Exploring Large Scale Peer-to-Peer Live Streaming Tolopgies

Authors: Chuan Wu, Baochun Li, Shuqiao Zhao

ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 4, No. 3., August 2008

Presented by Klara Nahrstedt

CS598KN, Fall 2017

Outline

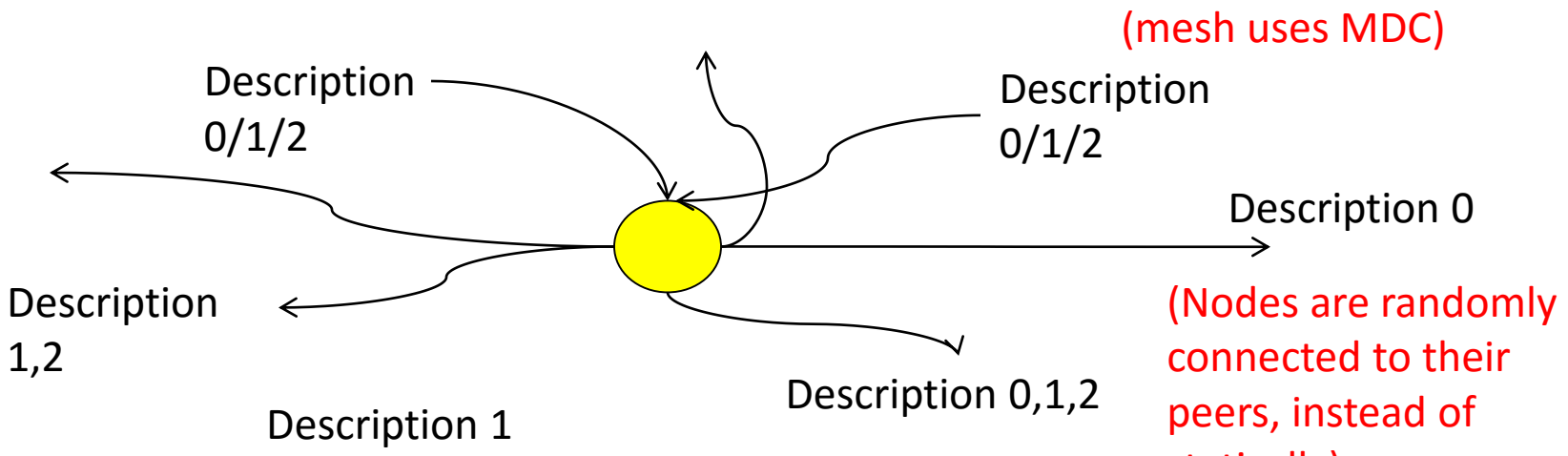
- Administrative (no class on October 19 (Thursday)
  - Presentation and review will be due on October 31
- Motivation and Background
  - Mesh topology
  - PPLive
- UUSee Description
- Problem Description
- Measurement Metrics to Evaluate P2P Live Streaming Toplogies
- Selected Measurement Results
- Conclusion

Motivation

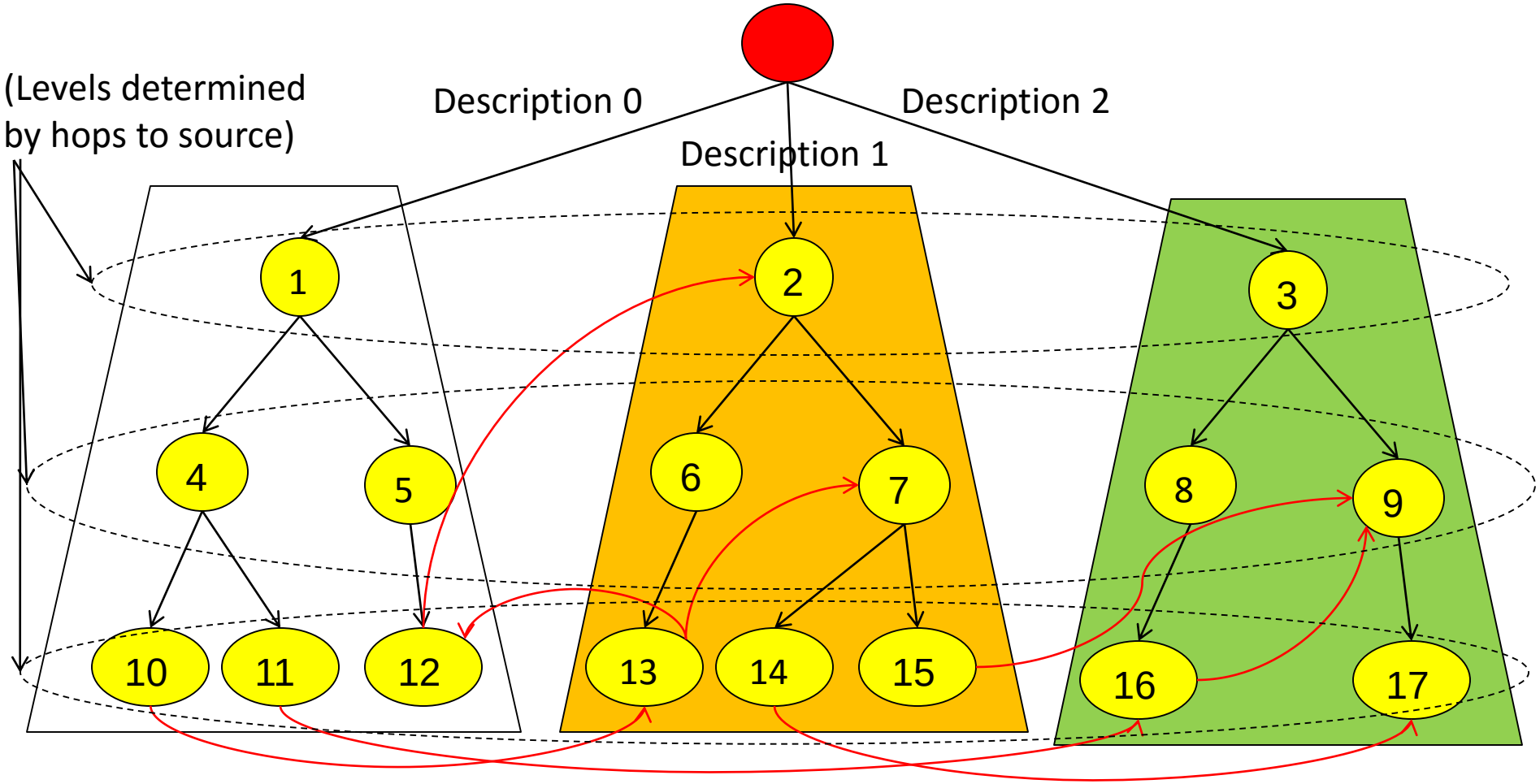
- P2P communication paradigm for multimedia streaming apps has been successfully deployed in the Internet
- Examples of P2P Live multimedia streaming systems:
  - CoolStreaming (2005)
  - PPLive (2006, 2007)
  - TVAnt (2006, 2007)
- Goal of P2P streaming systems
  - Shift bandwidth burden from servers to end systems
  - Shift bandwidth burden from dedicated streaming servers by maximally utilizing peer upload bandwidth
- Approach
  - Allow buffering/caching chunks/data blocks at peers and exchange among themselves
- Result
  - Mesh-based design of topologies (instead tree-based topologies) with
    - better resilience to peer dynamics,
    - better scalability in flash crowds
    - More efficient use of bandwidth
    - Simplicity with respect to topology maintenance

Mesh-based streaming

- Basic idea
  - Report to peers the packets that you have
  - Ask peers for the packets that you are missing
  - Adjust connections depending on in/out bandwidth

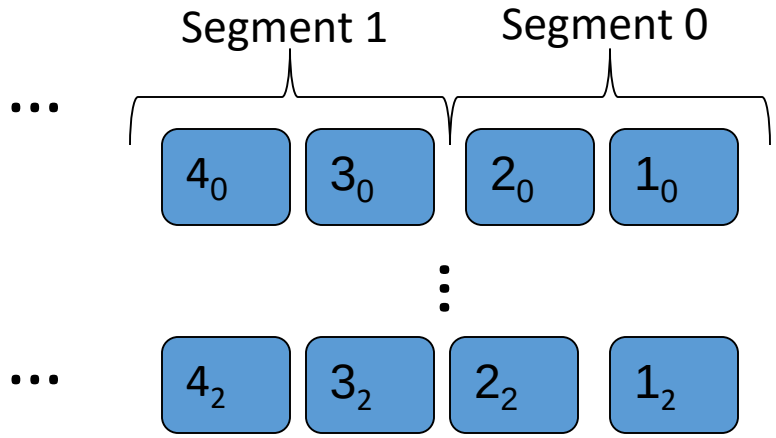


Content delivery

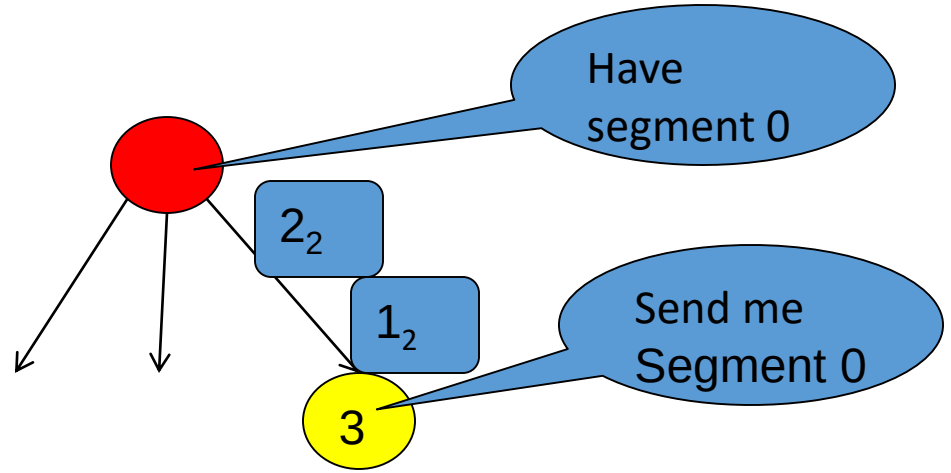


(1) Diffusion Phase (⟶) (2) Swarming Phase (↷)

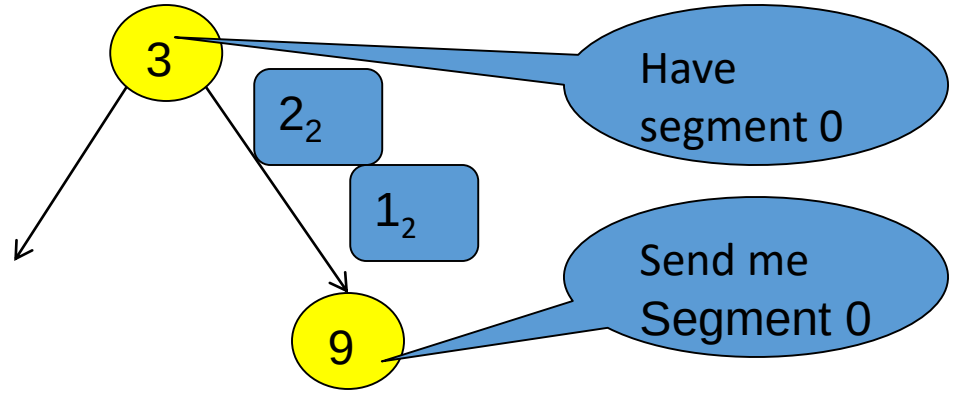
Diffusion Phase



- As a new segment (set of packets) of length *L* becomes available at source every *L* seconds
  - Level 1 nodes pull data units from source, then level 2 pulls from level 1, etc.
  - Recall that reporting and pulling are performed periodically



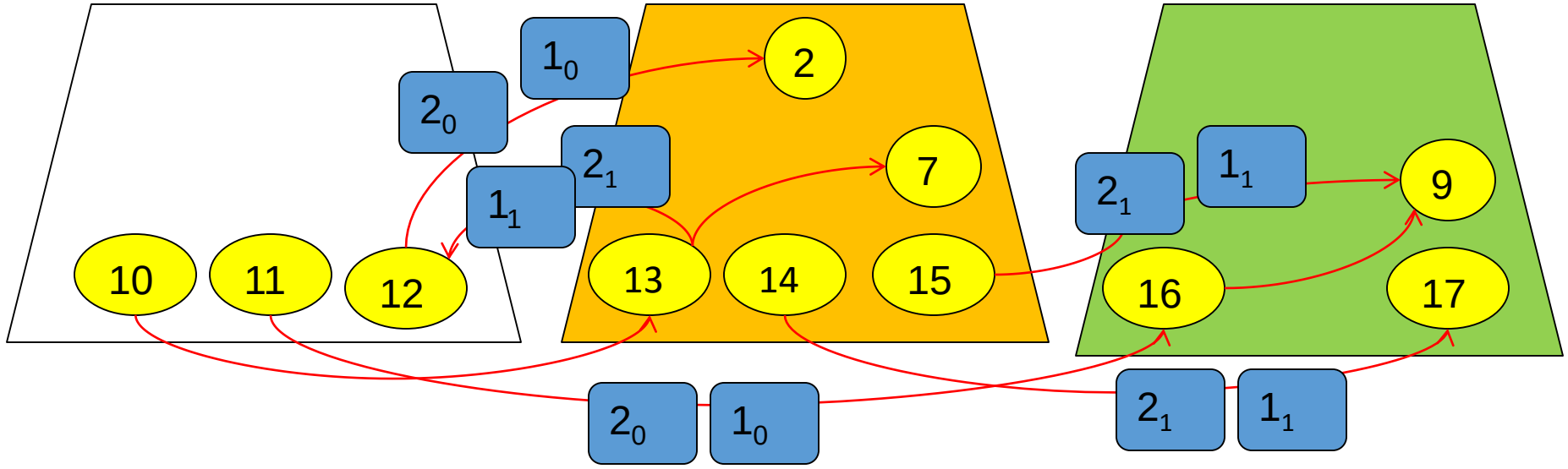
(during period 0)



(during period 1)

(drawings follow previous example)

Swarming Phase



- At the end of the diffusion all nodes have at least one data unit of the segment
- Pull missing data units from (swarm-parent) peers located at same or lower level
- Can node 9 pull new data units from node 16?
  - Node 9 cannot pull data in a single swarm interval

(drawings follow previous example)

PPLive

Case Study: PPLive

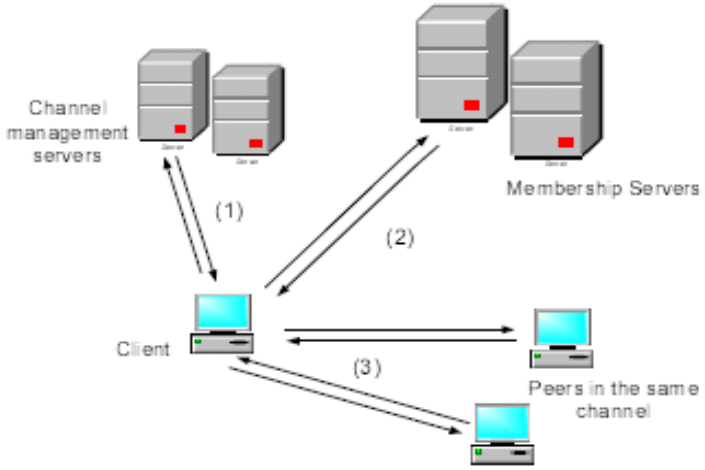
- Very popular P2P IPTV application
  - From Huazhong U. of Science and Technology, China
  - Free for viewers
  - Over 100,000 simultaneous viewers and 500,00 viewers daily (and increasing)
  - 100s of channels
  - Windows Media Video and Real Video format

PPLive Overview

Catalog Name	Number of channels
TV	52
Information	29
Sports	1
PhonenixTV	5
Movies	79
Teleplay	66
Entertainment	68
Cartoon	30
Game	28
Others	52
Summary	410

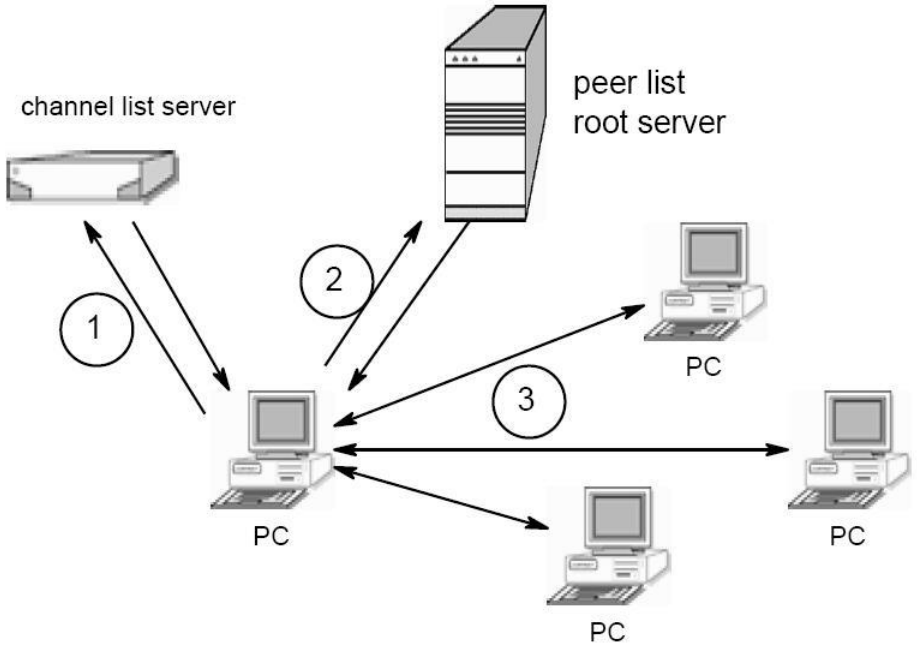
PPLive Design Characteristics

- Gossip-based protocols
  - Peer management
  - Channel discovery
  - TCP used for signaling
- Data-driven p2p streaming
  - TCP used for video streaming
  - Peer client contacts multiple active pe channel
  - Cached contents can be uploaded from a client peer to other peers watching the same channel
  - Received video chunks are reassembled in order and buffered in queue of PPLive TV Engine (local streaming)



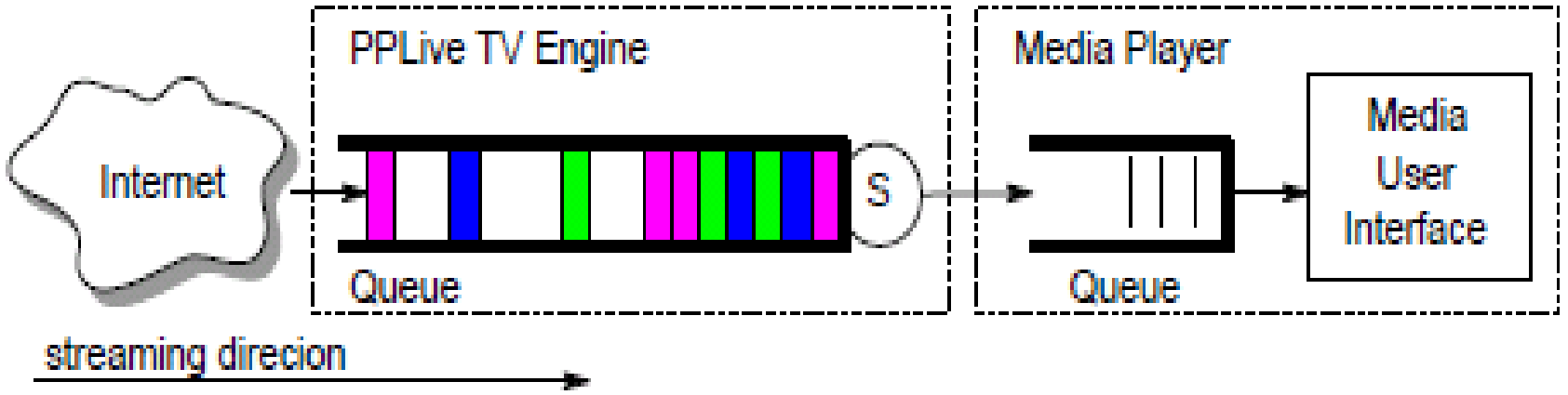
PPLive Architecture

1. Contact channel server for available channels
2. Retrieve list of peers watching selected channel
3. Find active peers on channel to share video chunks



Source: "Insights into PPLive: A Measurement Study of a Large-Scale P2P IPTV System" by Hei et al.

P2P Streaming Process



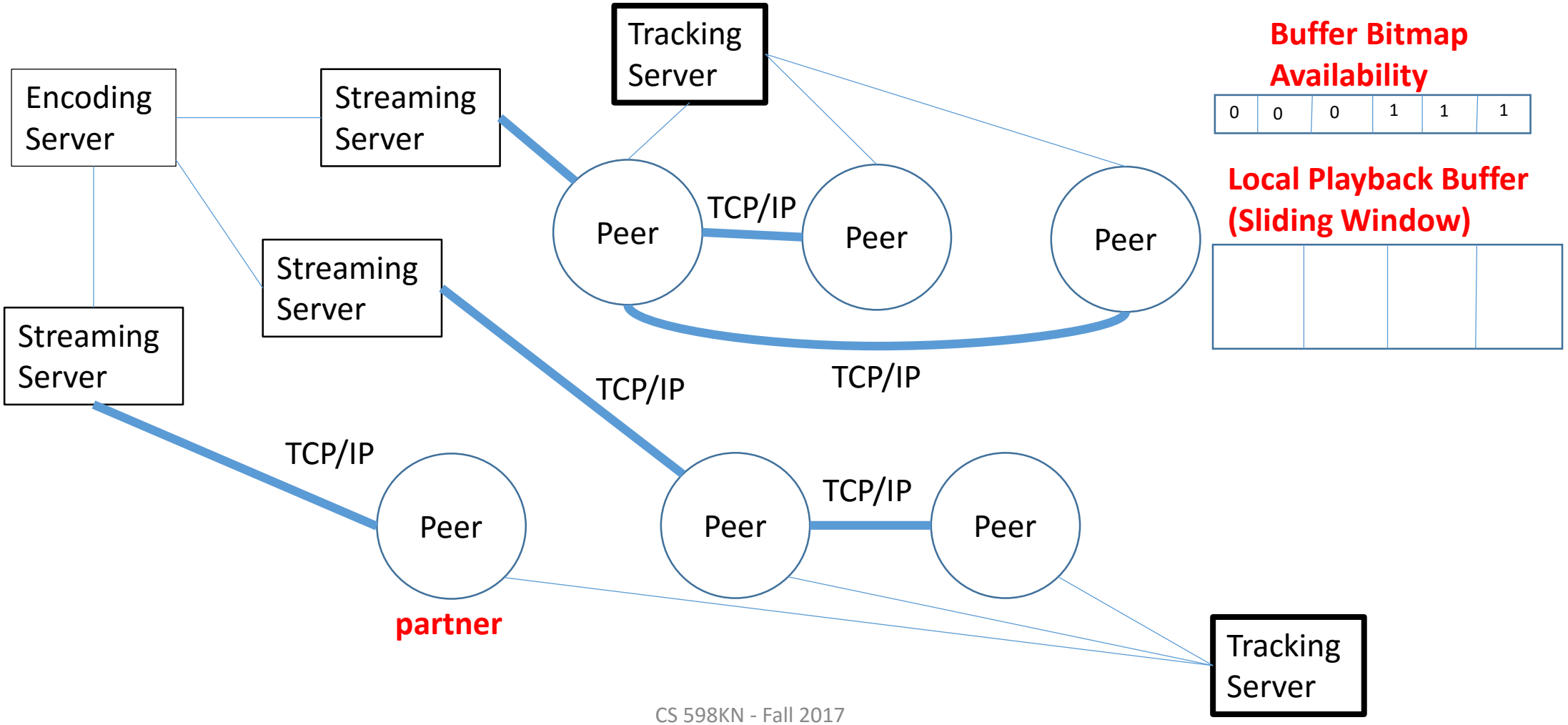
TV Engine – responsible for
- downloading video chunks from PPLive network
- streaming downloaded video to local media player

UUSee Inc

UUSee

- Leading P2P Streaming solution providers in China
- Contractual rights with CCTV
- Rich collection of legal content, encoded and broadcast live over the Internet
- Encoding of may channels at encoding server
- Media channels with high quality constant rate streams around 400 Kb/s using proprietary codec

UUSee Architecture



UUSee Functions

- Peer joins and does
  - Contact a local tracking server
  - Get a group of potential partners
  - Establish TCP/IP connection with these partners
  - Exchange buffer availability bitmap
- During streaming peer does
  - Measures TPC throughput of the connection
  - Executes estimation algorithm
  - Predicts partner's availability to serve itself
  - Ranks all known partners
  - Selects best 30 peers from which it actually requests media blocks
  - Employs "synchronous playback", i.e., each newly joined peer always starts to buffer media blocks that are to be played 20 seconds later than the current playback time of media channel at encoding server (all peers in the TV channel follow similar playback progress

UUSee Parameters

- UUSee partner buffer size – 500 media blocks
- Each block represents 1/3 second of media playback
- New peer starts media playback from the first buffered block after 20 seconds (if sufficient data is buffered; if not, re-buffering starts again)
- Initial startup delay at peers is usually 20 seconds (if re-buffering is needed, startup delay can be 40 seconds)
- Peers will stop filling up its buffer when buffer reaches around 75% of its total size.

UUSee Algorithms

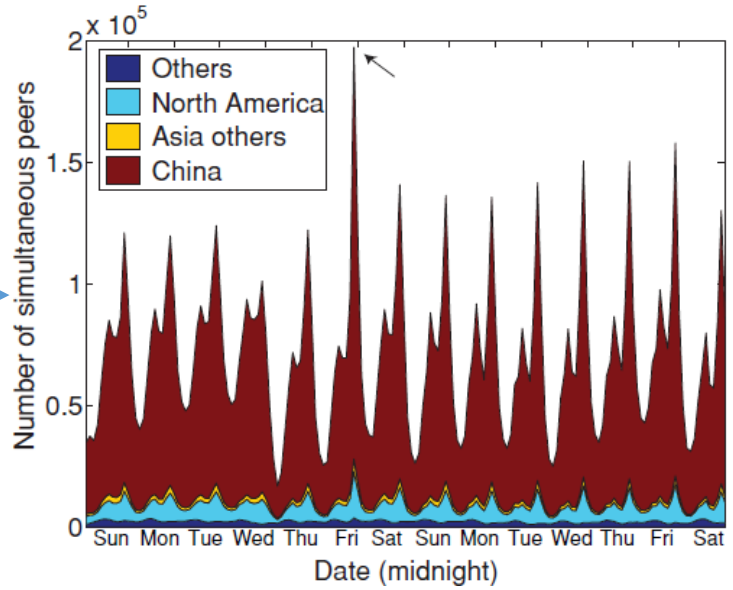
- During initialization phase, peer does
  - Estimate its maximum download and upload capacities
  - Each peer continuously estimates its aggregate instantaneous receiving and sending throughput from and to all its partners
  - If its estimated sending throughput is lower than its upload capacity for 30 seconds, it will inform one of the tracking servers that it is able to receive new connections from other peers.
  - Tracking servers keep list of peers that are able to accept new connections and bootstrap new peers with existing peers that are randomly selected from this set.
- During streaming phase, peer does
  - Keeps track of current playback buffer via buffer count
  - Neighbor peers recommend known peers to each other
  - Buffer count is used as important criteria for such recommendations (if a peer ‘i’ finds that other peer ‘j’ has a low buffer count, peer ‘i’ recommends known peers with larger buffer counts.

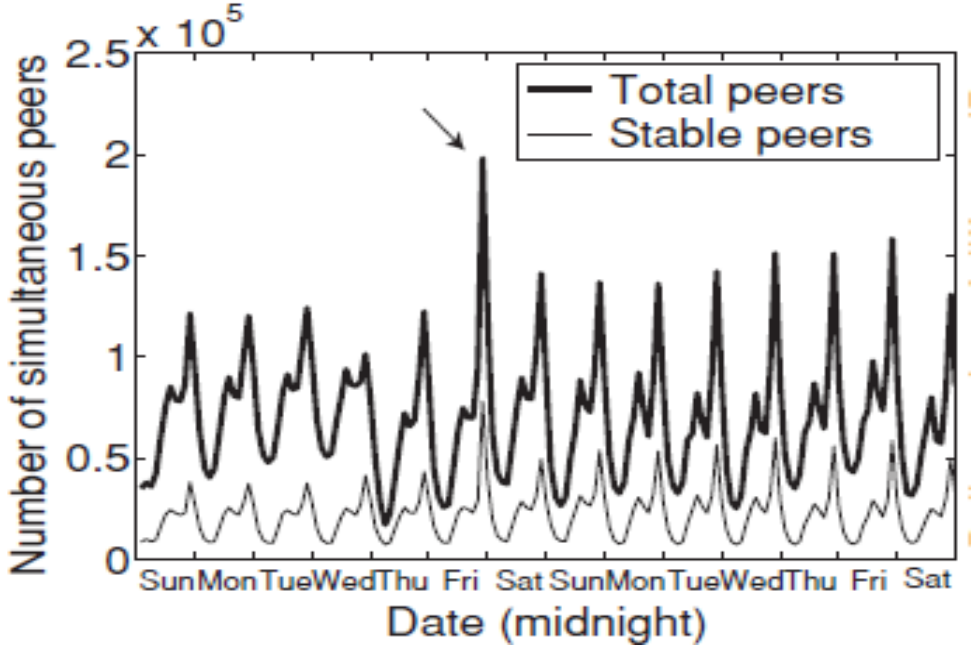
Magellan: Collection of Traces

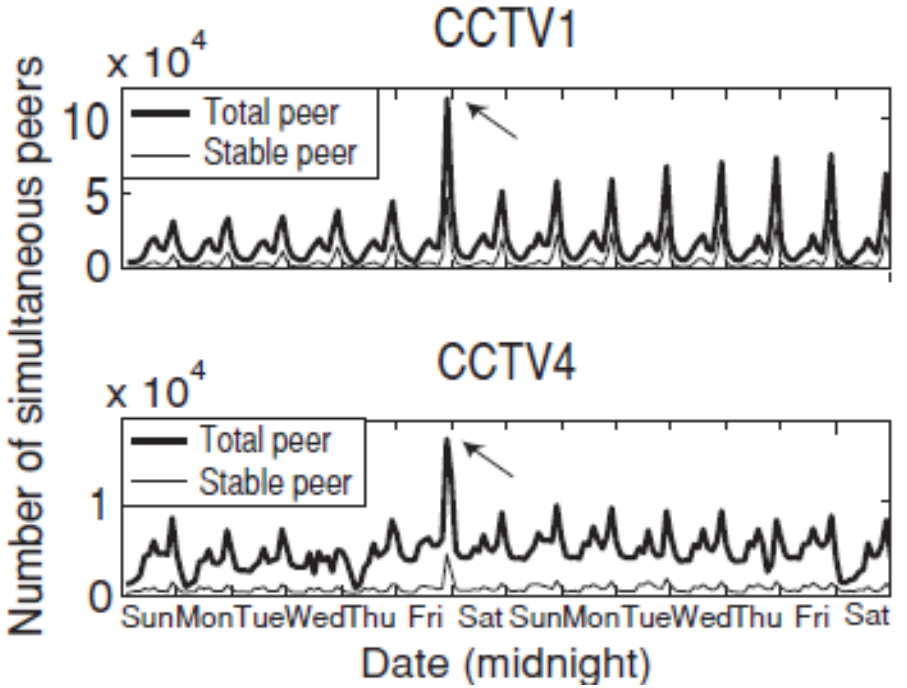
- **Instrument** and measure performance metrics and protocol parameters within UUSee streaming protocol
- Trace server collects measurement reports from existing peers
- Each report includes
  - IP address
  - Watched Channel
  - Buffer availability bitmap
  - Total download and upload bandwidth capacities
  - List of all peer's partners with their addresses, ports, number of blocks sent/received, throughput
- Reports are sent via UDP to trace server
- New peer sends reports after 20 minutes, subsequent reports are sent every 10 minutes (interest only in long-lived peers)
  - Long lived peers are more representative of P2P topologies
  - Transient peers may appear in partner lists

Scale of UUSee Topologies

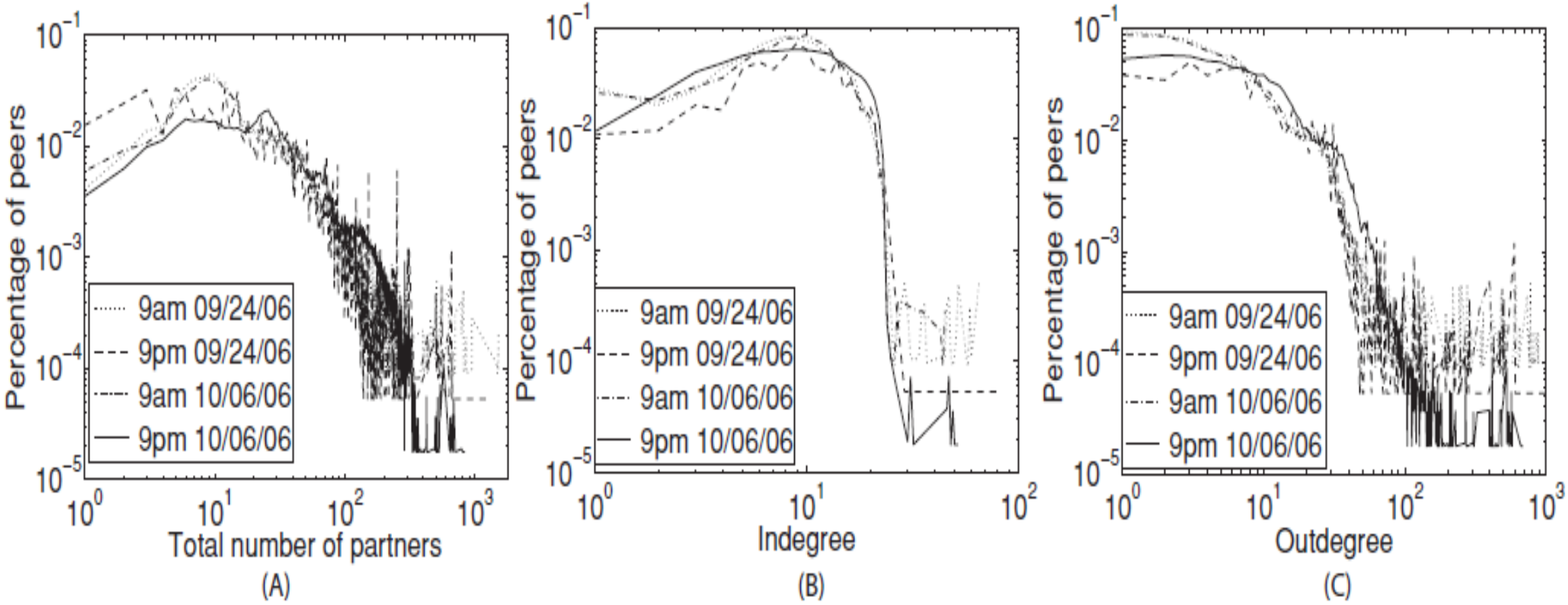
- Measure
  - Overall number of simultaneous peers
  - Number of Simultaneous Peers in Two Representative Channels
  - Number of Simultaneous Peers in Different ISPs and Areas







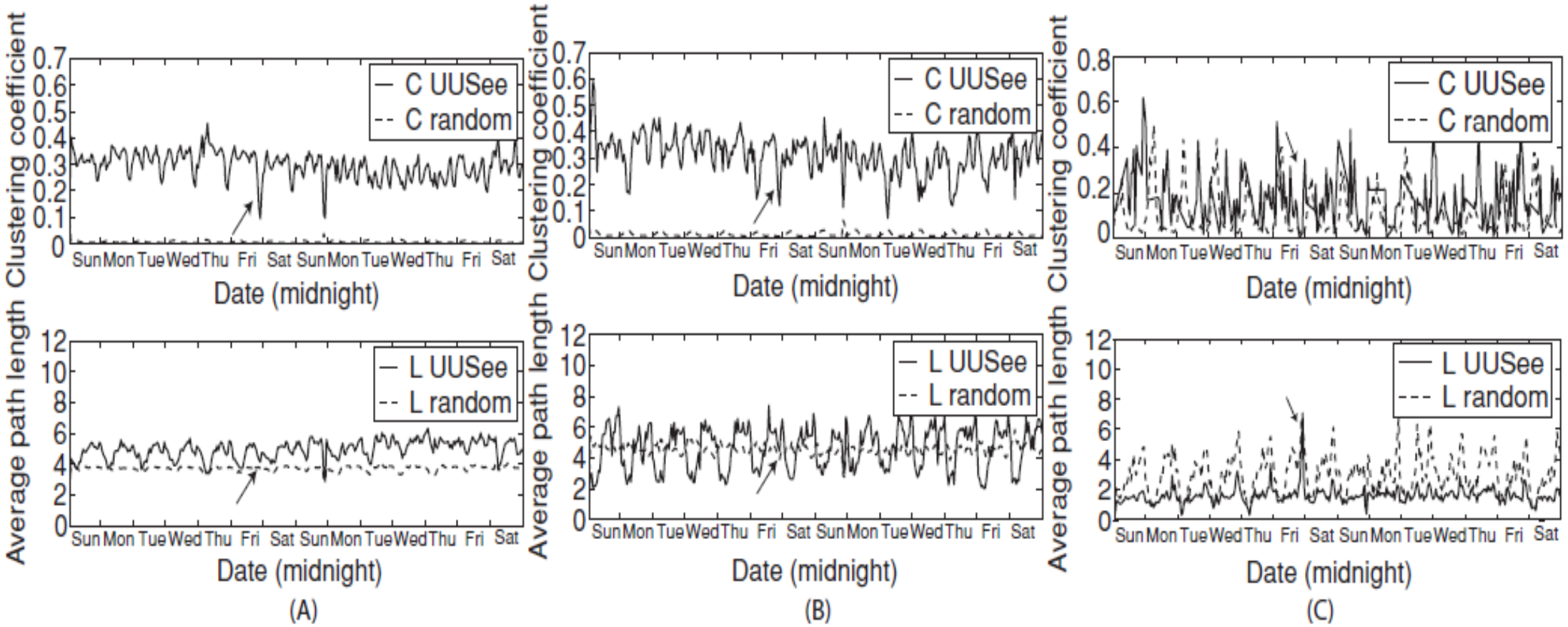
Degree Distribution of Stable Peers



Clustering

- Studies on Gnutella network point out that Gnutella exhibits "small world" properties, i.e.,
  - Peers are highly clustered with small pairwise shortest path lengths, as compared to random graph of similar peer numbers and link densities
- To investigate whether an undirected graph g is small-world graph, clustering coefficient is calculated as $C_g = \frac{1}{N} * \sum_{k=1}^{N} C_k$
  - N – total number of vertices in g
  - C_k – clustering coefficient for vertex 'k', calculated as proportion of edges between vertices within its neighborhood to the number of edges that could possibly exist between them.
  - Graph g is identified as small world if
    - (a) it has small average pairwise shortest path length l_g, close to that of corresponding random graph l_r;
    - (b) large clustering coefficient C_g , which is orders of magnitude larger than that of the corresponding random graph Cr.

Clustering (2)



A: small-world metrics for entire stable-peer graph, B: small-world metrics for ISP subgraph, C: small-world metrics for Area subgraph (Zhejiang Province)

Conclusion

- Many other metrics have been considered in addition to clustering and degree distribution
  - edge reciprocity (i.e., if there is edge from i to j, then there is also edge from j to i)
  - super-node connectivity (nodes with high sending throughput)
- Do you think these measurements revealed sufficient information about UUSee topology?
- How meaningful are these studies and measurements?