How Green is IP-Telephony?

Salman Abdul Baset[1], Joshua Reich[1], Jan Janak[2], Pavel Kasparek[2], Vishal Misra[1],
Dan Rubenstein[1], Henning Schulzrinne[1]
[1]Department of Computer Science, Columbia University, New York, USA
[2]Tekelec, Prague, Czech Republic
[1]{salman,reich,misra,danr,hgs}@cs.columbia.edu
[2]{jan,pavel}@iptel.org

ABSTRACT

With constantly increasing costs of energy, we ask ourselves what we can say about the energy efficiency of existing VoIP systems. To answer that question, we gather information about the existing client-server and peer-to-peer VoIP systems, build energy models for these systems, and evaluate their power consumption and relative energy efficiency through analysis and a series of experiments.

Contrary to the recent work on energy efficiency of peer-to-peer systems, we find that even with efficient peers a peer-to-peer architecture can be less energy efficient than a client-server architecture. We also find that the presence of NATs in the network is a major obstacle in building energy efficient VoIP systems. We then provide a number of recommendations for making VoIP systems more energy efficient.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques, Measurement techniques

General Terms

Performance, Measurement

Keywords

VoIP, energy efficiency, peer-to-peer

1. INTRODUCTION

We aim to understand and analyze the energy efficiency of Voice-over-IP (VoIP) systems. The core function of a VoIP system is to provide mechanisms for storing and locating the network addresses of user agents and for establishing voice and video media sessions over IP (often in the presence of restrictive network address translators (NATs) and firewalls). These systems also provide additional functionality such as voicemail, buddy lists, conferencing, and calling circuit-switched (PSTN) and mobile phones. From the perspective of energy efficiency, a VoIP system can broadly be classified according to two criteria: whether it is a primary-line phone service replacing PSTN and whether it uses a client-server (c/s) or a peer-to-peer (p2p) architecture. Vonage [11] and Google Talk [3] are examples of c/s architectures, while Skype [10] is an example of a p2p architecture. Of these, only Vonage is a primary-line phone service replacing PSTN service.

Recently, Nedevschi *et al.* [17] have developed models describing the relative power efficiency of c/s and p2p architectures for generalized network applications (e.g., file-sharing), and conclude that p2p approaches use system energy more efficiently than the c/s ones. Similarly, Valancius *et al.* [23] argue that building p2p nano-data centers on the Internet gateway devices provides energy savings over traditional centralized data centers. In both papers, the energy savings argument boils down to data center servers (1) needing cooling, network, and other overheads (measured by a multiplicative factor called *Power Utilization Efficiency - PUE*) and (2) having significant baseline power consumption (i.e., power consumption when idling). Typical data center PUEs range from 1.2–2, while the PUE of a peer is 1 (e.g., home air-conditioning is already running) and peers are on anyway, so processes running on peers escape this baseline cost.

We examine the relative energy efficiency of c/s and p2p VoIP systems, and find, intriguingly, that the energy consumption of a peer does not need to be very large in order for a p2p architecture to be *less* energy efficient than a c/s one, warranting further investigation. We also consider the impact of whether a VoIP system is used as an always-on PSTN replacement or as a communication addendum such as Skype. We find that in both cases the energy consumption of edge devices dominate. Finally, we demonstrate restrictive NAT settings to be a major source of energy waste.

Our paper begins by presenting common configurations of deployed c/s and p2p VoIP systems (Section 2). We devise a simple model for analyzing the energy efficiency of c/s and p2p VoIP system architectures (Section 3). This model enables a systematic comparison of c/s and p2p VoIP systems. We then present measurements for c/s and p2p components in Section 4 which we apply to the models developed Section 5. We conclude in Section 6 with recommendations to improve the energy efficiency of VoIP systems.

2. VOIP SYSTEM ARCHITECTURE

We present an overview of the main functionalities provided by VoIP systems and describe how they are typically

Green Networking 2010, August 30, 2010, New Delhi, India.
Copyright 2010 ACM 978-1-4503-0196-1/10/08 ...$10.00.

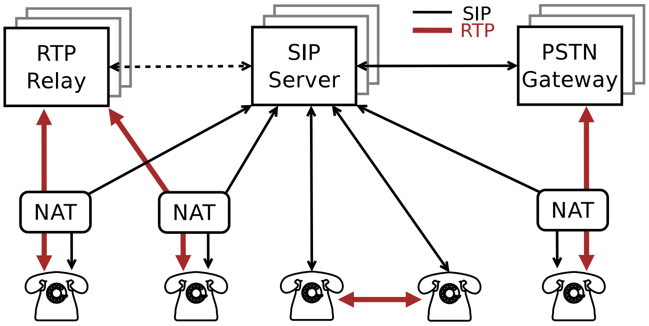

Figure 1: C/S ITSP architecture.

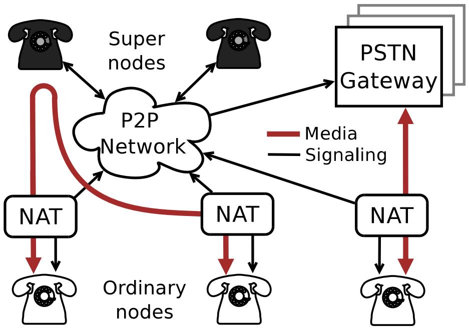

Figure 2: P2P VoIP architecture.

implemented in c/s and p2p VoIP systems. We then describe in more detail the architecture of a typical Internet telephony service provider (ITSP) and Skype, which are representative examples of c/s and p2p VoIP architectures, respectively.

The main functionalities in a VoIP system are:

- **Signaling** - storing and locating the reachable address of the user agents, and routing calls between user agents.
- **NAT keep-alive** - sending and processing user agent traffic to maintain state at the NAT devices for receiving incoming requests and calls.
- **Media relaying** - sending VoIP traffic directly between two user agents or through a relay. Relaying is necessary when one or both of the user agents are behind a restrictive NAT/firewall which prevents establishment of a direct VoIP connection.
- **Authentication, authorization, accounting** - verifying that a user agent is permitted to use the system and tracking usage for billing purposes.
- **PSTN connectivity** - Establishing calls between VoIP clients and PSTN phones using managed gateways.
- **Other services** - such as voicemail, buddy list storage, video calls, and conferencing.

Of the services listed above, signaling, NAT keep-alive, and media relaying lend themselves most easily to a p2p implementation. Consequently in VoIP systems (including Skype) of which we are aware, all but signaling, NAT keep-alive, and media relaying functionality are implemented on centralized servers. As we will see in Section 5, the relative energy consumption of c/s and p2p VoIP systems will be determined by the relative efficiency of c/s and p2p implementations of signaling, NAT keep-alive, and media relaying.

2.1 C/S ITSP Architecture – T-ITSP

We surveyed three c/s ITSPs to obtain information about their server systems, subscriber populations and characteristics of the network traffic. Based on this survey, we present an overview of the largest of these whose architecture is typical for an ITSP. We refer to this ITSP as *T-ITSP* in order to preserve its anonymity.

T-ITSP uses an infrastructure based on open protocols, namely SIP [20] for signaling and RTP [19] for media. It uses a SIP proxy and registrar implementation based on SER [8]. The SIP registrar stores the reachable address of user agents, whereas the proxy server forwards signaling requests between user agents. Users access the system (e.g., place calls) predominantly through hardware SIP phones. Most such phones are audio-capable only, although some also support video. The vast majority of hardphones are connected to the broadband Internet through a home/office router, which is typically configured to act as a NAT/firewall. Over 90% of SIP signaling is done over UDP. User agents connect to SIP servers, perform SIP digest authentication, and register their reachable address every 50 minutes to receive incoming calls, a process we refer to as a *registration* event.

Because most existing NAT devices maintain UDP bindings for a short period of time [16], hardphones behind NATs need to periodically refresh the binding in order to reliably receive incoming calls. The hardphones achieve this by sending a SIP NOTIFY request every 15 s to the SIP server, which replies with a 200 OK response. While wasteful, method proved to be the only reliable way of maintaining NAT bindings.

To establish a call, the user agents send SIP INVITE requests to the SIP proxy servers, which then forward these requests to the destination user agents. The vast majority of hardphones are behind NATs/firewalls and a large proportion of these devices use default settings that prevent user agents from establishing direct VoIP calls. Consequently, T-ITSP needs to operate RTP relay servers to relay these calls, thereby consuming additional energy and network bandwidth. T-ITSP also maintains a number of PSTN servers for calling phones in the traditional telephone network. T-ITSP does not encrypt signaling or media traffic. Figure 1 illustrates T-ITSP's architecture.

2.1.1 Traffic

T-ITSP has a total subscriber base of 100 k users. The peak call arrival rate is 15 calls per second (*CPS*) and peak active calls in the system are 8000. Approximately 60% (or 4,800) of the peak calls were to subscribers within the ITSP; the rest being routed to PSTN/mobile phones. Hardphones register their network address with T-ITSP's SIP registrar every 50 minutes and send a SIP NOTIFY message every 15 s to maintain the NAT binding. For 100 k subscribers, these statistics imply that the SIP registrar needs to process 33 registration events and 6,667 NOTIFY events per second.

Feature	T-ITSP	Skype
User agents	Hardphone	Softphone
Signaling	Centralized	P2P+Centralized
NAT keep-alive	Centralized	P2P
Media relaying	Centralized	P2P
PSTN	Centralized	Centralized
Voicemail	Centralized	Centralized
Buddy list	Centralized	Centralized

Table 1: Architecture: T-ITSP vs. Skype

In Section 4, we extrapolate these peak numbers for a large subscriber base.

2.2 P2P VoIP Architecture – Skype

We present an overview of Skype [14] which is representative of a p2p VoIP system. Skype is not advertised as a primary-line phone service. There are two types of nodes in a Skype network, super nodes and ordinary nodes. The super nodes form the Skype overlay network, with ordinary nodes connecting to one or more super nodes. Super nodes, which are chosen for their unrestricted connectivity and high-bandwidth availability, are responsible for signaling, NAT keep-alive and media relaying. Skype encrypts signaling and media traffic to prevent super nodes from eavesdropping. Skype managed-servers provide functionality for authentication, buddy list and voicemail storage, and calling PSTN and mobile phones. Figure 2 shows an illustration of a p2p VoIP system. Table 1 compares the distributed and centralized features of the T-ITSP and Skype.

3. POWER CONSUMPTION MODEL

We present a model for understanding the power consumption of c/s and p2p VoIP system architectures. We focus on signaling, NAT traversal, and media relaying as they are accomplished using managed servers in the former but through super nodes in the later. Let N be the total number of online ITSP subscribers and let λ_{INV} be the peak number of calls per second these subscribers make and d be the average call duration. These calls are either to other subscribers of the VoIP provider or to PSTN or mobile phones. Let p_v be the percentage of VoIP calls. Of these, let p_{relay} be the proportion of calls that need a relay.

3.1 Client-Server

As discussed in Section 2.1, a c/s VoIP architecture has dedicated servers for handling the signaling, NAT traversal, and media relaying traffic. Signaling traffic includes registration of user agent network addresses with the SIP registrar and call signaling for establishing media sessions. Let λ_{REG} and λ_{INV} denote the peak number of SIP registration events and calls per seconds, respectively, that N user agents generate. The NAT traversal traffic (SIP NOTIFY in T-ITSP) is sent by the user agents to refresh NAT bindings and ensuring reliable receipt of incoming calls. Let λ_{NAT} be the rate of these NAT traversal messages per second. λ_{NAT} will be significantly lower for signaling over TCP than over UDP. In most c/s VoIP systems, signaling and NAT traversal are handled on separate servers from those of media-relaying.

Let $S(\lambda_{REG}, \lambda_{INV}, \lambda_{NAT}, PROTO)$ represent the number of signaling servers needed to handle the peak signaling and NAT traversal load under a particular transport protocol $PROTO$. The $PROTO$ may be UDP, TCP, or TLS. An advantage of using permanent TCP connections between user agents and SIP servers is that it reduces the frequency of the traffic to maintain NAT bindings. However, maintaining hundreds of thousands of TCP or TLS connections on a server is costly in terms of the memory needed [21]. Let $M(\lambda_{INV}, d, p_v, p_{relay})$ represent the number of media relay servers needed to relay calls. Let w_s and w_m denote respectively the wattage consumed by signaling and media servers at the peak load. Let c be the system's PUE and r_s and r_m be the redundancy factor used for signaling and media servers. Then the power consumed by the signaling and media-relay systems is given as follows:

$$w_{c/s} = (Sw_s r_s + Mw_m r_m)c \qquad (1)$$

3.2 Peer-to-Peer

Recall from Section 2.2 that there are two types of nodes in a p2p communication system, (1) super nodes that forward signaling and routing traffic from other super nodes and ordinary nodes, and relay a call between nodes with restrictive network capacity (2) ordinary nodes that do not participate in the overlay routing and connect to one or more super nodes. Let N_S be the number of super nodes in the p2p system with a total population of N subscribers. In contrast to c/s systems, where it is easy to attribute the energy consumption of signaling, NAT traversal and relaying, it is non-trivial to do so for super nodes in p2p systems. We consider two reasonable accounting strategies (which apply as well to energy accounting on phones and network devices):

- **delta** - count only the additional power drawn by the signaling and relaying functions of the super node machine above that of the baseline power consumption of the machine.
- **prop** - in addition to delta, attribute a fraction of the system baseline power consumption proportional to the time the CPU is woken to handle signaling, NAT traversal and media relaying traffic.

For simplicity, assume that each super node sends and receives λ_{MAINT} messages per second to maintain the overlay, and receives $\frac{1}{N_S}$ of the total registration, call invites, and NAT traversal. Each super node relays at maximum one call at a time. A node may use a secure transport protocol such as TLS or DTLS for non-media relaying traffic. Let w_{base} denote the baseline wattage drawn by the super node machine. Let w_{Δ} denote the wattage drawn by the overlay maintenance, registration, signaling, NAT traversal, and media relaying functionality. Let p be the proportion of time the CPU is woken by the client if *prop* accounting policy is chosen or zero for the *delta* policy. Then the power consumed by p2p super nodes is:

$$w_{p2p} = (w_{\Delta} + w_{base}p)N_S \qquad (2)$$

3.3 Comparison Issues in C/S and P2P VoIP Systems

We highlight the broader issues in comparing c/s and p2p VoIP systems.

3.3.1 PSTN Replacement

The most important consideration from the comparison perspective is whether the systems are used as a replacement for the always-on PSTN system. For an IP-based c/s or p2p system that replaces PSTN as the primary-line phone service, the user agents must always be reachable (or powered on) to receive incoming calls. The total energy consumed by such systems is the sum total of the energy consumed by always-on user agents and servers if any.

In contrast, systems like Google Talk and Skype run as a software application on a desktop, laptop, or a mobile device. When comparing these architectures, it is important that we examine the power consumed by the machines providing the core functionality (servers in c/s, super nodes in p2p) and not the difference in energy consumed by the user agents.

3.3.2 Network Costs

C/S and p2p communication systems have a different network footprint as in the later, nodes (or user agents) have to exchange data to maintain the p2p network. Edge and core routers likely incur an energy cost for forwarding traffic for p2p and c/s communication systems. However, these costs are harder to quantify as the edge and core routers are always on. Although, an analysis similar to [17] can be used, we focus on quantifying the energy usage of the system itself and not the network.

4. MEASUREMENTS & RESULTS

In this section, we describe a set of experiments for measuring selected components of c/s VoIP infrastructure and Skype. Our power measurements were taken using a Watts-up .NET power meter [13]. The meter provides 0.1 W precision and claims accuracy to 1.5% of the measured value.

4.1 Client-Server SIP System

Based on the architecture and load information of T-ITSP, we set up a test bed comprising of two servers, the first for handling signaling and NAT traversal workload, and the other for handling media relaying. Our goal was to measure the power consumption of these servers under peak load, and extrapolate the number of servers needed and the power consumed based on peak workload, using the model developed in Section 3.1. Although, this extrapolation may be considered an over simplification, it still provides useful insights into the energy consumed by large scale c/s VoIP systems.

4.1.1 Testbed Overview

The SIP server machine is a Dell PowerEdge 1900 server [2] with two quad-core 2.33 GHz Intel Xeon X5345 processors and 4 GB of memory. It is connected to load-generators with two Intel 82545GM Gigabit Ethernet controllers. The machine has 6 fans. It runs Debian Squeeze (snapshot from 26th February 2010) with Linux kernel 2.6.32. We installed the latest version of SIP-Router, an open source SIP server [8] on the machine and configured it with all the features an ITSP operating in the public Internet would need to use. The SIP server is configured to use 2.5 GB of memory and 16 processes (2 per core). We use MySQL 5.1.41-3 (from a Debian package) configured with 2 GB of query cache. We use SIPp [9] version 3.1.r590-1 to generate SIP traffic according to the model described in Section 2.1.1.

For RTP relay tests we used an IBM HS22 blade server [4] with 5 blades installed. One of the blades was used as an RTP relay server; remaining 4 blades and another two desktop-class PCs were used as RTP load generators. Each blade has two Intel Xeon quad-core CPUs running at 2.9 GHz and a 10 GigE Intel NIC with multiple hardware transmission and receive queues and Linux 2.6.31 kernel. We used the latest version of iptrtpproxy [5], a kernel-level RTP relay. The software relays RTP packets using iptables rules. We used a modified version of SEMS [7] to generate a large number of simultaneous RTP sessions.

4.1.2 SIP Server Measurements

We performed a number of measurements to figure out the maximum number of subscribers our SIP server can support. We desire to determine the maximum load on this server in three configurations: (1) signaling and NAT keep-alive (SIP NOTIFY) traffic carried over UDP as described in Section 2.1.1; (2) signaling traffic over UDP but without any SIP NOTIFY traffic; (3) signaling traffic over permanent TLS connections. The first configuration allows us to reason about the maximum ITSP-like workload a server can handle. The second configuration provides insights into peak ITSP-like signaling workload a server can handle, assuming there were no NATs. The third configuration is helpful from the perspective of comparing T-ITSP to Skype as Skype uses a TLS-like protocol to encrypt signaling and media traffic.

Before running any tests, we provisioned the database of the SIP server with 1 M unique subscribers. The baseline consumption of the server is 160 W. The machine has 6 fans; each fan consumes 10 W when running at full speed. The power consumption when all fans are removed and the machine is idle is 145 W. To see how CPUs contribute to the overall power consumption of the machine, we run 8 cpuburn [1] processes (one per core). The machine consumes 332 W when all cores are fully utilized.

For the first configuration, we found out that our server could handle T-ITSP's traffic mix for approximately 0.5 M users. Under this load, the number of calls (λ_{INV}), registrations (λ_{REG}), and NAT keep-alives (λ_{NAT}) events per second were 75, 166, 33 k, respectively, and the server consumes (w_s) 210 W. For the second configuration, in which there is no NAT traversal traffic, we found that our server could handle load for approximately 1 M subscribers. w_s was 190 W.

For the third configuration (signaling over TLS) there was no need to exchange frequent keep-alive messages over TCP connections to keep NAT bindings open, so λ_{NAT} was 0. With SIP over TLS, the SIP server uses 61 kB of memory per connection and one connection is needed per user agent. Consequently, memory becomes our bottleneck and a maximum of 43 k simultaneously connected user agents can be supported on a single SIP server. w_s was 209 W.

Based on these measurements, we extrapolate the number of servers needed for these configurations in Table 2. Compared to the first configuration, observe that eliminating the keep-alive traffic reduces the number of servers by half in the second configuration. Although the number of signaling servers needed for the third configuration increases approximately by a factor of 12 as compared to the first configuration, we believe that such limitation can be addressed by tuning SSL buffer, by increasing memory in our server or by using hardware SSL accelerators. We are addressing this issue in our ongoing work.

4.1.3 Media Relay Server

We managed to saturate the IBM blade with 15,000 simultaneous calls. Each call has a bit rate of 64 kbit/s or an

Transport	NAT keep-alive	100 k	1 M	10 M	100 M
UDP	YES NOTIFY/s	1	2	20	200
UDP	NO	1	1	10	100
TLS	NO	3	25	250	2500

Table 2: Signaling servers needed, by configuration.

% relayed calls	100 k	1 M	10 M	100 M
0%	0	0	0	0
30%	1	2	10	96
100%	1	4	32	320

Table 3: Media servers needed when relayed calls are 0%, 30%, and 100% of ITSP-ITSP calls.

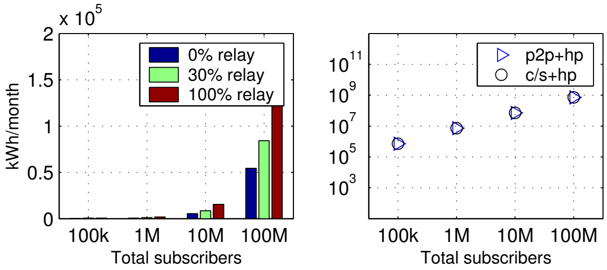

Figure 3: (a) kWh per month of running signaling and media servers as a function of number of subscribers based on T-ITSP workload (b) kWh per month of running a c/s and p2p system on a hardphone. The c/s kWh numbers include servers and hardphones.

aggregate bit rate of 960 Mbit/s. At this rate, the resource bottleneck appeared to be a single CPU core overloaded by ksoftirqd kernel thread. It is likely that even greater call volumes could be relayed by optimizing the multi-core scheduling of this machine using techniques such as [15]. At this workload, the media relay server consumed approximately 240 W (w_m). In Table 3, we extrapolate the number of relay servers needed as a function of user population and the number of calls that need relaying.

4.1.4 Hardware SIP Phones

We also performed measurements to determine the power consumption of a variety of SIP-based hardphones. We found that phones consume between 3 W to 6 W and observed that the phone power consumption does not change when placing a voice call.

4.2 Skype

We attempted to measure Skype's energy use as a softphone and as a super node.

4.2.1 Softphone

For several desktop machines running Windows XP and Windows 7, we did not observe any discernible change in the machine baseline power consumption when Skype was idle. The non-discernible change in the power draw when Skype is idle is partially attributed to the power meter we used which can only measure power up to tenth of a watt with an accuracy of 1.5%. When placing a voice call, we found that on average Skype consumes between 6 W to 8 W on a Windows XP and Windows 7 desktop machine. Similarly, for a video call, Skype consumed between 10 W to 20 W. For laptop machines running Windows XP and Max OS X, we found that Skype, on average, consumed between 1-2 W when placing a voice call. As with the desktop machines, Skype did not cause any discernible power increase when idling. We observed similar power draw behavior for other SIP-based software clients.

4.2.2 Super Node

Measuring Skype's energy draw as a super node is not straightforward. First we need a machine to transition to super node status. Since the Skype client itself decides whether to become a super node, we can only encourage this decision to be made by ensuring that the node has a public IP address, has sufficient bandwidth, and is lightly loaded (which we desired anyway given that we were trying to isolate what we assumed Skype's relatively low power consumption amidst the noise of the machine's hardware and OS). To this end, we ran a Skype client for a few hours on a machine with public IP address and good network connectivity. To determine if the Skype is relaying a call, we performed measurements using a traffic sniffer running on another machine which is connected to the same hub as the Skype machine. We assume a call is being relayed if the bit-rate was above a threshold [22]. Although, our meter readings indicated that there was a non-zero power increase, the difference measured was smaller than the measurement error reported by the power meter. Determining when a super node is handling signaling traffic is even harder to detect, and the power draw per event lasts for a shorter interval and is likely smaller in magnitude. We hope to address these challenges in future work. We did find that the machine can go to sleep when Skype is acting as a super node and relaying the call. The calls were either dropped or transferred to another relay; however, it is impossible for us to ascertain the status of those calls due to the closed nature of the Skype network.

5. DISCUSSION

Our model and measurements enable us to compute the power consumption of signaling, NAT keep-alives, and media relaying in c/s and p2p systems. Recall that for the T-ITSP workload that include signaling and NAT keep-alive traffic over UDP, our SIP server can handle this workload for 500 k subscribers, and consumes 210 W (w_S) under peak load. The RTP relay server under test consumed 240 W (w_M) and can relay 15 k calls, with each call having a bit-rate of 64 kbit/s. The number of active calls in the system for 500 k users are 24 k (extrapolating the number of active calls for 100 k users), requiring two relay servers to handle this load (one server can handle 15 k calls). Depending on the actual deployment, not all calls need relaying. Our conversations with various VoIP system providers suggest that using NAT traversal techniques like ICE [18] will likely bring down the relayed sessions under 30%. When relaying 30% of the 24 k calls, only one relay server is needed. We compute $w_{c/s}$ for both 100% and 30% relaying using our c/s model (equation (1)). We plug c (PUE) as 1.8, and $r_S = 1$ and

$r_M = 1$ in our model. For 100% and 30% relaying, the computed w_S is 1.242 kW and 0.81 kW, respectively. Observe that these numbers are approximate for the peak load and will be higher if the servers are under utilized.

Assuming delta accounting, p2p system will be more energy efficient than c/s when:

$$w_{\Delta} N_S < w_{c/s} \tag{3}$$

To solve (3) for w_{Δ}, we need to estimate the total number of super nodes in the system that can process signaling, NAT keep-alive and media relaying traffic. We estimate the number of super nodes to be 1% of the total user population, meaning that in a population of 500 k user agents, 5 k are super nodes. This assumption is reasonable since if 30% of the 24 k active calls (7.2 k) need a relay, a super node roughly relays one complete call at any instant. Thus, the power consumption per super node, w_{Δ}, is $\frac{0.81\,k}{5\,k} = 0.162\,W$ in order for c/s and p2p systems to be equivalent in terms of energy efficiency. When the servers are under utilized, say 50%, w_{Δ} is twice its original value (0.324 W). The small value of w_{Δ} suggests that if the super nodes were to consume more power than this value in order to handle the signaling, NAT keep-alives, and media relaying workload, a p2p system using super nodes will become energy inefficient as compared to a c/s VoIP system.

Due to the low precision of our power meter, we are not able to ascertain if Skype super node and relaying power consumption is close to w_{Δ}. However, we speculate that the power consumed by super nodes and relays running on desktop machines may likely be close to the w_{Δ} calculated above. The reason is that the CPU of a relatively unloaded machine running a Skype super node or relay may be woken often to service these requests, thus incurring the small power draw to cause it to go above w_{Δ}. On the contrary, handling an additional job on a loaded server causes almost no additional CPU wakeups. We plan to measure the Skype super node and relaying functionality using more precise power meters.

Figure 3(a) plots the kWh per month for running signaling and NAT keep-alives over UDP, and media relay servers that was calculated using (1). The figure illustrates that media relaying due to restrictive NATs is highly wasteful in terms of energy consumption. As an example, the kWh consumption (and the total number of servers) is approximately reduced by a third compared to the scenario when all calls are relayed (from 15,000 kWh to 5,000 kWh approximately, for 10 million users). Table 2 indicates that the number of signaling servers can be reduced by a factor of two when there is no NAT keep-alive traffic. The difference highlights the fact that NATs make the VoIP system wasteful in terms of energy consumption (and of course in terms of bandwidth).

In a VoIP system that replaces PSTN as the primary phone service, the hardphones, that on average consume 5 W, are always powered on. A question to ask is what is the relative power consumption of always on hardphones as compared with servers needed to handle peak load. In Figure 3(b), we plot the sum total of server and hardphone power consumption in kWh per month, denoted by 'c/s+hp'. We also plot the power consumption numbers for 'p2p+hp', in which servers contribute zero power. In both cases, the hardphone power draw was 5 W. As shown, the hardphone power consumption dominates the total power consumption of the VoIP system.

6. POWER EFFICIENT VOIP SYSTEMS

In this section, we present a number of optimization techniques that could help to make c/s ITSP infrastructure and p2p VoIP systems more energy efficient. From private conversations with a number of c/s ITSPs we surveyed for this paper, it is evident that the power efficiency of their infrastructure has not been a concern so far. It is typically other items on the bill, such as Internet connectivity and PSTN related fees, that dominate the overall cost of running an ITSP system. Nevertheless, we learn that energy consumption related costs is the only component of the overall bill that is getting more expensive [6].

Some of the techniques for reducing energy waste in VoIP systems are:

- Embed SIP user agents in the DSL/cable routers so that they can bypass the NAT inside the cable/DSL modem. The DSL/cable routers usually have a public IP address. Such embedded user agents will likely require no media relays and do not need to send NAT keep-alive traffic. This recommendation works well for user agents in a c/s VoIP system.

- Use persistent TCP connections among the user agents (p2p) and between the user agents and the SIP server. Persistent TCP connections require significantly less keep-alive traffic for maintaining NAT bindings.

- Use advanced NAT traversal techniques, such as ICE [18] to allow user agents to detect network conditions and use RTP relays managed by the ITSP only when absolutely required.

Techniques such as Wake-on-LAN and Wireless Multimedia Extensions [12] found in modern mobile computers may eventually lead to even more energy efficient always-on VoIP systems. Such devices could enter a power saving mode in periods of inactivity and be woken up remotely over the network upon arrival of an incoming call. Using power saving modes together with always-on VoIP is the focus of our ongoing work.

7. CONCLUSIONS & FUTURE WORK

We identified the key components that are implemented on servers in a c/s VoIP system and by super nodes in a p2p VoIP system (Skype). We presented a model for understanding power consumption of c/s and p2p VoIP systems. We performed a number of experiments to determine the server power consumption and number of servers needed for a given number of users and traffic load. Our models and measurements indicate that even when super nodes consume relatively small power for system operation, the p2p VoIP system can still be less energy inefficient than a c/s VoIP system. Our analysis and experiments showed that the presence of NATs is the main obstacle to building energy efficient VoIP systems. Our analysis also suggests that in c/s VoIP systems with always-on hardphones, the total power consumed is dominated by the power consumption of the hardphones.

In the future we plan to measure the power consumption of Skype clients in super node and client-only modes and investigate the impact of consolidated (relaying more than one call simultaneously) p2p media relays on overall system energy efficiency. In addition, we hope to build a model of

SS7 network to determine the relative power efficiency of PSTN and VoIP systems. The feasibility of running always-on VoIP clients on energy saving devices and with technologies like Wake-on-LAN and Wireless Multimedia Extensions is also a subject of future work.

Inefficient design of VoIP infrastructure may severely affect VoIP clients running on energy-constrained devices equipped with radio interfaces. Frequent communication will keep the radio interface active, draining the battery, and eventually rendering the device unusable. Designing a power efficient VoIP architecture for mobile devices is the focus of our ongoing work.

8. REFERENCES

[1] CPU burn [accessed March 2010]. http://pages.sbcglobal.net/redelm/.

[2] Dell Power Edge 1900 Server [accessed March 2010]. http://tinyurl.com/ye4z6pn.

[3] Google Talk [accessed March 2010]. http://www.google.com/talk/.

[4] IBM Blade [accessed March 2010]. http://ibm.com/systems/bladecenter/.

[5] iptrtpproxy [accessed March 2010]. http://www.2p.cz/en/netfilter_rtp_proxy/iptrtpproxy.

[6] ITSP private email communication, March 2010.

[7] SEMS [accessed March 2010]. http://iptel.org/sems.

[8] SIP Router Project [accessed March 2010]. http://sip-router.org/.

[9] SIPp [accessed March 2010]. http://sipp.sourceforge.net/.

[10] Skype [accessed March 2010]. http://www.skype.com.

[11] Vonage [accessed March 2010]. http://www.vonage.com.

[12] Wake-on-LAN [accessed March 2010]. http://en.wikipedia.org/wiki/Wake-on-LAN.

[13] Watts up .NET power meter [accessed March 2010]. https://www.wattsupmeters.com/.

[14] S. A. Baset and H. Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *Proc. of INFOCOM*, Barcelona, Spain, April 2006.

[15] M. Dobrescu. RouteBricks: Exploiting Parallelism To Scale Software Routers. In *Proc. of SOSP*, Big Sky, MT, USA, October 2009.

[16] B. Ford, P. Srisuresh, and D. Kegel. Peer-to-Peer Communication Across Network Address Translators. In *Proc. of USENIX*, Anaheim, CA, USA, April 2005.

[17] S. Nedevschi, J. Padhye, and S. Ratnasamy. Hot Data Centers vs. Cool Peers. In *Proc. of HotPower*, San Diego, CA, USA, December 2008.

[18] J. Rosenberg. Interactive Connectivity Establishment (ICE). RFC 5245, April 2010.

[19] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, June 2002.

[20] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, July 2004.

[21] C. Shen, E. Nahum, H. Schulzrinne, and C. Wright. The Impact of TLS on SIP Server Performance. In *Proc. of IPTCOMM*, Munich, Germany, August 2010.

[22] K. Suh, D. R. Figuieredo, J. Kurose, and D. Towsley. Characterizing and Detecting Relayed Traffic: A Case Study using Skype. In *Proc. of INFOCOM*, Barcelona, Spain, April 2006.

[23] V. Valancius, N. Laoutaris, L. Massoulie, C. Diot, and P. Rodriguez. Greening the Internet with Nano Data Centers. In *Proc. of CoNEXT*, Rome, Italy, December 2009.