Peer-to-Peer Systems and Applications

Lecture 8: P2P Video Streaming

TECHNISCHE UNIVERSITÄT DARMSTADT

Chapter 11:

Part IV. Peer-to-Peer-Based Applications

*Original slides provided by Osama Abboud and Julius Rückert (Technische Universität Darmstadt)

0. Lecture Overview

1. Introduction to Video Streaming
   1. What is Streaming?
   2. Live vs. on-Demand Streaming
   3. Content Dissemination
   4. Summary
2. P2P Streaming Concepts
   1. Topologies and Coordination
   2. Examples: PPLive, BitTorrent Live, Transit
3. Advanced Video Streaming Concepts
   1. Adaptive P2P Streaming using H.264 SVC
   2. Case Study: P2PStream
   3. Quality of Service vs. Quality of Experience
4. References

1. Introduction to Video Streaming

What is Streaming, Live vs. on-Demand Streaming,
Content Dissemination, Summary

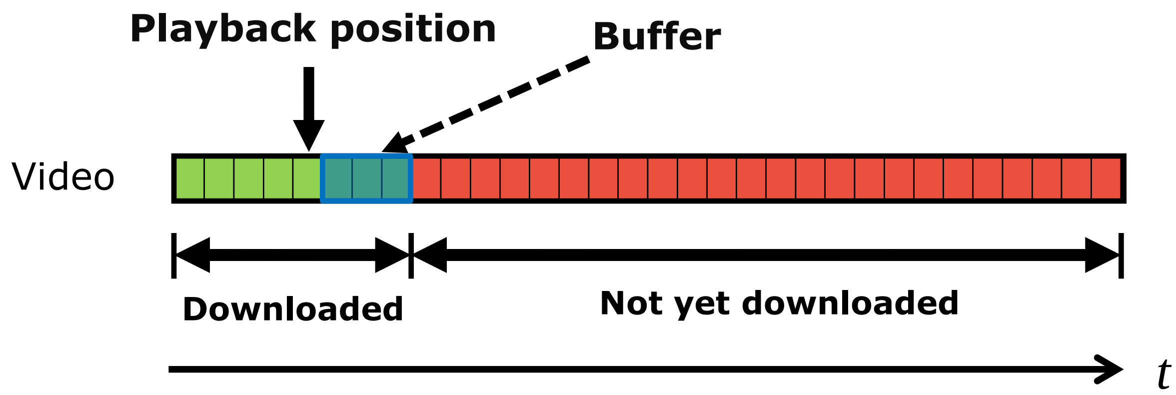
1.1. What is Video Streaming?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Main characteristics
  - Playback of video while it is transferred
  - Short waiting time before playback can start
- File sharing in contrast
  - Complete transfer of (video) file necessary for playback
  - Long waiting time before playback start possible
- Two main application scenarios
  - Live streaming
  - Video-on-Demand (VoD) streaming

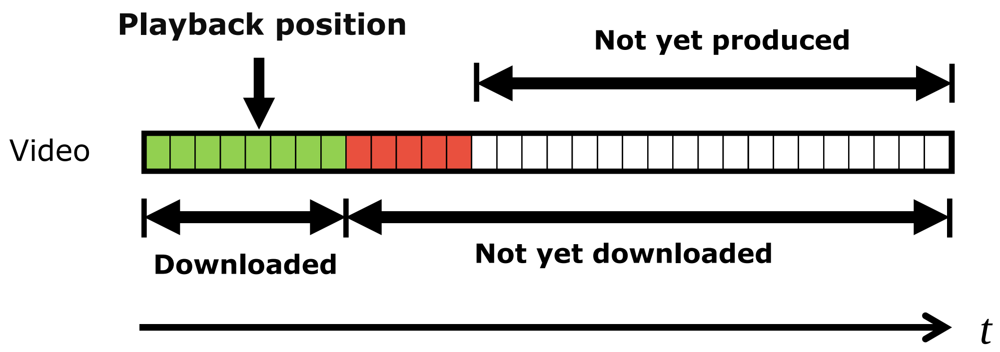
1.1. What is Video Streaming?

- General properties of streaming process
  - Buffer is used to account for jitter and bandwidth fluctuations
    - Buffer size: usually several seconds of video
  - For fluid playback, download rate must be higher than video bit rate
  - In case of buffer underrun, two strategies:
    - Pause playback (also called *playback stalling*)
    - Proceed with playback (e.g. by skipping frames)
      → Degraded video quality (type of degradation highly depends on video codec)



1.2. Live Streaming

- Traditional TV broadcasting scenario
  - Content provider distributes video to many receivers
  - Objective: deliver stream as "live" as possible
    - Live: as fast and synchronized as possible
- Content is often generated while streamed
  - Limited possibility to preload video data
  - Example: broadcasting of a soccer game



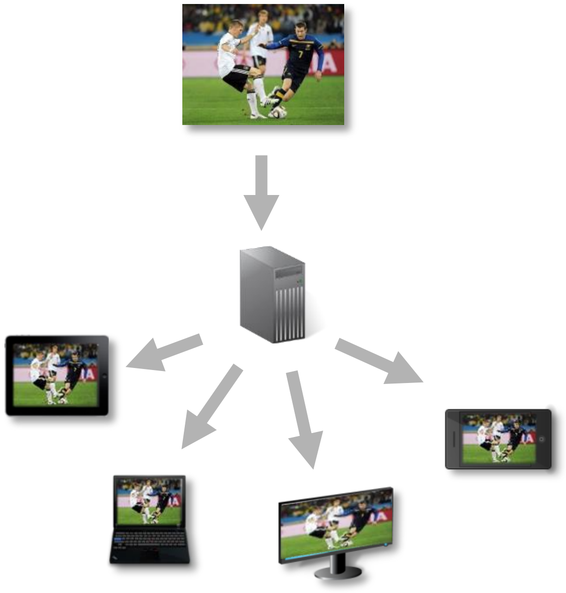
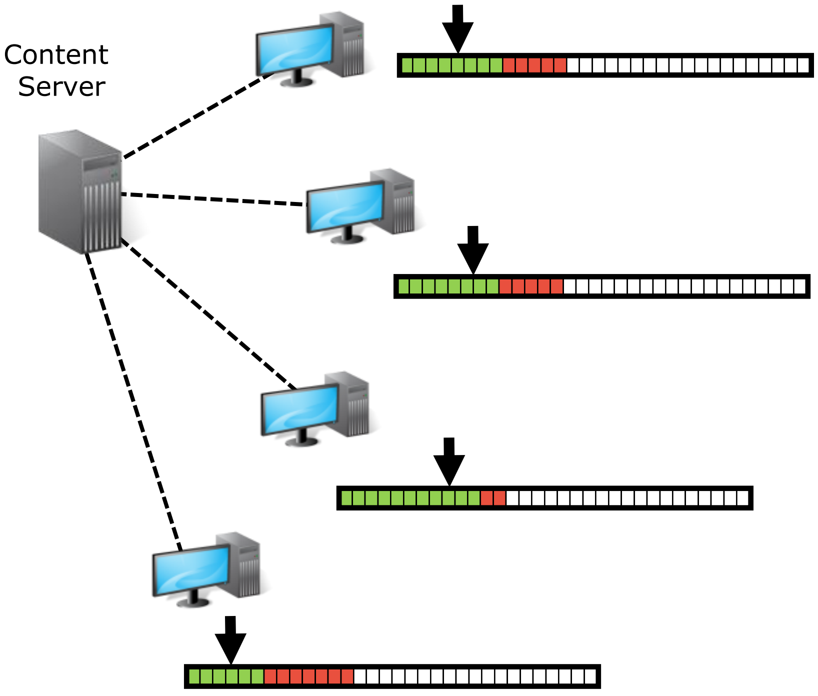


Image Source: http://www.focus.de/fotos/der-australier-garcia-r-kommt-frei-zum-schuss-philipp-lahm-l-und_mid_676372.html

1.2. Live Streaming

- Playback synchronization (being “live”)
  - Hard to achieve in reality
  - Metric: playback lag
    - Time between availability of video chunk and actual playback at client
  - Depends on characteristics of network, client, and video player
    - Transmission delay
    - Jitter
    - Packet losses
    - Client connectivity
    - Playback strategy
  - Clients never completely in sync

- Playback policy on missing data
  - Typically rather skip missing data
  - Pausing would increase playback lag



1.2. Video-on-Demand Streaming

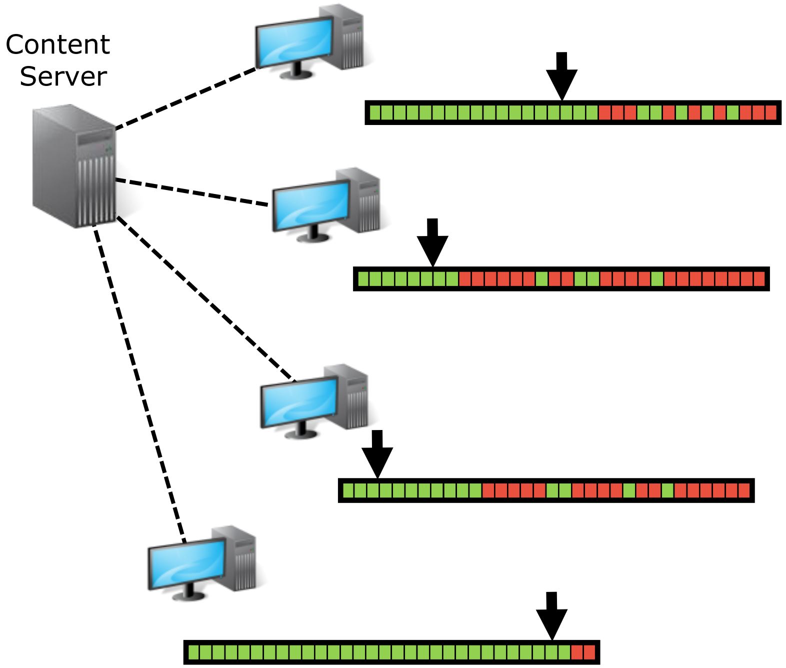
TECHNISCHE UNIVERSITÄT DARMSTADT

- Pre-generated content
  - Provider holds complete video files
  - Preloading of data possible
- Videos are streamed anytime and anywhere
  - Users stream video independent of each other
  - Typically few concurrent users for a single video
- Popular examples
  - YouTube, Vimeo (free, more short videos)
  - Netflix, Maxdome (paid, movie length, country-specific)

1.2. Video-on-Demand Streaming

- No playback synchronization
  - Clients naturally differ in playback
  - Synchronization not required

- Playback policy on missing data
  - Typically rather stall than skip data
  - Being "live" is not important

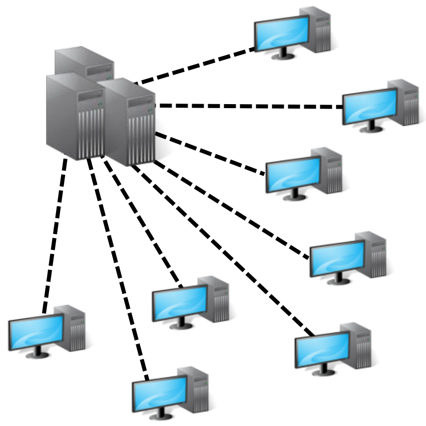


1.3. Content Dissemination

- Streaming source
  - Single entity (e.g. server, server farm, cloud service, single peer)
    - Stores and/or broadcasts content
  - Multiple sources are also possible, e.g. group video conference
    - Not considered here

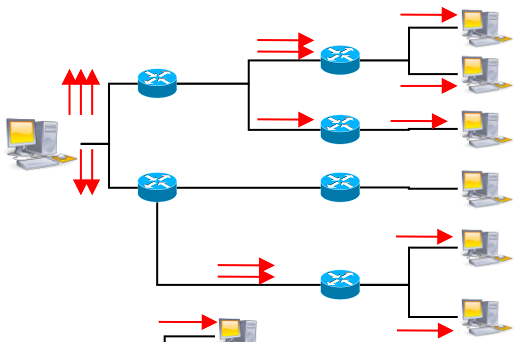
- Multiple clients retrieve content over the network

- Possible ways to deliver content
  - IP unicast
  - IP broadcast
  - IP multicast
  - Overlay multicast

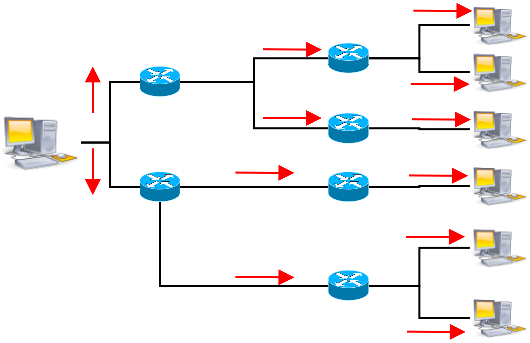


1.3. IP Unicast/Broadcast/Multicast

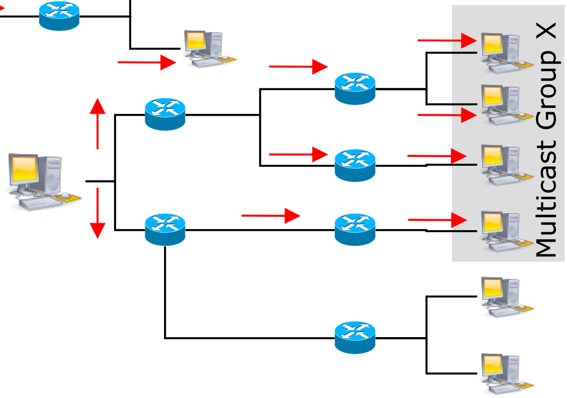
- Unicast
  - Point-to-point communication
  - Data delivered from sender to specific receiver
  - Streaming: replicated unicast necessary to send N copies
  - Problem: High bandwidth demands at sender



- Broadcast
  - Point-to-multipoint transmission
  - Indiscriminate transmission of data
  - Associated with traditional radio / TV transmission
  - Every connected receiver gets the data
  - Only scalable in small networks



- Multicast
  - Multipoint-to-multipoint transmission
  - Data replicated at IP level by routers
  - Cleanest solution from conceptual point
  - Can be an efficient solution for managed networks
    - e.g. IPTV offered by Internet Service Provider
  - Not deployed/usable at Internet-scale
    - Scalability: violates stateless principle of IP routers



1.3. Overlay Multicast

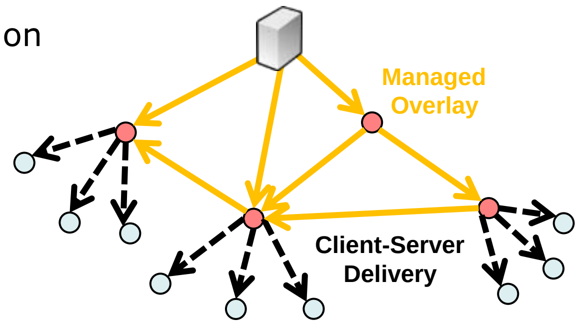
- Thus, is IP unicast the only feasible solution?
  - It is actually used for most streaming services today
  - Problem: It does not scale well with the number of clients
    - Required bandwidth at central entity and transmission:
      - E.g. HD video (1080p) bit rate: 3300 kbps,
        1 Gbps server bandwidth

        → max. number of clients = $\left\lfloor \frac{1Gbps}{3300\ kbps} \right\rfloor$ = 303 clients
    - → Solution is very costly for streaming provider and transport networks
- Alternative: overlay multicast / streaming overlays
  - Idea: Leverage distributed nature of Internet
  - Streaming source distributes video to a few clients or other servers
  - They contribute upload bandwidth to serve others
    - Goal: Offloading of streaming source
    - Load distributed using peer-to-peer mechanisms

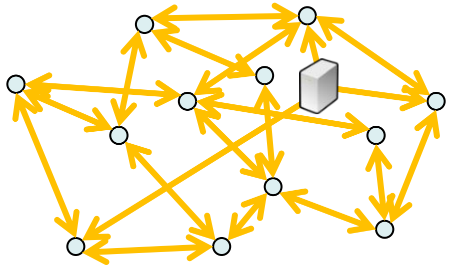
1.3. Overlay Multicast Types

TECHNISCHE UNIVERSITÄT DARMSTADT

- Managed overlays: Content Delivery Networks (CDNs)
  - Overlay between managed server machines (CDN nodes)
  - Clients are served by CDN nodes nearby in a client-server fashion
  - Approach can be classified as a proxy-based overlay
    - CDN nodes act as proxies at the edge of the overlay
    - Overlay delivery transparent to clients
  - Example for CDN network: Network of Akamai Inc. with > 200k CDN nodes deployed world-wide [SKL+14] , [NSS10]



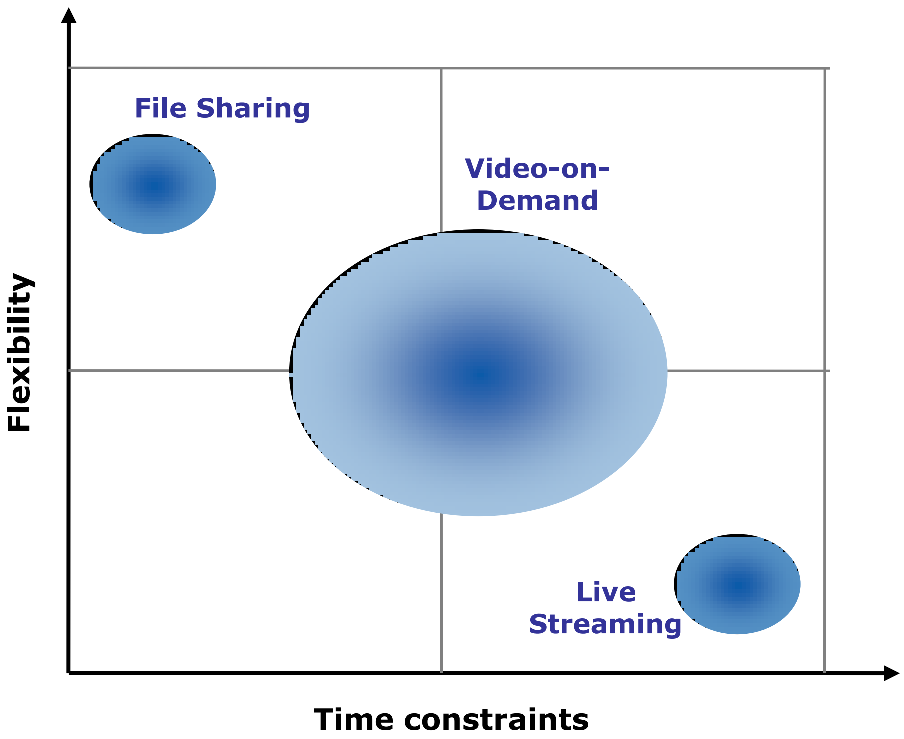
- Unmanaged: Fully P2P-based approaches
  - Clients become peers and contribute upload bandwidth to serve others
  - Clients are active part of overlay
    - Run P2P multicast software
    - Overlay delivery not transparent to clients
  - Example: PPLive, BTLive, Hive Streaming, Tribler, CoolStreaming, …



- Hybrid approaches exist
  - Combination of both approaches possible and rational
  - Example: Akamai NetSession [ZCL+13]

1.4. Summary

- Live streaming
  - Clients have nearly same playback positions
  - Suitable for multicast
- Video-on-Demand (VoD)
  - Watch while download
    - Start playback after short delay
    - Avoid playback stalling
  - Diverse playback positions
    - Multicast not applicable
- File sharing
  - No watch-while-download
    - Low quality requirements (download time)
  - “Random” download order possible



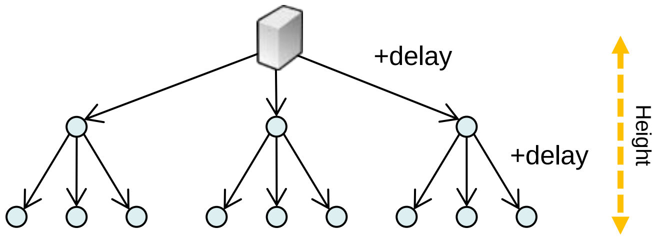
2. P2P Streaming Concepts

Topologies and Coordination,
Examples: PPLive, BTLive, Transit

2.1. Topologies and Coordination

TECHNISCHE UNIVERSITÄT DARMSTADT

- Single Tree
  - Peers form a hierarchical topology
  - Playback lag depends on height of tree
  - Video data is "pushed" down the tree
    - Receiving video chunks from parent
    - Forwarding chunks to all children
  - Coordination of transfer
    - Data paths are stable (assuming no node failures)
    - Here: push-based coordination very efficient

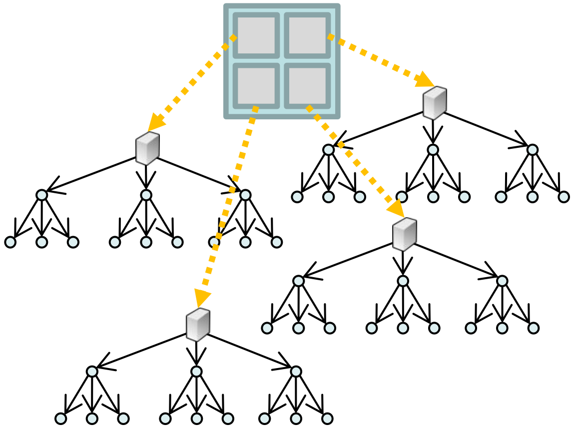


- Pros:
  - Minimal overhead for transmission coordination (push-based)
- Cons:
  - Maintenance of topology induces overhead
    - Tree should be as balanced as possible
    - Peer churn can disconnect complete sub-trees
  - Resource utilization and fairness
    - Only a few peers (inner nodes) can contribute upload bandwidth
    - Peers at the edge (leaf nodes) only consume

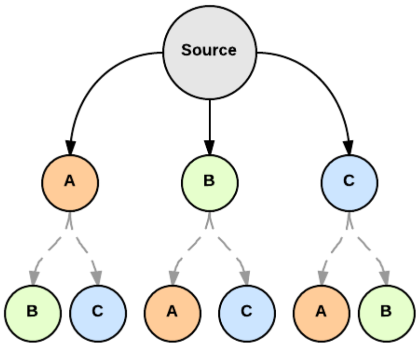
2.1. Topologies and Coordination

TECHNISCHE UNIVERSITÄT DARMSTADT

- Multi Tree
  - Peers form *n* independent tree topologies
    - Each peer is part of all topologies
    - A peer has an inner node position in at least one tree
  - Video stream is divided in *n* sub-streams
    - Each sub-stream is disseminated by one of the trees
    - Client receives all *n* sub-streams and merges them
  - Coordination of transfer: push-based
  - More resilient against peer churn
    - Failure of peer in best case only requires a sub-tree to be repaired (reason: inner-tree position in sub-tree)



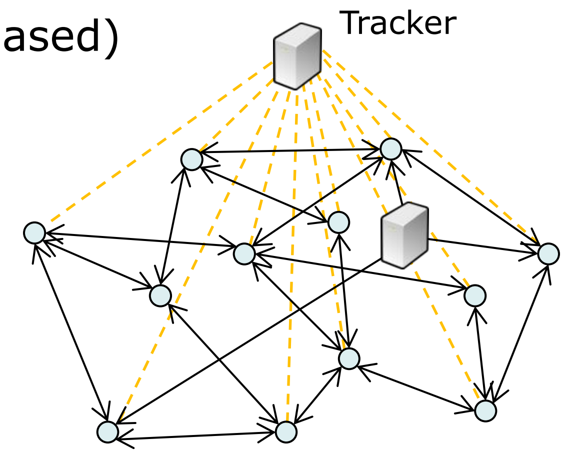
- Pros:
  - Minimal overhead for transmission coordination (push-based)
  - All peers can contribute
    - Contribution can be adapted to peer resources: High capacity peer can have more inner node positions
- Cons:
  - Still: overhead for maintenance of topology
    - Balancing of trees



2.1. Topologies and Coordination

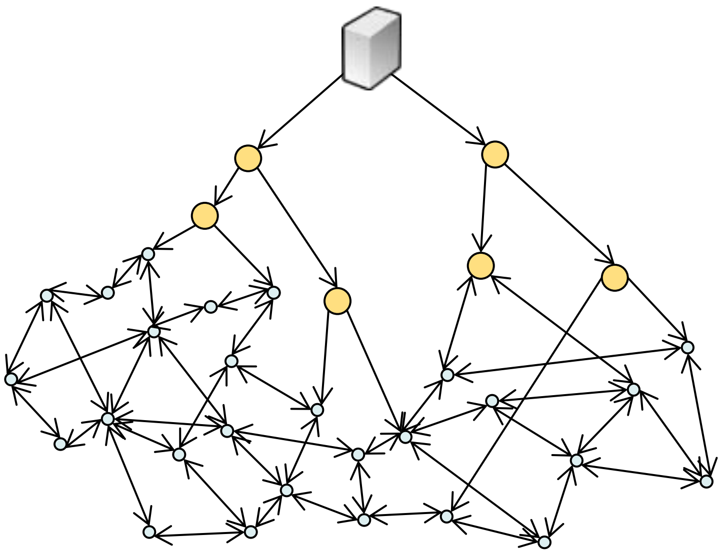
TECHNISCHE UNIVERSITÄT DARMSTADT

- Mesh
  - Peers form a (random) mesh (no parent-child relations)
    - A tracker holds information on available peers
    - Peers can retrieve new neighbors at tracker if needed
  - Video is divided into chunks/blocks
  - Peers exchange block maps, indicating data availability
  - Blocks explicitly requested from neighbors (pull-based)
    - No stable data paths (in comparison to tree)
- Pros:
  - Very robust against node failure
  - Minimal maintenance overhead
- Cons:
  - Coordination overhead due to pull transmission
  - Increased playback lag due to block maps exchange



2.1. Topologies and Coordination

- Hybrid Tree/Mesh Systems
  - Use advantages of both approaches
  - Example:
    - Form highly efficient tree between stable peers (“backbone”)
    - Form mesh between unstable peers and attach to tree
- Pros:
  - Fast dissemination of new chunks/blocks using tree
  - Can achieve good balance between overhead and stability
- Cons:
  - Stable peers need to be identified



2.2. Example 1: PPLive

- Most popular P2P live streaming system
  - 400,000 daily average users in 2006 (mostly from China)
- Live streams or movies according to schedule
- Free to use but proprietary implementation
- Important Properties
  - Mesh-/pull approach with tracker for neighbor discovery and monitoring
  - One overlay per channel
  - Each peer is connected to a large set of neighbors (60-100)
  - Video blocks are advertised to neighbors
  - Missing blocks are requested from other peers
  - Only if block not available from other peers, request from broadcasting source
  - Blocks are discarded shortly after being played
  - Up to 1 minute delay difference between users possible

Source: http://www.download.ba/program-image/pplive.jpg

Sources: [PKV+10, S07, VGL07]

2.2 Example 2: BitTorrent Live

- Novel P2P live streaming system
  - US patent filed by Bram Cohen, the inventor of BitTorrent [Co12]
  - Goal: Use BitTorrent concepts to allow anyone to provide own live streams
- BitTorrent Live: Public Beta
  - Ran by BitTorrent Inc. from March 2013 to February 2014
  - In this time: frequent updates were released and protocol was refined
  - Announcement in February 2014: the first version based on the protocol to be published as mobile application
- Details on system described in [RKH14]

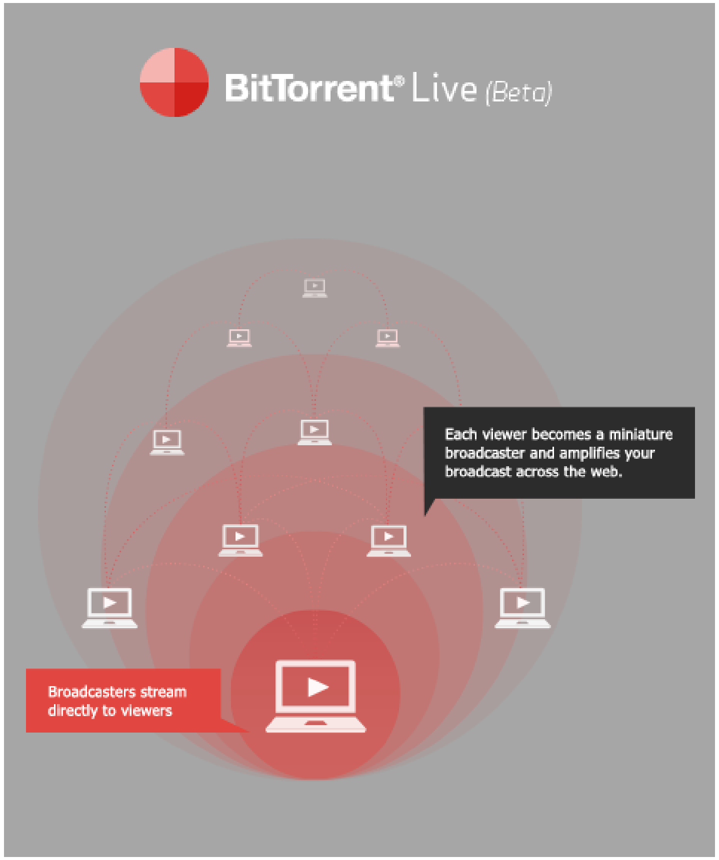
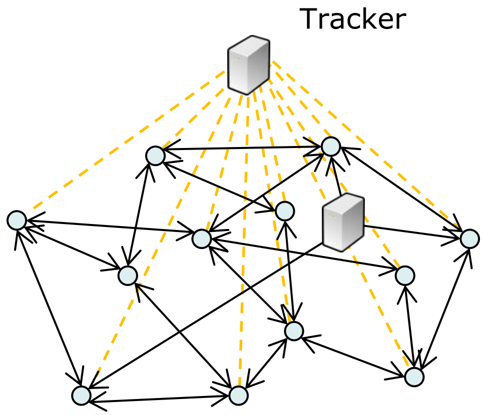


Image Source: http://live.bittorrent.com/

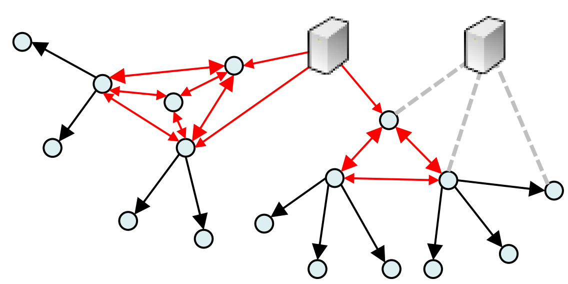
2.2 Example 2: BitTorrent Live

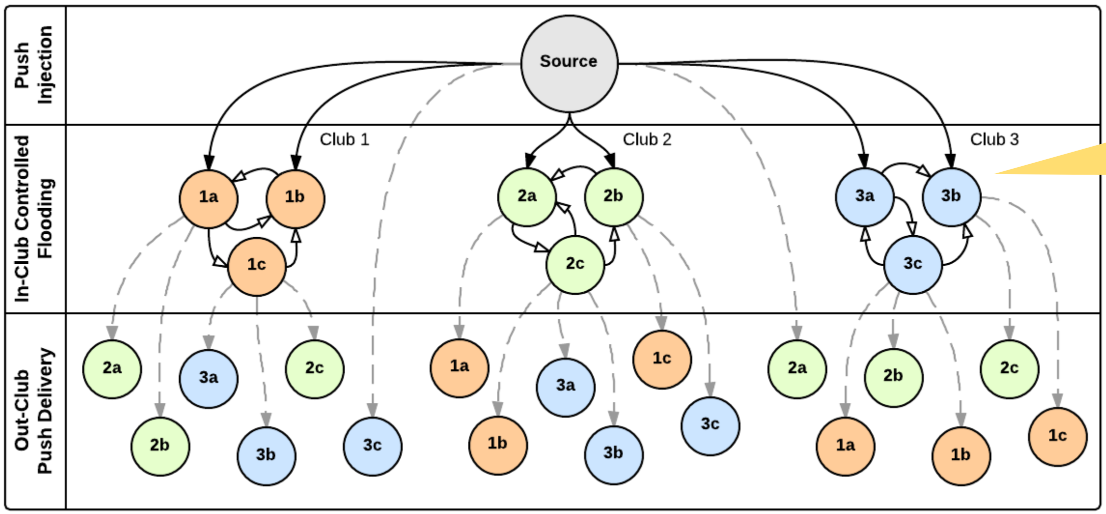
- Use of a tracker for peer discovery
  - Initially contacted by joining peers
  - Provides peers with a list of connected peers
  - Peers exchange peer lists to discover neighbors
- Joining
  - Tracker recommends the peer the clubs to join (to allow for load balancing between clubs)
  - Peer selects club and contact peers from tracker list (Using simple ping/pong messages)
  - Receiving peer answers with a list of other peers in the club and the number of its current up-/download connections
  - Joining peer randomly selects peers to connect to (among the peers with the lowest observed load)



2.2 Example 2: BitTorrent Live

- Peers organize in so called “clubs”
  - Source divides video stream in substreams
  - Each substream is distributed via its own club
  - Peers are member of a fixed number of clubs (e.g. 2), source is member of all clubs
  - Peers help distributing the substream within the club
  - Non-members of a club connect to members as leafs





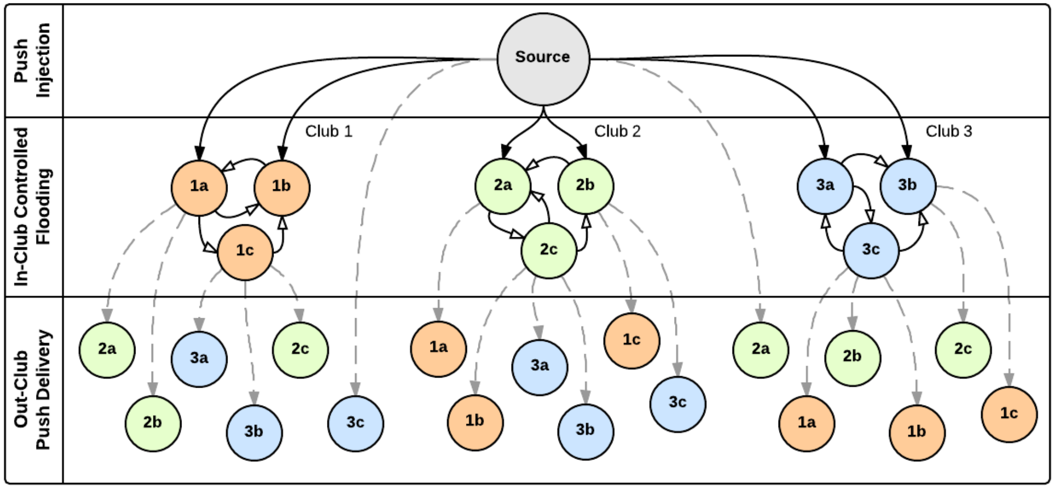
Controlled flooding inside club (mesh): push-based with announcement of chunk arrival to club members

More details on BitTorrent Live:

[RKH14] Rückert, Knierim, Hausheer: IEEE P2P, 2014.

2.2 Example 2: BitTorrent Live

- Content distribution
  - Source injects video blocks into clubs
  - Peer receives block for own club
    - Notify download neighbors inside club about block arrival
    - Flood/push new block to all upload neighbors inside club that did not send notification about block arrival
    - Push new block to upload neighbors outside the own club



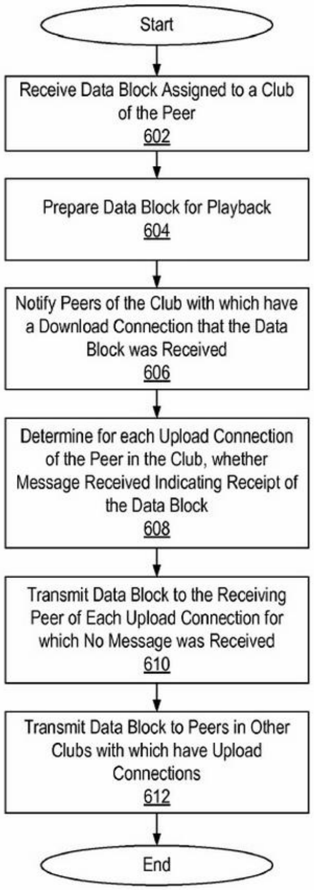
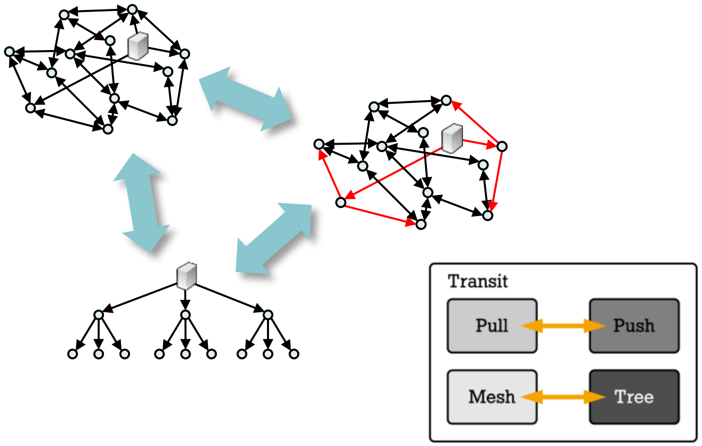


Image Source: [Co13]

2.2. Example 3: TRANSIT

- TRANSIT [WRR+14]:
  - Hybrid mesh/pull, multi-tree/push P2P live streaming system
  - Used to study overlay mechanism tradeoffs in quickly changing environments
  - Enable transitions between mechanisms

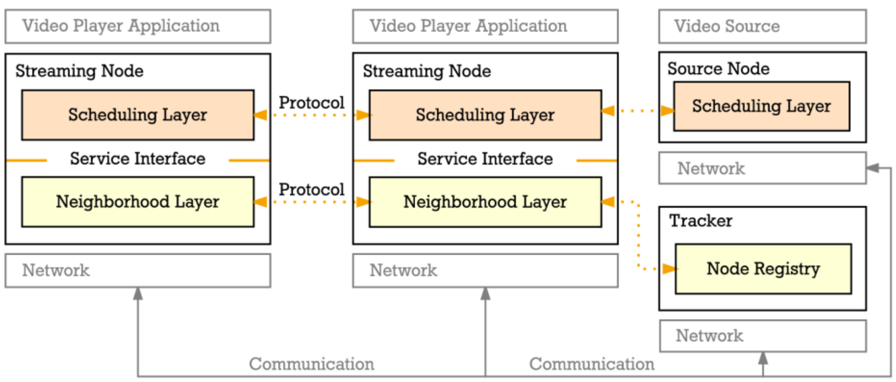


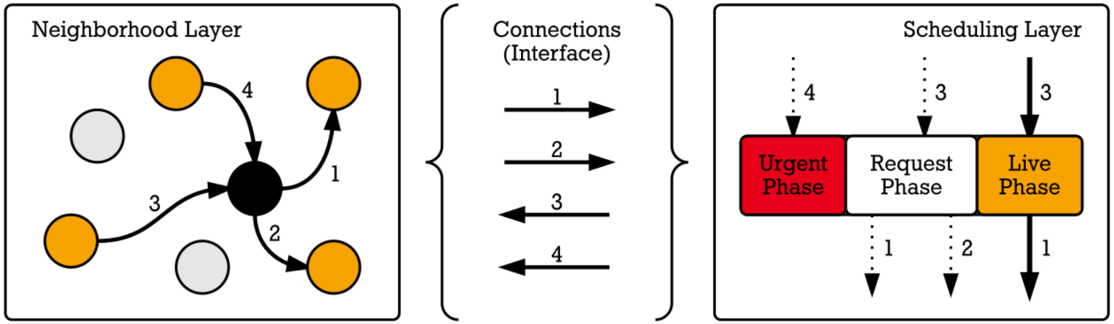
- System features
  - Supports single-/multi-layer videos
  - Two-layer mechanism separation
    - Neighborhood management
    - Scheduling of stream delivery

[WRR+14] Wichtlhuber, Richerzhagen, Rückert, Hausheer: In: IFIP Networking, 2014.

2.2. Example 3: TRANSIT

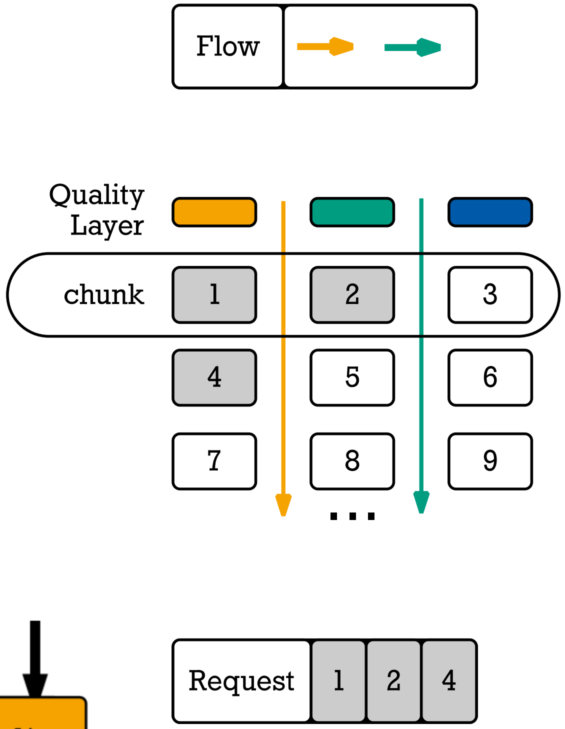
- Key properties of TRANSIT
  - Separation into scheduling layer and neighborhood layer
  - Multiple mechanisms on each of the layers
  - Well-defined protocol between mechanisms of same layer
  - Neighborhood connections offered using service interface to scheduling layer

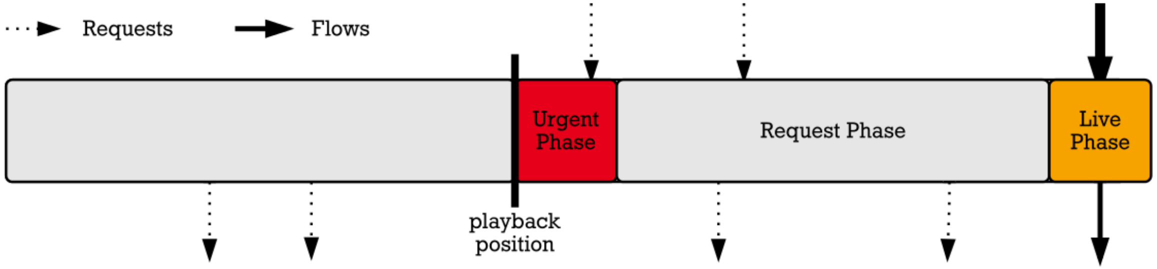




2.2. Example 3: TRANSIT

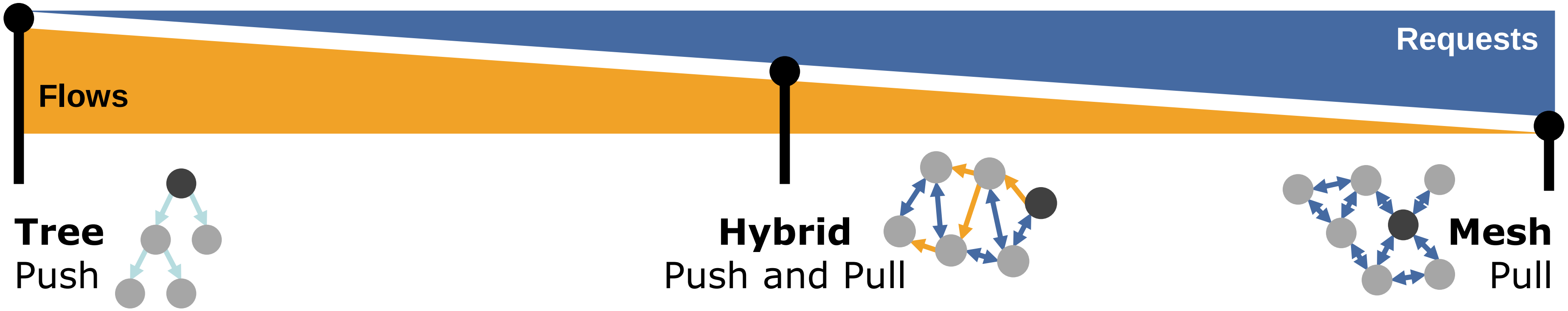
- Flows: multi-tree/push
  - Goal: Fast diffusion of new/recent blocks
  - Negotiated with unlimited lifetime
  - No retransmissions of lost blocks
  - Trees originating from the source
- Requests: mesh/pull
  - Goal: Use to retrieve missing blocks and before flows established
  - Address (multiple) individual blocks
  - Requires exchange of buffermaps





2.2. Example 3: TRANSIT

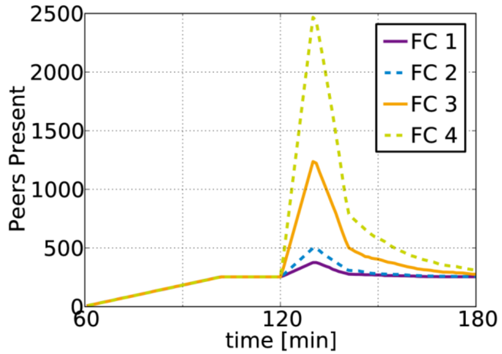
- Flows and requests allow to use the best of the two worlds
  - Any combination of push- and pull-based scheduling mechanism
  - Scheduling directly determines the active overlay topology
  - Neighborhood layer provides basic connections to scheduling
  - Ratio between flows and requests based on current conditions
    - Preference towards multi-tree/push delivery in stable conditions
    - Mesh/pull assures delivery of missing blocks and initial delivery

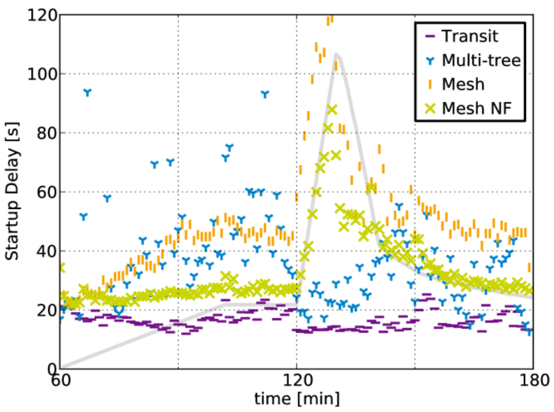


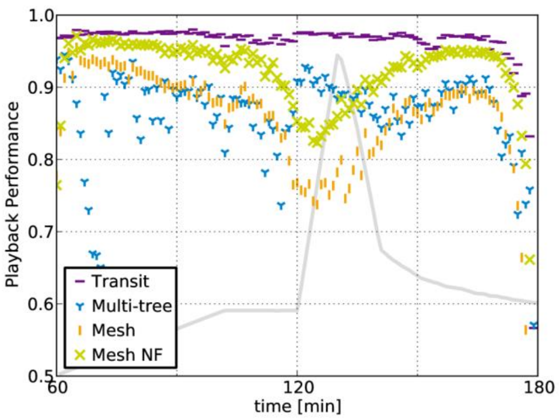
2.2. Example 3: Transit

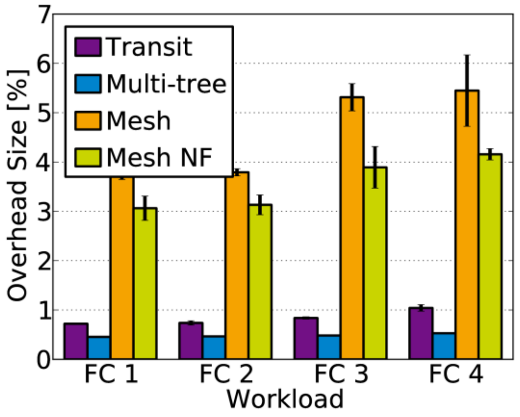
- Study of system characteristics
  - Here: focus on very dynamic scenario (i.e. flash crowd event)
  - High and stable playback performance
    - Startup delay for joining peers
    - Number and length of stalling events
  - Overhead near level of pure tree-based approach
    - Bandwidth and message overhead

Flash crowd scenario





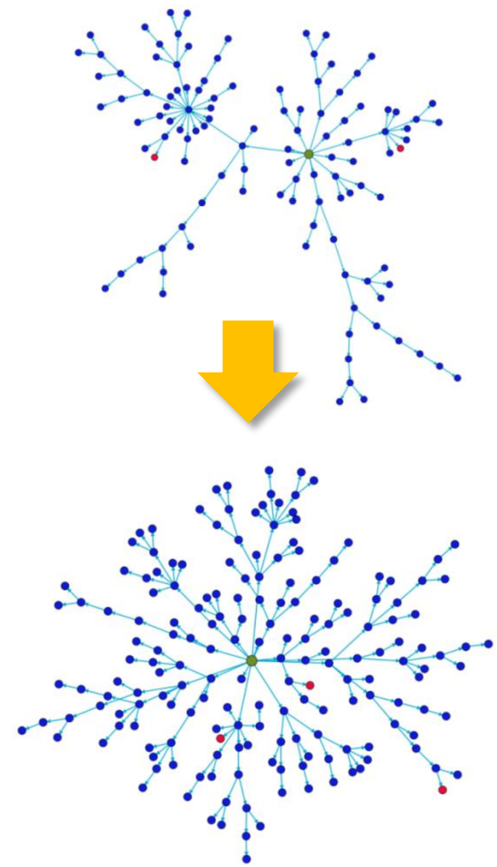


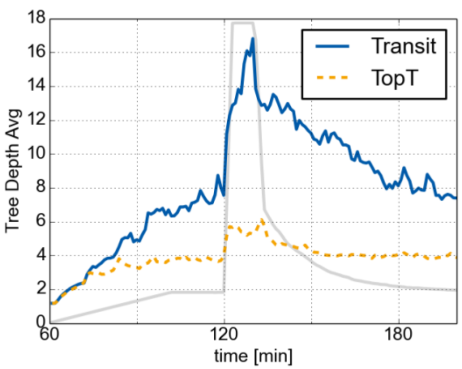


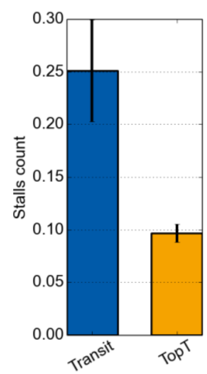
[WRR+14] Wichtlhuber, Richerzhagen, Rückert, Hausheer: In: IFIP Networking, 2014.

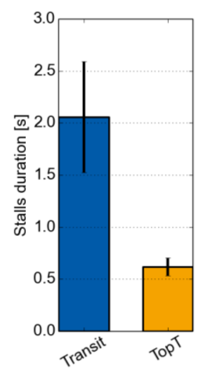
2.2. Example 3: TRANSIT

- Importance of the streaming topology
  - Typically, *flows* rather stable and account for >80% of data delivery
  - Optimizing the multi-tree structure of flows allows to further increase system performance
  - How should it be optimized?
    - Reduce flow tree heights
    - Balancing trees
    - Result: more balanced responsibility and avoidance of high dependency on individual peers









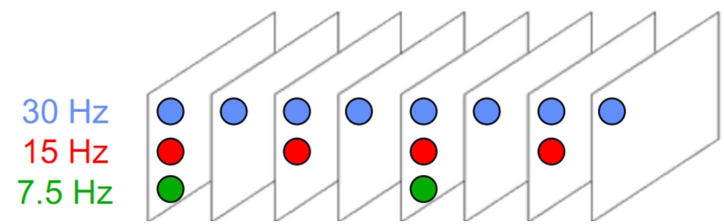
[RRL+15] Rückert, Richerzhagen, Lidanski, Steinmetz, Hausheer: In: IFIP Networking, 2015.

3. Advanced Video Streaming Concepts

Adaptive Streaming using H.264 SVC, Case Study: P2PStream
Quality of Service vs. Quality of Experience

3.1. Adaptive P2P Streaming Scalable Video Coding (SVC)

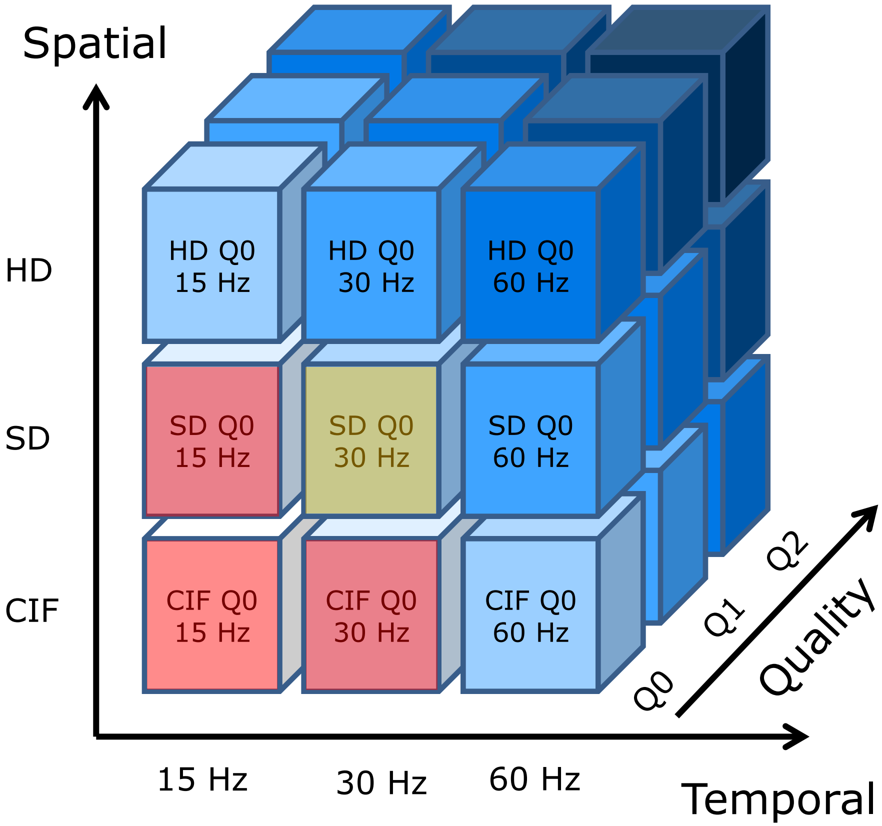
- Extension of the video standard H.264/AVC
  - Referred to as H.264/SVC
  - Extension to H.265 also exists: SHVC
- Multi-layer video codec
  - Video quality scalable by retrieving a sub-stream of the video
  - Allows controlled quality adaptation
- Scalable in three dimensions
  - Video resolution (spatial)
  - Frame rate (temporal)
  - Image quality (quality)



Picture source: http://www.hhi.fraunhofer.de/en/departments/image-processing/image-communication/video-coding/svc-scalable-extension-of-h264avc/

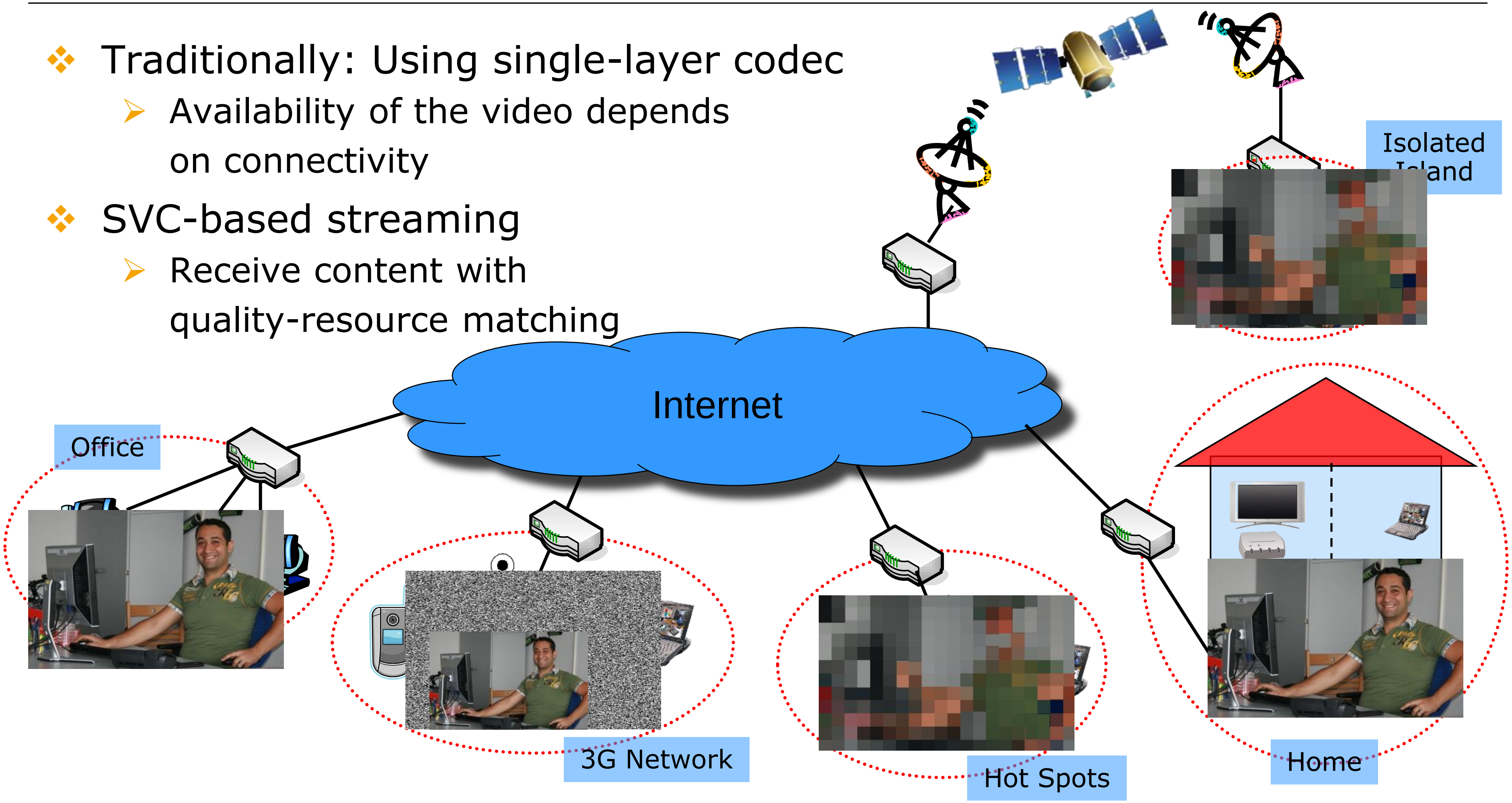
3.1. Adaptive P2P Streaming
The SVC Cube Model

- Layer described by triple (d,t,q)
  - (0,0,0) – Base layer
  - (2,2,2) – In this case full quality
  - Layers depend on all lower layers
  - (1,1,0) for example depends on
    - (0,0,0), (0,1,0), (1,0,0)



3.1. Adaptive P2P Streaming

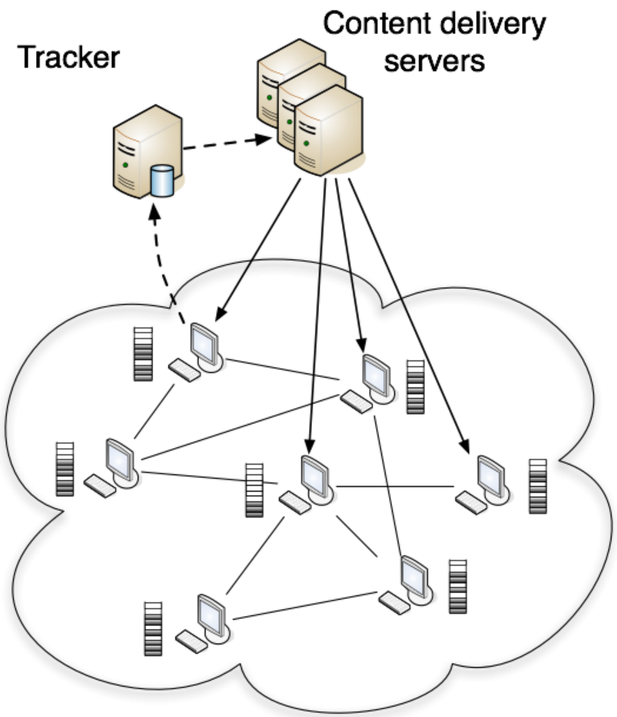
- Traditionally: Using single-layer codec
  - Availability of the video depends on connectivity
- SVC-based streaming
  - Receive content with quality-resource matching



3.2. Adaptive P2P Streaming
Case Study: P2PStream

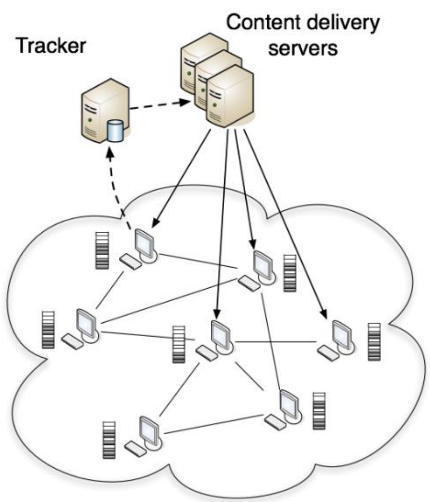
- Video-on-Demand (VoD) streaming system
  - Clients not synchronized
  - Potential number of clients much smaller than for live systems

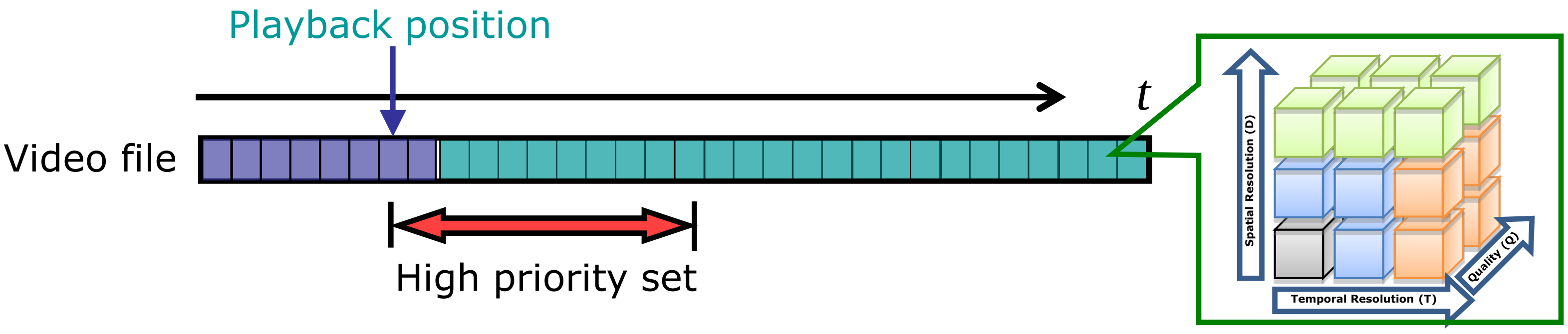
- P2PStream: System features
  - Mesh-/pull-based
  - Entities:
    - Tracker
    - Content delivery servers
    - Clients / Peers
  - Heterogeneous clients
  - Quality adaptation using SVC



3.2. P2PStream: The P2P VoD System Design

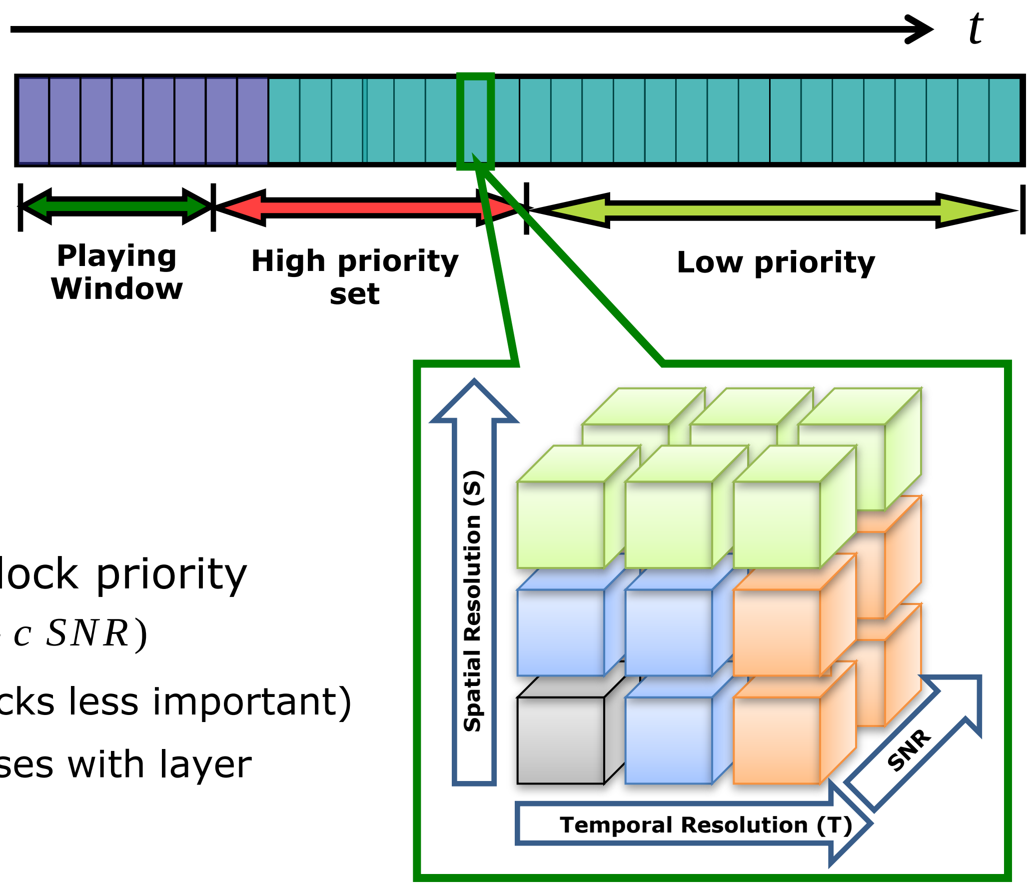
- Multi-source download
  - Mesh-based pull approach
  - Tracker with contact information of the peers
- Peer-assisted architecture
  - Peers exchange video data
  - Content delivery servers provide minimum QoS
- Video streaming
  - Video divided into pieces (time domain)
  - Pieces divided into blocks (SVC cube model)





3.2. P2PStream: Scheduling Strategy

- Stream divided into three regions:
  - Playing window
  - High priority set
  - Low priority set

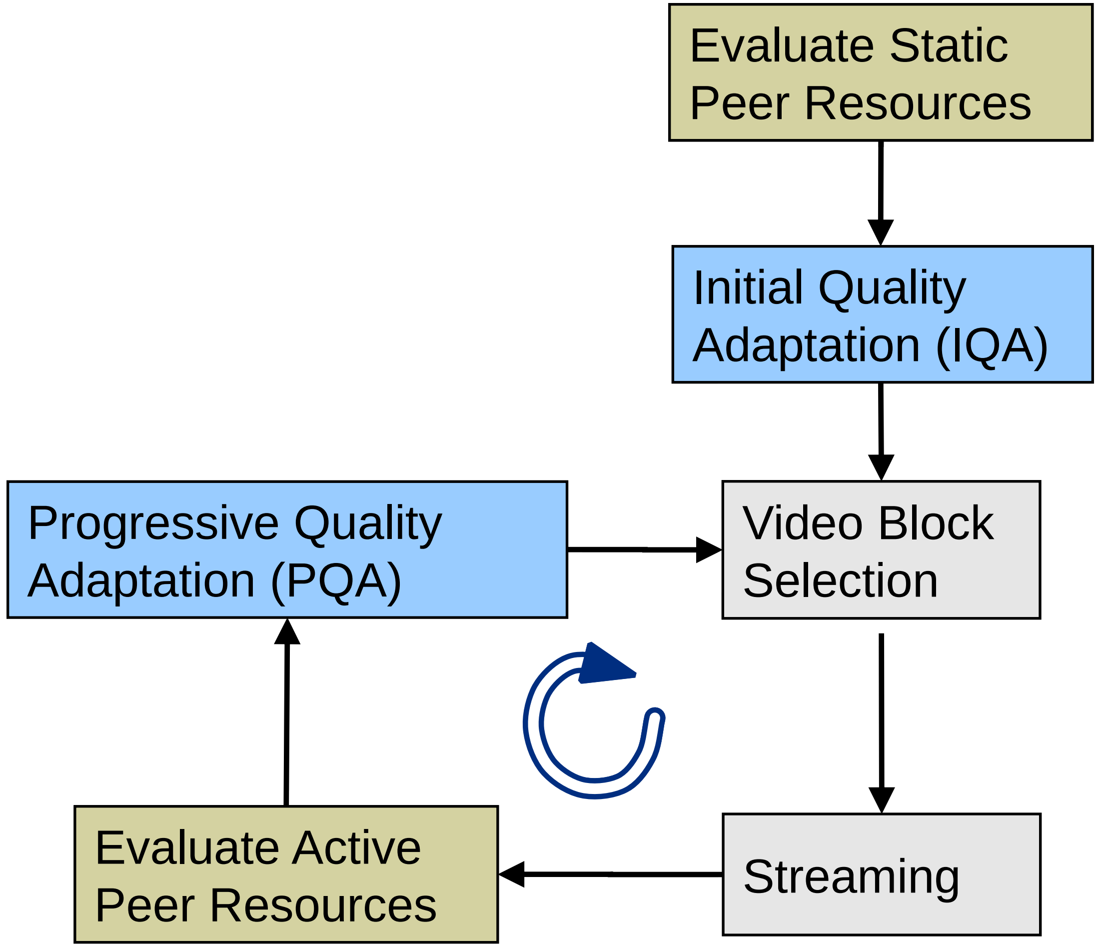


- How to choose blocks inside buffering window?
  - Mathematical modeling of block priority

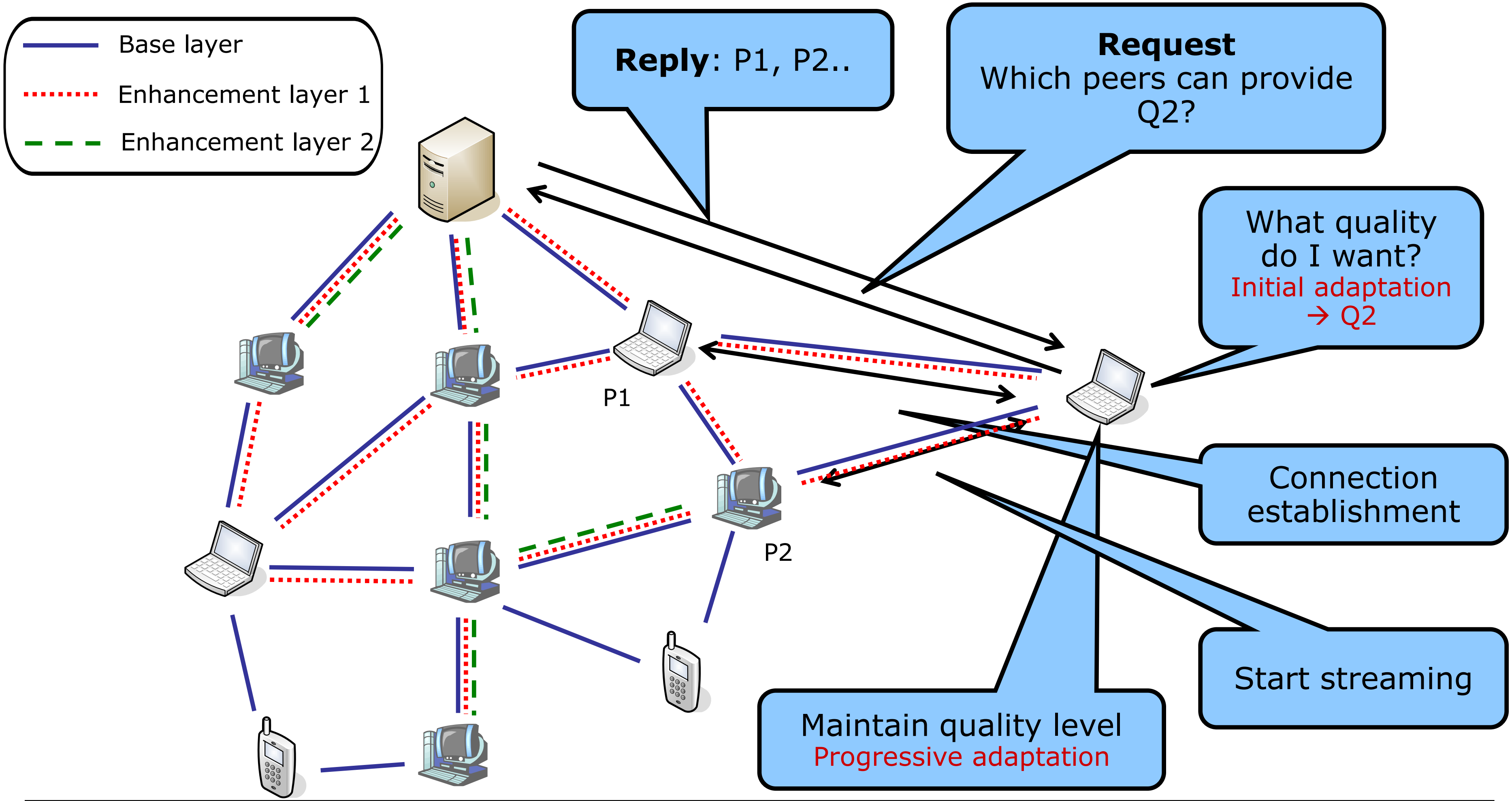
$$\text{Priority} = -A\,t - B(a\,S + b\,T + c\,SNR)$$

    - Decreases with t (later blocks less important)
    - High for base layer, decreases with layer

3.2. P2PStream: Adaptation Steps

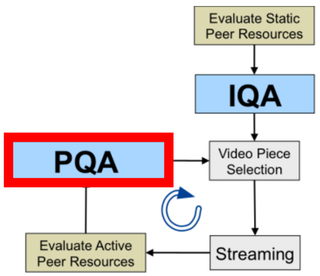


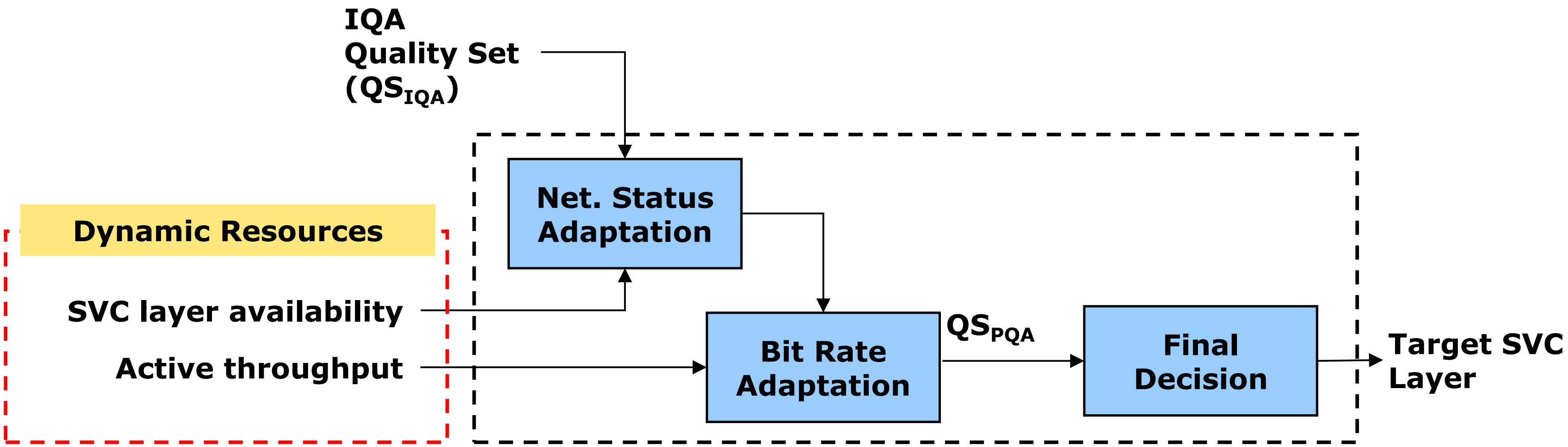
3.2. P2PStream: How does it work?



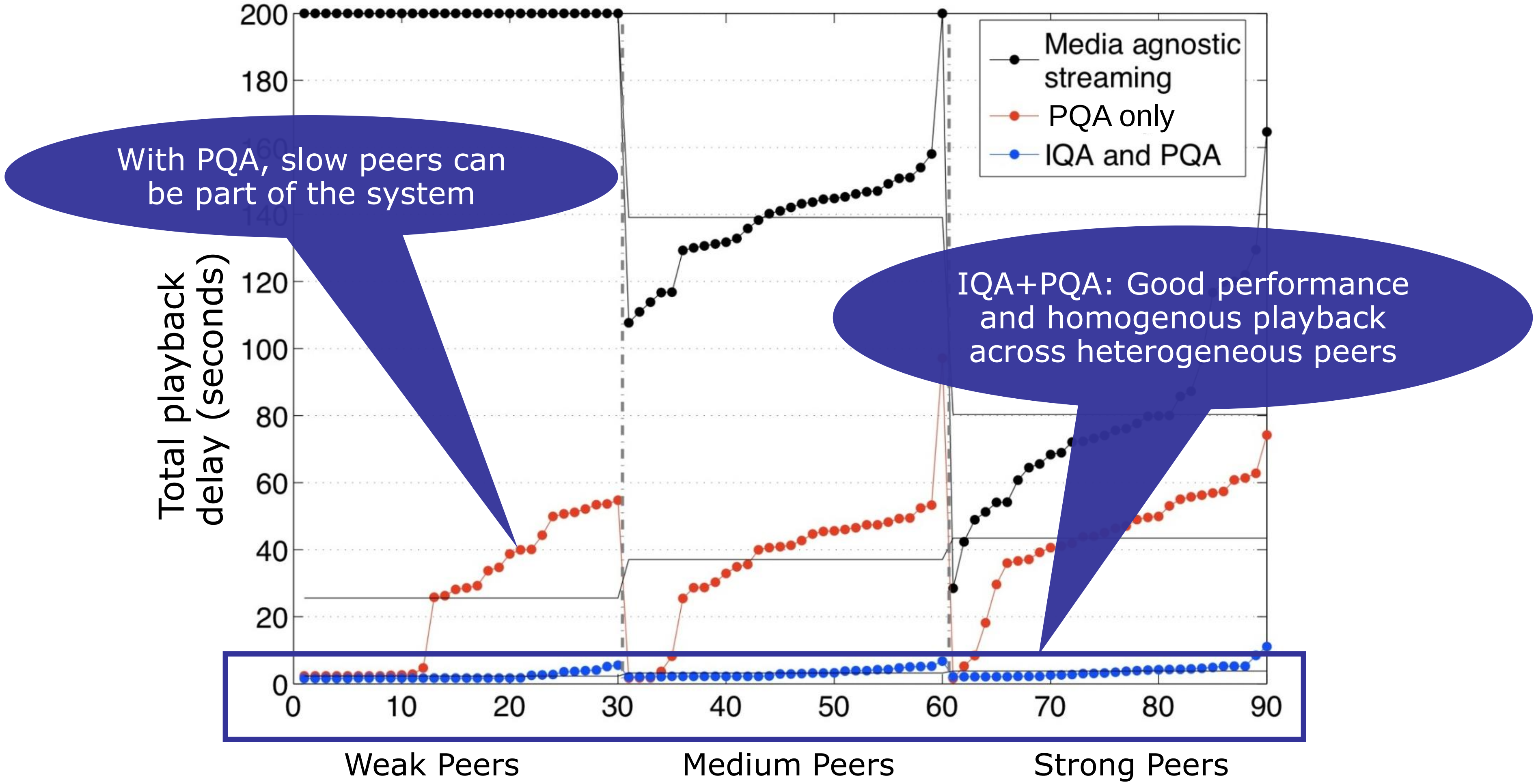
3.2. P2PStream: Progressive Quality Adaptation (PQA)

- Goal
  - Adapt to dynamic network resources
  - Anticipate possible stalls and avoid them

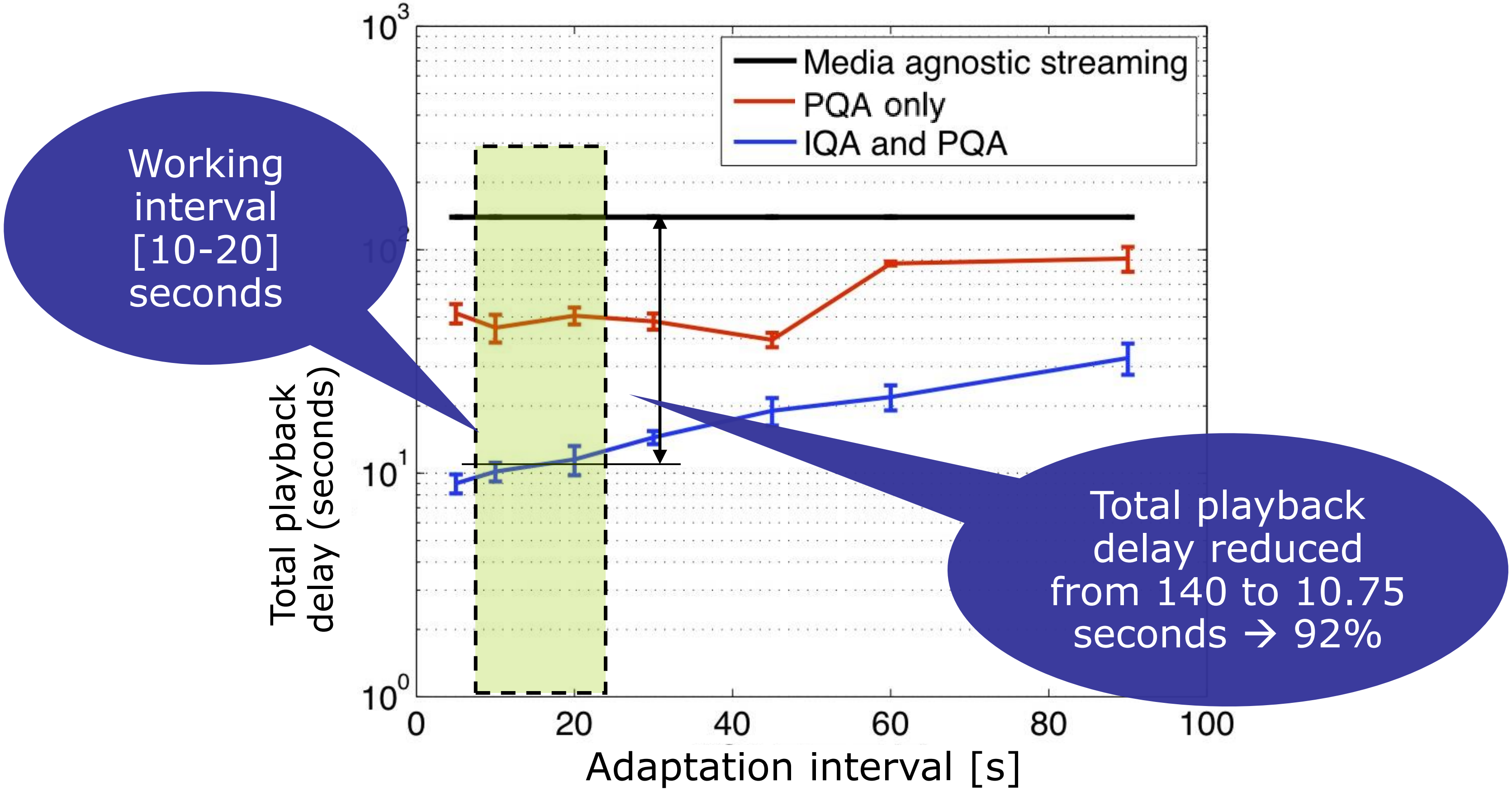




3.2. Impact of Adaptation on Playback Delay

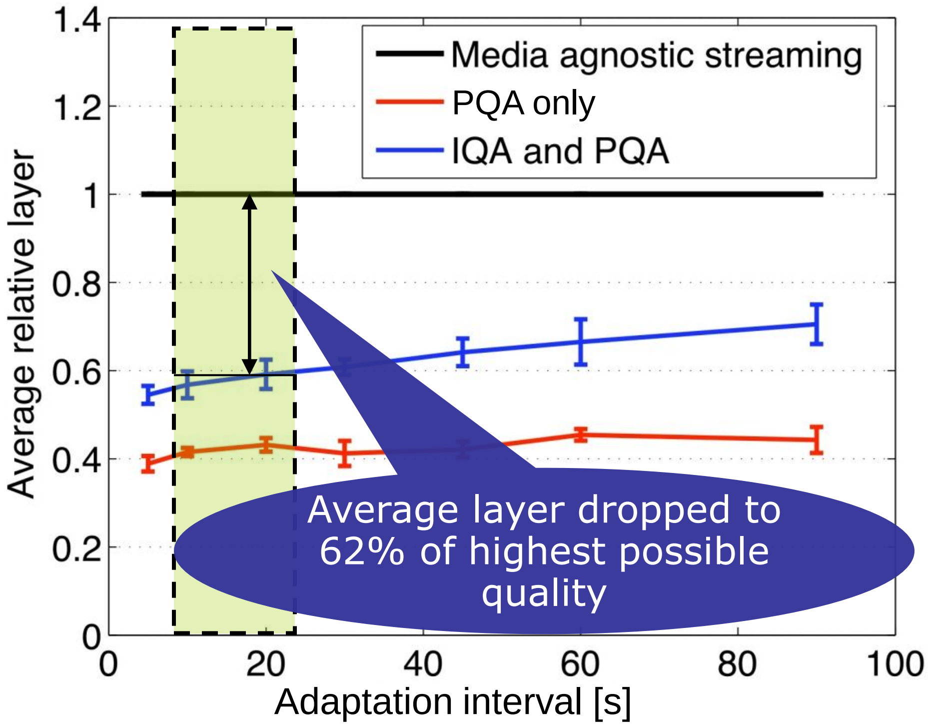


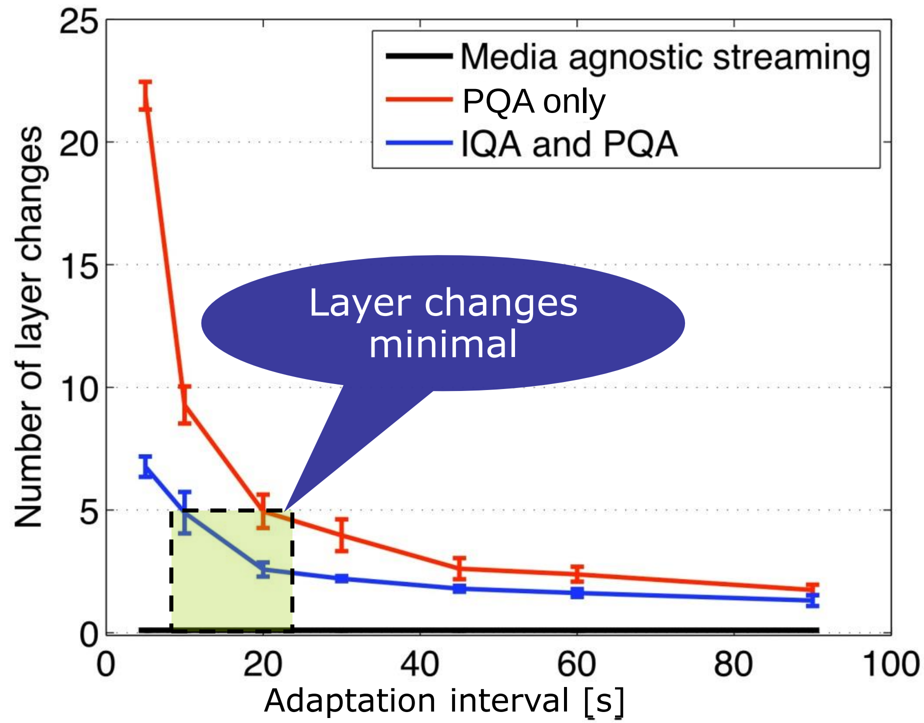
3.2. Impact of Adaptation Interval on Playback Delay



Increasing adaptation interval → larger playback delay

3.2. Impact of Adaptation Interval on Video Quality



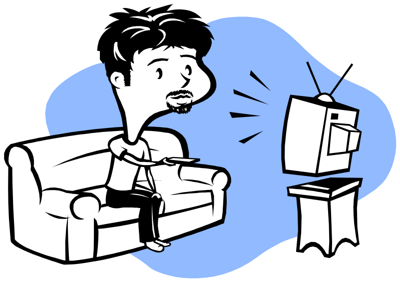


Adaptation maintains good SVC quality with low layer oscillations

3.3. QoS vs. QoE

- Quality of Service (QoS)
  - Metrics: e.g. throughput, bandwidth, delay, packet loss
  - Describe technical aspects of service
  - Problem: No direct correlation to user experience
    - E.g. same packet loss rate can result in different video qualities

- Quality of Experience (QoE)
  - Central question: What is the user experience during video streaming?
  - Aspects: video quality, stalling, change of quality
  - But also: environment, light conditions, distance to screen



3.3. Quality of Experience (QoE)

- Objective: Quantification of user experience

- Subjective QoE
  - Based on user studies
  - Only statistical results
  - Standardized testing procedures (e.g. by ITU)
    - Users rate quality on 5-point scale (excellent to bad)
    - Result: Mean Opinion Score (MOS)

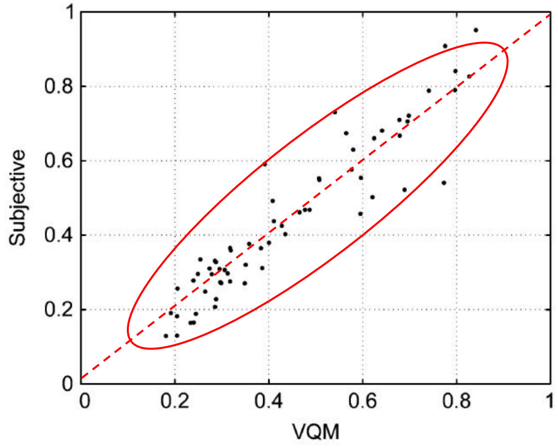
Source: http://mmspg.epfl.ch/3diqa
[Accessed Feb. 2., 2012]

- Objective QoE
  - Modeling characteristics of the human visual system
  - Estimation of quality perceived by end-user
  - Subjective tests used as benchmark
  - Example: VQM [PW04]

3.3. Video Quality Metric (VQM)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Properties
  - State-of-the-Art objective QoE metric
  - Usable with a wide range of video codecs, qualities, and bit rates
  - Quality score: 0 - 1
    - 0 - no perceived impairment
    - 1 - maximum perceived impairment
- Independent Evaluation
  - By the Video Quality Expert Group (VQEG)
  - Strong correlation with subjective test results
- Standardization
  - Part of ITU recommendation (ITU-T J.144)
  - Standardized by ANSI (ANSI T1.801.03-2003)



Extract of the results from evaluation by the VQEG (NTSC - 480i - 640x480) [PW04]

4. References

- [AZPS11] O. Abboud, T. Zinner, K. Pussep, R. Steinmetz: On the Impact of Quality Adaptation in SVC-based P2P Video-on-Demand Systems. In *ACM Multimedia Systems,* 2011.
- [Co12] B. Cohen: Peer-to-Peer Live Streaming. US Patent Application, 2013. Available online: http://de.scribd.com/doc/132418122/bittorrent-live-patent
- [NSS10] E. Nygren, R. K. Sitaraman, J Sun: The Akamai Network: A Platform for High-performance Internet Applications. In: ACM SIGOPS Operating Systems Review 44(3) (2010).
- [PKV+10] Piatek et al.: Contracts: Practical contribution incentives for P2P live streaming. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation*, 2010.
- [PW04] M. H. Pinson, S. Wolf: A new Standardized Method for Objectively Measuring Video Quality. In IEEE Transactions on Broadcasting, 50(3):312 – 322, 2004.
- [RKH14] J. Rückert, T. Knierim, D. Hausheer: Clubbing with the Peers: A Measurement Study of BitTorrent Live. In: IEEE International Conference on Peer-to-Peer Computing (P2P), 2014.
- [RRl+15] J, Rückert, B. Richerzhagen, E. Lidanski, R. Steinmetz, D. Hausheer: TopT: Supporting Flash Crowd Events in Hybrid Overlay-based Live Streaming. In: IFIP Networking, 2015.
- [S07] Sentinelli et al.: Will IPTV ride the peer-to-peer stream? In *IEEE Communications Magazine*, *45*(6), 86-92, 2007.
- [SKL+14] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, M. Jain: Overlay Networks: An Akamai Perspective. In: Advanced Content Delivery, Streaming, and Cloud Services. John Wiley & Sons, 2014
- [VGL07] Vu, et al.: *Mapping the PPLive Network: Studying the Impacts of Media Streaming on P2P Overlays*, 2007.
- [WRR+14] M. Wichtlhuber, B. Richerzhagen, J. Rückert, D. Hausheer: TRANSIT: Supporting Transitions in Peer-to-Peer Live Video Streaming. IFIP Networking, 2014.
- [ZCL+08] G. Zhai, J. Cai, W. Lin, X. Yang, W. Zhang: Three Dimensional Scalable Video Adaptation via User-End Perceptual Quality Assessment. In *IEEE Transactions on Broadcasting*, 54(3):719 –727, 2008.
- [ZCL+13] M. Zhao, A. Chen, Y. Lin, A. Haeberlen, et al.: Peer-Assisted Content Distribution in Akamai NetSession. In ACM IMC 2013.