Smart Proxying for Reducing Network Energy Consumption

Rafiullah Khan, Raffaele Bolla
DITEN Department
University of Genova (UNIGE)
Via Opera Pia 11a, 16145 Genova, Italy
Email:{rafiullah.khan, raffaele.bolla}@unige.it

Matteo Repetto, Roberto Bruschi
CNIT
Research Unit of University of Genova
16145 Genova, Italy
Email: {matteo.repetto, roberto.bruschi}@cnit.it

Maurizio Giribaldi
Infocom s.r.l,
P.zza Alessi 2/7
16128 Genova, Italy
Email: maurizio.giribaldi@infocomgenova.it

***Abstract*—Many network-based applications require full time connectivity of the hosts for fate sharing and responding to routine applications/protocols heart-beat messages. Past studies revealed that network hosts are most of the time unused or idle but still kept powered on just to maintain network connectivity. Thus, significant energy savings are possible if the network hosts can sleep when idle. Unfortunately, present low-power sleep modes cannot maintain network connectivity and results in applications state loss, thus preventing users from enabling power management features. A proxy-based solution was recently proposed in the literature that allows network hosts to sleep and still maintain their network standings over Internet while requiring minor changes to applications/protocols. The Network Connectivity Proxy (NCP) can handle some basic network presence and management protocols like ICMP, DHCP, ARP etc on behalf of sleeping hosts and wake them up only when it is truly necessary. This paper addresses NCP generic architecture and its requirements, main NCP responsibilities, different NCP types and their characteristics and some possible solutions to preserve open TCP connections for the sleeping hosts. It also describes key challenges in the design and implementation of NCP and proposes possible solutions. NCP can result in significant network energy savings up to 70% depending on the hosts time usage model.**

***Index Terms*—Green networking, energy efficiency, smart proxy, power measurement.**

I. Introduction

Green technology is one of the main challenges for the today's world to step into a sustainable society, that can fulfill the needs of today and have the capability to meet those of tomorrow. Main motivation for green technologies is reducing the global CO_2 footprint [1]. Energy efficiency in ICT became a high priority objective due to increase in energy price, continuous growth of Internet-connected edge devices, high data rates and increase in the number of services offered by telecoms and Internet Service Providers (ISP). Thus, green networking is a primary research topic for network engineers to maintain the same performance and QoS of current infrastructures while reducing energy consumption [2].

Recent research shows that energy consumption of network end hosts is higher than servers and data centers and is continuously growing due to the advancement in technology and the increase in the number of network-based devices like PCs, laptops, tablets, smart phones, game consoles, set-top boxes etc. US Environmental Protection Agency (EPA) estimated that network hosts consume about 2% of the overall US electricity requirement [3].

Recent survey [4] revealed that about 60% to 70% of the offices PCs are left powered on all the time even outside work hours just to maintain network connectivity for remote access, Instant Messaging (IM), Voice-over-IP (VoIP) clients, file sharing, security updates or other administrative management reasons. Most of network applications and protocols require full-time network connectivity to receive/respond to routine heart beat messages [5]. Present power management sleep modes such as ACPI S3 and S4 states cannot maintain network connectivity, thus network hosts are left powered on to maintain full-time connectivity even when the hosts are idle, which leads to significant wasted energy consumption [6].

This paper discusses the concept of Network Connectivity Proxy (NCP) that has the capabilities to maintain full network presence for sleeping hosts. The NCP should be always powered on and can be located on host's Network Interface Card (NIC) or other network entity that remains always powered up such as routers, thus does not cause any increase in the network energy consumption. NCP allows the network hosts to sleep and still maintain their standing over Internet by responding to applications/protocols heart-beat messages on behalf of sleeping hosts. The basic idea of proxy was already proposed in literature [2], [5], [7], [8]. This paper addresses NCP generic architecture and requirements, characteristics of different NCP types based on its functionality and location, possible solutions to preserve open TCP connections for the sleeping hosts and some other basic requirements and key challenges in the design and implementation of the NCP.

The rest of the paper is organized as follows. Section II describes the energy waste of ICT and current approaches for low energy networking. Section III describes generic proxy architecture, proxy requirements and types. Section IV addresses some possible solutions to manage open TCP connections. Section V addresses key challenges in the proxy implementation. Section VI addresses expected energy savings and finally, section VII concludes the paper and defines future directions for proxying.

II. Current Approaches to Reduce Network Energy Waste

With the great advancement in the field of Information and Communication Technology (ICT) and continuous growth of Consumer Electronics (CE), the aggregate energy consumption

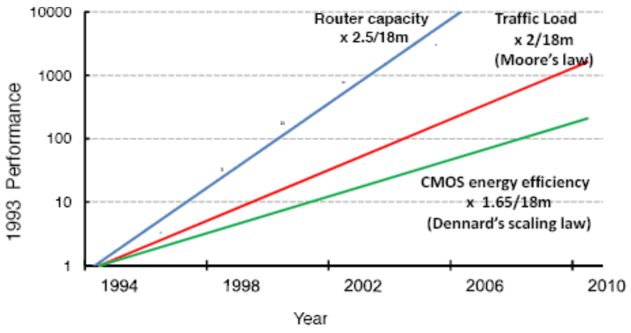


Fig. 1. Evoluation of router capacity vs. traffic volumes (Moore's law) and energy efficiency of silicon.

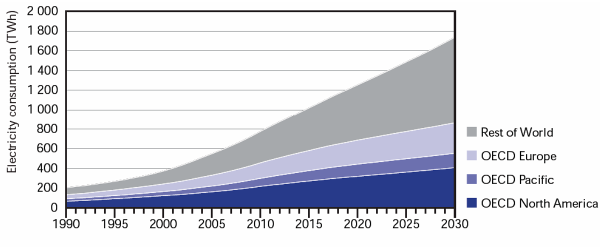
became a prime concern for the researchers. The silicon technologies (e.g., CMOS) improve their energy efficiency by a factor of 1.65 every 18 months as suggested by Dennard's scaling law. Fig. 1 shows that this improvement is low compared to routers capacities (increase factor of 2.5/18 months) and data traffic volumes increase which follows Moore's law and doubles after every 18 months [1]. The increasing energy consumption has two negative impacts: economic and environmental. Economic impact is obvious as the energy prices are continuously increasing. Environmental impact comes from the power plants that generate the electricity. These power plants consume some precious natural resources and most of them emit green house gases in the environment [1].

The International Energy Association (IEA) estimated energy consumption of ICT in Europe, North America, Pacific and rest of the world as shown in Fig. 2. It is clear from Fig. 2 that ICT and CE equipment energy consumption in 2012 is almost double compared to year 2005, and an increase of almost three times is expected by year 2030 with the current increase rate of energy requirement. The Global e-Sustainability Initiative (GeSI) estimated an increase in carbon footprint of ICT for year 2020 to about 320 Mtons of CO_2 emission as shown in Fig. 3. The network infrastructure of mobile communication and fixed wired narrow-band networks cause most significant contributions to greenhouse gases.

Recent surveys revealed that most of the high capacity routers and high data rate links are under-utilized most of the time, thus causing considerable energy wastage. Several ideas are proposed in the literature to reduce total network energy consumption. These approaches can be classified into four different categories.

A. Re-engineering

Re-engineering focuses on introducing and designing more energy efficient elements for network devices. It consists of new silicon (e.g., ASICs, FPGAs, network/packet processors, etc) and memory technologies (Ternary Content-Addressable Memory (TCAM), etc) for packet processing engines and adoption of pure optical switching architecture [1]. Thus, re-engineering focuses on optimization of internal device structure and reduction of complexity level.



Source: IEA estimates.

Fig. 2. Estimated energy consumption of ICT and CE equipment.

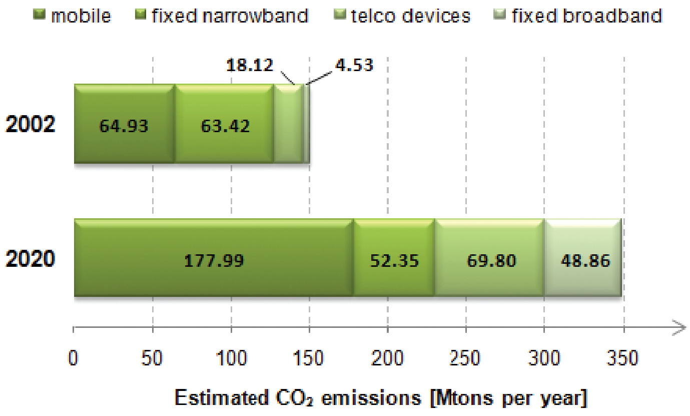


Fig. 3. GeSI estimated emission of CO_2.

B. Hierarchy of hardware components

This approach introduces a hierarchy of power management architectures for network hosts to reduce the energy consumption [9]. It integrates several different power level computing platforms into a single multi-tiered device. The device can operate at only one tier or power level at a specific time. The device uses low power tier for simpler tasks or under low load conditions. It switches to higher power tiers if the processing load increases. This approach significantly reduces energy consumption and it can be very useful for mobile devices that operate on battery power but at the cost of increasing the device complexity.

C. Dynamic rate adaptation

Dynamic rate adaptation is based on varying network device capabilities (e.g., link bandwidths, computational capacity of packet processing engines, etc.) according to current traffic loads and service requirements. Thus, dynamic rate adaptation makes device energy efficient that consumes energy proportional to its utilization [6]. It reduces working rate of the packet processing engines by changing clock frequency and/or by changing voltage of processors or by throttling the CPU clock. Thus, it can be performed by using two power management capabilities of the network devices called dynamic scaling and idle logic. Dynamic scaling scales the voltage/frequency to reduce or increase work rate according to current load condition. Idle logic reduces power consumption by rapidly turning off some sub-components in case of no activity and re-waking them up when some new activities are received.

D. Smart Sleeping (Proxying)

Smart sleeping or proxying approach puts network devices into sleep modes when they are idle and wakes them up when it is truly necessary [10]. A special smart proxy can be designed to maintain network presence for the sleeping hosts [11]. Sleeping hosts consume much less power and higher energy savings can be achieved.

III. Network Connectivity Proxy

Many users leave their PCs powered on even outside working hours when the PCs are largely idle, thus wasting considerable amount of energy. There are two main reasons behind this:

1) To maintain network connectivity for remote access, IM, VoIP clients, security updates or other administrative management reasons.
2) To share its resources (e.g. P2P file sharing).

A network connected host maintains its IP address and its availability over the Internet. The proxy-based solution will maintain network connectivity for the hosts when they are going to sleep, thus it is important to understand the characteristics of a fully connected network host [8]. A network connected host should:

1) Respond to ARP requests in order to allow MAC level communication in the network.
2) Maintain its IP address. In case of DHCP, it should periodically generate DHCP lease requests in order to maintain its IP address.
3) Respond to network diagnostic messages such as ping that uses ICMP to detect the presence of host over the network.
4) Maintain application-level reachability and allow new TCP connections to be established by responding to TCP SYN packets.
5) Maintain existing TCP connections.
6) Generate/respond to the network-based applications and protocols periodic heart beat messages [7].

At present, ACPI standard sleep modes such as S3 (standby: suspended to RAM) and S4 (hibernation: suspended to disk) states are not able to maintain network connectivity. As a result, these power saving features are not usually used by the users. Power consumption of hosts can be reduced through slow clock (dynamic rate adoptation) or stopped clock (sleep) operation. Unfortunately, remote host access in stopped clock mode is not possible as it is effectively sleeping [3]. The NCP is designed for this purpose that implements key network presence capabilities and maintains network standing for the hosts when they enter into sleep mode. NCP starts functioning on behalf of sleeping hosts and wake them up only when their resources are required [5]. This is possible as modern NICs have the capability to recognize magic packet and trigger host wake up. The magic packet is used for the remote wake up of sleeping host.

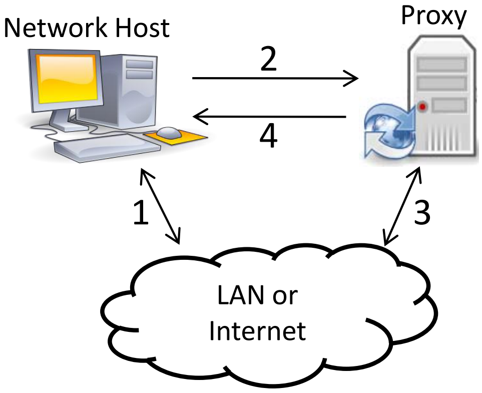


Fig. 4. NCP functional view: 1:host awake, 2:host going to sleep, 3:proxy perform on behalf of host, 4:proxy wake up the host.

Fig. 4 shows a generic functional diagram for NCP. When the host becomes idle and inactivity timer expires, it transfers its network presence to the proxy and goes to sleep. The proxy starts functioning on behalf of that host and receives/responds to routine network traffic for the sleeping host. It wakes up the host when a packet or TCP connection request is received that requires host resources and returns the presence state back to the host.

A. General Requirements

The NCP has some general requirements in order to allow network hosts to sleep and save energy and still maintain their standings over the network [8]. Some basic requirements of NCP are as follows:

1) It should always remain fully powered up and present in the network.
2) It should be in the same network/subnet as the host for which it is covering. This will make easy for the NCP to discover the power state of hosts and waking them up using wake on LAN (WOL) packets.
3) It should be present on network devices which always remain powered up such as routers/switches so that it will not cause any significant increase in network energy consumption.
4) It should be able to know about the power state of hosts.
5) It should be able to act on behalf of sleeping hosts by using their IP addresses.
6) It should be able to handle simple network discovery and management protocols such as ARP, ICMP, DHCP etc on behalf of sleeping hosts.
7) It should be able to preserve existing TCP connections for the sleeping hosts by generating and responding to control messages.
8) It should be able to allow new TCP connection requests for sleeping hosts.
9) It should be able to buffer incoming packets to give enough time for the sleeping hosts to fully recover from their stand-by modes. Buffers can also be used to increase the host sleep time specially during the P2P

files download process. When the NCP buffer is almost full, it wakes up the host, transfers data to the host and then the host goes back again to sleep.
10) Generate/Respond to routine application/protocol messages as required by the application/protocol.

B. Modes of Operation

There are two different ways to operate the network connectivity proxy [12].

1) Uncoordinated or Invisible Proxying: Uncoordinated or invisible proxy doesn't advertise its presence in the network. It invisibly works in the network and guesses host's power state from the traffic analysis. When the host goes to sleep, proxy invisibly starts functioning for it. The invisible proxy requires host traffic to directly pass through it as it guesses host's power state from the traffic analysis. It does not require any change to hosts and/or application servers. Since it doesn't communicate with the host, it cannot verify if the host permanently leave the network while sleeping.

2) Coordinated or Cooperative Proxying: Coordinated or cooperative proxy on the other hand announces its presence in the network and communicates directly with the network hosts. In cooperative proxying, NCP and host exchange two kinds of messages: application specific and wakeup/sleep messages. Application specific messages are used to register host's applications and services at the proxy. These contain information about the application connections and application routine messages. Wakeup/sleep messages are required to trigger proxy when host goes to sleep and wake-up the host when proxy receives a new connection request or a packet that needs host resources. Thus, when a host goes to sleep based on power saving policies or external user commands, it informs the proxy about the power state change and transfers all proxyable information like its MAC and IP address, port number of all open sockets and connection sessions information, TCP connection sequence and acknowledge numbers, etc. to the proxy. The cooperative proxy can be located at any place in the network and requires a software on the host that can be used for communication with the proxy.

C. Types by Location

The proxying functionality can be placed in different devices [13].

1) Self Proxy: In case of self proxying, NCP is located within the network host on its NIC as shown in Fig. 5. It can be invisible or cooperative proxy. NCP can use subset of applications/protocols semantics with unique characteristics that will generate responses without need of complex decision process. NIC will proxy automated responses for network-based applications and protocols and will allow host to sleep when idle. This NCP has the advantage of sharing with the host the same MAC and IP address. Modern NICs have an auxiliary power source from the motherboard which allows NICs to remain powered up even when the host is sleeping. NIC consumes very low power, thus self proxying will not cause any significant increase in network power consumption. Since NIC has limited memory and computational power, thus it will be difficult to implement a complete stand alone proxy covering many applications.

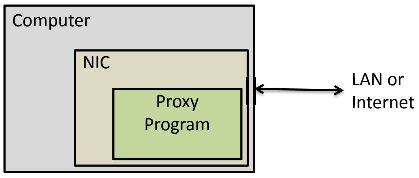


Fig. 5. Self proxy.

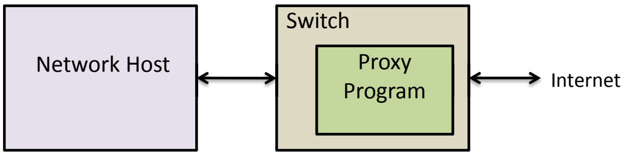


Fig. 6. Switch proxy.

Agarwal et al. [14] presented a USB based low power architecture, named 'Somniloquy', that augments network interfaces to enable sleeping PC to be responsive to network traffic. Somniloquy can overcome the above limitations of self proxying by embedding a secondary low power processor in the PC's network interface that runs an embedded operating system. Somniloquy has significant memory and processing capability and impersonates sleeping host by using same MAC and IP address. Its flash storage is used to temporary store data before being transferred in large chunks to the sleeping PC. It has 64 MB DRAM and 2 GB flash that makes easier to implement the proxying functionality.

2) Router/Switch Proxy: NCP can also be located on network router/switch as shown in Fig. 6. Switch proxy can be both, cooperative and invisible to the network. NCP will use the WOL packet to wake up the sleeping host when its resources are required. Since router/switch always remain powered up in the network, the NCP will not cause any significant increase in the network power consumption. Better than host NIC self proxying but still most of the routers/switches have limited memory and computational power which will make difficult to implement complete stand alone proxy covering many applications for all of the network hosts.

3) Third Party Proxy: Third party proxy is a proxy located on any other host in the network as shown in Fig. 4. This NCP can only be cooperative and it requires communication with the network hosts. It can have high memory and high computational power but may increase the overall network power consumption. Third party proxy can only be beneficial if it is covering for very large number of network hosts. Different approaches can be adopted to reduce network power consumptions such as allowing NCP host itself to go to sleep when idle or not covering for any host. Network hosts should send first WOL packet and transfer proxyable state information to NCP host before going to sleep.

4) Endpoint Proxy: The proxying functions are located on remote endpoints of the host. When the host is going to sleep,

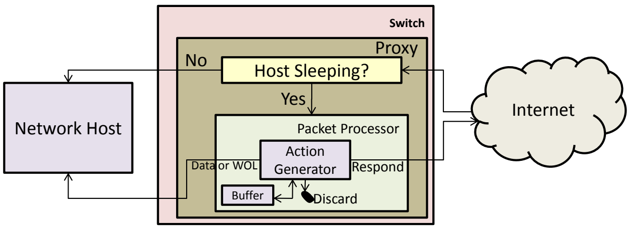


Fig. 7. Generic proxy design.

it will inform the application endpoints about its power state change with proxying request. The endpoints will apparently disconnect application connections with the local host but will preserve their state. When the host wakes up, it will contact the endpoints and re-establish the proxied connections. Endpoint proxy does not require any addition or change in the network hardware but requires software modifications to both sides of the connection. Endpoint proxy can solve the general NCP problem of preserving TCP connections but every application needs to have its own endpoint proxy. Irish and Christensen in [15] proposed similar approach to preserve generic TCP connections for the sleeping hosts.

5) Dual Proxy Approach: Dual proxy approach uses two different proxies with different set of functionalities. It can be used in the scenarios where single proxy does not have the capability to meet all the requirements. Section V-D proposes dual proxy approach to implement the proxying functionalities over different networks/subnets.

D. General Design

The generic proxy block diagram is shown in Fig. 7. The coordinated proxy has a socket connection to the network host on which special control packets and power state information flows. When the network host receives user command to go to sleep, it transfers the power state change information and other proxyable state information such as IP address, port numbers of all open ports, sequence number of all open TCP connections etc to the proxy. The invisible proxy requires the proxy be on the path between the host and Internet. The proxy simply directs packets to the network host when it is awake otherwise it directs them to the packet processing part [16]. When the host enters into sleep mode, the proxy starts intercepting the packet directed to the sleeping host and performs the appropriate action. Proxy generates simple ARP and ICMP reply packets on behalf of sleeping host when it receives the ARP or ICMP request packet. When proxy receives packet directed to listening port but no appropriate action is defined in proxy, it will sent WOL packet to wakeup the sleeping host.

As shown in Fig. 7, the packet processing unit receives all the packets directed to network host when it is sleeping. It performs one of the following task depending on the type of packet received.

1) Directly respond to packets such ARP, ICMP etc.
2) Wake up the host

Destination MAC	Source MAC	Type	6 bytes of 1, followed by dest. MAC repeated 16 times
6 Bytes	6 Bytes	2 Bytes	102 Bytes

Fig. 8. Generic format of WOL packet.

3) Discard the packet
4) Buffer the packets for later processing by the host when it is awake.

The packet processing unit also generates the network protocols/applications heart beat messages at periodic intervals on behalf of sleeping host.

E. Wake on LAN

The wake on LAN (WOL) packet which is also known as ‘magic packet’ was invented by IBM in 1996 for the remote waking up of a network connected PC through its NIC. All modern NICs have an auxiliary power connection from the motherboard. This allows NICs to remain powered up even when the host is sleeping. NICs also have the connection to the wake up interrupt line on the motherboard [3] used for the host wake up. These NICs have the ability of pattern matching for the received packets to trigger the host wake-up. The magic packet is used for this purpose and it is sent in a MAC frame to the sleeping host or broadcasted to all network hosts using the network broadcast address. Thus, WOL packet wakes up the host from ACPI S3 or S4 sleep states to fully powered on state (S0). It contains in its payload six bytes of all 1’s followed by the target computer’s MAC address (48-bit) repeated 16 times. The generic format of the magic packet is shown in the Fig. 8. The following are some of the basic requirements of magic packet:

1) It requires target computer MAC address.
2) It does not provide any delivery confirmation.
3) It requires target computer to be in the same subnet/network. WOL packet can also be sent over Internet but requires to create new inbound firewall rule for the router to accept packets directed to specific port and broadcast them within the network. This increases security risks such as denial of service attacks.
4) It requires target computer NIC to support pattern matching and wake on LAN.

IV. Managing TCP Connections

Many network-based applications such as telnet, ssh, and other resources sharing applications establish permanent TCP connections between client and server. TCP connection is maintained even when no data is flowing by generating and responding to periodic TCP keep alive messages. The main challenging task for proxy is to get control of the host’s open TCP connections when the host enters into sleep mode and return the TCP connections back to the host when it wakes up without any connection drop. The proxy should be able to send TCP keep alive messages with the correct sequence number according to the last received packet by the server. The

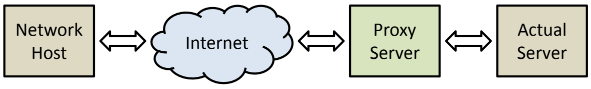


Fig. 9. Proxy server.

proxy should have the knowledge of sequence numbers and acknowledgments for every open TCP connection of the host. Below are some approaches that may be adopted to preserve the TCP connections.

1) Re-writing Applications: The simplest way to preserve the TCP connection is to re-write applications at both sides, host and application server, so that they establish TCP connections only when they have data to transfer. Another possibility can be allowing hosts to preserve the application state at the application server when there is no active data transfer or host is entering into sleep mode. This approach is usually not appreciated as it requires changes to both hosts and application server and it will also introduce some latency.

2) Proxy Servers: Proxy servers can be used as a new entity at the application server side that embeds the capabilities of preserving TCP connections/applications state as shown in Fig. 9. Apparently, the proxy server is proxying the actual application server instead of the network hosts. It can be designed to perform a subset of tasks performed by the real server. It intercepts all the requests directed to the application server and check if it can fulfill the requests, otherwise forward it directly to the actual server. The main motivation of proxy server is to improve performance by reducing load on the real server and having request filtering capability.

The proxy server maintains the TCP connection with the real server on behalf of the sleeping hosts. When the host wakes up, it contacts the proxy server and re-establish the previous connection. The proxy server can be located close to the server or it can act as part of the server. Every application needs to have its own proxy server having the capability to preserve the TCP connections with the actual server on behalf of sleeping hosts.

3) Splitting TCP connections: The proxy splits TCP connections between the host and the server as shown in Fig. 10. The host informs the proxy to preserve TCP connections before going to sleep. During the sleep time, TCP connection between the host and proxy is dropped but between proxy and server will be still maintained. When the host wakes up due to user activity or proxy WOL trigger, it will re-establish its TCP connection with the proxy and will start using the existing TCP connection with the server. Due to the limitations of self/switch proxying, it will be difficult for proxy to manage many TCP connections for different applications/hosts. Also this scheme requires all TCP connections to be established through proxy and it requires the proxy to be cooperative.

Christensen in [17] proposed the concept of split TCP connection with the introduction of new 'shim' layer between the application and socket interface. This approach uses the existing TCP software socket layer and does not propose any changes to application. This shim layer 'fakes out' the application server to see the TCP connection established all the time. When the host enters into sleep mode, the shim layer at client informs the shim layer at server to drop the TCP connection. When the host wakes up, the client shim layer informs the server shim layer to re-establish the connection with the socket information. This TCP splitting requires additional software both at client and server but no changes to the TCP.

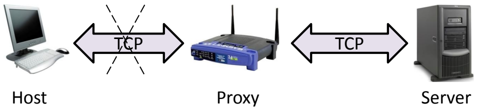


Fig. 10. Split TCP connection.

4) Greening of TCP/IP: TCP connection can also be preserved for the sleeping host by introducing a new green TCP/IP with a connection sleep option. This requires introducing a new connection sleep option in the TCP header. Sleep option in client green TCP/IP will inform the server green TCP/IP when the host is going to sleep. Server will keep the connection open but it will not send any data/ACK to the sleeping host [15]. Server also blocks the socket associated with the TCP connection to prevent excessive data queue at the server. When the host wakes up, it informs the server and data flows can be resumed. The green TCP/IP is backward compatible with normal TCP/IP as the normal TCP/IP ignores any option it does not understand.

V. NCP Key Challenges

This section addresses some of the main limitations and key challenges and proposes some possible solutions to be adopted in the designing of proxy.

A. Memory and processing power requirements

This is the main limitation for self proxy and switch proxy. Current NICs and switches don't have enough memory and processing power to implement complete standalone proxy covering for many applications and network hosts. Third party proxying or dual proxy approach can solve this problem but it will increase the energy consumption as NCP is required to be powered on all time. Third party proxying approach is only useful if it is covering for large number of network hosts. The power consumption can be reduced if the third party proxy can also sleep during idle times and woken up by network hosts using WOL packets before entering into sleep mode.

B. NCP location within network

The invisible proxy requires sleeping host to be directly connected with NCP or traffic of the sleeping host should pass through NCP. This limitation can only be addressed with the help of cooperative proxy. The host informs the NCP about the power state change along with proxyable information before entering into the sleep mode. The NCP broadcasts the ARP reply to the entire network and associate its MAC address with the IP address of the proxied host. The NCP will broadcast another ARP reply to re-associate proxied host MAC address with host's IP address when the host comes out of the sleep state.

C. Host power state awareness

The NCP should be aware of the host presence and its power state. Since the invisible proxy makes decisions from the host's traffic analysis, it is usually very difficult to know about the host presence in the network after sleeping. The NCP should be able to know if the host permanently leaves the network and stops proxying for it. This problem may be solved if the NCP wakes up the host at periodic intervals just to check its presence in the network. However, this results in un-necessary wake up and reduces the potential energy savings. Modern NICs that support DMTF's Alert Standard Format (ASF) Specification 2.0 have the capability to respond to ARP when the host is sleeping, thus making easy for the proxy to check for host presence in the network.

D. NCP over differnt subnets/networks

The NCP can cover for network hosts that are located in the same network/subnet. When the NCP receives a packet that requires the sleeping host's resources, it wakes up the host using a WOL packet. The WOL packet contains the target host MAC address in its payload. It is virtually impossible to have MAC-level communication between hosts in different networks/subnets or to acquire the MAC addresses of hosts from other subnets using ARP requests. Thus, the host needs to communicate its MAC address to the proxy before going to sleep. This problem can be addressed by using the cooperative dual proxy approach and a concept similar to 'ARP proxy' as shown in Fig. 11. The ARP proxy on router allows the main proxy to communicate with the hosts of other networks/subnets without being aware of the existence of subnets. When host A wants to sent a packet to the main proxy, it broadcasts an ARP request to know the MAC address of the main proxy. The APR proxy on router will provide ARP response on behalf of the main proxy with its own MAC address. After the router receives a packet from the host A, it forwards it to the main proxy. The router should implement some additional proxying functionality that will enable it to communicate directly with the main proxy and get the information about all proxied sleeping hosts from it. Whenever the router receives a packet for the sleeping host A, it will forward it to the main proxy. In the same way, the main proxy can wake up the sleeping host A using WOL packet with the help of low functioning proxy at the router. Since this scheme involves traffic forwarding, it may increase the security risks.

E. Application independent proxy

This is one of the challenging tasks that need to be addressed in the design of proxy. Every application has its specific heartbeat/keep alive messages that need to be received and responded at periodic intervals in order to keep the connection open. The proxy needs to know the application and its session data like ports, types of heartbeat messages, etc in order to maintain the application's state for sleeping hosts. Due to very large number of network-based applications and their continuous increase, it would be practically impossible to implement a proxy specific for each of them. The proxy should be able to transparently discover what need to be proxied and should require no modification to the applications.

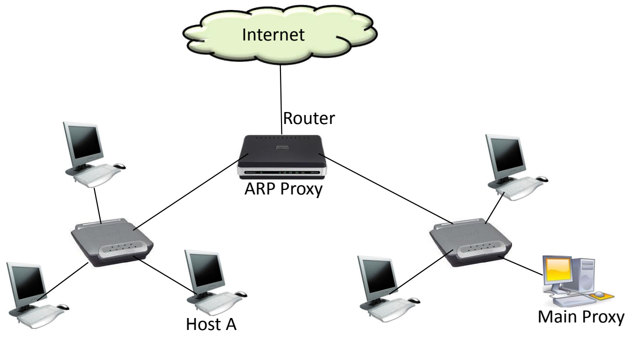


Fig. 11. Proxying over different subnets.

F. NCP support for mobile hosts

The NCP should also be configured to allow host mobility between networks. This possibility can be addressed with an approach similar to 'Mobile IP' [18] and the proxy itself acts as 'Home Agent' for the mobile host. Mobile IP allows host to move from one network to another while maintaining its permanent IP address. The mobile host is always identified by its home address regardless of its current location in the Internet. When the mobile host enters a new network, it is assigned a care-of address which is associated to its home address through a binding at the home agent. This scheme will require the proxy to be strictly cooperative.

The NCP should also be able to support wireless mobile hosts and handle with the connection disruption in wireless environments which is the main design goal of invisible proxy. Wireless mobile hosts may suffer from loss of connectivity that results in dropping connections with the endpoint. The proxy design should also take into consideration the wireless environment and propose solutions to handle short term connection drops due to weak signal or any other reasons. The cooperative NCP should periodically check if the mobile host is alive, otherwise transparently start proxying for it for predefined short time. If the host does not connect again to the network within this short time, the proxy assumes that host has permanently left the network and clears the proxied resources.

VI. Expected Energy Savings

This section estimates possible energy saving by adopting the NCP to allow network hosts to sleep during idle times. The real energy savings estimation requires the knowledge of all network-based devices, their active and sleep power consumption and average use per day [19]. According to the Survey presented in [4], almost about 60 to 70% people leave their computers powered on at home and in the offices all the time only to maintain network connectivity for resources sharing, remote access and for other network centric applications. A normal desktop computer on the average requires 4 to 6W in the sleep mode and 60 to 80W of electricity to be fully powered on. A full time powered on computer on the average requires almost 650 kWh of electricity every year. On the other hand, a computer powered on only for 8 hours and sleep for the rest of the day requires almost 250 kWh of electricity. The average cost of electricity in Europe is around 22 cent/kWh. Thus, a full time powered on PC consumes around 145 Euro/year and NCP power managed PC consumes around 55 Euro/year of electricity. Thus, a single computer on the average can save around 90 Euro every year that accounts to almost 60% savings. An organization with 10,000 PCs and 60% powered on all the time, will be able to save about 0.5 million Euro every year. Thus, enabling power management in PCs can result in billion of Euro savings in the Europe alone every year.

VII. Conclusions

This paper has discussed some possible network energy saving approaches and particularly has focused on the network connectivity proxy. The basic concept of NCP has been described with reference to the previous work in literature. Briefly, the generic structure of NCP and its main requirements and responsibilities have been described. Also different types of NCP along with benefits and limitations of each were addressed. The existing solutions to preserve open TCP connections for the sleeping hosts require modification to TCP/IP stack. This paper addressed some possibilities to preserve TCP connections. The main motivation of this paper is to present a complete sketch of NCP to point out the key challenges in its design and implementation and to propose some possible solutions. NCP can result in significant network energy savings about 60 to 70% depending on the network host's time usage model. Thus, NCP has the potential to provide great environmental and economic savings in the field of ICT.

Acknowledgment

This work was partially supported by the European Commission in the framework of the FP7 ECONET (low Energy COnsumption NETworks) project (contract no. INFSO-ICT-258454).

References

[1] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," in *IEEE Communications Surveys and Tutorials (COMST)*, vol 13 no. 2, pp. 223-244, May 2011.

[2] M. Gupta and S. Singh, "Greening of the Internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03*, pp. 19-26, New York, 2003.

[3] K. Christensen, P. Gunaratne, B. Nordman, and A. George, "The Next Frontier for Communications Networks: Power Management," in *Computer Communications*, Vol. 27, No. 18, pp. 1758-1770, Dec. 2004.

[4] J. Roberson, C. Webber, M. McWhinney, R. Brown, M. Pinckard, and J. Busch, "After-hours Power Status of Office Equipment and Energy use of Miscellaneous Plugload Equipment," in *Lawrence Berkeley National Laboratory, Berkeley, California. Report# LBNL-53729-Revised*, 2004.

[5] K. Sabhanatarajan, A. Gordon-Ross, M. Oden, M. Navada, and A. George, "Smart-NICs: Power Proxying for Reduced Power Consumption in Network Edge Devices," in *IEEE Computer Society Annual Symposium on VLSI, ISVLSI '08*, April 2008.

[6] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI '08*, USENIX Association Berkeley, CA, USA, 2008.

[7] M. Allman, K. Christensen, B. Nordman, and V. Paxson, "Enabling an Energy-Efficient Future Internet Through Selectively Connected End Systems," in *Sixth Workshop on Hot Topics in Networks (HotNets-VI)*, November 2007.

[8] M. Jimeno, K. Christensen, and B. Nordman, "A Network Connection Proxy to Enable Hosts to Sleep and Save Energy," in *IEEE International Conference on Performance, Computing and Communications Conference (IPCCC)*, Dec. 2008.

[9] J. Sorber, N. Banerjee, M. D. Corner, and S. Rollins, "Turducken: Hierarchical Power Management for Mobile Device," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services, MobiSys '05*, 2005.

[10] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft, "Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation, NSDI '09*, USENIX Association Berkeley, CA, USA, 2009.

[11] K. Christensen and F. Gulledge, "Enabling Power Management for Network-Attached Computers," in *Int. Journal of Network Management*, Vol. 8, No. 2, pp. 120-130, March-April 1998.

[12] J. Klamra, M. Olsson, K. Christensen, and B. Nordman, "Design and Implementation of a Power Management Proxy for Universal Plug and Play," in *Proceedings of the Swedish National Computer Networking Workshop (SNCW 05)*, Sept. 2005.

[13] B. Nordman and K. Christensen, "Improving the Energy Efficiency of Ethernet-Connected Devices: A Proposal for Proxying," in *Ethernet Alliance White Paper*, Sept. 2007.

[14] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, "Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage," in *6th ACM/USENIX Symp. On Networked Systems Design and Implementation (NSDI 09)*, Boston, MA, USA, April 2009.

[15] L. Irish and K. Christensen, "A Green TCP/IP to Reduce Electricity Consumed by Computers," in *Proceedings of IEEE Southeastcon*, pp. 302-305, April 1998.

[16] P. Werstein and W. Vossen, "A Low-Power Proxy to Allow Unattended Jabber Clients to Sleep," in *Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT '08*, Dec. 2008.

[17] C. Gunaratne, K. Christensen, and B. Nordman, "Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed," in *Int. Journal of Network Management*, Vol. 15, No. 5, pp. 297-310, September/October 2005.

[18] S. Cheshire and M. Baker, "Internet Mobility 4x4," in *Proceedings of SIGCOMM*, California, USA, August 1996.

[19] B. Nordman, M. A. Piette, and K. Kinne, "Measured Energy Savings and Performance of Power-Managed Personal Computers and Monitors," in *Proceedings of American Council for an Energy-Efficieny Economy (ACEEE) summer study on energy efficiency in buildings*, 1996.