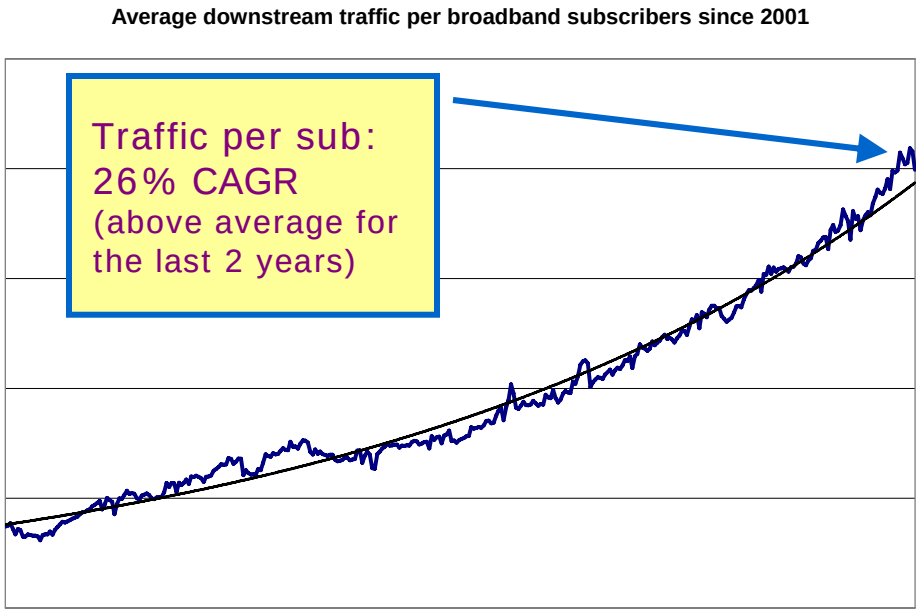
CPM: Adaptive VoD with *C*ooperative *P*eer Assist and *M*ulticast

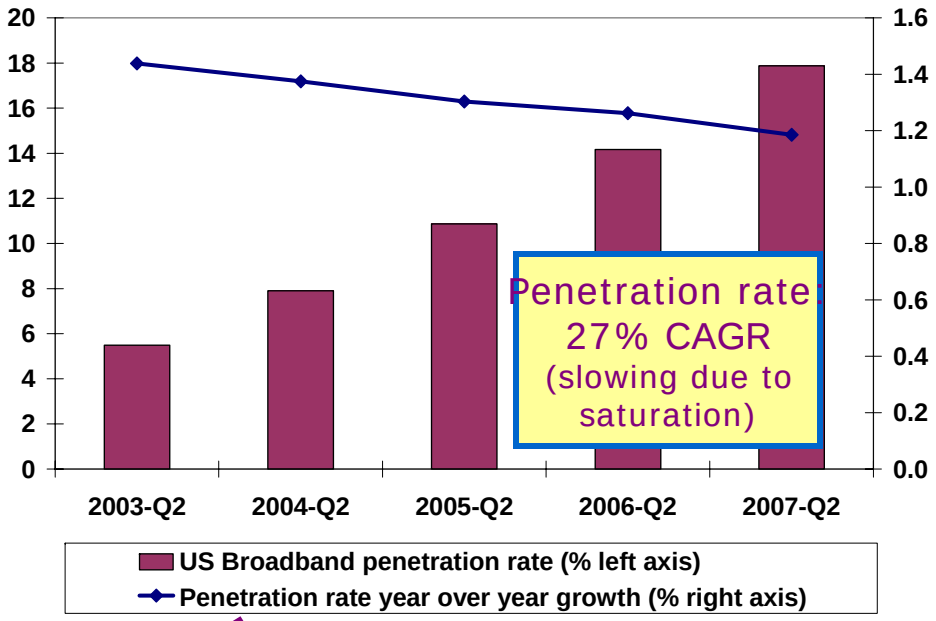
Vijay Gopalakrishnan, Bobby Bhattacharjee, K. K. Ramakrishnan, Rittwik Jana and Divesh Srivastava

AT&T Labs Research, NJ USA

September 4, 2008

Traffic Growth on the Internet
Penetration Rate & Traffic per subscriber





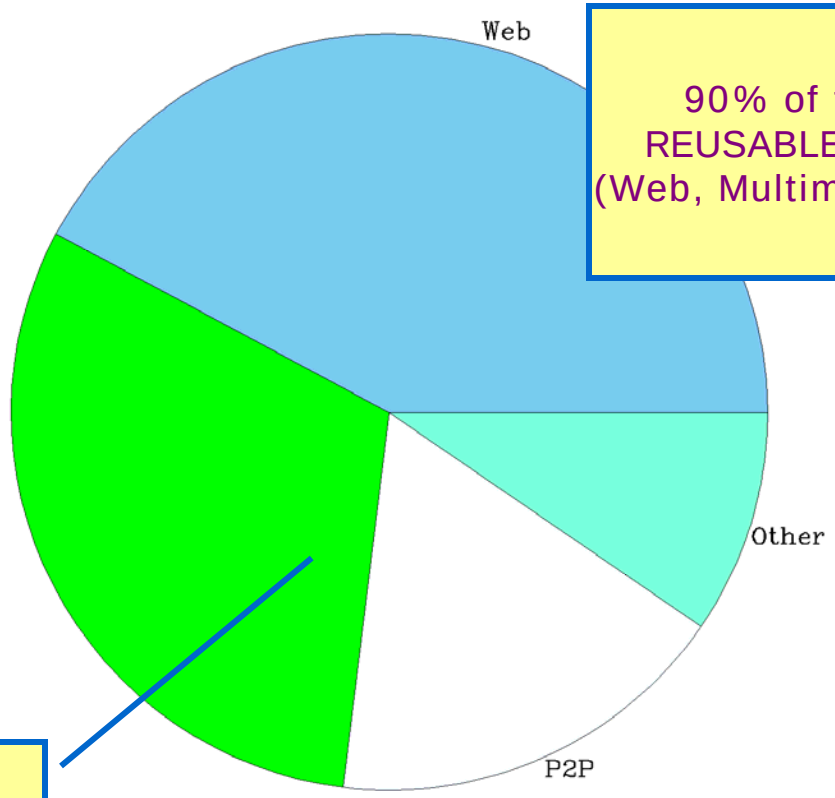
Overall compound average growth rate (CAGR) for last several years: 60% / year

Current Application Mix on the Internet (U.S.)
Busy hour is 1/3 Explicit Multimedia

Web: HTTP traffic with a non-video mime type.

Explicit Multimedia: Video/Audio HTTP traffic and streaming video protocols such as RTSP, RTMP.

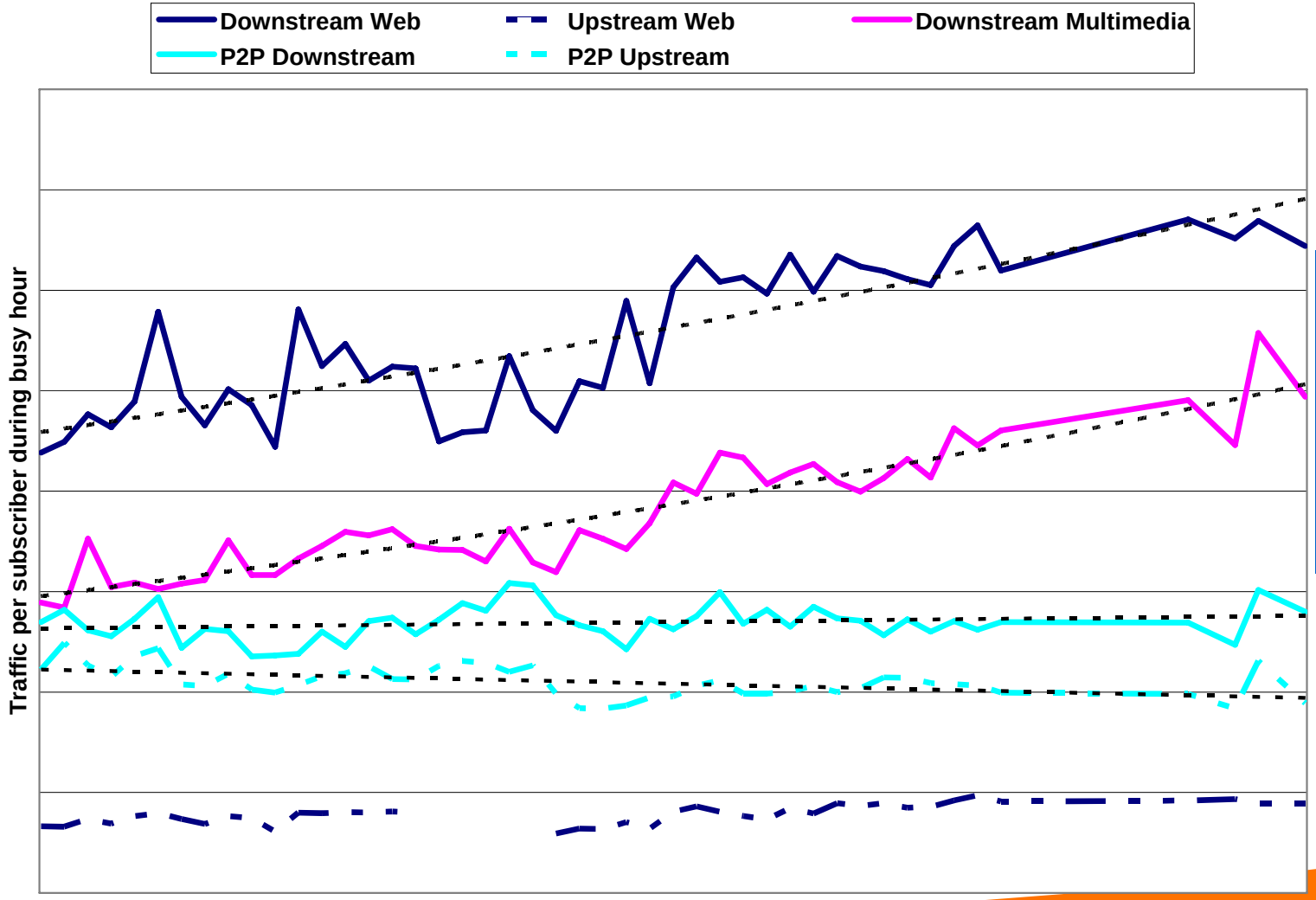
P2P: File Sharing applications such as BitTorrent, Gnutella, etc. (at least 50% video: next slide)



90% of total is REUSABLE content (Web, Multimedia, P2P)!

Explicit Multimedia: already 31%

Content Growth over the last Year
Multimedia/sub: +76% CAGR



Strong growth of Multimedia and Web traffic per subscriber. Multimedia: +76% CAGR

Dealing with Growth of Content

- Content growth is driving network evolution
- Viewing Content that has traditionally been on analog/broadcast systems (over-the-air, cable) is also shifting to platforms with IP connectivity
  - 57% of all TV viewing in the US is “time shifted” (TiVO data)
    - DVR, Internet
  - With IP based delivery, this will become the norm
    - Users will demand more pull-based interactive viewing experience
- Service providers and Content Providers are investing to meet this challenge
- Service providers will have to evolve their delivery mechanisms
- Content delivery solutions need to take advantage of information across layer boundaries

Scalable and Efficient Content Dissemination

- P2P applications have been very popular for file/content sharing
  - P2P client downloads content from other clients willing to upload
  - Peers can be located in lots of different ways: a web site ("tracker"), DHTs, social networking site, etc.
- For a service provider, we examine whether to
  - Use P2P mechanisms and applications as they are currently designed, or
  - Couple P2P mechanisms with techniques for scalable content dissemination that a service provider may have available
    - Servers in the network
    - Caching in the network
    - Multicast
    - Cooperative Peers
- Use one or another mechanism for particular type of content or situation? OR, can we find a unified way of using these mechanisms in a cooperative way for content dissemination that:
  - Scales: large numbers of users; large content library
  - Meet a range of user viewing requirements
  - Robust to varying degrees of popularity

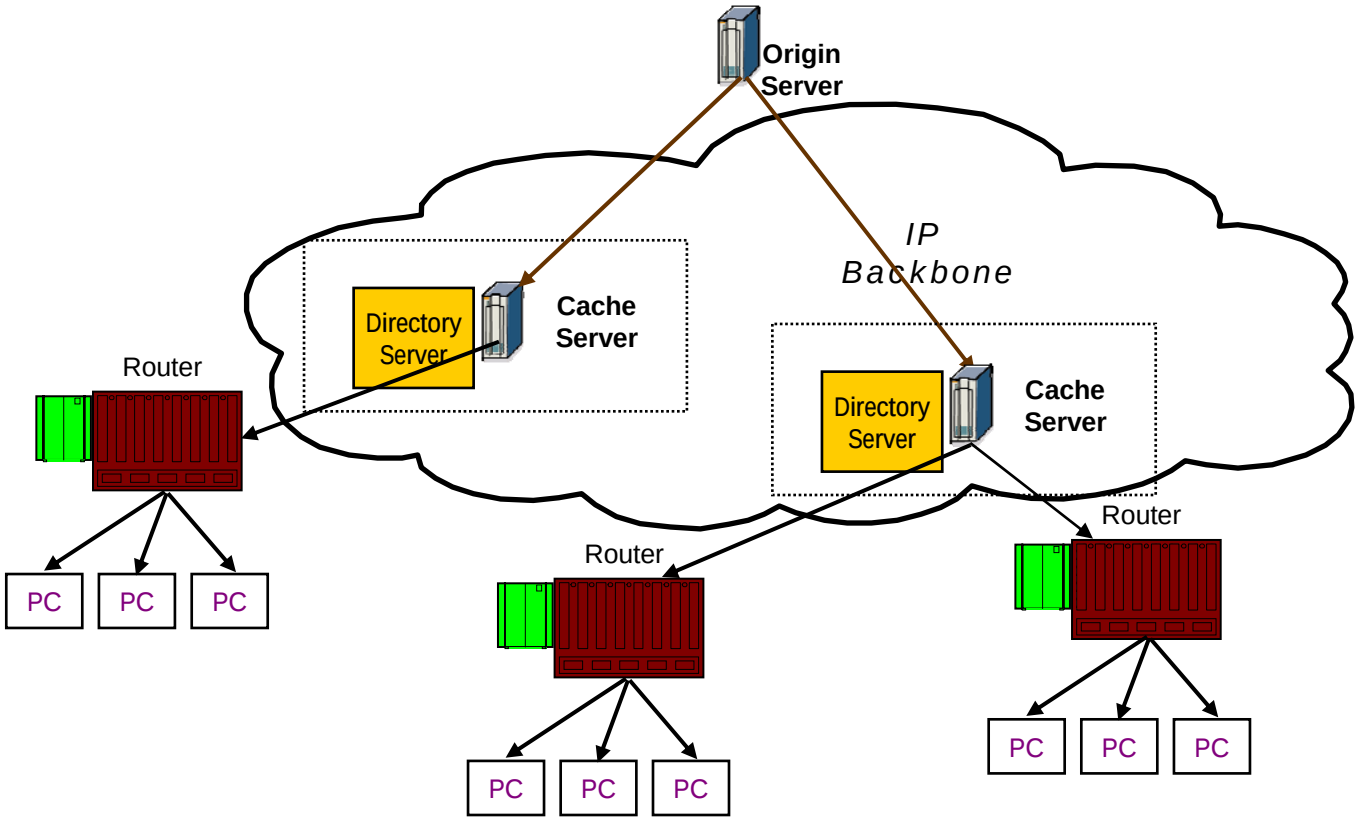
Applicability of Individual Approaches for on-demand access

- Unicast from a server works well in certain situations
  - Requests for rare content
  - Unicast can provide quick response to a user request
    - Can adapt quality to individual user's B/W availability
- Peer-to-peer between user devices
  - Good if upload bandwidth from user is large
  - Potential to both decrease content download times (user benefit) and reduce backbone route miles (ISP benefit) by downloading from nearby peers
- Multicast of a particular piece of content to large number of consumers can be very resource-efficient
  - Works well for access to popular content, especially with bursty requests
  - Can offer reduction in network bandwidth to deliver content with large number of concurrent users
  - Difficult to serve unpopular content, requests spread out over time
- How can we work well under all these different situations?

Our Approach for Video-on-Demand

- Unified approach to provide efficient support for VoD in a service provider environment using
  - Multicast: resource usage decoupled from population
  - Caching at the clients
  - Peer-to-peer that is topology aware
  - Server unicast if needed
- Key Goal: Adaptive and Flexible – dynamically exploit the most appropriate mechanism to deliver the content
- Good user experience
  - Fast start while decoupling user-perceived performance from popularity
  - Maintain quality – minimal (goal is to approach zero) user perceived interruptions while watching arbitrary length content

Network Infrastructure for CDNs



Content likely distributed to servers at PoPs (“cache servers”)

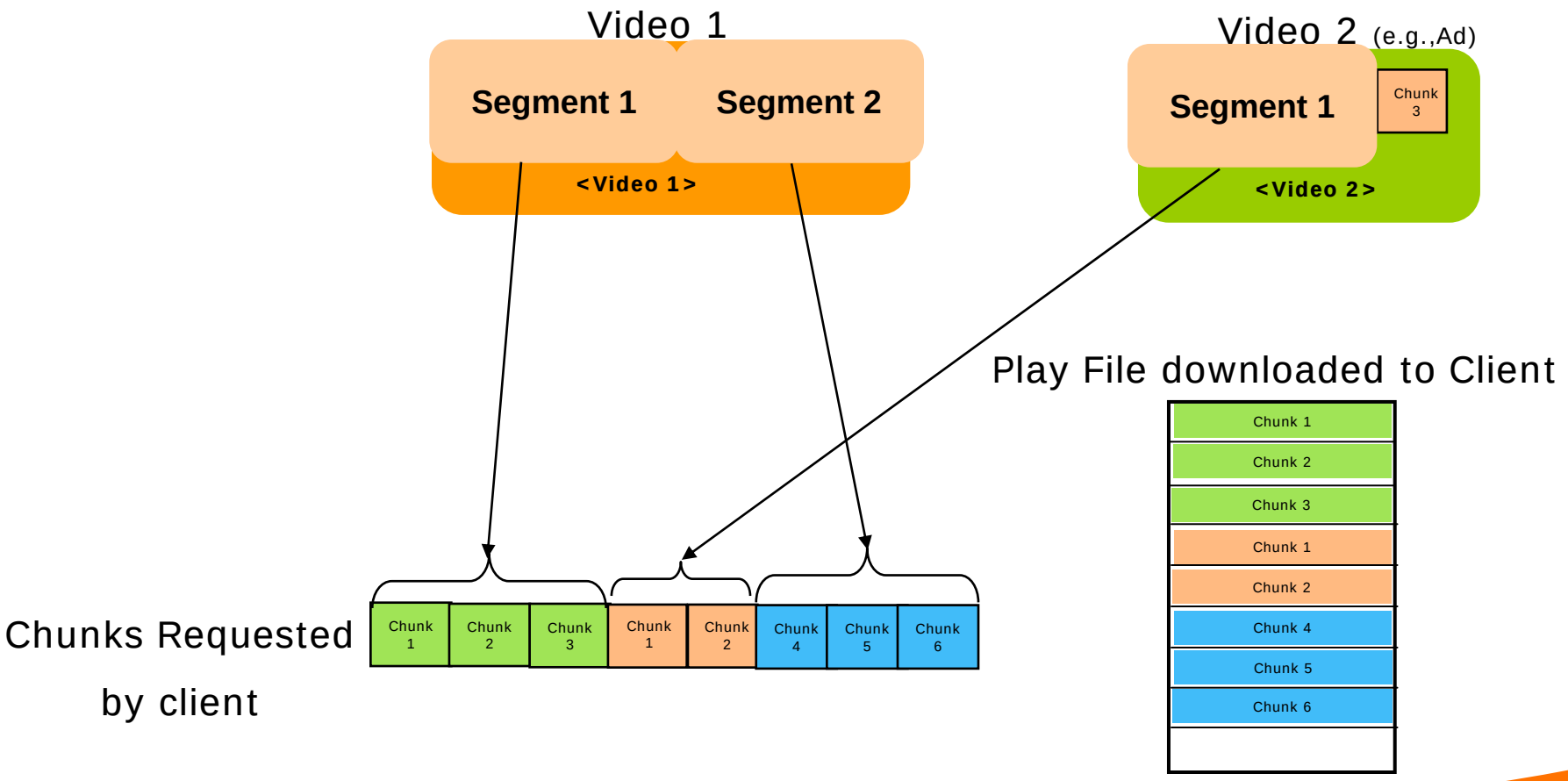
Focus: distribution of content from servers at PoPs to consumers in the local metro area

- limited uplink bandwidth from each home; small storage available @home

Data Model

- Video consists of a sequence of segments
- Segments comprise a sequence of ‘chunks’
- Chunks are the smallest addressable unit
  - Clients send requests for (sequence of) chunks
  - Chunks may be relatively small, on the order of 30 secs.
- Chunk size considerations
  - Typical viewing size of small videos
  - Amortization of overheads
  - Resilience to download failures from a serving node
- Client would be provided with a “play file” that identifies the chunks that constitute the requested pieces of content
  - Play file downloaded from origin server
    - meta-data that includes suitable tags and chunk UIDs
  - Client requests chunks following the play file

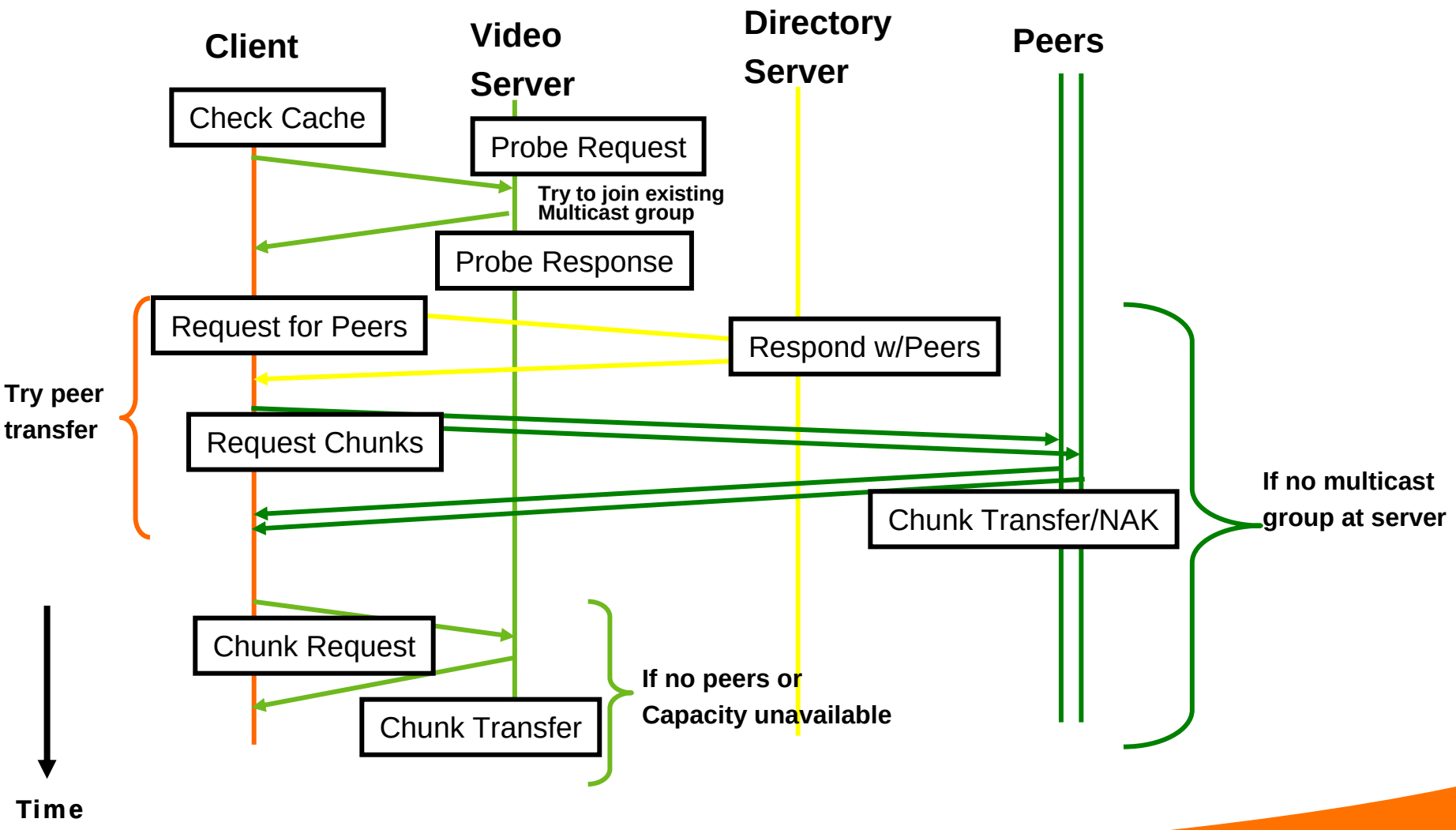
Accessing Content



Overview of Approach

- Decision making is performed on an individual chunk basis
  - Adapt to the current conditions on a dynamic basis
- We first exploit local cache to serve up content, especially for the initial segments of the video content
  - Enables fast start-up
- Client computes chunk's current "deadline"
- Query server if multicast group for chunk exists – client added to the multicast group
  - Server *multicast* is preferred first
- Second choice is to obtain content from peers
  - Directory server provides list of peers
    - Can be made "network friendly"
  - Local peers are favored over peers that are more hops away
- Third choice is to contact server if suitable peer is not found that can serve up the chunk
  - Server scheduling mechanisms attempt to batch chunk requests so as to exploit multicasting of chunk to multiple receivers

Overview of Approach



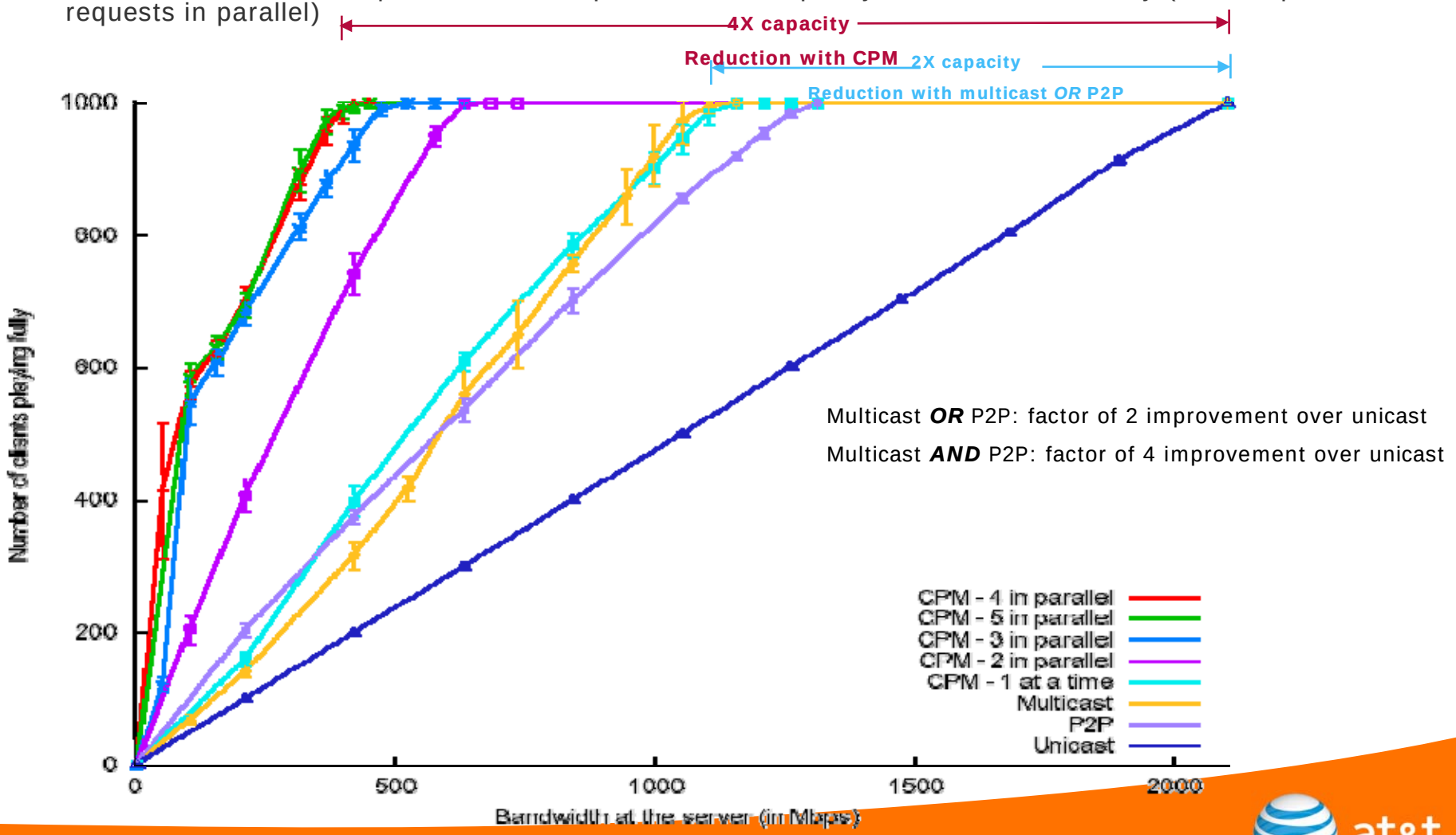
at&t

Further Evaluation using Synthetic Workload

- Understand behavior of CPM with parameter variations
- Environment has 1 server, a directory server, 1000 clients
- Clients request from a library of 888 videos each 30 minutes in length
  - 60% of the requests go to small subset (e.g., 8) videos
  - 30% requests go to medium popularity (80) movies
  - 10% requests go to rare (800) movies
  - Video playout rate 2 Mbps
- Clients have 1 Mbps nominal upload bandwidth
  - Clients pre-populated by server with 10 chunks (5 min) of startup of video (~75 MBytes)
    - Each client pre-populated with only one movie from the popular set
- Inter-arrival time of request for video
  - Base case for arrival of requests for video
    - 50% of requests within initial 5 minutes
    - Remaining 50% requests uniformly distributed over the next 20 minutes
- We vary almost every parameter (burstiness of request inter-arrival time, popularity, chunk size, uplink bandwidth)
  - Viewing model: start to finish of video (no FF/REW)

Estimate of Server Capacity

- Number of clients able to play out video without underrun for a given server capacity
  - Unicast increases linearly until all the clients are served
  - Multicast resource usage grows more slowly (packs clients into a group and re-use of group) – 1.07 Gbps – less than P2P (1.29 Gbps) and slightly better than CPM getting 1 chunk at a time
  - When CPM fetches multiple chunks from peers, server capacity reduces substantially (0.43 Gbps @ 4 chunk requests in parallel)

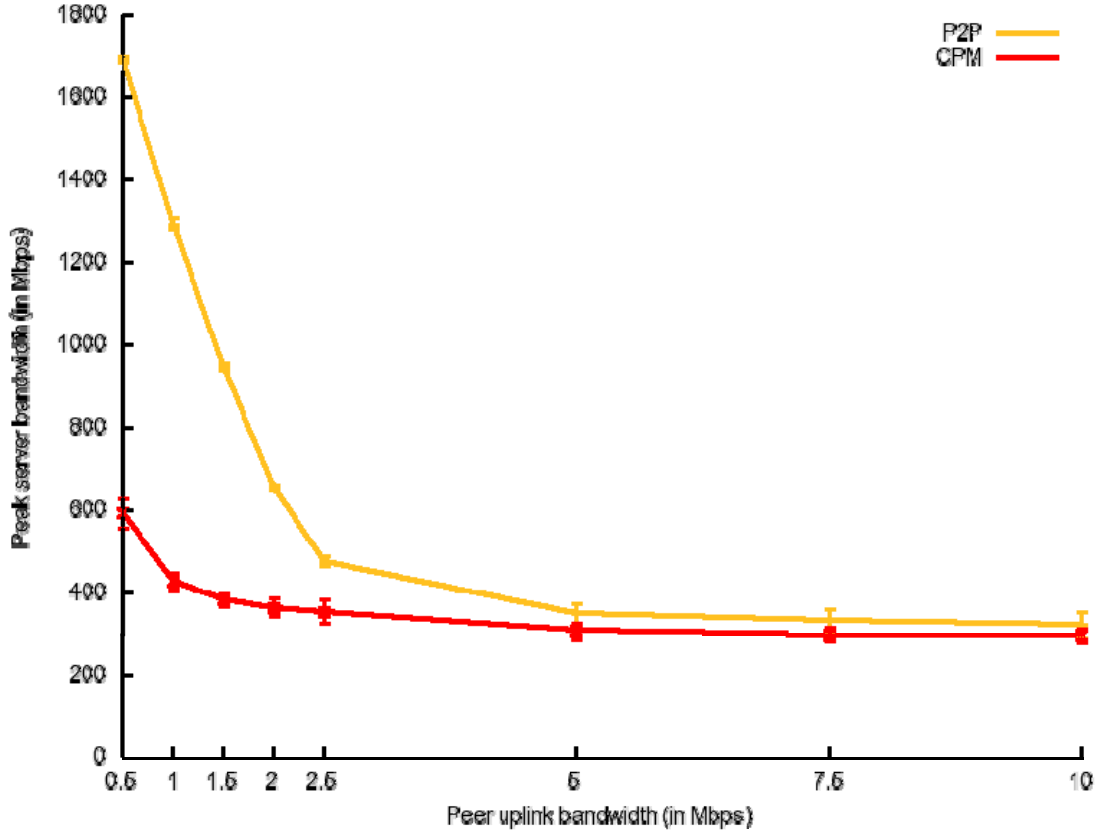


Multicast ***OR*** P2P: factor of 2 improvement over unicast

Multicast ***AND*** P2P: factor of 4 improvement over unicast

at&t

Sensitivity to Peer Upload Bandwidth

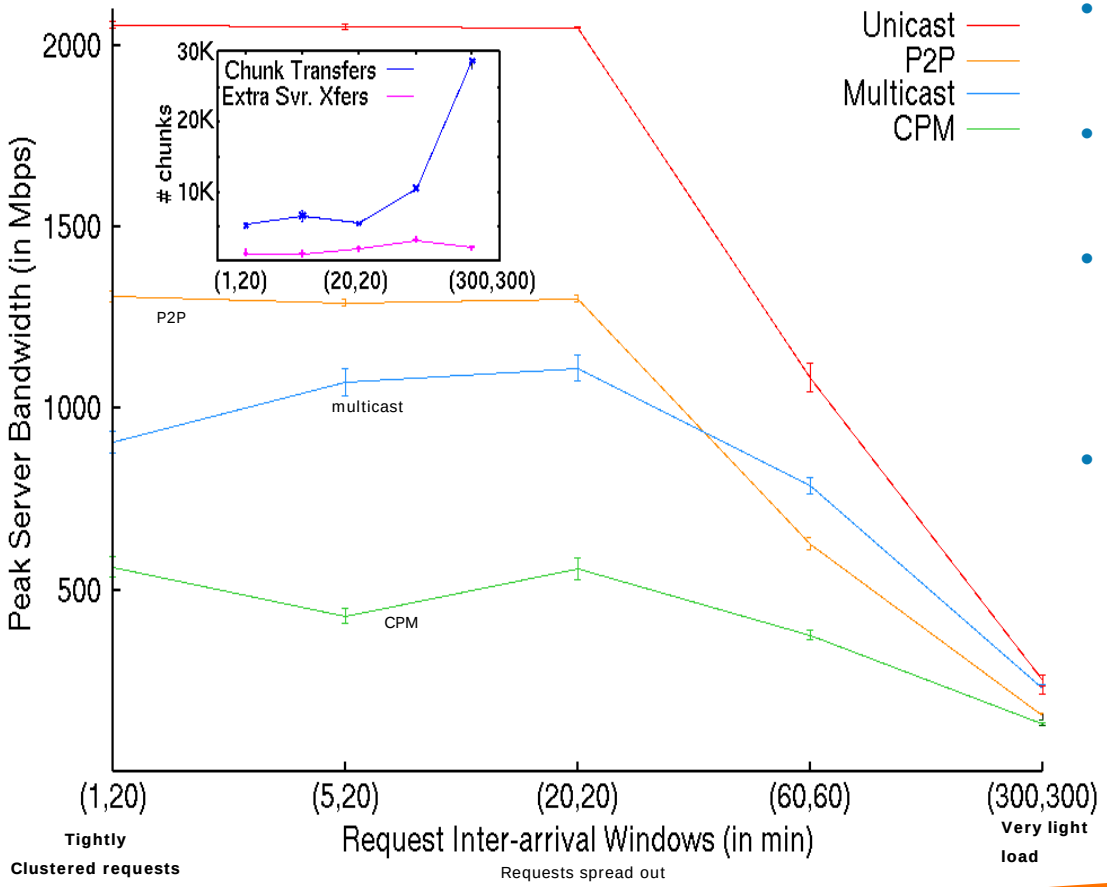


- Default case was when peer upload bandwidth (= 1 Mbps) limited to only ½ video playout rate (= 2 Mbps)
- With increase in peer upload capacity, difference between CPM and P2P diminishes

- Even when the upload bandwidth is constrained, CPM still works well
  - We can manage/control the application, and protect other applications that require bandwidth to the home
- With typical P2P environment, quality may suffer; typical usage now – systems tend to use considerable portion of uplink capacity

Sensitivity to Request Inter-arrival Time

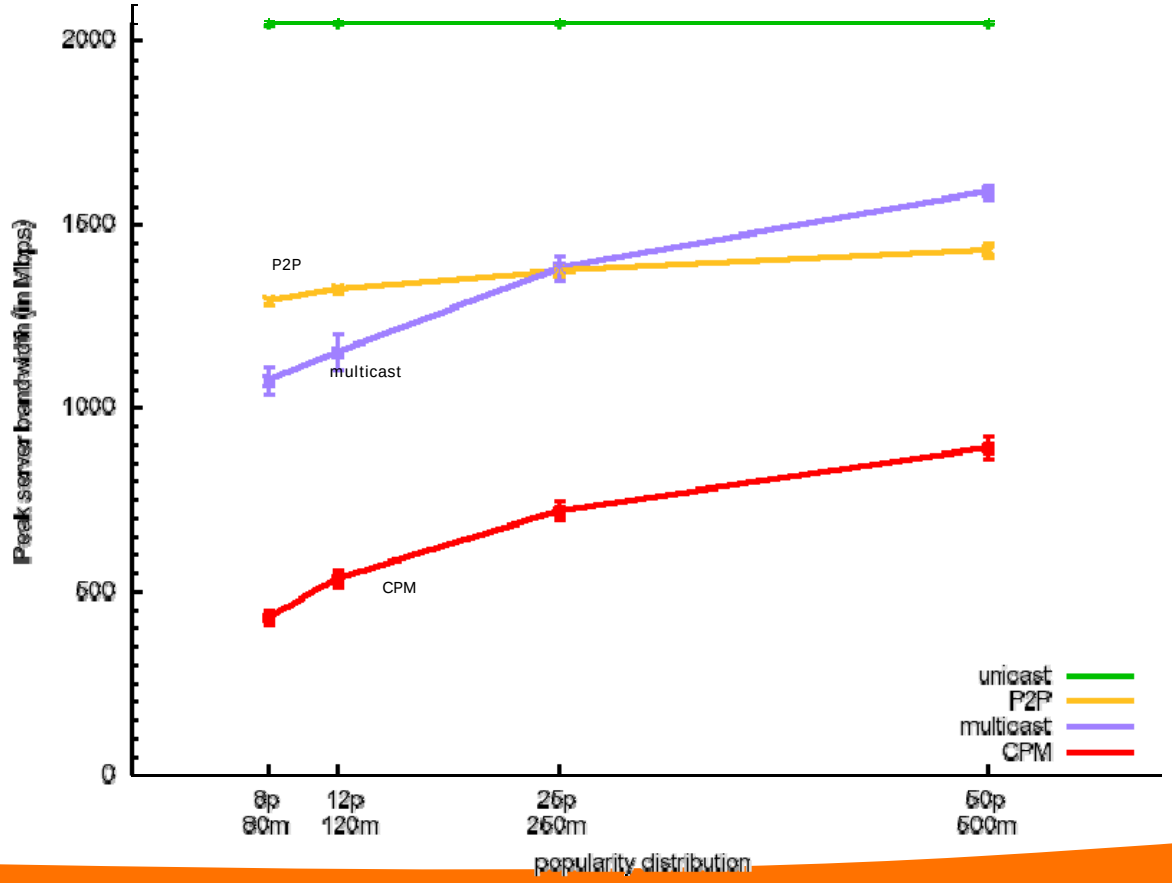
- Default - each client played only one 30 minute movie. (5,20):
  - 50% of the clients arrived in 1st 5 minutes
  - Remaining 50% of the clients arrived in the 1st 20 minutes
- Varied Inter-arrival time intensity (a,b) over wide range



- Unicast – doesn't matter – server bandwidth drops only when client requests go away
- Multicast – increase in server bandwidth until a large # clients complete playing their video
- P2P server capacity requirements are similar in behavior to unicast – except P2P peers reduce server load
  - As requests get sparse, P2P becomes better than multicast
- CPM improves over all others
  - Able to hand impulse load through server multicast
  - Handle sparse requests through peer transfers
  - At intermediate arrival intensity (5,20), CPM is aided by peers. (20,20): slight increase in duplicate server transfers
  - Subsequently, peer transfers help

Sensitivity to Popularity of Content

- CPM gains from multicasting popular content
  - We vary the popular set from 8 to 50 (chosen by 60% of users)
  - Medium popular content varies from 80 to 500 (chosen by 30% users)
  - Rare content – stays put at 800 (chosen by 10% of users)



- Server bandwidth for uninterrupted viewing with 1000 clients
- Multicast – increase in server bandwidth as popular set ⇑
- P2P server capacity increases, but more slowly as popularity is more diffuse
  - Fewer copies of popular content is pre-populated
- CPM is better but also pays in server bandwidth as popularity decreases
  - Primarily reflecting inability to aggregate requests

Summary

- Video viewing will increase and increasingly be “on demand”
- Dealing with content growth will need solutions that exploit multiple delivery techniques at network and application layer
- Cooperative peer-peer and multicast (CPM) for serving on-demand video
  - Exploits Multicast, P2P, client and network caches to provide a high-quality service
- CPM has substantial benefits
  - For the user: Viewing experience assured even when system load is high
  - For the service provider: reduces server and bandwidth resource requirements
- CPM adapts to diverse deployments
- Approach exploits information about the environment – network topology