Wireless Network Interface Energy Conservation for Bottlenecked First Mile Networks

Surendar Chandra
surendar@acm.org

ABSTRACT

Placeshifting systems stream videos from the home to a single remote user using the limited upstream capacity of the home broadband link. We analyze the behavior of two placeshifting systems each using two types of broadband networks. We show that the duration between packets did not depend on the way that the servers were sending the packets through the bottleneck link. Even though both of these systems used TCP, the duration between packets did not follow the round trip times either. Instead, it depended on the particular broadband network. Our analysis shows how the bottlenecked first mile network leads to predictable packet delivery at the remote client. Paradoxically, it also leads to shorter periods and a single packet within each data burst. We discuss the limitations imposed by this behavior on a client side energy saving mechanism. We also describe techniques that allow the placeshifting servers to better operate with client side WNIC energy saving mechanisms.

Categories and Subject Descriptors

C.2.m [**Computer Systems Organization**]: Miscellaneous

General Terms

Experimentation

Keywords

Placeshifting, TCP streaming, first mile bottleneck, energy saving

1. INTRODUCTION

The wireless network interface (WNIC) consumes a significant portion of the laptop energy reserves [12, 1]. They support various power states with varying network functionality. Prior efforts [4, 9, 11, 3, 16, 14] exploited the low power states to conserve energy.

Energy conservation schemes are effective for isochronous streaming media because of their predictable network behavior [3]. However, prior systems did not consider the effects of network bottleneck on packet predictability. Without any bottleneck, the packet dynamics at the client are predominantly driven by the way packets were sent from the media server. For wireless users, the bottleneck link can either be in the first mile or in the last mile network. With a bottlenecked last mile network, there are no idle durations that can be exploited for energy conservation. Hence, we explore streaming scenarios using a bottlenecked first mile network.

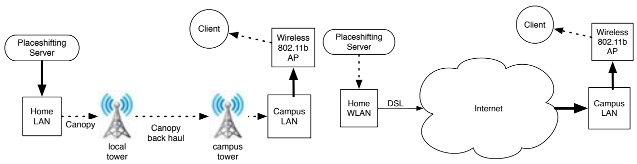
We investigate placeshifting systems that allow users to watch the video from a place that is different from the originally intended location. For example, when traveling, these systems allow a single user to remotely watch premium TV programs that they had already subscribed at their home (and thus avoid paying twice). Systems such as the EchoStar Slingbox/TV Everywhere, Sony LocationFree and TV2Me offer hardware devices that attach to the TV output. The upcoming EchoStar ViP 922 DVR will also natively offer this Slingbox functionality. Client software is available for laptops, smart phones and handheld gaming consoles. Sony and EchoStar also make dedicated wireless tablets that are designed to watch the LocationFree and Slingbox streams, respectively.

The placeshifting server uses the under-provisioned upstream capacity of the broadband link; the Speed Matter's "*Survey of Internet speeds*" estimates that the average upload speed in the US is 1.1 Mbps. The Slingbox consumes over eight Mbps to locally stream standard definition (SD) streams. Note that placeshifting access is one-to-one and is different from peer-to-peer (P2P) scenarios where the same object was available elsewhere; P2P users can circumvent the low upload capacity by either accessing the object from a server or swarm from a number of different locations [5].

We analyzed the network traffic created by EchoStar Slingbox PRO and Sony LocationFree LF-B20 placeshifting devices. Both these devices streamed using the TCP protocol with no user configurable way to choose UDP streaming. We used the Motorola Canopy WWAN as well as a DSL service as our first mile broadband network. We watched three different TV programs that were stored in a DVR for repeatable experimentation. We watched the streams from a laptop as well as a Sony PSP handheld.

Since the placeshifting devices used TCP, we expected that the duration between packets will depend on the round trip times (TCP self-clocking [8]). Instead, they depended on the first mile network as well as on the distribution of packet sizes. We show that the bottleneck allows the streams to be received far more periodically than when directly receiving the stream from the placeshifting servers themselves. However, the bottleneck can only allow one packet in a given data burst. Placeshifting devices can increase the duration between packets by using the largest packet size that will also not be automatically fragmented by the bottleneck network. However, while using large packets, jitter can lose large amounts of data; the client will lose the entire packet if the WNIC was in an energy saving state at the start of a packet. These observations have important implications on client side WNIC energy saving mechanisms.

NOSSDAV'10, June 2–4, 2010, Amsterdam, The Netherlands.



(a) Canopy Wireless WAN (b) DSL

Figure 1: First mile (bottleneck link) scenarios

We analyze the streams in §2 and analyze energy saving mechanisms in §3. We discuss prior research in §4 and conclude in §5.

2. PLACESHIFTING STREAM ANALYSIS

We analyze the placeshifting streams with an eye towards understanding factors that affect energy conservation (§3).

2.1 System setup

For our experiments, we used the Canopy and DSL broadband services (Fig. 1). The Motorola Canopy is a wireless last mile broadband and a back-haul service that behaves similar to the IEEE WiMax network. The shared bandwidth between the Canopy back-haul towers was seven Mbps. The network was configured 3:1 for downstream and upstream traffic. However, this configuration was elastic; when there was no downstream traffic, the upstream packets can use the entire seven Mbps of bandwidth. We used two hops to directly reach the campus's wired LAN. We also used the AT&T DSL service which offered six Mbps downstream and 768 Kbps upstream; the bandwidth was capped at the home DSL modem. The University did not have any special peering arrangements with the DSL provider. For local viewing, we used the Ethernet connection for the Slingbox and the integrated access point for LocationFree streams. We refer to these network scenarios as *Canopy*, *DSL* and *Direct*, respectively. Note that we expect to fully use all of the upstream capacity because the owner is remote and is not using the home broadband link for other purposes. Besides, any simultaneous use from within the home will predominantly use the downstream capacity, leaving the upstream for placeshifting purposes.

The placeshifting *Server* was attached to an EchoStar ViP 622 HD DVR which allowed us to replay the same stream for the various experiments. We used the Slingbox PRO and Sony LocationFree LF-B20 placeshifting devices. Both of these devices use two TCP channels to independently transmit the audio and video information. We watched the Slingbox stream using a Mac OSX laptop and the LocationFree stream using a Sony PSP handheld device. Though the LocationFree can stream videos at up to three Mbps, the PSP was only capable of receiving streams at about one Mbps (corresponding to the Sony *Preset 5*). The clients used a dedicated 802.11b access point operating on an isolated wireless channel to remotely watch and control the TV programs. The placeshifting devices as well as the viewer clients are proprietary and closed. Hence we analyze the packets transmitted by these devices using client-side tcpdump traces. While using the Slingbox, we collected tcpdump traces as well as watched the streams using the same Mac laptop. While watching the LocationFree streams, we collected tcpdump traces using a Mac laptop that was associated to the same wireless AP as the Sony PSP LocationFree client.

We watched three different TV programs for about ten minutes each. We watched a high definition (HD) BBC *Planet Earth* program. Note that our placeshifting devices do not stream in HD. The newer Slingbox PRO-HD placeshifting *Server* requires at least 1.5 Mbps upstream network speed for HD streaming, currently not supported by our DSL provider. We watched the SD cartoon show *Caillou*. We also surfed through a set sequence of live over-the-air (OTA) HD local channels; we changed the channel, waited for the stream to stabilize (DVR exhibited a channel change latency of several seconds) and then waited for three seconds before switching the channel. Such surfing frequently changes the scenes and adversely affects the compression ratios achieved by MPEG like encodings. We refer to these traces as *Planet*, *Caillou* and *Surf*, respectively. We performed the remote experiments around 4:00 PM on a weekday; we expected some congestion induced by traffic from campus and Internet users. We repeated the experiments under each scenario and report the representative results.

For lack of space, we only illustrate the first 150 seconds using the *Planet* stream for the rest of the paper.

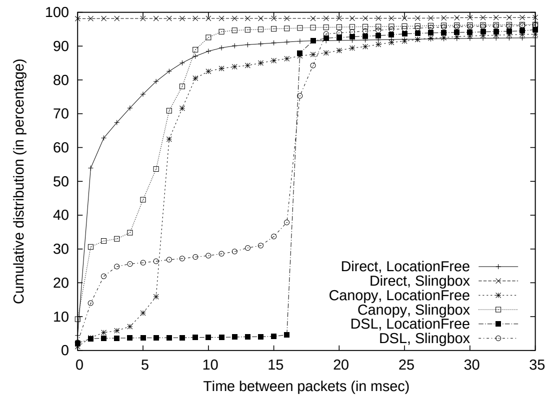


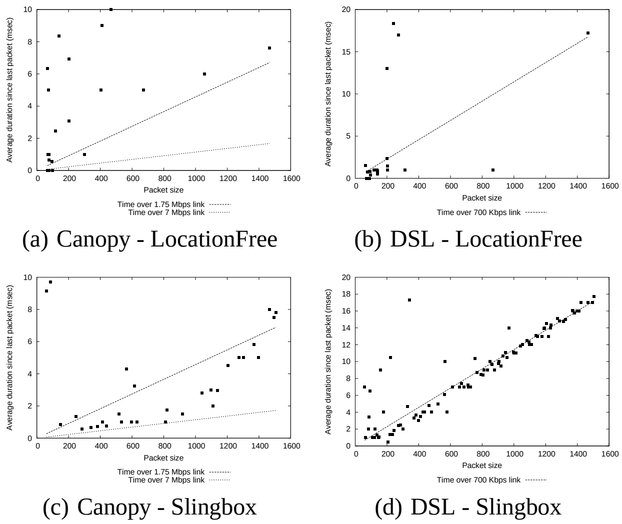
Figure 2: Duration between packets (Planet)

2.2 Duration between packets

First, we plot the distribution of the duration between packets for the various networking scenarios for the *Planet* stream in Fig. 2. While using the in-home *Direct* network, the Slingbox sent 97% of the packets within a msec of each other. For the LocationFree device, 45% of the packets were received within a msec of each other while 90% were received within 15 msecs of each other.

For the *Canopy* network and LocationFree, only 15% of the packets were received less than seven msecs of each other with 50% of the packets being received at seven msecs. However, with the Slingbox, 30% of the packets were received within less than a msec and 85% of the packets were received within seven msecs. Using the *DSL* network, we note that for the LocationFree service, 90% of the packets were received within 16 msecs and for the Slingbox, about 40% of the packets were received within 16 msecs.

The observed system behavior was different between *Direct* and the various broadband networks. We expected the placeshifting *Server* to produce data frames at regular intervals (say every 33 msec for the 30 fps stream). Our results showed that the duration between packets did not follow the stream frame rate. TCP streams are self-clocking [8] and the time between bursts is controlled by the receipt of the acknowledgments. The downstream bandwidth is much higher than the upstream link and the acknowledgment packets were likely delivered with little congestion. Hence we measured the round trip times using the time between sending feedback traffic from the client to the placeshifting *Server* and the corresponding ACK as 34.37 msec and 14.54 msec for *DSL* and *Canopy* networks, respectively. Thus, the time between packets cannot be attributed to the TCP self clocking mechanism either.



(a) Canopy - LocationFree

(b) DSL - LocationFree

(c) Canopy - Slingbox

(d) DSL - Slingbox

Figure 3: Packet size and duration between packets

2.3 Distribution of packet sizes

In general, video streams are expected to be a variable bitrate with packet sizes depending on the frame type (e.g. I, P or B). However, TCP is a reliable data streaming protocol. Hence, video frames written by the placeshifting *Server* need not be transmitted immediately as an IP datagram; the actual network packet sizes are influenced by the current congestion window and the receipt of acknowledgments. While using the *Direct* link (not illustrated for lack of space), only 79% of the LocationFree packets and 58% of the Slingbox packets were streamed at the largest size. The remaining LocationFree packets were uniformly distributed between 66 and 1,466 bytes. For the Slingbox, 25% of the traffic used 134 byte packets, 8% used 286 bytes, and another 8% used 1,362 bytes. Using the *DSL* link, about 95% of the LocationFree packets and 62% of the Slingbox packets were streamed at the largest packet size. Also, the Slingbox transmitted 16% of the traffic using 150 byte packets and another 10% of the traffic used 302 byte packets. On the other hand, using the *Canopy* network, 92% of the LocationFree packets and 58% of the Slingbox traffic used the largest packet sizes. The remaining LocationFree packet sizes were uniformly distributed between 66 and 1,465 bytes. The Slingbox streamed 22% of the packets of size 150 bytes, 5% packets of size 286, 4% of size 382, 4% of size 1,274 and another 5% of size 1,370.

2.4 Packet size and duration between packets

Next, we plot the relationship between the stream packet size and the duration between the current and prior packet for watching the *Planet* stream in Fig. 3. We also show reference lines that plot the time taken to transmit a particular packet using the nominal bandwidth of the bottleneck link. The specific bottleneck bandwidth value affects the slope of this reference curve; a higher bandwidth will shift the reference line towards the x-axis. Note that it is not usually possible to send a packet faster than the reference duration (except for packets which were buffered and delayed in the Internet cloud and hence can be momentarily sent faster than the reference duration). It is possible to observe a larger duration either because the placeshifting server was deliberately choosing the larger interval or when the real bottleneck link was elsewhere.

We chose a reference line of 700 Kbps for the *DSL* network and a range of 1.75 Mbps and seven Mbps for the *Canopy* network (soft provisioned). If the average duration between packets was correlated to the reference line, then that either means that the packets were being automatically buffered and delayed at the home end of the broadband link or that the placeshifting server was shaping the traffic to arrive at the appropriate times to the broadband link. Our analysis (§2.2) shows that the behavior of the system was different for each network; there is no indication that the placeshifting system itself was explicitly performing any traffic shaping.

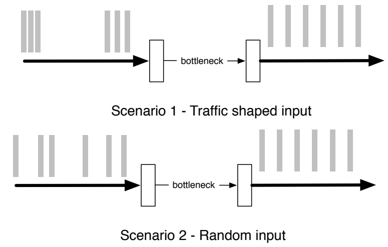


Figure 4: Bottleneck link automatically shaping network traffic

First we focus on the *Canopy* network and the LocationFree stream (Fig. 3(a)). On average, the maximum size packets required about eight msecs since the previous packet. This value closely matches the reference line for 1.75 Mbps; the time between packets matched the time required to transmit the packet. Similarly, while using the Slingbox device, the largest and most popular packets required about eight msec which was also the time required to transmit this packet through a 1.75 Mbps link. However, smaller packets were received at durations which were less than the reference 1.75 Mbps link. On the other hand, these smaller packets were received within the duration for a seven Mbps link; the *Canopy* network allowed momentary stream bandwidths to exceed the 1.75 Mbps limit. Note that many small packets required more than ten msecs since the last packets and hence were not illustrated because of our choice of the scale of the y-axis.

Next, we analyze the behavior while using the *DSL* network and the LocationFree device (Fig. 3(b)). Most of the duration between packets were strongly correlated with the time to transmit the packet through the DSL bottleneck. Over 95% of the packets were received as the largest packet and was strongly correlated with the reference line. Similarly, we analyzed the behavior while using the Slingbox in Fig. 3(d). We note a strong correlation between the packet size and the time to receive the packet. Interestingly, the correlation remains even though the higher error rates could be expected to adversely affect the stream periodicity; the bottleneck network dominates the duration between packets.

As the bottleneck link bandwidth reduces from *Canopy* (1.75 Mbps) to the *DSL* (700 Kbps) network, the time to transmit the packet through the bottleneck link dominates the duration between packets. Thus, the distribution of the packet sizes decides the distribution of the duration between packets, irrespective of how they were sent by the placeshifting server.

The TCP on the placeshifting server can control the duration between packets by controlling the packet sizes; smaller packets will be received more frequently and larger packets less frequently. We illustrate this behavior in Fig. 4. In Scenario 1, the incoming traffic was sent in bursts while in Scenario 2, the packets arrive at the bottleneck in random fashion. However, in both scenarios, if the bottleneck was saturated, then the egress traffic from the bottleneck is equally spaced with the spacing depending on the network bandwidth of the bottleneck and the packet size. For a saturated 700 Kbps upstream *DSL* link, this duration amounts to 16.77 msec between successive 1,468 packets. If the placeshifting server in-

Trace	Device	Type	Idle duration (%)	Energy (%)
Planet	LocationFree	Direct	86.24	26.15
		Canopy	82.58	29.49
		DSL	88.56	24.02
	Slingbox	Direct	13.36	89.18
		Canopy	81.32	30.63
		DSL	90.85	21.92

Table 1: Energy savings potential

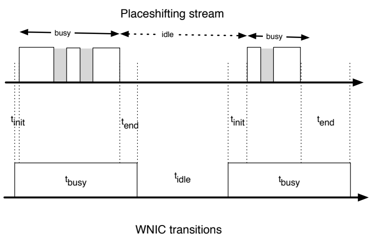


Figure 5: Client side energy management

creased the periodicity beyond this duration, then it will reduce the network throughput. Energy savings mechanisms that need to increase the duration between packets without reducing the network throughput will require traffic shaping mechanisms that are beyond the bottleneck link (towards the client). Approaches that pace the placeshifting server by delaying the ACKs [16] will reduce the effective network throughput. Note that we wish to watch the TV programs at the maximum possible fidelity (which requires a 700 Kbps stream in the *DSL* scenario). Next, we show the importance of these observations on client-side energy savings mechanisms.

3. CLIENT SIDE ENERGY SAVING

Client side energy saving mechanisms exploit the idle durations inherent in bursty media streams.

3.1 Energy saving potential

First we tabulate the percentage of idle time with respect to the total stream duration for the various scenarios in Table 1. We also show the amount of energy consumed by optimally transitioning a Wavelan WNIC [7] to *sleep* instead of the *idle* state. We note that the streams exhibit large amounts of idle durations. While using the *DSL* network, about 88.56% to 90.85% of the durations was idle. This corresponds to potential energy savings of between 76% and 78% over a policy that used the Wavelan *idle* state; there is room for significant energy savings. Note that, *Direct* network scenarios with the Slingbox offer as little as 13% idle duration (requiring about 90% of the energy for leaving the WNIC in the *idle* state).

3.2 Energy saving policy

Energy saving policies need to predict the burst durations so that the WNIC can be transitioned to a low power state at the end of the burst as well as transition the WNIC back to an active state to receive the next burst. Fig. 5 illustrates a scenario where the WNIC remains active during the t_{busy} interval. The policy needs to minimize both t_{init}, the duration when the interface is active and waiting for streaming packets and t_{end}, the duration when the interface is still active even though the data burst is complete. Any data that was delivered while the interface was in a low power state will not be received by the client. However, unlike UDP based

Stream	Single (%)	big prev. small cur. (%)	big prev. big cur. (%)	small cur. (%)
***Canopy*, Slingbox(Wait = 3 msec)**				
Planet	67.63	31.14	0.30	0.92
Surf	68.30	30.62	0.16	0.91
***DSL*, Slingbox(Wait = 4 msec)**				
Planet	71.01	24.56	0.00	4.40
Surf	74.56	20.52	0.02	4.88
***Canopy*, LocationFree(Wait = 3 msec)**				
Planet	94.71	1.51	3.37	0.40
Surf	94.62	1.36	3.38	0.63
***DSL*, LocationFree(Wait = 4 msec)**				
Planet	96.35	2.30	0.09	1.24
Surf	97.74	1.61	0.01	0.63

Table 2: Subsequent packet size distribution

streaming scenarios [3], TCP will retransmit these lost packets. The first mile bottleneck mitigates the loss of periodicity imposed by these retransmissions. Note that the bursts themselves might exhibit idle durations (illustrated by gray shade). When these idle durations within a data burst approach the idle duration between packet bursts, energy saving becomes difficult; we prefer systems that exhibit a clear distinction between the idle and busy durations.

3.2.1 Choosing the policy durations

We chose the *Sleep* and *Wait* parameters (Fig. 5) as follows: the WNIC transitioned to a low power state during *Sleep*. Given the prevalence of large packets (§2.3) and their effect on the bottleneck link (§2.4), we choose the time to transmit the large packet through the bottleneck link as the *Sleep* parameter. The busy duration (t_{busy}) (Fig. 5) is calculated dynamically: after a new packet is received, the WNIC remains in the active state for *Wait* duration. If no packets were received during the *Wait* duration, the WNIC is then transitioned into a lower power idle state for (*Sleep* - *Wait*) duration; packets received during idle duration are lost.

To choose the *Wait* duration, we analyzed packets that were received close together in order to understand whether the subsequent packets were small or large. Larger packets require longer *Wait* durations; affecting our ability to achieve good energy savings. We analyzed the percentage of times when a single packet was received. When multiple packets were received, we investigated three different scenarios: 1) the percentage of packets in which the prior packet within the active burst was large and the subsequent packet was small, 2) when the prior packet was large and the subsequent was also large and 3) times when the first packet was small and tabulated them in Table 2. Note that for LocationFree, about 95% of the packets were single with about 3.4% receiving two large packets in succession. Hence our energy saving policy will transition the WNIC to a *Sleep* state immediately after receiving a packet (without actively idling for the *Wait* duration during the first *Sleep* interval). For the Slingbox, $\frac{2}{3}$ were single while about a third saw a small packet follow a larger packet. These smaller packets can be received with a moderately small *Wait* duration of about four msec.

3.3 Performance metrics

The total energy consumed by the WNIC depends on the time spent as well as the magnitude of power consumed in the idle and active power states ($\sum t_{idle} * P_{idle} + \sum t_{active} * P_{active}$). The power values are specific to a particular WNIC card. We generalize the energy consumed by reporting the relative *Sleep* duration as compared to an optimal policy. The optimal policy will

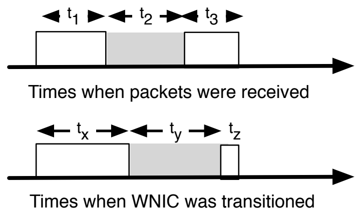


Figure 6: Relative energy consumption metrics

Sleep whenever the network is idle; it transitions the WNIC to an idle state after receiving a packet and then transition back to receive the next packet without losing any packets. Conversely, the achieved *Sleep* duration can be higher than 100% as a particular policy can sleep longer while losing packets. In the worst case, a policy can *Sleep* throughout the experiment, losing all packets and yet achieving higher *Sleep* duration than the optimal policy. We also report the relative *Active* duration as compared to the optimal policy. Again, the *Active* duration can be less than 100%. The optimal policy will achieve a *Sleep* value of 100%, *Active* value of 100% with no packet loss. We also specifically compare the energy consumed while using the Wavelan WLAN card [7]. We consider the time taken to transmit packets using an IEEE 802.11b network with effective throughput of five Mbps.

We describe these metrics with an illustration (Fig. 6): consider a scenario in which two packets, each requiring t_1 and t_3 were received with a duration of t_2 between them. An optimal policy would remain active for t_1 and t_3 and remain in a sleep state for t_2 and require an energy of $(t_1 + t_3) * P_a + t_2 * P_s$ (where P_a and P_s is the power required to remain in the active and sleep states, respectively). For the Wavelan card [7], $P_a = 1425mW$ and $P_s = 177mW$. Consider a policy that maintained the WNIC card in active states for durations t_x and t_z while sleeping for t_y between these active states. We report the relative *Sleep* duration as $\frac{t_y}{t_2}$ and relative *Active* duration as $\frac{(t_x+t_z)}{(t_1+t_3)}$. Note that the relative *Sleep* interface is over 100% because the second packet will be lost by this policy (the WNIC woke up too late to receive the packet).

3.4 Energy savings using the Canopy network

First, we analyze the energy saving for using the *Canopy* network (Table 3) which offered sufficient bandwidth for the placeshifting service. We choose *Sleep* durations of four, five and six msecs and *Wait* durations of one, two and three msecs. 50% of the packets were received in seven msecs (§2.2); *Sleep* value of six msecs and a *Wait* duration of one msec allow the system to sleep and then be ready to receive the next packet.

Energy savings and the amount of lost packets were similar between the placeshifting servers (Table 3) even though their packet dynamics were different. Note that our policy for LocationFree avoids the initial *Wait* interval in an active state. For the LocationFree and *Planet* stream, even though the *Sleep* value of six msec and *Wait* value of one msecs achieved 98% of the sleep duration while being active for 107% of the optimal active duration, it also lost 78% of the incoming packets. The best energy savings were achieved by a policy that used a *Sleep* value of four msecs and a *Wait* duration of three msecs. This policy lost 20% of the packets while remaining online for 356% of the optimal policy. For a Wavelan card, this translates to about 71% of the energy consumed without energy savings (optimal policy required 30%).

3.5 Energy savings using the DSL network

Next, we analyze the energy savings while using the *DSL* network (Table 4). In §2.2, we noted that 90% of the packets were received in 16 msecs. Hence, a policy that used a *Sleep* value of 15 msecs and a *Wait* duration of three msec might be promising. However, for the LocationFree and *Planet* stream, we observe that we achieved a *Sleep* duration of 93% and an active duration of 151% of the optimal policy while losing 24% of the incoming packets. For a Wavelan card, this policy consumed 29.55% of energy (as compared to 24.02% for the optimal policy). On the other hand, a *Sleep*

Sleep	**Wait (msec)**	**Sleep (%)**	**Active (%)**	**Energy (%)**	**Lost (%)**
LocationFree					
4 msec	1	87	157	38.87	72
	2	62	279	58.62	44
	3	45	356	71.20	20
5 msec	1	95	122	33.15	83
	2	73	225	49.86	54
	3	58	297	61.58	35
6 msec	1	98	107	30.81	78
	2	80	191	44.40	59
	3	66	256	54.96	51
Slingbox					
4 msec	1	86	157	40.69	74
	2	52	306	66.64	46
	3	23	434	88.86	24
5 msec	1	94	126	35.17	79
	2	64	256	57.92	49
	3	39	364	76.72	34
6 msec	1	97	110	32.44	79
	2	72	221	51.80	54
	3	50	313	67.86	41

Table 3: Energy savings for *Canopy* and Planet

Sleep	**Wait (msec)**	**Sleep (%)**	**Active (%)**	**Energy (%)**	**Lost (%)**
LocationFree					
13 msec	3	85	214	36.38	33
	4	82	238	38.89	14
	5	80	252	40.43	11
	6	78	263	41.63	9
14 msec	3	89	183	32.98	20
	4	87	199	34.74	15
	5	85	211	36.03	12
	6	84	221	37.11	10
15 msec	3	93	151	29.55	24
	4	91	165	31.10	19
	5	89	178	32.52	17
Slingbox					
13 msec	3	75	338	42.56	45
	4	63	457	52.78	20
	5	55	542	60.11	11
	6	47	614	66.37	9
14 msec	3	76	332	42.01	32
	4	67	422	49.82	19
	5	60	495	56.05	15
	6	52	566	62.15	12
15 msec	3	80	296	38.92	36
	4	71	382	46.24	23
	5	64	450	52.15	20

Table 4: Energy savings for *DSL* and Planet

duration of 13 msecs and a *Wait* duration of five msecs achieved *Sleep* durations of 80% of the optimal value while remaining active for 252% as compared to the optimal policy while only losing 11% of the packets. Further analysis (not shown) showed that we already experience packet loss of around 9% in *Direct* scenarios with acceptable streaming performance. We observed poorer results for the Slingbox; a policy that used *Sleep* of 15 msecs and *Wait* of three msecs achieved a *Sleep* duration of 80% of optimal while losing 36% of the packets. For a Wavelan, this policy consumed about 39% of the energy.

4. RELATED WORK

Balakrishnan et al. [2] analyzed the effects of network asymmetry on TCP performance and reduced the amount of TCP traffic on the constrained reverse network. However, our system operates by primarily sending its data over the constrained reverse network; our path for TCP acknowledgments is unconstrained.

Saroiu et al. [10] analyzed the bottleneck bandwidths between Gnutella peers. Xu et al. [15] investigated the effects of peer asymmetry on P2P streaming systems. Chen et al. [5] analyzed the behavior of IPTV systems while operating using asymmetric peers. Unlike these systems, placeshifting systems cannot choose many clients in order to achieve high bandwidth at the client.

Wang et al. [13] presented a comprehensive analysis of TCP streaming. However, they did not consider network bottlenecks.

Prior work explored energy saving under different network scenarios. Mohapatra et al. [9] developed an integrated power management mechanism that considered the various levels of energy management possible in mobile devices. Our work primarily investigates the effects of a first mile bottleneck on the energy consumption. Earlier, we [3] investigated the energy implications of receiving UDP Microsoft media, Quicktime and Real streams. We then developed [4] an application specific server side traffic shaping mechanism to improve energy savings. Similarly, Shenoy et al. [11] developed a proxy based traffic shaping mechanism. Gundlach et al. [6] also developed a proxy based traffic shaping mechanism. In this work, we show that the bandwidth limitation on the server side plays an important role in shaping the network traffic. However, given the single user operating nature of placeshifting systems, such proxies are unlikely to be widely deployed. Yan et al. [16] investigated energy saving for TCP based downloading systems. They delayed the ACKs in order to transparently force the server to shape the traffic for better energy saving. We show that these techniques do not work in our scenario where the duration between packets are dictated by the bottleneck link; traffic shaping will need to be implemented after the bottleneck network.

5. DISCUSSION

We show that the placeshifting servers themselves did not produce significant stream periodicity. Instead, the first mile bottleneck naturally shapes the stream to be periodic, especially when the bottleneck capacity is stable. Large packets can increase this duration. On the other hand, large packets are likely to lead to large amounts of data loss since the WNIC will lose the whole packet if it remained in the idle state at the start of the packet reception. We show that the bottleneck bandwidth is stable for DSL networks, even when the packets subsequently traverse the Internet. We showed that the placeshifting systems chose large packet sizes, an artifact of the data streaming nature of the TCP protocol. Except for any jitter introduced by the Internet, this traffic shaping was more accurate than relying on the underlying video periodicity. Even with the use of client side proxies, prior efforts [4] only achieved energy savings for the loss of 30% to 50% of the packets. Paradoxically, our system cannot transmit more than one packet within a single burst. Server side traffic shaping cannot address this concern. We either require client side traffic shaping mechanisms (unlikely for single user placeshifting) or be forced to use lower stream quality and entirely remove the network bottleneck.

6. REFERENCES

[1] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta. Wireless wakeups revisited: energy management for voip over wi-fi smartphones. In *MobiSys '07*, 2007.

[2] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz. The effects of asymmetry on tcp performance. In *MobiCom '97*, pages 77–89, New York, NY, USA, 1997. ACM.

[3] S. Chandra. Wireless network interface energy consumption: implications for popular streaming formats. *Multimedia Syst.*, 9(2):185–201, 2003.

[4] S. Chandra and A. Vahdat. Application-specific network management for energy-aware streaming of popular multimedia formats. In *USENIX Annual Technical Conference*, pages 329–342, Monterey, CA, June 2002.

[5] Y.-F. Chen, Y. Huang, R. Jana, H. Jiang, M. Rabinovich, J. Rahe, B. Wei, and Z. Xiao. Towards capacity and profit optimization of video-on-demand services in a peer-assisted iptv platform. *Multimedia Sys. Jour.*, 15(1):19–32, Feb. 2009.

[6] M. Gundlach, S. Doster, H. Yan, D. K. Lowenthal, S. A. Watterson, and S. Chandra. Dynamic, power-aware scheduling for mobile clients using a transparent proxy. In *ICPP '04*, pages 557–565, Montreal, Canada, Aug. 2004.

[7] P. J. M. Havinga. *Mobile Multimedia Systems*. PhD thesis, Univ. of Twente, Feb. 2000.

[8] V. Jacobson. Congestion avoidance and control. In *SIGCOMM '88*, pages 314–329, 1988.

[9] S. Mohapatra, R. Cornea, N. Dutt, A. Nicolau, and N. Venkatasubramanian. Integrated power management for video streaming to mobile handheld devices. In *MULTIMEDIA '03*, pages 582–591, 2003.

[10] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *MMCN 2002*, Jan. 2002.

[11] P. Shenoy and P. Radkov. Proxy-assisted power-friendly streaming to mobile devices. In *MMCN 2003*, pages 177–191, Jan. 2003.

[12] M. Stemm, P. Gauthier, D. Harada, and R. H. Katz. Reducing power consumption of network interfaces in hand-held devices. In *MoMuc-3*, Princeton, NJ, Sept. 1996.

[13] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia streaming via TCP: An analytic performance study. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(2):1–22, May 2008.

[14] Y. Wei, S. M. Bhandarkar, and S. Chandra. A client-side statistical prediction scheme for energy aware multimedia data streaming. *IEEE Transactions on Multimedia*, 8(4):866–874, Aug. 2006.

[15] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer-to-peer media streaming. In *ICDCS '02*, pages 363–371, 2002.

[16] H. Yan, S. A. Watterson, D. K. Lowenthal, K. Li, R. Krishnan, and L. L. Peterson. Client-centered, energy-efficient wireless communication on ieee 802.11b networks. *IEEE Trans. Mob. Comput.*, 5(11):1575–1590, 2006.