Fig. 1. A generic system diagram for a DONet node.

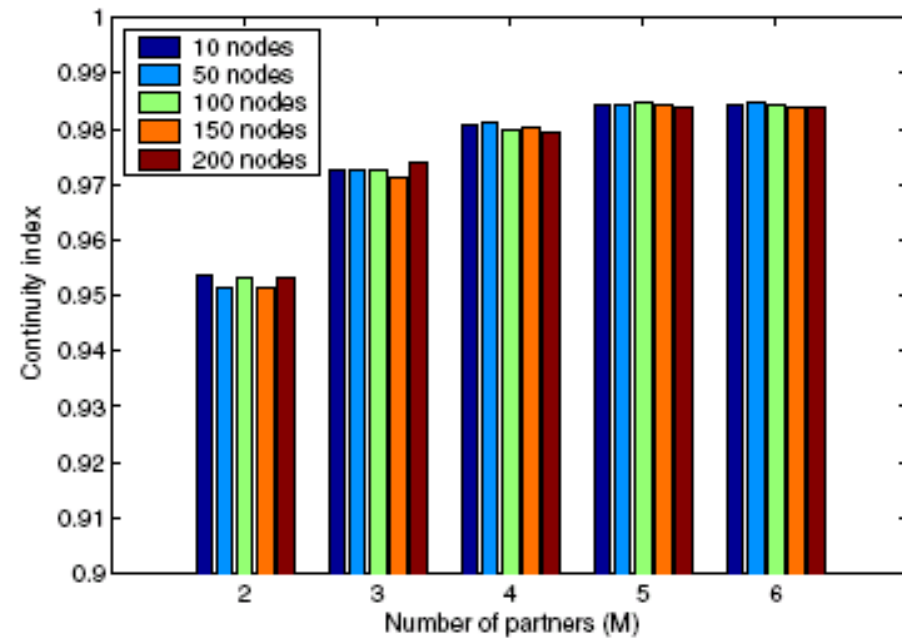


Fig. 7. Continuity index as a function of the number of partners.

Robustness of performance

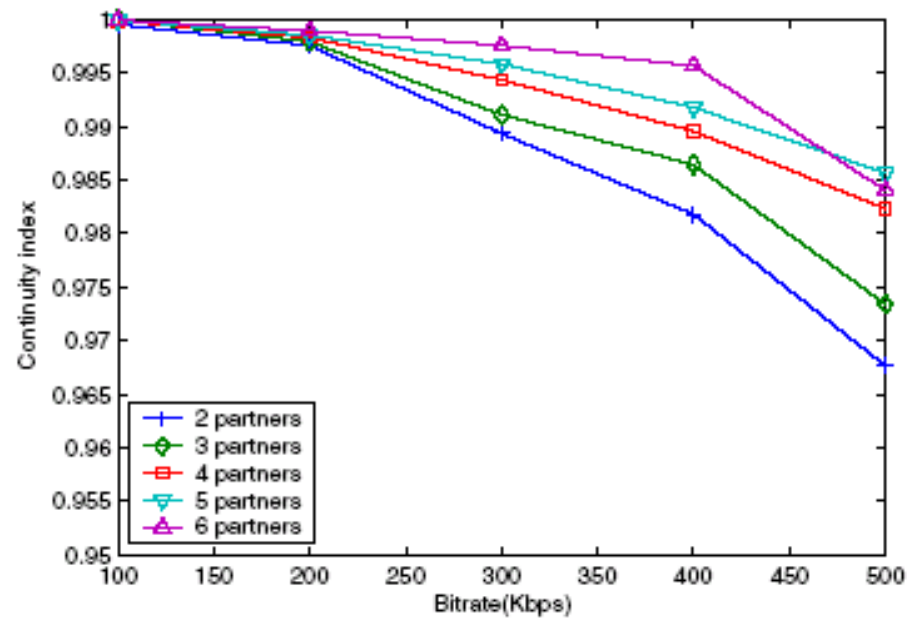


Fig. 8. Continuity index as a function of the streaming rate. Overlay size = 200 nodes.

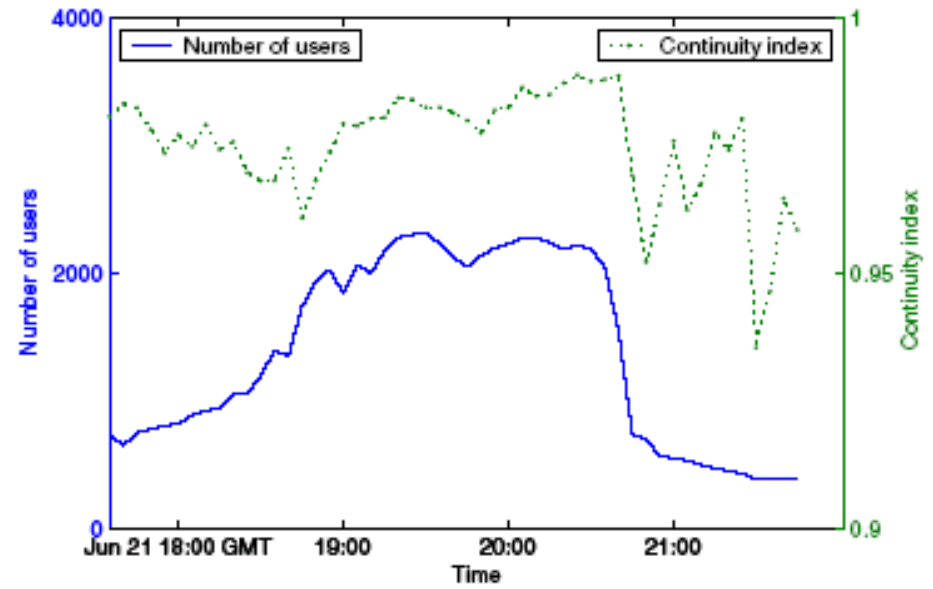


Fig. 14. Number of users and Continuity index over time.

P2P VoD

- Several companies doing P2P VoD in China, since 2007
 - Viewer population reached 100s Ks quickly
 - Playback rate ~400Kb/s
 - Initially, 20+% bandwidth supported by servers
 - After optimization, around 10-12% from servers
- P2P VoD is harder than P2P streaming
 - Some less popular movies may have fewer peers watching
 - Peers viewing same movie may not be synchronized
 - Need to support “VCR” operations – pause, skip forward etc
- What is the secret?
- We collaborated with PPLive, and wrote a ACM Sigcomm paper on the design of a P2P VoD system

Y Huang, ZJ Fu, DM Chiu, JCS Lui and C Huang, "Challenges, Design and Analysis of a Large-scale P2P VoD System", ACM Sigcomm 2008

Other results on P2P VoD

- In P2P VoD, we show how to do replication if peers' storage of videos is limited: try to load balance, and make peers store content as different as possible
- We showed that if both Server and P2P are used to support VoD, then you should try to serve hot contents by server and long tail content by peers
- We have done many other work on P2P systems, see <http://home.ie.cuhk.edu.hk/~dmchiu/publications.html>

Summary

- Content distribution is an important service in Internet
 - Great effort put into IP multicast, to create a more efficient infrastructure for content distribution, but it has various limitations
 - Current solution is based on CDN, and to a smaller extent P2P, which is similar to “sharing economy”
 - P2P system design is innovative
- Question:
 - Can we fix the key problems with IP multicast, and create a better network infrastructure for content distribution?