Capacity Analysis of MediaGrid: a P2P IPTV Platform for Fiber to the Node (FTTN) Networks

Yennun Huang, Yih-Farn Chen, Rittwik Jana, Hongbo Jiang, Michael Rabinovich,
Amy Reibman, *Fellow, IEEE*, Bin Wei, and Zhen Xiao, *Senior Member, IEEE*,

Abstract— **This paper studies the conditions under which P2P sharing can increase the capacity of IPTV services over FTTN networks. For a typical FTTN network, our study shows a) P2P sharing is not beneficial when the total traffic in a local video office is low; b) P2P sharing increases the load on FTTN switches and routers in local video offices; c) P2P sharing is the most beneficial when the network bottleneck is experienced in the southbound segment of a local video office (equivalently a northbound segment of an FTTN switch); and d) sharing among all FTTN serving communities is not needed when network congestion problems are solved by using some other technologies such as program pre-caching or replication. Based on the analytical results, we design and implement the MediaGrid platform for IPTV services which monitors FTTN network conditions and decides when and how to share videos among peers to maximize the service capacity. Simulations and bounds both validate the potential benefits of the MediaGrid IPTV service platform.**

Keywords : IPTV, P2P, Content distribution network, FTTN,FTTP, xDSL, Video-on-Demand.

I. Introduction

Internet TV (IPTV) represents a prime opportunity to integrate video, voice, and data onto a single IP network. It promises to offer viewers an innovative set of choices and control over their TV content. Two major U.S. telecommunication companies, AT&T and Verizon, have recently announced significant investments to replace the copper lines in their networks with fiber optic cables to create sufficient capacity for delivering many TV channels to residential customers. The LightSpeed project in AT&T, for example, aims to reach about 18 million U.S. households with a high-bandwidth digital network in the next three years. The trend is similar in Europe and Asia. Major cities in Japan, for example, already provide high-speed networks which allow customers to obtain video over IP. In China, regulatory changes are being considered to allow the re-broadcast of traditional TV content over IPTV infrastructures.

There are two types of IPTV deployment: Fiber to the Node (FTTN) and Fiber to the Premise (FTTP). In FTTN, fiber optic cables are used to connect the central hub of a network service provider to a neighborhood node within 3000 feet of customer homes, on average. Copper wires are then used to connect the node to each individual home. It is expected to provide 20 to 25 Mbps network capacity. This is a significant improvement over the current network infrastructure where copper wires are commonly used in the last 6000 feet and the available bandwidth is limited to $1.5 - 6.0$ Mbps. In contrast, FTTP brings fiber directly to each individual customer home and can provide up to 39 Mbps bandwidth. The deployment of FTTP can be expensive, however. It is estimated that FTTP requires approximately five times the capital investment of FTTN. In addition, the deployment of FTTP can take four times as long as that of FTTN. Consequently, the current plan for AT&T is to deploy FTTP only in new construction and multi-dwelling units. In contrast, FTTN is more flexible and cost efficient, making it most suitable for existing neighborhoods. The 20-25 Mbps bandwidth offered by FTTN is typically enough to support several high-quality TV streams as well as high-speed Internet and Voice over IP (VoIP) services. For this reason, we will focus on a capacity model and algorithm design to provide insight into engineering a high-capacity IPTV network. We will use the LightSpeed project from AT&T as an example. However, the analytical model presented in the paper can be applied to IPTV services over FTTP and xDSL networks in general.

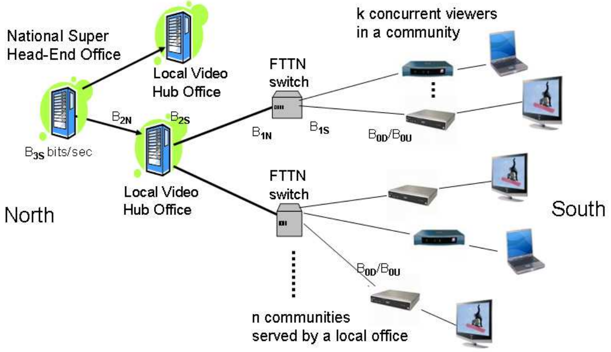


Fig. 1. System Model

The network architecture of project LightSpeed consists of a small number of national super head-ends and a large number of local video hub offices as shown in Figure 1. The super head-ends disseminate broadcast videos and on demand videos to local video hub offices, which in turn distribute video content to the customers.

To provide video on demand (VOD) service, the service provider can install multiple media servers at various locations across a content distribution network (CDN) to disperse the bandwidth contention; however, the costs of deploying and maintaining these extra servers can be very expensive. One

Y. Huang, Yih-Farn Chen, Rittwik Jana, Amy Reibman and Bin Wei are with AT&T Shannon Research Laboratory. Michael Rabinovich and Hongbo Jiang are with Case Western Reserve University. Zhen Xiao is with IBM Research Labs. The contact author is Yennun Huang: yen@research.att.com.

solution is to use a peer-to-peer (P2P) communication system. In such systems, end users (i.e., peers) interested in file sharing participate as both clients and servers, typically through an application overlay network. When a user locates an interesting file from another user, the downloading happens directly between the two without going through a central server. While early P2P systems and P2P analysis are mostly used for file downloading [4] [5], recently there have been several efforts on using the peer-to-peer approach to support live streaming [3] [8] [12] [13] [16] and VOD streaming [1] [6] [7] [11]. The P2P approach avoids the deployment problem of IP multicast service as well as the network bottleneck at the video server.

In this paper, we discuss and analyze the conditions under which P2P video sharing can increase the maximum number of concurrent viewers served by a local video hub office taking particular care in modeling the constraints of the underlying physical infrastructure. Based on these results, we develop the MediaGrid IPTV platform and simulate MediaGrid P2P streaming algorithms to maximize the number of concurrent viewers.

The contributions of the paper are the following: 1) This is the first paper which studies the benefit of P2P sharing for IPTV services over FTTN/xDSL Networks; 2) Because the service bottleneck is most likely to be in the local video hub office, our analytic results indicate that P2P sharing for viewers within a local community has the greatest potential benefit, while P2P sharing for all viewers globally may actually decrease the service capacity of FTTN/xDSL networks; 3) We analyze the conditions under which P2P sharing for IPTV services becomes beneficial; 4) Given the network constraints for a good-quality IPTV service, we design the MediaGrid platform which monitors the network conditions and decides when and how to share videos among viewers; 5) We develop a comprehensive simulation model to validate the analytical results and the benefits of the MediaGrid P2P platform.

The rest of the paper is organized as follows. We start by providing a detailed description of the IPTV system model and derive capacity bounds based on this model. Simulation results are presented in Section IV. Section V ends with a concluding remark.

II. FTTN Network Model

As shown in Figure 1, video streaming servers are organized in two levels - a local video hub office which consists of a cluster of streaming servers or proxies to serve viewers directly and national super headend (SHE) offices which can distribute videos to local video hub offices based on existing policies or on demand. We focus on the video on demand (VOD) function of the IPTV system. Each local office connects to a set of FTTN switches through optical fiber cables. Each FTTN switch connects a community of IPTV service customers through twisted-pair copper wires.

A community consists of all homes which are connected to the same FTTN switch. The uplinks (north-links) of service routers in the local office connect to national SHE offices by high-speed optical fiber networks. The parameters used throughout the paper are detailed below.

B_{0D},download bandwidth into a home (22 Mbps); B_{0U}, upload bandwidth out of a home (1 Mbps); B_{1S}, total capacity of south-bound links (downlinks) of an FTTN switch (24 Gbps); B_{1N}, capacity of the north-bound link (uplink) of an FTTN switch (1.24 Gbps); B_{2S}, maximum throughput in a local office, determined by capacities of service routers, optical network cables and/or streaming servers in a local office (up to 200 Gbps); B_{2N}, maximum capacity of the north-link of service routers in a local office (10 Gbps); u, average streaming bit rate for a video (must be at least the video encoding bit rate, 6 Mbps for high-definition programs and 2 Mbps for standard definition programs); k, maximum number of concurrent viewers supported by an FTTN switch; b, fraction of viewers in a community who get videos from peers, not from a local office; n, maximum number of communities connected to a local video hub office; N, maximum number of concurrent viewers supported by a local office; S_c, number of viewers who receive videos from peers within the same community; S_a, number of viewers who receive videos from peers in other communities.

A. Network Constraints for Conventional IPTV Services

To provide a good-quality IPTV service, the following network conditions must be met. First, the download bandwidth to the home must be greater than the streaming rate: $B_{0D} \geq u$. Second, the downlink and uplink bandwidths of an FTTN switch must each be able to support k concurrent viewers in a community: $B_{1S} \geq uk$ and $B_{1N} \geq uk$. Third, the total number of communities served by a local office is bounded by the total downlink throughput in a local office: $B_{2S} \geq nB_{1N} \geq nku$ or $n \leq B_{2S}/(ku)$. Fourth, the maximum number of concurrent viewers supported by a local video office is nk.

The traffic on the uplinks of a local office (bounded by B_{2N}) depends on the video distribution policy used by an IPTV service provider. To reduce the load on the national offices, popular videos are distributed to local offices during off-peak hours. In this case, an IPTV service administrator can apply a distribution policy where the most popular videos are available in local video hub offices so that the uplinks of local video offices will not become a bottleneck.

Example 1
Using AT&T Project Lightspeed as an example, we assume that a local office with a cluster of 100 video servers can support up to $100G$ bps streaming throughput. Given the networking constraint, $B_{0D} = 22$ Mbps, which is greater than the 6 Mbps encoding rate of a high-definition video, the maximum load on the south-link of an FTTN switch to support 192 concurrent viewers is 1.152G bps, which is much smaller than $B_{1S} = 24$ Gbps. Similarly, the maximum load on the north-link of an FTTN switch is 1.152 Gbps. However, this almost saturates B_{1N}, the capacity of the link. The maximum number of communities served by a local video office, n, is 87 (=$100G/1.152G$) and the maximum number of concurrent viewers supported by a local video office is $87 \times 192 = 16704$.

Note that in this example, the bottleneck is B_{1N}, the link capacity between a local office and an FTTN switch.

B. Network Constraints for MediaGrid IPTV services

To simplify the discussion, the constraint analysis concentrates only on thethe none-or-all case. Each peer gets data entirely from either the server or peers. The task of serving video streams on a P2P network can be shared by several peers. However, the analysis presented here can be easily extended to the case where a viewer can get part of a video from his/her peers and part from the local video office server.

1) **Case 1: P2P sharing among peers within a community:** Among k concurrent viewers in a community, $S_c = kb$ of them will get videos from peers within the same community, and $k - S_c$ will get videos from servers. The following constraints must be satisfied for good quality IPTV services:

$$\begin{aligned} C1: B_{1S} &\geq (k + S_c)u = k(1+b)u \\ C2: B_{1N} &\geq (k - S_c)u = k(1-b)u \\ n &\leq B_{2S}/[(k(1-b)u)] \\ N &= nk \leq kB_{2S}/[(k(1-b)u)] \end{aligned}$$

P2P viewers gets video from peers in its community, so the uploaded P2P video traffic is uS_c. Therefore, the total traffic generated by P2P sharing for the south-links (downlinks) of an FTTN switch is $2S_c$. In this case, the capacity of the southbound links (downlinks) of an FTTN switch must be greater than the sum of video streaming traffic $(k - S_c)u$ coming from video servers and the total P2P video traffic $2S_cu$ as shown in constraint C1. Given the increased upload traffic, P2P sharing within a community may not be feasible if the downlink bandwidth of an FTTN switch is the bottleneck. However, P2P sharing decreases the load on the uplinks of the FTTN switch as shown in constraint C2. Therefore, P2P sharing for IPTV within a community will have the most benefit if the infrastructure bottleneck is on the uplink bandwidth of an FTTN switches.

Example 2

Let $S_c = 92$ out of 192 viewers get video from peers in their community. Then, the maximum load in the south-link of an FTTN switch is $(k + S_c)u = 1.704G$ bps, compared to 1.152 Gbps in example 1. The maximum load on the north-link of an FTTN switch is $(k - S_c)u = 600M$ bps, compared to 1.152 Gbps in example 1. The maximum number of communities supported by a local office, $n = 167$ (=$B_{2S}/[(k(1-b)u)] = 100G/600M$), compared to 87 communities in example 1. The maximum number of concurrent viewers supported by a local video office, $N = 167 \times 192 = 32064$, compared to 16704 in example 1.

This example shows that P2P sharing reduces the load between a local office and an FTTN switch (the north-link of an FTTN switch) and therefore reduces the possibility that B_{1N} is a bottleneck. This example also shows that without upgrading existing network infrastructure, P2P sharing can significantly increase the number of concurrent viewers that can be served by a local video office.

2) **Case 2: P2P sharing among peers in a local hub office:** In the second case, we consider P2P video sharing among all viewers served by a local office. Within a community, S_c viewers get all their videos from peers within the same community and S_a viewers get the entire or part of their videos from peers in other communities, where $S_c + S_a = kb$. The following constraints must be satisfied to guarantee good quality IPTV service:

$$\begin{aligned} B_{1S} &\geq = (k + S_c + S_a)u \geq k(1+b)u \\ B_{1N} &\geq (k - S_c + S_a)u \geq k(1-b)u \\ n &\leq B_{2S}/[(k - S_c + S_a)u] \\ N &= nk \leq kB_{2S}/[(k - S_c + S_a)u] \end{aligned}$$

The maximum traffic in the uplink of an FTTN switch occurs when each of the S_a viewers get its entire video from peers outside its community. To maximize the capacity of a local office, we balance the video sharing traffic among peers in all communities. When an equilibrium has been reached, each community will upload S_au bps to other communities and receive S_au bps from other communities for P2P video sharing. Thus, the total video streaming download traffic in the uplink (north link) of an FTTN switch is $(k - S_c - S_a)u$ bps coming from the local office servers and S_au bps coming from peers in other communities while the upload traffic is S_au bps to support peers in other communities. Therefore, in balance, the total traffic on the uplink of an FTTN switch is $(k - S_c - S_a)u + 2S_au$.

From the above constraints, increasing P2P sharing among peers across all communities (i.e. increasing S_a) increases the traffic on both the uplink and the downlinks of an FTTN switch, but reduces the load on the uplink of a local office. So, if B_{2N} is the bottleneck, applying P2P technology for peers in all communities of a local office is beneficial. However, even in this case, an IPTV service provider could apply other content distribution technologies (i.e. caching or replication) to distribute video files from national offices to the local offices to reduce the load on the uplink of the local office. So, P2P sharing among all communities may not be needed.

3) Bottleneck Observations: From the analysis of the above three cases, we can derive the following conclusions:

- P2P technology is useful when some of the network links in FTTN switches or local video hub offices are the bottleneck.
- If B_{1S} is the bottleneck, P2P sharing does not help because any peer sharing increases the downlink traffic of an FTTN switch.
- If B_{2N} is the bottleneck, P2P sharing among viewers in all communities of a local office helps to reduce the load on B_{2N}. However, if a service provider can apply other technologies to distribute video files from national SHE offices to the local offices, P2P sharing may not be needed.
- If B_{1N} or B_{2S} is the bottleneck, P2P sharing within a community reduces the load on the north link of an FTTN switch and its local office. In this case, P2P sharing within a community helps to reduce the load on these congested links. However, P2P sharing across

communities increases the possibility that B_{1N} or B_{2S} is a bottleneck. Therefore, P2P sharing across communities should not be used when B_{1N} or B_{2S} is the bottleneck.

C. Bounds for Maximum Concurrent Viewers Supported by a Local Video Hub Office

Since the benefit of P2P sharing among all communities is limited, we only consider P2P sharing within a community in the following study. The number of concurrent viewers supported by a local video office, $N(= nk)$, can be increased either by increasing n or k. In practice, the number of communities connected to a local hub office, n, is determined by the number of POPs (point-of-presences) connected to a local office and the number of available locations to install fibers and FTTN switches. Increasing n requires significant planning and capital investment to rent/buy new space to install IPTV network equipment. Therefore, to increase N, it is more economical to increase k. In the following, we assume that n is fixed at the time of deployment. Our MediaGrid P2P streaming technology focuses on increasing k to maximize N, the total number of concurrent viewers served by a local office.

Without P2P sharing, the number of concurrent viewers is

$$N_{nop2p} = nk_{nop2p} \leq \frac{B_{2S}}{u} \text{ and} \quad (1)$$

$$k_{nop2p} \leq \min\left(\frac{B_{1S}}{u}, \frac{B_{1N}}{u}\right) \quad (2)$$

When MediaGrid P2P sharing is applied, let k_{p2p} be the number of viewers in a community and b be the fraction of these viewers who receive videos from peers instead of streaming servers. From Constraints 1 and 2, we have

$$N_{p2p} = nk_{p2p} \leq \frac{B_{2S}}{(1-b)u} \text{ and} \quad (3)$$

$$k_{p2p} \leq \min\left(\frac{B_{1S}}{(1+b)u}, \frac{B_{1N}}{(1-b)u}\right) \quad (4)$$

Normally, b increases as k increases, since as k increases, there is a better chance that a video has already been viewed and stored in the set-top boxes of some peers, and a better chance it can be downloaded from a peer. We assume a model where z is a constant and

$$b = zk_{p2p} \text{ when } zk_{p2p} < 1, \text{or}$$
$$b = 1 \text{ otherwise.}$$

We pick representative values of $n = 30$ and $z = 1.4e - 4$ in Eqn.(2) and Eqn.(4) to see the effect on viewer capacity.

Figure 2 shows two capacity surfaces plotted against varying uplink,B_{1N} and downlink,B_{1S} throughputs. The inclined P2P surface shows the maximum number of users that can be supported as per Eqn.(4) bounded by the minimum k_{p2p}. The other surface (No P2P) shows the service capacity of the local video office using a centralized FTTN architecture. As shown, P2P sharing is not beneficial when B_{1S} value is small.

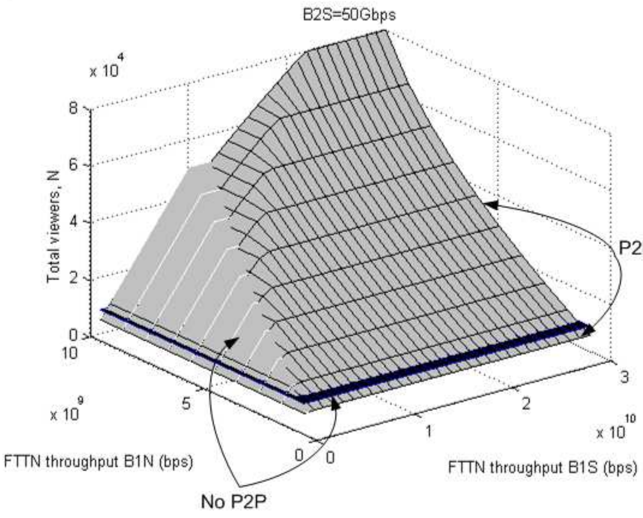


Fig. 2. Capacity Bounds in a local video hub office where n=30, z=1.4e-4

Therefore, it is crucial to determine the threshold value of B_{1S} and allow P2P sharing only when the capacity of the south link of a FTTN switch is greater than the threshold value. In practice, the values of b and k_{p2p} depend on a number of factors such as the sharing policy, video distribution, client storage, client bandwidth constraints and video access patterns, etc.

From the above discussions, we design the MediaGrid P2P streaming algorithms to maximize the number of concurrent viewers in a community by monitoring bandwidth constraints, the network traffic and the availability of videos in peers. In the next two sections, we study the maximum number of concurrent viewers supported by a local office by simulating the MediaGrid P2P streaming algorithm on a FTTN/xDSL network environment.

III. MediaGrid P2P Algorithm

There are three key novel aspects of our MediaGrid video sharing algorithm. First, it explicitly takes into account the capacity limitations of the underlying physical network infrastructure, not the overlay network of peers as other systems. Second, to avoid a long delay before the start of the requested program, the algorithm allocates enough supporting peers to collectively serve a video at or above the stream encoding rate, while at the same time ensuring that each peer upload rate does not exceed its uplink capacity. Third, unlike existing streaming P2P approaches that deal with peer disconnections by special stream encodings that either add redundancy or gracefully degrade stream quality in response to failures (the approaches we call *passive error handling*), in MediaGrid, the local office either substitutes a failed peer with a different peer or serves the missing content itself, thereby utilizing *active error handling*.

Because our analysis shows that P2P sharing is mostly beneficial within a community and is actually likely to be detrimental for cross-community or global sharing, we will concentrate here on P2P sharing within a single community. In MediaGrid, the local office server maintains full information

```
ProcessRequest(Stream S, Viewer V)
// Request for stream S received from viewer V
   If the FTTN switch of the viewer satisfies Constraint 1
   with $S_c = S_c + 1$ and $k = k + 1$
      PeerSet = SelectPeers(S)
      // Find a set of peers in the viewer's community to serve request
      If PeerSet $\neq \emptyset$
         Send message UsePeers(EncodingRate(S),PeerSet) to viewer V
         terminate
      endif
   endif
   if there is enough total office download capacity
   and bandwidth of link to the viewer's FTTN switch
      Serve stream S to viewer V directly
   else
      Send RequestDenied() to viewer V
end

ReplacePeer(Viewer V, Peer P)
// Request to replace a failed peer P from viewer V
   Let S be the stream being viewed by V,
   and n be the number of peers serving the stream to V.
   P' = FindPeer(S);
   if P' $\neq$ NULL
      send ReplacementPeer(P') to V
   elseif there is enough total office download capacity
   and bandwidth of link to the viewer's FTTN switch
      send UseOffice() to V // In response, V will request stream segments
            that peer P used to serve from the office directly
   endif
end
```

Fig. 3. MediaGrid Server Algorithm

about each community, including which viewers are currently watching which programs, which programs are available for upload from each peer, the current uplink and downlink bandwidth consumption by each peer and each FTTN switch, and the total load on the office communication links. Viewers send to the office two kinds of requests: ProcessRequest requests a stream and ReplacePeer indicates that one of the peers serving them failed. The office processes these messages according to the MediaGrid Server algorithm shown in Figure 3.

When the client receives UsePeers message, it divides the streaming rate equally among the peers[1], calculates which stream segments to request from each peer and the necessary buffering before starting to render the stream, and sends the appropriate requests to the peers. The details of this algorithm are omitted for brevity.

The most intricate part of the algorithm is the implementation of the SelectPeers and FindPeer functions. The difficulty arises if a peer able to serve rare content is already assigned to serve more commonly available content. Because of the severely limited uplink capacity, it will not be able to serve more than one video. This leads to the need to dynamically reassign peers to viewers. This reassignment will occur transparently to the viewer. For example, consider peers P1, P2, P3, and P4 and let P1 and P2 have two streams, A and B, and P3 and P4 only have stream A. Assume a viewer requested stream A and the server chose peers P1 and P2 to serve it. If another viewer requests stream B, without dynamic reassignment, Peers P1 and P2 are unable to serve it because their capacity is used up; the server would have to serve the new viewer directly. However, if the server reassigns the first viewer to peers P3 and P4, the second viewer can be served by P1 and P2. Selecting peers for a given request may trigger cascading reassignment of previously selected peers, or global scheduling of all currently served streams among all the peers, which could cause long delays potentially exceeding clients' buffer capacity. To avoid such global scheduling our SelectPeers algorithm resorts to heuristics. The main points of the SelectPeers algorithm are:[2]

- Given the current values for the peer uplink capacity (1 Mbps) and stream encoding rates of 2 Mbps and 6 Mbps, we assume peer uplink capacity is less than the stream coding rate.
- For ease of management, the algorithm attempts to select the fewest peers with sufficient aggregate uplink capacity for serving the stream. In particular, together with the previous assumption, this means that each peer can upload to only one viewer at a time.
- When more than enough *free* (i.e., unassigned) peers with the requested stream exist, we select n peers, where $n = Rate(S)/B_{0U}$, as follows. Let $S_{max}(P)$ be the *most* popular stream among streams stored at peer P. We select n peers P with the *least* popular streams $S_{max}(P)$, among all the candidate peers. The rationale is that peers whose most popular stream is not very popular will probably not need to serve a future request.
- When not enough free peers exist, we attempt to free up additional peers by reassigning the streams they currently serve to other peers. Let $\mathcal{P}$ be the set of busy peers that have the requested stream and m be the number of peers we need to free up. The algorithm tries to free the m peers in the decreasing order of the popularity of the streams they are currently serving. The rationale behind this heuristic is that the more popular the currently served stream is, the more likely it is to find another peer that could take over this stream's delivery. Thus, the algorithm sorts peers in $\mathcal{P}$ in the decreasing order of popularity of the streams they are currently serving, and then for each peer P in the list, attempts to find an idle peer that has the stream being served by P. This step completes once the first m peers are freed up, or until it went though the entire candidate list $\mathcal{P}$, in which case the algorithm returns a failure (and empty peer set).
- If the overall set of n peers for a requested stream have been found, the viewers of reassigned streams are notified to switch to new peers.

IV. Simulation

A. Simulation Model

To study the impact of MediaGrid P2P delivery on system scalability and capacity, we performed an event-driven simulation study. Based on our earlier analysis, we simulate a system comprised of the local office and one community.

[1] Future work will explore dividing the streaming rate unequally.

[2] The FindPeer algorithm is a specialization of SelectPeer to one peer and is not discussed further.

We assume the same behavior for every node in the community: an idle peer requests a stream with probability of 2% every time tick. A time tick occurs every minute. A peer may download only one stream at a time. When a peer issues a request, it selects one of 1000 programs according to Zipf's popularity distribution. Each stream lasts 60 minutes and has a data rate of either 2 Mbps (standard definition TV) or 6 Mbps (HDTV). Once downloaded, the program remains available at the peer for a period called the stream time-to-live (stream TTL). A peer may be turned off and on by its user. An operational peer is turned off with probability 0.1% on every time tick, and a non-operational peer is turned on with probability 0.5% on every tick. This means that every peer on average spends five times longer on than off.

B. Simulation results

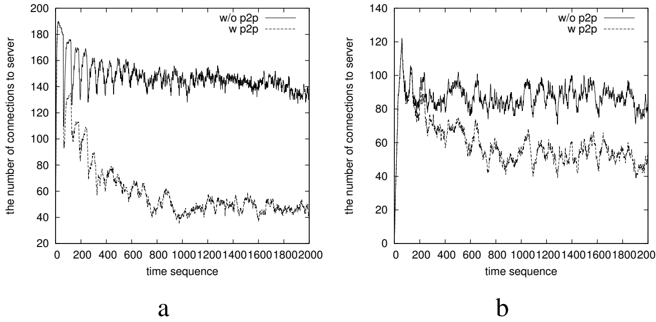


Fig. 4. The number of concurrent connections to local office from one community, a) Standard TV, normal demand (1/50 requests per minute per idle user), b) HDTV, normal demand.

We begin by examining the behavior of the system utilizing today's components characteristics: the FTTN switch supporting the community size of 192 users, user download bandwidth of 22 Mbps and upload bandwidth of 1 Mbps, and the office-to-FTTN link bandwidth of 1.244 Gbps. Figure 4 shows the number of viewers served by the local office for standard definition TV programs under normal demand, and for HDTV under normal demand. Fewer concurrent connections at the office implies greater system scalability because the office can serve more communities without overload. This figure shows that, for standard-definition TV, P2P delivery reduces the load on the local office by over 50%. However, the benefits diminish greatly for high-definition TV, because existing peer uplink capacity is not adequate for serving HDTV, since one stream delivery requires cooperation of six operational peers.

The above result might seem discouraging because HDTV will be more prevalent in the near future. However, since the hardware will also improve, we next explore how hardware improvements impact these results. Because the previous graphs show the system stabilizes after 1000 minutes, each point in the graphs below represents the average values observed between 1500 and 2000 minutes.

Figure 5(a) shows the total number of concurrent viewers in the community that the system can support with and without P2P, as a function of the total number of users in the community. Note that because the request rate is the same in both cases, the difference between curves is due to denied user requests. The graph shows that the system without P2P saturates at a community size around 500, while P2P shows no saturation[3]. In fact, a larger community size makes it more likely to find six operational peers for delivering a given stream, which mitigates the HDTV problem that limited P2P benefits. Figure 5(b) explains the underlying reason for this phenomenon by splitting the total concurrent viewers into those served by the office and by the peers. It shows that the load on the office grows very slowly and never exceeds around 220 concurrent streams. The additional demand due to a larger community size is absorbed by other peers. Thus, we can scale the HDTV delivery just by increasing the number of downstream ports on an FTTN switch, even if the peer link capacity stays the same. Since it is highly likely that HDTV will replace standard TV in the near future, we concentrate on HDTV for the rest of our simulation study.

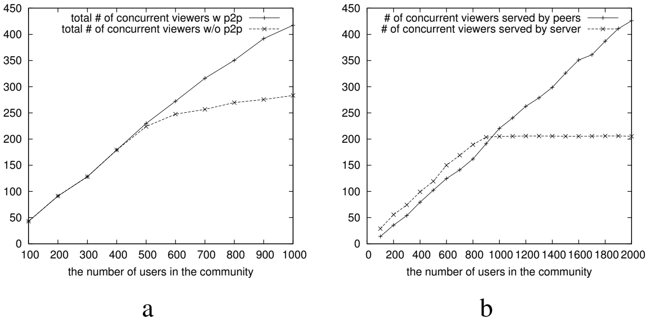


Fig. 5. Fulfilled HDTV stream requests, a) The total number of concurrent requests in a community that can be supported by the local office, b)The concurrent requests served by the office and by the peers.

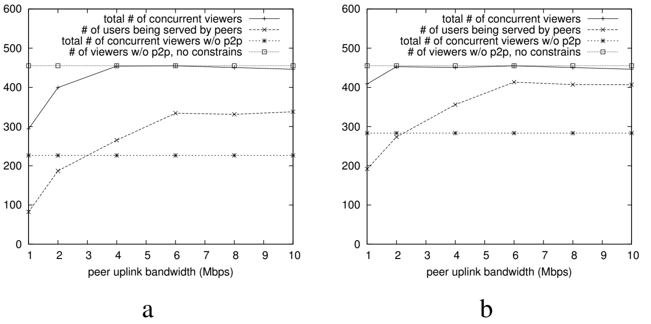


Fig. 6. The effect of uplink bandwidth on HDTV delivery, a) 200 minutes TTL, b) 1000 minutes TTL.

Next, we examine how HDTV delivery and P2P benefits are impacted by potential bandwidth growth in peer uplinks as well as in the link between the office and the FTTN switch. The community size is 1000 for these experiments. Figure 6 shows, for different stream TTL values (i.e., durations of retention in peer's cache), the concurrent requests a system can serve with and without P2P, and the number of concurrent requests served by peers in the case of P2P delivery. To provide

[3] Slight growth beyond 500 users is due to the repeated viewings of previously downloaded programs, which are satisfied from caches of set-top boxes.

a basis for comparison, the figure also shows the number of requests served by an ideal system with unconstrained capacity. Any difference with the ideal case indicates denied requests.

We can draw several conclusions from these results. First, a higher peer uplink bandwidth mitigates limited TTL and causes a rapid increase in P2P benefits. For TTL of 200 minutes, doubling the uplink bandwidth to 2 Mbps more than doubles the number of concurrent requests served by the peers, and increasing it to 4 Mbps eliminates any denied requests. For TTL 1000, just increasing the uplink bandwidth from 1 to 2 Mbps allows the overall system to serve virtually all requested streams. This result also highlights the tradeoff between the peer's storage capacity and uplink bandwidth. The system can serve all the demand by either having peers with enough storage to retain downloaded TV programs for 1000 minutes and an uplink bandwidth of 2 Mbps, or by having peers with enough storage to only maintain streams for 200 minutes but with 4 Mbps uplink bandwidth.

Second, the number of requests served by the peers increases when the uplink bandwidth increases all the way to 6 Mbps (the stream data rate), meaning that the office can connect more communities. This result indicates the benefits of providing the uplink bandwidth of the stream data rate. At the same time, expanding bandwidth beyond the stream data rate did not provide any additional benefit in our experiments, even though a peer could in this case serve more than one request simultaneously. Finally, only P2P delivery allows the system to adequately serve a community of this size, as the traditional system will have to deny over half the user requests.

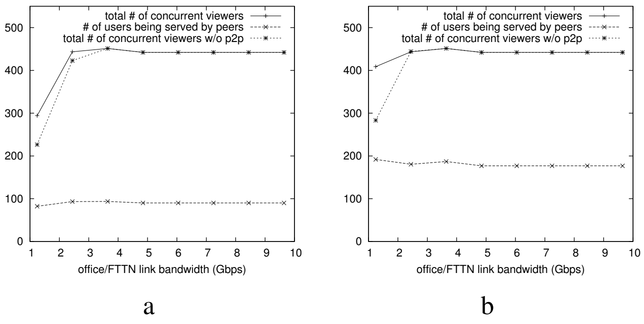


Fig. 7. The effect of office-to-FTTN bandwidth on HDTV delivery, a) 200 minutes TTL, b) 1000 minutes TTL.

Finally, we examine the impact of increasing the bandwidth of the office-to-FTTN link. Figure 7 shows the number of concurrent requests served by the system with and without P2P for different stream TTL values, as well as the number of concurrent requests served by peers in the P2P case, assuming the fixed peer uplink capacity of 1 Mbps. We can see that doubling the capacity of the office-to-FTTN connection (from 1.2 Gbps to 2.4 Gbps) allows the system to serve virtually all the demand for a given community size even without P2P delivery. Of course, the peers still help because they reduce the load on the office and allow the office to serve a larger number of communities. Also, only P2P allows the system to scale with the growth of the community. Thus, P2P allows the system to immediately benefit from more ports in the FTTN switches without corresponding increases in the FTTP's uplink bandwidth.

V. Conclusions

As IPTV services are deployed around the world, scaling the networking infrastructure to serve an increasing number of customers under a reasonable cost structure becomes a top priority to the broadband service providers. Given the constraints of the IPTV service infrastructure, we analyzed the network conditions under which P2P video sharing is beneficial. We also designed the MediaGrid platform and its P2P sharing algorithm to monitor these constraints and maximize the service capacity. Finally, we developed analytic and simulation models to compute and compare P2P video sharing benefits for an IPTV service provider. Our work demonstrates that the service providers should invest in client devices that have sufficient storage and bandwidth for supporting effective P2P video sharing.

For future work, to help us gain valuable insights and to validate our designs, we plan to deploy the MediaGrid system in real IPTV infrastructures.

References

[1] Simon Sheu, Kien A. Hua, Wallapak Tavanapong, *Chaining: A Generalized Batching Technique for Video-on-Demand Systems*, in Proc. of the IEEE Int'l Conf. On Multimedia Computing and System, Ottawa, Ontario, Canada, June 1997, pp. 110–117.

[2] Duc A. Tran, Kien A. Hua, Simon Sheu, *A New Caching Architecture for Efficient Video-on-Deman Services on the Internet*, in IEEE Symposium on Applications and the Internet (SAINT 2003), 2003.

[3] Duc A. Tran, Kien A. Hua, Tai T. Do, *A Peer-to-Peer Architecture for Media Streaming*, in IEEE Journal on Selected Areas in Communication, Special Issue on Advances in Overlay Network, 2003.

[4] X. Yang and G. de Veciana, *Service Capacity of Peer to Peer Networks*, Proc. of IEEE INFOCOM, 2004.

[5] M. Adler, R. Kumar, K. Ross, D. Rubenstein, T. Sue, and D. Yao, *Optimal Peer Selection for P2P Downloading and Streaming*, Proc. of IEEE INFOCOM, 2005.

[6] Yang Guo, Kyoungwon Suh, Jim Kurose, Don Towsley *P2Cast: P2P Patching Scheme for VoD Service*, in WWW 12th, Budapest, Hungary, 2003.

[7] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, *A measurement study of peer-to-peer file sharing systems*, in Proc. of ACM/SPIE on Multimedia Computing and Networking (MMCN'02), San Jose, CA, USA, January 2002.

[8] Eveline Velos, Virgilio Almeida,Wagner Meira, Azer Bestavros, Shudong Jin, *A Hierarchical Characterization of a Live Streaming Media Workload*, in IEEE IMW'02, 2002, pp. 117-130.

[9] Kien A. Hua, Mounir A. Tantaoui, *Cost Effective and Scalable Video Streaming Techniques*, in Borko Furht, Oge Marques (eds.): Handbook of Video Databases, CRC press, 2002.

[10] Yang-Hua Chu, Sanjay G. Rao, and Hui Zang, *A Case for End System Multicast*, in Proc. of ACM SIGMETRICS 2000, pp. 1-12.

[11] Yang Guo, Kyoungwon Suh, Jim Kurose, and Don Towsley, *A Peer-to-Peer On-Demand Streaming Service and Its Performance Evaluation*, in Proc. of IEEE Int. Conf. on Multimedia Expo (ICME'03), 2003.

[12] H. Deshpande, M. Bawa and H. Garcia-Molina, *Streaming Live Media over a Peer-to-Peer Network*, Stanford database group technical report (2001-20), Aug. 2001.

[13] S. Banerjee, Bobby Bhattacharjee, and C. Kommareddy, *Scalable application layer multicast*, in Proc. Of ACM SIGCOMM 2002, Pittsburgh, PA, 2002, pp. 205-217.

[14] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole, *Overcast: Reliable multicasting with an overlay network*, in Proc. of the Fourth Symposium on Operating Systems Design and Implementation, 2000, pp. 197-212.

[15] Christos Gkantsidis and Pablo Rodriguez. Network Coding for Large Scale Content Distribution. In Infocom 2005.

[16] Mayank Bawa, Hrishikesh Deshpande, Hector Garica-Molina, *Transience of Peers and Streaming Media*, in ACM SIGCOMM Computer Commuications Review, January, 2003, vol. 33, pp. 107-112.

[17] Anwar Al Hamra, Ernst W. Biersack, Guillaume Urvoy-Keller, *A Pull-Based Approach for a VoD Service in P2P Networks*, HSNMC 2004, pp. 995-1006.

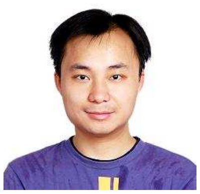
[18] M. Hefeeda and A. Habib and B. Bhargava, *Cost-profit analysis of a peer-to-peer media streaming architecture*, Technical report, CERIAS TR 2002-37, Purdue University, June 2003, http://citeseer.ist.psu.edu/article/hefeeda02costprofit.html

[19] *http://www.veritest.com/clients/reports/microsoft/ms_media_services.pdf*

Yennun Huang is the Director of Dependable Distributed Computing and Communication Research in AT&T Labs. He received his Ph.D. and M.S. in Computer Science from University of Maryland and B.S. from National Taiwan University. His current research interests are dependable computing, P2P, distributed middleware and systems, web security, content management, IPTV and mobile services.

Robin Chen received his Ph.D. in Computer Science from University of California, Berkeley, an M.S. in Computer Science from University of Wisconsin, Madison, and a B.S. in Electrical Engineering from National Taiwan University. His current research interests include mobile computing, P2P incentives, IPTV, and World Wide Web. Robin is a member of ACM, IEEE, and IW3C2, the International World Wide Web Conference Committee.

Rittwik Jana, Senior Technical Specialist at AT&T Labs Research. He received a B.Engg degree in electrical and electronics from the University of Adelaide, Australia, in 1994. He received a Ph.D. degree from the Australian National University in 1999. He worked as an engineer at the Defense Science and Technology Organization (DSTO), Australia from 96 to 99 and as a member of technical staff at AT&T Labs-Research, New Jersey from 99 to date. He has been continuously working in the area of mobile and wireless communications for ten years covering aspects from physical layer modem design to application layer software development. His primary expertise falls in the areas of wireless receiver design, mobile service platform design and wireless channel modeling. He has served as a reviewer and a program committee member for numerous IEEE conferences and journals.



Hongbo Jiang received the B.S. and M.S. degrees from Huazhong University of Science and Technology, China. He is currently working toward the Ph.D. degree in Computer Science at Case Western Reserve University, Clevealand, OH. His research concerns computer networking, especially algorithms and architectures for high-performance networks and wireless networks.

Michael Rabinovich is a Professor of Computer Science at Case Western Reserve University, which he joined in 2005 after eleven years at AT&T Labs - Research. His interests revolve around Internet and distributed systems. He holds PhD in Computer Science from the University of Washington.

Amy R. Reibman received the Ph.D. in electrical engineering from Duke University in 1987. From 1988 to 1991, she taught Electrical Engineering at Princeton University. In 1991 she joined AT&T and is currently a Technical Consultant in the Communication Sciences Research Department at AT&T Lab-Research. In 1998, she won the IEEE Communications Society Leonard G. Abraham Prize Paper Award. She was the Technical co-chair of the IEEE International Conference on Image Processing in 2002. She became elected IEEE Fellow in 2005 for her contributions to video transport over networks.

Dr. Reibman's research interests include video compression systems for transport over packet and wireless networks, and video quality metrics.

Bin Wei received the Ph.D. in Computer Science from Princeton University and has been part of the research community at AT&T Labs since then. He has been working in the areas of computer architecture, visualization, multimedia, and mobile computing. He has served as a member of the organizing committee and as a technical co-chair for a number of international conferences and workshops, also as a guest editor for technical journals. Currently, he is interested in the topics on novel applications and enabling technologies.

Zhen Xiao received his Ph.D. from Cornell University in January 2001. After that he worked as a senior technical staff member at AT&T Labs for five years where he received the "Research Excellence Award". He is now a Research Staff Member at IBM T.J. Watson Research Center. His research interests include SIP, multimedia, IPTV, Grid computing, Web technologies and content delivery, security and dependability, and reliable multicast. He is a senior member of IEEE.