CUBS: Coordinated Upload Bandwidth Sharing in Residential Networks

Enhua Tan[1], Lei Guo[2], Songqing Chen[3], and Xiaodong Zhang[1]

[1]Dept. of Computer Science & Eng.
The Ohio State University
Columbus, OH 43210
{etan,zhang}@cse.ohio-state.edu

[2]Yahoo! Inc.
701 First Avenue
Sunnyvale, CA 94089
lguo@yahoo-inc.com

[3]Dept. of Computer Science
George Mason University
Fairfax, VA 22030
sqchen@cs.gmu.edu

Abstract—

Millions of residential users are widely served by cable or DSL connections with modest upload bandwidth and relatively high download bandwidth. For the increasingly important and demanding P2P applications such as VoIP, BitTorrent, and Internet streaming, stable or high upload bandwidth is required. Inadequate upload bandwidth degrades the performance of these applications among residential users. On the other hand, our Internet measurements show that plenty of idle upload bandwidth (from 50% to 80%) is always available in a local residential network. Based on this observation, we propose a system prototype to Coordinate Upload Bandwidth Sharing (CUBS) among neighboring residential users. Specifically, the idle upload bandwidth of neighbors can be used upon a request from a demanding user. Since it has become a common practice to deploy wireless access points in a residential user's home, we have built CUBS by leveraging the support from the wireless networks. In CUBS, to discover and manage idle bandwidth, a localized overlay is constructed by the cooperative users. CUBS is application independent as the bandwidth sharing is implemented at the network layer. CUBS is also ISP transparent because the sharing of neighbors' bandwidth does not demand any additional bandwidth supplies. We have evaluated the CUBS system prototype with experiments in Internet. The experimental results demonstrate that CUBS can effectively improve the performance of upload intensive applications by more than 30%.

I. INTRODUCTION

With the significantly improved transmission speed and falling prices, residential users have widely subscribed to broadband services, such as cable or DSL, for Internet accesses. Recent studies by Internet World Stats [1] show that there are over 70% Internet users of U.S. population, among which the number of broadband subscribers has passed 60% in November 2005 [2], and this number keeps increasing. In addition, residential users commonly have the 802.11 WLAN deployed at home to access the Internet. For example, Jupiter Research estimates 65% of the U.S. households use Wi-Fi access points [12].

A. Problems Caused by Limited Upload Bandwidth

For a residential user using broadband connections, the download and upload bandwidth limits are normally asymmetric, with a substantially lower upload bandwidth. For example, a subscriber to DSL service can pay a flat fee to get up to 1.5 Mbps download bandwidth, but only up to 384 Kbps upload bandwidth. While many residential users are satisfied with the fast download speed offered by the broadband services for their download-dominant Web accesses, more and more residential users participate in increasingly demanding and dependable P2P-based applications that prefer equal upload to download bandwidth in principle. For example, BitTorrent, the traffic of which accounts for more than 50% of Internet traffic, applies a "tit-for-tat" policy to encourage each user to contribute to other users by uploading content [16], and a peer that could not upload sufficiently fast would suffer slow download. Similarly, in P2P-based Internet live streaming applications, the slow upload to other users leads to significant streaming quality degradation. For a Skype conference call, the conference host is required to relay packets for other conference attenders, which also demands substantial and stable upload bandwidth.

B. Would An Upload Bandwidth Increase Effectively Address the Problems?

To answer the above question, our study starts by investigating upload bandwidth availability and distribution patterns in residential networks. Having observed 25 residential networks connected through large broadband service providers by active probing from Planet-Lab for 21 days, we have had two main findings summarized as follows. First, compared with the download channel utilization (most users utilize the download bandwidth fully and frequently), the upload channel utilization is much more bursty and unbalanced. For example, less than 20% users highly demand upload bandwidth in the cable network. This confirms the reported P2P user behavior [28]: only a small percentage of peers make contributions in P2P communications by uploading or relaying contents to others. Second, we find that there always exists idle upload bandwidth in a residential network. Overall, about 50% to 80% upload bandwidth is idle in our observations. These two findings imply that even if ISPs provide more upload bandwidth for each user or reset the upload and download bandwidth limits equally, it may still not satisfy the increasing demand of active P2P users while more idle upload bandwidth in the network would be underutilized, or the download channel would be more congested. Thus, to simply increase the upload bandwidth for each individual user is not a cost-effective way to address the problems.
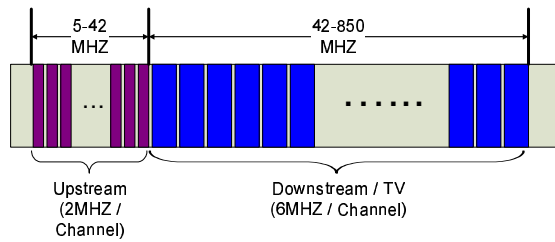
Fig. 1. Cable Network Bandwidth Allocation (U.S.)

C. Our Solution: Coordinated Upload Bandwidth Sharing (CUBS)

A major objective of this study is to enable the Internet to provide residential users with demanded upload bandwidth for their P2P related applications without demanding additional bandwidth or infrastructure support from ISPs. Motivated by our measurement results, we have built a system to Coordinate the Upload Bandwidth Sharing (CUBS) in a residential network. Since residential users commonly have wireless access points deployed at home for Internet accesses, CUBS is built with the support of wireless accesses. In CUBS, an overlay network is formed to monitor the availability of the upload bandwidth from all members. Once a user is unsatisfied with the upload bandwidth from its local Internet access, she can utilize the idle upload bandwidth from the neighboring networks via wireless accesses. Such bandwidth sharing is fair and flexible via our distributed idle bandwidth discovery and management mechanisms. CUBS has two merits. First, it is application independent as the bandwidth sharing is implemented at the network layer. Second, CUBS is ISP transparent as the sharing of neighbors' bandwidth does not demand additional bandwidth supply or infrastructure support from ISPs. We have comprehensively evaluated our implemented CUBS prototype. The experimental results demonstrate that CUBS can significantly improve the performance of upload intensive applications without affecting the performance of neighboring networks.

The remainder of the paper is organized as follows. We introduce the background of residential networks in section II and present our Internet measurement results in section III. We describe the design of CUBS in section IV. Based on the implemented prototype, we evaluate CUBS in section V. Related work is presented in section VI and we make concluding remarks in section VII.

II. BACKGROUND

Many applications, such as VoIP, P2P-based Internet streaming, BitTorrent-like download, heavily rely on user's upload bandwidth to obtain satisfied quality of service. In this section, we first briefly overview the basic background regarding bandwidth caps in residential network, and then present a set of experiments to demonstrate why insufficient upload bandwidth and congested upload channel can significantly degrade the performance of these applications perceived by end users in residential networks.

A. Bandwidth Allocation in Residential Networks

In order to have high speed accesses to the Internet, most residential users today are using cable or DSL modems connecting to ISPs. Commonly, for a Internet subscription in residential networks, the download bandwidth a user gets is always much larger than the upload bandwidth (For example, 7 Mbps / 512 Kbps in a cable network) since normal residential users usually download more contents than that they upload. ISPs may also provide symmetric upload/download rate subscriptions for business users with a higher price. Figure 1 shows the typical bandwidth allocation for the upstream and downstream (or TV) channels in U.S. cable network [3]. The upstream channel is 2 MHz, ranging from 5 to 42 MHz, while the downstream channel is much larger (6 MHz), and has a much wider range to select (42 to 850 MHz) besides the allocated channels for TV programs.

A downstream or upstream channel is shared within a residential neighborhood through setting download and upload bandwidth caps over the cable modem by the ISP. Bandwidth caps are used by ISPs in order to limit the speed of Internet accesses. The total physical speed for the downstream or upstream channels is much larger than the allocated download or upload speed for each end user. For example, DOCSIS 1.x [4] can provide 38 Mbps downstream speed and limits the upstream speed to 9 Mbps for each channel. For DSL networks, the upstream channel is not shared among neighbors, but the ISPs also have bandwidth cap settings in order to provide a similar access speed to end users because the distance of the DSL modem to the DSL Access Multiplexer (DSLAM) affects the physical transmission speed substantially (ranging from 800 Kbps to 25 Mbps).

Such a bandwidth limit policy thus leads to the following unbalanced demand/supply situation: when most users are not uploading, active users cannot use more than their upload caps although there is idle upload bandwidth available in the same network or neighboring networks. This situation has caused several performance problems as we demonstrate in the next subsection. As the bandwidth cap is often implemented with token bucket, one might suggests to increase the token bucket size so that more upload bandwidth can be utilized for a longer duration by active users. However, this approach is infeasible at all in practice because most uploads last for hours while the token bucket size cannot be that large.

B. Problems in Residential Networks

In this subsection, through experiments, we quantitatively study existing problems of delay-sensitive applications and P2P applications in residential networks.

1) Congested residential upload channel directly deteriorates delay-sensitive application's performance: VoIP is a dependable service on the Internet today. VoIP applications are very sensitive to end-to-end transmission delay (for example, one-way transmission time should be less than 150 ms in order to have satisfactory conversation quality [8], [27]). For VoIP applications in residential networks, the upstream voice packets can be queuing in the long queue of cable/DSL modem [17] and result in a large transmission latency when
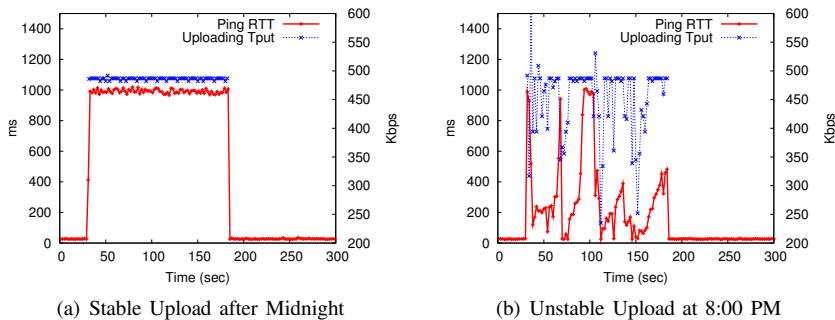
(a) Stable Upload after Midnight

(b) Unstable Upload at 8:00 PM

Fig. 2.   Round Trip Time Varies along Upload

Fig. 3.   BitTorrent Download Time Decreases with Larger Upload Bandwidth

the upload bandwidth is fully occupied by other applications such as P2P streaming. That is, when a user in a home network runs a VoIP client, the upload traffic from other active applications on the same computer or from other users in the same network, can easily delay the user's upstream voice packets due to the congestion in upload channel.

To demonstrate how upload traffic can deteriorate the voice quality, we experiment by using a ICMP `ping` session together with an Iperf [5] upload session on a computer directly connecting to a cable modem. The cap of upload bandwidth for this subscription is 500 Kbps. The Iperf session uses the TCP protocol to continuously generate upload traffic. We run the experiments at different times of a day. Figure 2 shows the ping round trip time along with the upload throughput. On Figure 2, the left y-axis represents the round trip time in milliseconds, while the right y-axis represents the upload bandwidth in Kbps. As shown by Figure 2(a), for a relatively stable upload session that is performed after midnight when most neighboring users are inactive, the ping round trip time increases from 28 ms to 1,000 ms steadily after the Iperf session starts to upload. However, a round trip time larger than 300 ms is not acceptable for VoIP sessions. We also conduct the same experiments at 8:00 PM in the evening (peak time). Figure 2(b) shows that when the achievable upload rate is varying due to the competition from other users from the same residential network, the ping round trip time varies significantly, resulting in a larger standard deviation (jitter), 274 ms in this case. In fact, when the upload bandwidth is highly demanded during peak time, an upload session is often unstable and thus further deteriorates the VoIP quality by increasing the transmission jitter.

2) Insufficient residential upload bandwidth hurts the performance of P2P applications: For most P2P applications, users are encouraged to upload to other users in order to improve the scalability and download performance. The more a user contributes (by uploading), the more likely the user gets a better service quality. For example, in the widely used BitTorrent protocol, a tit-for-tat mechanism is engaged in order to suppress free riders [16], [20]. Although such a design provides an effective incentive mechanism, for residential BitTorrent users, this has posed a problem due to the insufficient upload bandwidth. As a result, the perceived download performance by residential users could have been constrained by the cap of upload bandwidth.
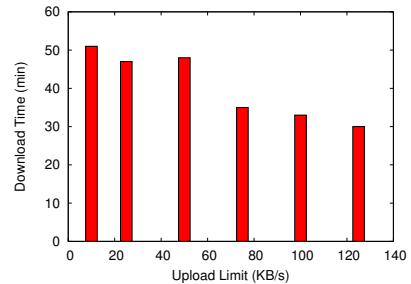
To study the effect of insufficient upload bandwidth to download performance, in our experiments, we run a BitTorrent client by restricting its upload bandwidth to a gradually increasing threshold. As we can observe from Figure 3, the download time generally decreases when the upload bandwidth threshold increases. The trend is not strictly linear, since for the torrent we tested, there existed more than 2,000 seeds (totally about 8,000 peers) that could help increase the download speed for peers with low upload bandwidth. In general, a slow upload speed results in being penalized for download. In [26], the authors also show that the download speed increases linearly when the upload capacity is less than 200 KB/s (and sub-linear growth after that).

III. USERS' UPLOAD BANDWIDTH USAGE

In order to investigate the upload bandwidth usage of broadband network users, we have collected a large number of IP addresses of P2P users, by using a Gnutella network crawler. We then reversely map each IP address to its corresponding domain name, and identify the top 7 ISPs presented in these domain names: Charter, Comcast, Cox, Road Runner, Ameritech, Pacbell, and Verizon. The first four provide cable services while the rest three provide DSL connections. Then we randomly pick up 3 to 4 IPs from each ISP, and scan the subnet corresponding to the IP address. In order to detect available upload bandwidth of an end host, we send ICMP echo request ping packets to the end host that can respond with ICMP echo reply packets [13], [17]. For the subnets we scanned, the responding hosts account for up to 78.8% of the subnet IP range. We saturate the upload channel of the destination by sending probing packets every 10 ms (or smaller for hosts with higher upload bandwidth) for a very short duration (typically 5 seconds), so that we can estimate the available upload bandwidth by observing the number of losses in the received packets. The packet size is 1,500 bytes. The measurement is repeated every 30 minutes for each host in a subnet.

Because ICMP ping packets directly compete with the existing upload traffic, the raw estimation of available bandwidth based solely on probing packet losses may overestimate the available bandwidth. To investigate the accuracy of the raw estimation method, we did Internet experiments by tuning the available upload bandwidth of the probing target host in our control. In the experiments, the available upload bandwidth is
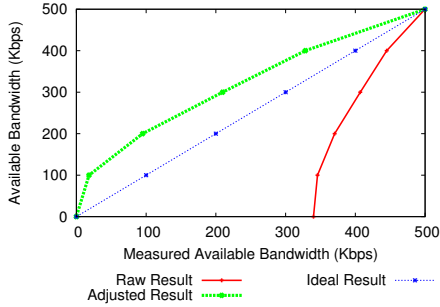
Fig. 4. Illustration of Our Available Bandwidth Estimation Adjustment Method

ISP	# of Measured IPs	Median Upload Capacity (Kbps)	Average Idle Upload Bandwidth (%)
Charter	70	487	62
	66	261	93
Comcast	63	1578	64
	52	1593	75
Cox	36	544	69
	32	536	71
Road Runner	35	444	77
	34	359	72
Ameritech	159	343	52
	79	352	57
PacBell	146	377	58
	148	352	57
Verizon	193	558	59
	16	379	83

set to 0, 100, 200, 300, 400, 500 Kbps. The distribution of the raw estimation results shown in Figure 4 (rightmost curve) confirmed that the raw estimation method *overestimates* the available bandwidth, especially when the available bandwidth is not sufficient. On the other hand, the experiments results show that the raw estimation results have a close to linear distribution. Based on this observation, we propose a raw estimation adjustment method by linearly mapping the raw estimation results from the numerical value range of $[min, max]$ to $[0, max]$. min and max are the minimum and maximum values of the raw estimation results, which can be obtained when the measurements are repeated for a number of times. The leftmost curve in Figure 4 shows the distribution of our adjusted estimation results, which sugguests that the adjusted estimation method is much more accurate than the raw estimation method. In fact, our adjusted estimation results slightly *underestimate* the available bandwidth, which implies that our available bandwidth estimation method is on the *conservative* side.

We have probed 25 subnets with 2,040 IP addresses in 21 days (during February and March, 2008) altogether, with 23 senders distributed over Planet-Lab, and 2 senders locally. Table I summarizes the statistics through our measurements for the 14 subnets in a period of 21 days. For each ISP, we present the results of 2 subnets. In this table, the last column shows the average idle upload bandwidth (in percentage of the median upload capacity) during the probing period. Compared with cable services, DSL services have a higher upload bandwidth utilization. In the following, due to the page limit, we present the details of 6 subnets, three from cable and three from DSL, starting from cable service providers.

Figure 5(a) shows the *average* and the *standard deviation* of the *available* upload bandwidth of a Comcast subnet. There is an amount of 64% of upload bandwidth available on average for this Comcast subnet, and the average available bandwidth shown on Figure 5(a) does not vary too much from day to day. Figure 6(a) further shows the CDF (Cumulative Distributed Function) of the average available upload bandwidth of each host, and the corresponding minimum and maximum available upload bandwidth of that host. The figure confirms that less than 20% of the hosts in this subnet use more than half of the upload bandwidth on average. Also, note that for users of Comcast, the upload bandwidth of some subscribers could reach 1.6 Mbps to 2.4 Mbps.

Figure 5(b) shows that in a Road Runner subnet, 77% of upload bandwidth is unused on average. The largest cap of the upload bandwidth for users in this subnet is around 480 Kbps. Correspondingly, Figure 6(b) demonstrates that less than 10% of the hosts use more than half of the upload bandwidth on average.

Figure 5(c) shows that in a Cox subnet, there is an amount of 69% of upload bandwidth unused on average. Notice that although the largest upload cap for users in this subnet is around 550 Kbps (comparable to that in the Road Runner subnet), the available bandwidth changes significantly from time to time. Consider what we have observed from Figure 5(a) and 5(b), we tend to believe that users in this subnet is more active than users in other subnets.

For DSL users, Figure 7(a) shows the available upload bandwidth among the active hosts in an Ameritech subnet during 21 days. As we can see, the average available upload bandwidth shows little variation over time, and it is about 52% of the maximum upload capacity, which is quite stable across the 21 days we measured.

Figure 8(a) further shows the CDF of the average available upload bandwidth of each host, and the corresponding minimum and maximum available upload bandwidth of that host. The average idle upload bandwidth varies from each other for the measured hosts. Our conjecture is that for DSL networks, the different distance to the telephone company leads to different physical upload (and download) speed for each host.

We perform similar experiments for other residential networks serviced by DSL. Figure 7(b) and Figure 7(c) show the corresponding results in a Pacbell subnet and a Verizon subnet. The average available upload bandwidth is 58% for Pacbell, and 59% for Verizon. We can also observe staircases in Figure 8(c), which suggests that subscribers probably have paid different fees to subscribe for different upload (and download) rates in DSL as well. In this Verizon network, the upload caps are around 350 Kbps and 750 Kbps.

Figure 9 shows the cumulative distributions of average probing ping round-trip time (RTT) for the measured residential networks. This RTT distribution only reflects the RTT when
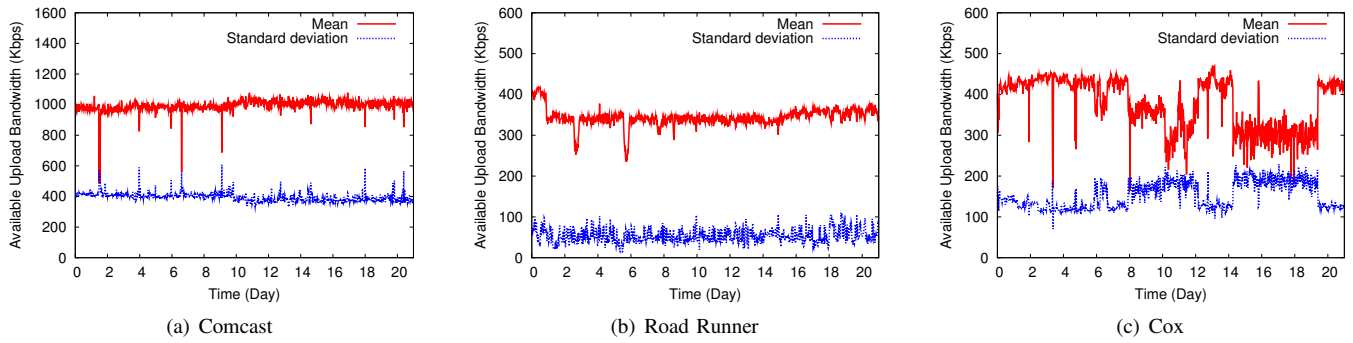
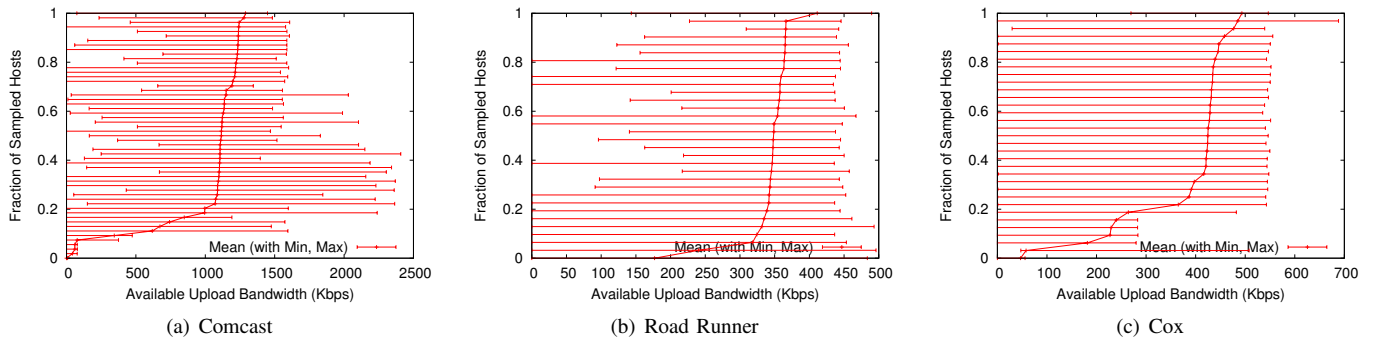Fig. 5. Cable: Available Upload Bandwidth during Measurement Period (day)

(a) Comcast

(b) Road Runner

(c) Cox

Fig. 6. Cable: Average Available Upload Bandwidth for Each Host (CDF)

(a) Comcast

(b) Road Runner

(c) Cox
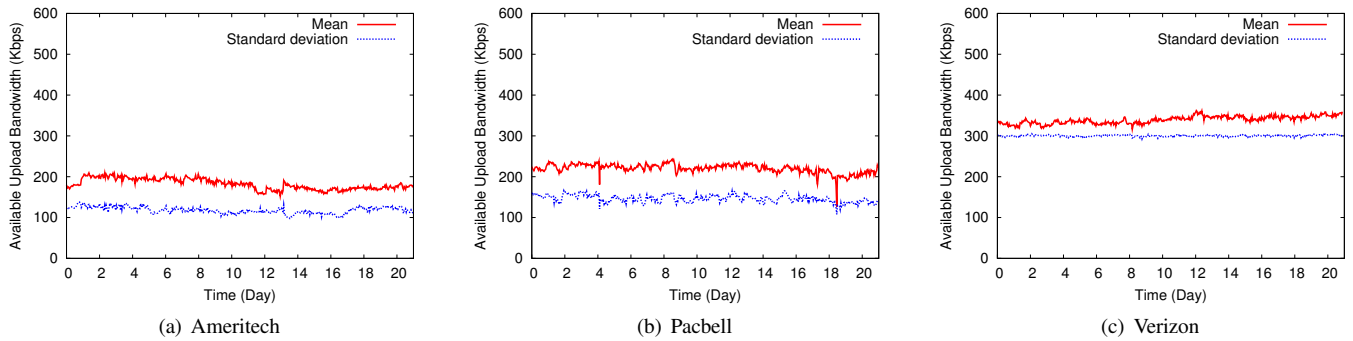
Fig. 7. DSL: Available Upload Bandwidth during Measurement Period (day)

(a) Ameritech

(b) Pacbell

(c) Verizon
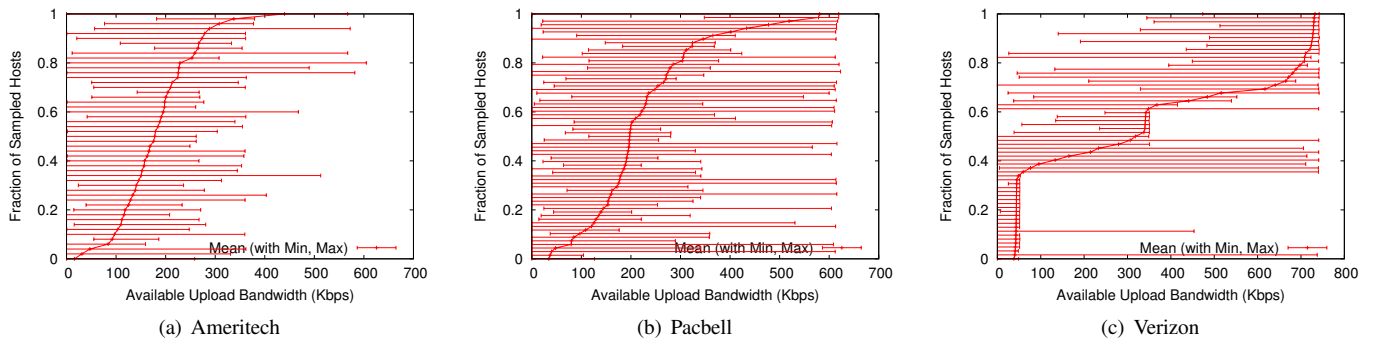
Fig. 8. DSL: Average Available Upload Bandwidth for Each Host (CDF)

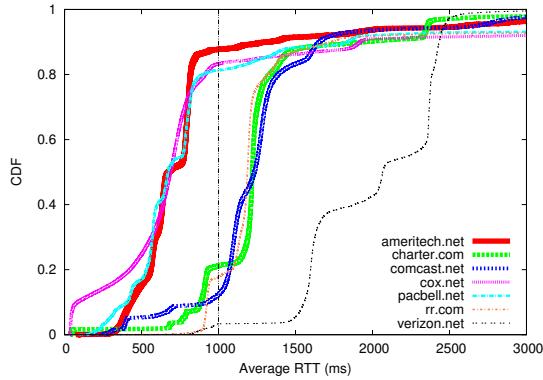(a) Ameritech

(b) Pacbell

(c) Verizon

Fig. 9. Probing Ping Round-Trip Time (Only for Saturated Upload Channel)

the upload channel of the probing host is saturated because of our probing packets. As shown in the figure, in most of the networks, the round-trip time is larger than 1 second, which suggests that the lengths of downstream/upstream queues of the residential networks are very large [17]. These results validate the result shown in Figure 2, that is, the congested upload channel of residential networks can lead to high upstream latency and variation, which will degrade the performance of delay-sensitive and/or throughput-sensitive applications.

In summary, through a 21 day measurement study of 25 residential networks supported by top 7 ISPs in United States, our measurement results show that: *(1) In broadband networks, there exists plenty of unused upload bandwidth, ranging from about 50% to 80% of the entire upload capacity on average; (2) For the cable networks we have observed, less than 20% users have been fully utilizing their upload bandwidth, which suggests that the upload channel is utilized in an unbalanced way.*

IV. SYSTEM DESIGN

While various applications, particularly P2P applications, may have been adversely affected by the limited residential upload bandwidth as we have shown in section II, the measurement results in last section show that there is a significant portion of upload bandwidth available in a residential cable or DSL network on average. These observations motivate us to leverage such idle upload bandwidth to respond to the increasing demand of P2P applications. For this purpose, we design CUBS, a scheme for *Coordinated Upload Bandwidth Sharing* in a residential network. CUBS is application independent as the bandwidth sharing is implemented at the network layer.

A. An Overview of CUBS

CUBS leverages existing widely deployed WLANs in a residential network (including cable and DSL connections) for upload bandwidth sharing without demanding any infrastructure support. Today, in a residential network, most subscribers use a wireless AP (access point) to provide Internet accesses for multiple computers at home. In this work, we call this AP as the user's local AP. In addition to the local AP, a wireless station can often connect to a few neighboring APs. A previous study in 2006 shows that the number of wireless APs that can be detected in residential areas ranges from
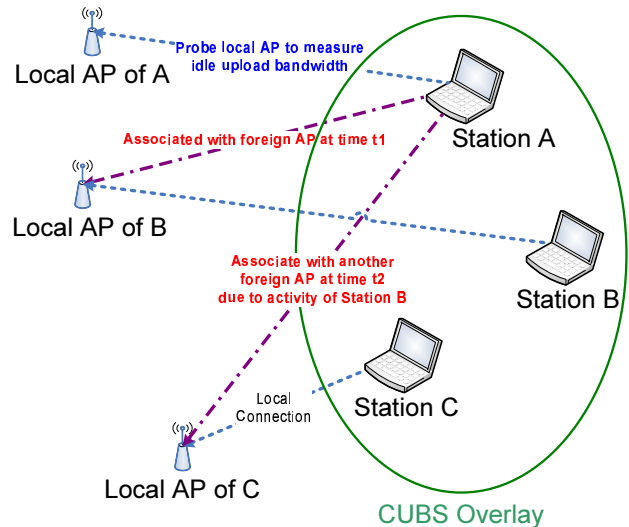

Fig. 10. Illustration of CUBS system

2 to 20 [12]. CUBS thus leverages this fact to effectively share upload bandwidth among residential neighboring users via wireless networks. In CUBS, we assume each station that wants to participate in CUBS bandwidth sharing has two network interfaces: one wired/wireless interface used to connect to its local AP, and another wireless interface used to connect to neighboring APs.

To facilitate the bandwidth sharing, a CUBS overlay is formed by the participating stations using their local APs. Via this overlay, each station can exchange information including the bandwidth availability, network performance, and access credentials of its local AP. With the help of CUBS overlay, a station demanding additional upload bandwidth can easily find accessible neighboring APs that have idle bandwidth by querying on the overlay. By connecting to the local AP via one interface and associating with the most idle neighboring AP (called *foreign AP*) when possible via another interface, a station will have accesses to possibly doubled upload bandwidth with these two interfaces. Notice that CUBS can share the upload bandwidth from the same ISP or different ISPs (multihoming) as long as there is idle bandwidth from neighboring APs.

To avoid aggressively using neighboring AP's upload bandwidth, CUBS restricts the use of upload bandwidth on the neighboring AP when it detects competitions between itself and the neighboring AP owner. To avoid competitions with the download traffic of the neighboring AP owner, CUBS restricts download dominant connections to use the local AP only based on traffic direction prediction.

Figure 10 illustrates the typical operations of a CUBS system. Stations A,B,C form the CUBS overlay. Station A tries to use the neighboring local AP of Station B at a time, and later associates with the local AP of Station C when Station B actively uses its upload bandwidth. We will explain details in the following subsections.

B. CUBS Overlay Operations

In principle, the CUBS overlay works like common P2P networks, but with the unique purpose of sharing AP information instead of files. The main challenge for the CUBS

Procedure 1 CUBS Workflow

---

global station S; /* local station */

daemon()
1: /* periodical operations */
2: S.localAP.avail-upload-bw ← S.localAP.upload-capacity - (sum of upload bandwidth usages of S.localAP)
3: S.neighborAPs ← scanned neighboring APs with acceptable signal strength and wireless bandwidth
4: **if** number of active S.neighborNodes < Threshold-num-overlay-neighbors **then**
5:     update-neighbor-nodes()
6: **end if**
7: S.foreignAP.avail-upload-bw ← update from S.foreignAP owner
8:
9: /* operations triggered by conditions */
10: **if** S.foreignAP owner starts exclusively using S.foreignAP **then**
11:     associate-with-neighbor-AP()
12: **end if**
13: For a new connection, predict its upload-to-download traffic ratio. Put download dominant connections to local connectivity and route other connections to available connectivities based on traffic load.

update-neighbor-nodes()
1: /* Query tracker */
2: Query tracker with the detected AP-IDs of S.neighborAPs
3: S.neighborNodes ← list of stations using S.neighborAPs
4:
5: /* Connecting to neighbor nodes */
6: **for** sta in (S.neighborNodes) **do**
7:     **if** sta has connection with S **then**
8:         continue
9:     **end if**
10:     **if** sta has public IP or port forwarding on NAT **then**
11:         Establish bidirectional connections with sta
12:     **else**
13:         Use tracker to relay messages
14:     **end if**
15: **end for**

associate-with-neighbor-AP()
1: max ← 0
2: **for** ap in (local AP of S.neighborNodes) **do**
3:     **if** (ap is in S.neighborAPs) AND (S has access on ap) **then**
4:         **if** idle upload bandwidth of ap > max **then**
5:             max ← idle upload bandwidth of ap
6:             new-assoc-AP ← ap
7:         **end if**
8:     **end if**
9: **end for**
10: S.foreignAP ← new-assoc-AP

---

overlay is to update/query the sharing information at run time. In fact, CUBS has an inherent localization characteristic: each station is only interested in the information of the neighboring APs, which implies that each station will be only interested in exchanging information with geographical nearby stations connecting to the neighboring APs. Because of this localization property, CUBS is designed as an unstructured P2P overlay in order to minimize the query hops for performance optimization.

Bootstrap For a new node joining CUBS overlay, it needs to bootstrap and find other nodes. To speed up this process, CUBS uses a tracker to record the local AP-ID and IP address of each overlay node in a cache. The local AP-ID is the unique identity of the node's local AP, which is a pair of ESSID and AP MAC address.

Overlay Connections Upon joining the CUBS, the new node queries the CUBS tracker with the detected AP-IDs of

neighboring APs, and the tracker returns the IP addresses of the nodes associated with the AP-IDs if it hits in the cache. After that, the new node connects to the neighboring nodes so that they can exchange AP information directly as neighbors in the CUBS overlay. For stations without port forwarding enabled on NAT, the tracker can be used to relay messages. After the establishment of the neighbor connection, if either node can not detect the local AP of the other node, the connection will be terminated, in order to prevent a node joining CUBS with no local AP to share with other nodes.

Idle Bandwidth Discovery In order to enable idle upload bandwidth discovery and management, each CUBS node needs to periodically (for example, 30 seconds) measure the upload bandwidth rates of itself on its local AP and neighboring AP being used, and collects the upload bandwidth rates from other CUBS users of its local AP via CUBS overlay neighbor connections. Then the node can calculate the available upload bandwidth by subtracting the sum of the upload rates from the upload capacity of its local AP. The calculated result serves as an important metric for CUBS neighbors to discover idle upload bandwidth for sharing.

Failure Recovery If a node gets offline or has a power failure, while its local AP is still alive, the nodes using its local AP will continue their usages. Meanwhile the other nodes will not be able to discover this AP and share its upload bandwidth until the AP owner returns to CUBS overlay. When the local AP itself fails, the related nodes notice the failure and will attempt to associate with another neighboring AP for bandwidth sharing.

C. CUBS Node Operations

With the help of CUBS overlay, a newly joined CUBS node can choose the neighboring AP with the highest available upload bandwidth and an acceptable signal strength for sharing upload bandwidth. In order to associate with the neighboring AP, this node will request the chosen neighboring AP owner to grant accesses for the AP. After associating with the chosen neighboring AP (foreign AP), the node grows its upload bandwidth by branching its upload traffic to the local AP through one network interface and the foreign AP through another wireless interface.

To utilize the idle bandwidth discovered via CUBS, for a new TCP connection initiated by the node, CUBS modifies the `connect` system call and binds the connection to one of the available network interfaces based on the traffic load. Because most of the UDP traffic is for VoIP or other delay-sensitive applications, CUBS node simply routes the UDP packets to the interface connecting to its local AP to minimize the delay.

D. Fair Usage of the Upload Bandwidth of Foreign AP

A CUBS node carefully controls the usage of the upload bandwidth of the foreign AP in order to prevent the degradation of the upload performance for the foreign AP owner. For example, a node can engage `Traffic Control (tc)` [6] to limit the usage of the upload bandwidth to 70% of the upload capacity on the foreign AP.

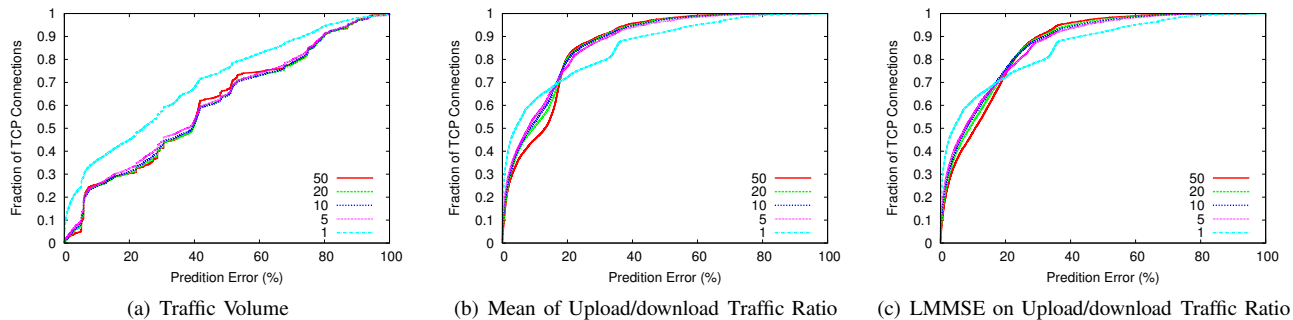| (a) Traffic Volume | (b) Mean of Upload/download Traffic Ratio | (c) LMMSE on Upload/download Traffic Ratio |

Fig. 11.   Prediction Difference Distributions for Different Prediction Methods

More importantly, in order to let the foreign AP owner be able to fully use the upload bandwidth when it becomes active, CUBS needs to detect this situation and responds quickly. As a rule of thumb, if the owner starts to actively use the upload bandwidth and detects that the available upload bandwidth of the AP becomes scarce, the owner can conclude that there might exist bandwidth competitions on the AP, and then signals other users of the AP to refrain from competitions. Such a user then decreases its usage to the half of its current usage, and observes if the owner still signals for further actions. If so, it restrains further usage of the foreign AP, and tries to find other CUBS neighboring APs for upload bandwidth sharing. That is, CUBS initiates hand-off only when it is necessary to guarantee the AP owner's performance. Notice that if two CUBS nodes are competing on a foreign AP while none of them are owners, the two nodes can detect this situation and continue their usage since the owner is still idle and is able to provide sharing bandwidth.

Fair bandwidth usage also serves as an incentive mechanism to encourage residential users to share bandwidth over CUBS, since the sharing user can still get its full upload bandwidth when it is active. The next subsection details on how to enable the sharing to be fair on the download side.

E. Prediction-based Download Bandwidth Utilization of Foreign AP

For a new TCP connection, if it is upload intensive or bidirectional, the connection can be bound to the interface associated with the foreign AP. Otherwise, this connection should be bound to the interface connecting to local AP, in order to avoid excessive usage of the foreign AP's download bandwidth. However, this requires prediction of a connection's upload to download traffic ratio before it is established. In order to do so, we have designed and evaluated several history-based prediction methods.

We first analyze the packet header trace of Dartmouth residential wireless network (in Apr 2004) [23], and find that for all the TCP connections to a remote host, the upload-to-download traffic ratio varies within a small range. Then we did trace-driven evaluations for several prediction methods. The first method predicts upload-to-download traffic ratio based on the total upload and download traffic of latest n connections to the same remote host. n is the history size, which we set to 1, 5, 10, 20, and 50 in our evaluations. The second method uses the mean of the upload-to-download traffic ratios of the latest n connections for prediction. The third method is based on LMMSE [18] estimation for the upload-to-download traffic ratios of the latest n connections. LMMSE is used for Internet traffic prediction.

Figure 11 shows the distributions of the difference of predicted ratio to real ratio (the smaller the better) for the three methods. As shown on Figure 11, the second method is simple, but is much more accurate than the first method: for more than 80% connections, the difference is less than 20%. The third method is more complex, however, achieves comparable performance to the second method. And when the size of history increases above 1, the difference distributions are quite close to each other.

With all the above considerations in designs, Procedure 1 shows the workflow for our CUBS system. Each station runs a daemon for background operations. The daemon also dynamically determines the traffic routing through the available network interfaces.

F. Discussions

In addition to the basic operations implemented in our prototype system, the following features can also be applied in CUBS for better performance or security considerations.

Overlay-controlled User Authentication: In order to prevent any unauthorized usage of a shared AP in CUBS, the owner of the shared AP can enable/disable the association of a guest node by controlling the allowed MAC addresses using the AP through the web service provided by the AP. The owner adds the MAC address of an overlay neighbor to the access control list when the connection to the neighbor is established, and removes the corresponding MAC address upon the departure of a neighbor. The AP owner can also use the access control list to avoid abusive usage of the shared bandwidth by removing malicious neighbors.

Multiplexing One Wireless Interface to Use Multiple APs: When a user only has one wireless network interface, it can still share bandwidth in the CUBS overlay with the help of FatVAP-like [22] multiplexing mechanism. FatVAP allows one wireless interface to connect to multiple APs at the same time by fast switching among APs, thus can be useful for CUBS users with only one wireless interface to share upload bandwidth among neighboring APs.

Mesh Networks for Sharing Long Distance APs: CUBS node currently only connects to neighboring APs to harness

TABLE II
BitTorrent Download Experiment on Planet-Lab

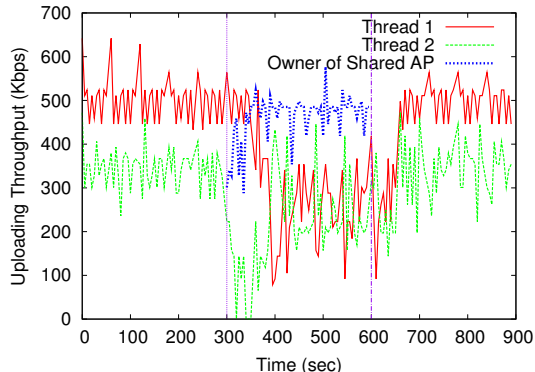| Balance Method	Download Bytes (MB)	Upload Bytes (MB)
1 Interface	261.2	199.0
Load-based	345.5	392.9
Hash-based	342.9	388.2

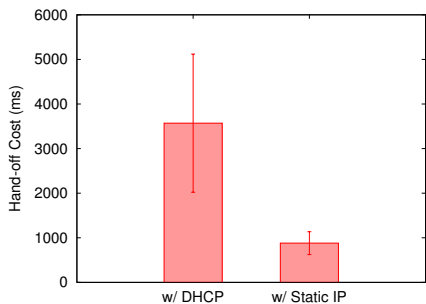Fig. 12. Idle Bandwidth Sharing on CUBS Overlay

Fig. 13. Hand-off Cost Comparison

more available bandwidth. In fact, CUBS nodes can form a wireless mesh network and route a upload packet to the Internet via an AP which can be far away from the originator of the packet. With the mesh network, upload bandwidth sharing is no longer limited among neighboring CUBS nodes, but can be shared among any nodes in the mesh network.

V. Performance Evaluation

We evaluate CUBS prototype based on the setup shown in Figure 10. To emulate the residential network, we use NISTNet [1] to control the Internet connections of the APs. We configure the upload and download capacity of the emulated residential networks to be 500 Kbps and 7 Mbps, which are typical bandwidth caps for a cable subscription.

We first evaluate BitTorrent download with only one interface connecting to the local AP, or with two interfaces connecting the local AP and a CUBS neighboring AP at the same time. The client we use is the original BitTorrent client (Linux version). We deploy 1 seed and 18 peers over Planet-Lab. The download lasts for one hour. For the download with multiple interfaces, we try to balance the traffic in Linux with the load-based balancing mechanism supported by the kernel through routing table configuration [7]. We also

[1] http://www-x.antd.nist.gov/nistnet/

implement hash-based mechanism [19] to balance the traffic by intercepting the `connect` system call and binding the new TCP connection to different interfaces according to the hash value of TCP destination. The results in Table II indicate that both methods can increase the average upload throughput. As a result, the average download throughput is improved by more than 30% compared with only using one network connection.

To evaluate the overlay functions on available upload bandwidth discovery and monitoring, as well as demonstrate the fair usage of foreign AP's upload bandwidth, we conduct the following experiments. Station A accesses both its local AP and a CUBS neighboring AP owned by station B (B-AP) at the same time. Station A restricts the utilization of B-AP's upload bandwidth up to 70% of its capacity with `tc`. From the overlay, station A gets the status of idle upload bandwidth on B-AP detected by station B. We run two upload threads with Iperf at station A periodically for 900 seconds. Meanwhile, station B is idle for the first 300 seconds and then starts uploading for a duration of 300 seconds. The result in Figure 12 shows that it takes station B about 40 seconds (since the idle upload bandwidth is detected every 30 seconds) to upload at a full speed after station A finds it is competing with the owner of B-AP and then ceases to utilize B-AP (notice that during this competing period, station A doesn't occupy but share the upload bandwidth with station B). After station B finishes uploading, station A quickly detects the available bandwidth and re-utilizes B-AP again.

We also evaluate the cost of hand-off among shared APs in CUBS. We periodically switch between two APs while using `ping` to observe packet losses. Each run has 20 hand-offs. As shown in Figure 13, when we use DHCP to get a new IP address in a newly associated AP, the average cost (duration of packet losses) is 3.5 seconds with 1.5 seconds standard deviation. Instead, when we use a static IP address, the cost is reduced to less than 1 second. This result suggests we can decrease the hand-off cost by caching the allocated DHCP addresses for each AP, and reusing the cached IP address to re-associate with the corresponding AP. The chance of IP address conflicting with existing stations is low because the leased DHCP address is often the same for the same MAC address.

VI. Related Work

Researchers have made efforts to understand the similarities and differences between the broadband networks and other networks at different layers. Claypool et al. used 47 broadband hosts to study the broadband access queue sizes [15]. Lakshminarayanan et al. measured the TCP throughput and latency from 25 broadband hosts to other hosts [24]. Sinha et al. showed that P2P applications created most of the upstream traffic in broadband networks [29]. A recent study performed a large-scale measurement of 1894 broadband hosts from 11 cable and DSL providers in North American and Europe [17]. Consistent with previous findings, work [17] confirmed that many cable links show high variation in link bandwidths over short timescales and packets transfer over cable suffer high jitter as a result of cable's time-slotted access policy while DSL links show large last-hop delays. Work [14] characterized

the traversing traffic by residential customers. Balakrishnan et al. [11] observed poor TCP download performance due to limited upstream channel bandwidth and suggested ACK congestion control and ACKs-first scheduling on the router.

Li et al. [25] studied the access point queueing size as residential broadband users commonly have 802.11 networks deployed at home. Work [30] characterized upload traffic for P2P applications in a WLAN. Multihoming has also been studied to effectively utilize the limited bandwidth in small enterprise or residential networks [31], [19], [9], [10], [21]. Guo et al. considered practical multihoming for small enterprise through using DSL or cable connectivities at the same time with a network device performing NAT and dynamic DNS resolution [19]. Thompson et al. proposed PERM [31] for end users to leverage neighbor's WLAN to improve the Internet connectivity. PERM focused on scheduling flows based on predicting flow round-trip time or traffic volume, for example, for web browsing or large file downloading flows. In our CUBS system, flow scheduling is not based on flow traffic volume prediction, since our target is on P2P upload traffic, which is usually difficult to be predicted. Kandula et al. proposed FatVAP [22] to aggregate available bandwidth from accessible APs with only one wireless card by utilizing Power-Saving Mode and time-switching among associated APs. FatVAP focused on increasing the download bandwidth for a wireless station with little considerations on the traffic direction and the fair usage of the shared APs. In contrast, our work focuses on mitigating the bottleneck effect of upload bandwidth for both upload and download applications in the last mile among residential users, which is a serious and timely issue for various applications.

VII. Conclusion

The pervasive usage of cable or DSL broadband networks in residential areas effectively benefits a lot of people by relieving the last mile access problem. However, the limited upload bandwidth allocation at each subscriber has shown to be problematic to the performance of various delay-sensitive or P2P applications. While the upload demand from a subscriber is frequently unsatisfied due to a low upload bandwidth cap, in a residential network, the total upload bandwidth offered by its ISP is often significantly underutilized. Therefore, in this paper, without demanding additional bandwidth supply from ISPs or infrastructure support, we present CUBS to enable a residential user to share available idle upload bandwidth of its neighbors in a coordinated manner to improve the performance of various applications. CUBS flexibly shares the idle upload bandwidth while providing fairness guarantees to all users in the same residential network. Our experiments based on a prototype implementation demonstrate the effectiveness of CUBS.

Acknowledgment

References

[1] http://www.internetworldstats.com/am/us.htm.
[2] http://www.websiteoptimization.com/bw/0511/.
[3] Cable modem tutorial. http://www.cable-modems.org/tutorial/.
[4] Data Over Cable Service Interface Specifications (DOCSIS). http://en.wikipedia.org/wiki/DOCSIS.
[5] Iperf. http://dast.nlanr.net/Projects/Iperf/.
[6] Linux traffic control. http://lartc.org/.
[7] Routing for multiple uplinks/providers. http://lartc.org/howto/lartc.rpdb.multiple-links.html.
[8] One way transmission time. *ITU-T Recommendation G.114*, 2003.
[9] A. Akella, B. Maggs, S. Seshan, R. Sitaraman, and A. Shaikh. A measurement-based analysis of multihoming. In *Proc. of SIGCOMM*, Karlsruhe, Germany, August 2003.
[10] A. Akella, S. Seshan, and A. Shaikh. Multihoming performance benefits: An experimental evaluation of practical enterprise strategies. In *Proc. of USENIX Annual Technical Conference*, Boston, MA, 2004.
[11] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz. The effects of asymmetry on TCP performance. In *Proc. of MobiCom*, 1997.
[12] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular Internet access using in situ Wi-Fi networks. In *Proc. of MobiCom*, 2006.
[13] R. Carter and M. Crovella. Measuring bottleneck link speed in packet-switched networks. Technical Report 1996-006, 15, 1996.
[14] K. Cho, K. Fukuda, H. Esaki, and A. Kato. The impact and impliactions of the growth in residential user-to-user traffic. In *Proc. of SIGCOMM*, 2006.
[15] M. Claypool, R. Kinicki, M. Li, J. Nichols, and H. Wu. Inferring queue sizes in access networks by active measurement. In *Proc. of the 4th Passive and Active Measurement Workshop (PAM)*, Antibes Juan-les-Pins, France, April 2004.
[16] B. Cohen. Incentives build robustness in BitTorrent. In *Proc. of P2PEcon*, 2003.
[17] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *Proc. of IMC*, 2007.
[18] Y. Gao, G. He, and J. C. Hou. On exploiting traffic predictability in active queue management. In *Proc. of INFOCOM*, 2002.
[19] F. Guo, J. Chen, W. Li, and T. Chiueh. Experiences in building a multihoming load balancing system. In *Proc. of INFOCOM*, 2004.
[20] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurement, analysis, and modeling of BitTorrent-like systems. In *Proc. of IMC*, Oct 2005.
[21] A. Habib and J. Chuang. Improving application QoS with residential multihoming. In *Computer Networks*, volume 51, August 2007.
[22] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi. FatVAP: Aggregating AP backhaul capacity to maximize throughput. In *Proc. of NSDI*, 2008.
[23] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD trace set dartmouth/campus/tcpdump (v. 2004-11-09), Nov. 2004.
[24] K. Lakshminarayanan and V. Padmanabhan. Some findings on the network performance of broadband hosts. In *Proc. of IMC*, 2003.
[25] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki. Measuring queue capacities of IEEE 802.11 wireless access points. In *Proc. of BROADNETS*, Raleigh, NC, September 2007.
[26] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent? In *Proc. of NSDI*, 2007.
[27] S. Ren, L. Guo, and X. Zhang. ASAP: an AS-aware peer-relay protocol for high quality VoIP with low overhead. In *Proc. of IEEE ICDCS*, 2006.
[28] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. Biersack. Performance limitations of ADSL users: A case study. In *Proc. of the 8th Passive and Active Measurement Conference (PAM)*, April 2007.
[29] A. Sinha, K. Mitchell, and D. Medhi. Flow-level upstream traffic behavior in broadband access networks: DSL versus broadband fixed wireless. In *Proc. of IEEE International Workshop on IP Operations & Management (IPOM)*, 2003.
[30] E. Tan, L. Guo, S. Chen, and X. Zhang. SCAP: Smart caching in wireless access points to improve P2P streaming. In *Proc. of ICDCS*, June 2007.
[31] N. Thompson, G. He, and H. Luo. Flow scheduling for end-host multihoming. In *Proc. of INFOCOM*, Barcelona, Spain, 2006.