Energy Consumption of Always-On Applications in WCDMA Networks

Henry Haverinen
Nokia Enterprise Solutions
henry.haverinen@nokia.com

Jonne Siren
Nokia Enterprise Solutions
jonne.siren@nokia.com

Pasi Eronen
Nokia Research Center
pasi.eronen@nokia.com

***Abstract*— Always-on applications, such as push email and voice-over-IP, are characterized by the need to be constantly reachable for incoming communications. In the presence of stateful firewalls or NATs, such applications require "keep-alive" messages to maintain up-to-date connection state in the firewall or NAT, and thus preserve reachability.**

In this paper, we analyze how these keep-alive messages influence battery lifetime in WCDMA networks. Using measurements in a 3G network, we show that the energy consumption is significantly influenced by the Radio Resource Control (RRC) parameters and the frequency of keep-alive messages. The results suggest that especially UDP-based protocols, such as Mobile IPv4 and IPsec NAT Traversal mechanisms, require very frequent keep-alives that can lead to unacceptably short battery lifetimes.

I. INTRODUCTION

One of the main benefits of a mobile phone is being reachable at any time, either via call or text messaging (SMS). In addition to these traditional circuit-switched services, there is an increasing number of TCP/IP based applications that require the terminal to be constantly attached to the radio network, and being able to receive TCP/IP traffic. Examples of such "always-on" applications include push e-mail, instant messaging, and IP-based voice and video telephony.

Many always-on applications do not send or receive traffic constantly; instead, the network interface is idle most of the time. To ensure acceptable battery lifetimes, 3GPP WCDMA networks have a number of Radio Resource Control (RRC) states that allow smaller energy consumption when there is no traffic.

In theory, 3GPP WCDMA seems ideally suited for always-on applications. However, real 3G networks almost always include a network element that is not part of the official 3GPP architecture: a stateful "middlebox"—either a firewall or a Network Address Translator—between the mobile terminal and the Internet. Typically, middleboxes create state based on packets sent by the terminal, and remove the state once it has been unused for some time. This implies that connections have to be initiated by the terminal, and keep-alive messages have to be sent to prevent the middlebox from removing the state.

In this paper, we describe how the keep-alive messages used by always-on applications interact with RRC power management, and measure the influence of RRC parameter settings in a 3G network. The results show that the energy consumed by keep-alive messages can lead to unacceptably short battery lifetimes, especially when IPsec or Mobile IP is used. We also give recommendations how to configure the RRC and middlebox parameters to reduce power consumption.

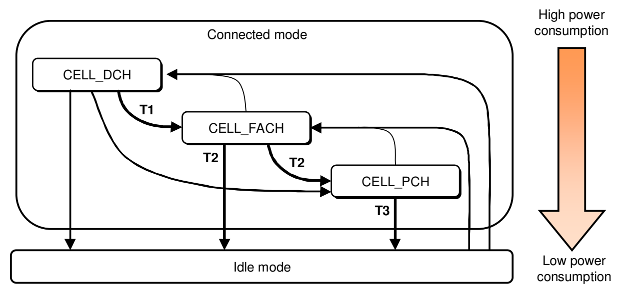

Fig. 1. Overview of WCDMA Radio Resource Control (RRC) state machine.

The rest of this paper is organized as follows. Section II gives a short overview of the RRC protocol. Section III describes how typical stateful middleboxes and keep-alive mechanisms operate. Section IV first presents our measurement setup, and then the results of power consumption measurements in a 3G network. Section V discusses the implications of these results, and gives recommendations how to reduce power consumption. Finally, Section VI describes related work, and Section VII contains our conclusions and directions for future work.

II. RADIO RESOURCE CONTROL (RRC) PROTOCOL

The Radio Resource Control (RRC) protocol, defined in [1], is responsible for assigning radio resources between the terminal and the network. The RRC states in WCDMA packet data are illustrated in Fig. 1, and are described below:

CELL_DCH (Dedicated Channel). In this state, a dedicated channel is allocated to the terminal. This state is used for data transmission, except when the amount of data is very small. It achieves maximum throughput and minimum delay, but at the cost of power consumption. According to Wigard et al. [16], the typical power consumption is 200–400 mA.

CELL_FACH (Forward Access Channel). In this state, the phone shares the channel with other phones. This state is used when there is not much traffic to transmit. The battery consumption is roughly half of the consumption in the CELL_DCH state.

CELL_PCH (Paging Channel). This optional state offers the lowest current consumption, around 1–2 percent of the CELL_DCH state. In this state, the terminal is not able to send or receive packets, but can be paged if there are downlink packets; the terminal will then enter either the CELL_DCH or CELL_FACH state. Not all network implementations currently use the CELL_PCH state.

Idle mode. In this state, the phone does not have an RRC connection. The terminal can still have an IP address (i.e., a PDP context) and it can be reached by paging. The battery consumption is similar to CELL_PCH state.

An additional state, URA_PCH, exists in the specifications but is not currently used. This state is similar to CELL_PCH and further improves performance when there is mobility.

State transitions are based on traffic volume and inactivity timers T1, T2, and T3 managed by the Radio Network Controller (RNC), shown in Fig. 1. The names T1, T2, and T3 are not officially used in 3GPP specifications but they have been established in WCDMA parlance.

T1 is an inactivity timer that is used in the CELL_DCH state, and is reset whenever there is traffic. The T1 value may depend on the DCH data rate. In the RNC implementation we used, the default values were 5 seconds for 8–32 kbit/s, 3 seconds for 64 kbit/s, and 2 seconds for 128 kbit/s and faster.

T2 is an inactivity timer in the CELL_FACH state; the state machine will enter either the CELL_PCH state (if used) or idle state after being inactive for T2 seconds. In the same RNC implementation, the default value for T2 was 2 seconds.

T3 is a timer used in CELL_PCH. After staying in the CELL_PCH for T3 seconds, the RRC connection will be released. This is typically a very long timer (several minutes or even tens of minutes).

The inactivity timers T1 and T2 define the time after which the phone transitions to less power-consuming states. However, as moving from one state to another requires explicit signaling and involves delay, using extremely short timer values is not feasible either. Thus, the timer values represent a trade-off: the shorter the timers are, the more delay the user will experience e.g. in web browsing [3].

When an always-on application sends periodic keep-alive messages in an otherwise idle terminal, the usage of RRC states depends on the RNC's configuration and its features. Three different cases are possible:

1) If the RNC supports the CELL_PCH state, then the sequence is Cell_FACH → Cell_PCH → Cell_FACH → Cell_PCH etc.
2) If CELL_PCH is not supported but the RRC connection can be created directly to the CELL_FACH state, the sequence is Cell_FACH → Idle → Cell_FACH→ Idle etc.
3) If the RNC does not support 1 or 2, then the sequence is Cell_DCH → Cell_FACH → Idle → Cell_DCH → Cell_FACH → Idle etc.

For this paper, we measured cases 1 and 3.

III. Stateful middleboxes

A. Firewalls and Network Address Translators (NATs)

Stateful middleboxes, either in the form of firewalls or Network Address Translators (NATs) are commonly used in today's Internet. Although NATs are generally not needed for IPv6, we expect that firewalls will be.

Product	TCP timeout	UDP timeout
Check Point NG FP2 firewall	60 min	40 s
Cisco IOS router NAT	1440 min	300 s
Cisco PIX firewall	60 min	120 s
Juniper Netscreen firewall	30 min	60 s
Nokia IP VPN gateway	60 min	120 s
ZyXEL Prestige 660 ADSL router	60 min	60 s
ZyXEL ZyWALL 70 firewall	150 min	180 s

Table 1. Default connection state timeouts (source: product manuals)

In a typical NAT configuration, a local network uses IP addresses from a private IP address range (e.g., 10.x.x.x). A Network Address Translator in the local network is connected to the Internet with at least one publicly routable IP address. As terminals send traffic from the local network, the NAT translates the local IP address to the public address(es). Usually, the TCP/UDP port numbers are also modified. The NAT keeps track of active connections so that it is able to translate inbound packets to the correct local addresses.

Stateful firewalls do not modify packets passing through them, but also maintain state of active connections in similar fashion. Typically, only hosts "inside" the perimeter are allowed to initiate new connections (i.e., cause firewall to create connection state). Packets coming from the "outside" are dropped unless they belong to an existing connection, or are otherwise explicitly permitted.

Both NATs and stateful firewalls usually rely on timers to remove unneeded connection state. The timer is reset whenever there is traffic, so the state will expire if the connection is idle for a long time. Table 1 shows the default timeout values for some common NAT and firewall products.

B. Keep-alive Mechanisms

If the terminal needs to be reachable from outside the local network, then the expiration of connection state during idle periods becomes a problem. In order to reset the expiry timers in the middleboxes, many protocols include a keep-alive mechanism by which the terminal can send "dummy" packets when there is no regular traffic to be sent.

Table 1 shows that the timeouts for TCP are significantly longer than for UDP. While many UDP-based protocols do not require long-lived sessions with idle periods, there are several examples of protocols that do: for example, Mobile IPv4 and IPsec NAT Traversal, and Session Initiation Protocol (SIP).

Usually Mobile IPv4 encapsulates packets between the mobile node and the Home Agent (HA) using IP-in-IP encapsulation. However, IP-in-IP encapsulation does not work through typical NATs (and is often blocked by firewalls), and thus Mobile IPv4 NAT Traversal [11] specifies a UDP-based encapsulation. For similar reasons, RFC 3948 [7] specifies a UDP-based encapsulation for IPsec ESP packets.

In Mobile IPv4 NAT Traversal, keep-alive messages are ICMP ("ping") messages sent to the HA. In IPsec NAT Traversal, a special NAT-keepalive packet is sent to the VPN gateway. In SIP, the keep-alive messages can provided by, e.g., STUN (Simple Traversal Underneath NATs) [8].

Interval [s]	Avg. current 2G [mA]	Avg. current 3G [mA]
20	29	34
40	16	24
150	9.1	16
300	7.3	14
Infinite	5.2	6.1

Table 2. Current consumption by keep-alive interval

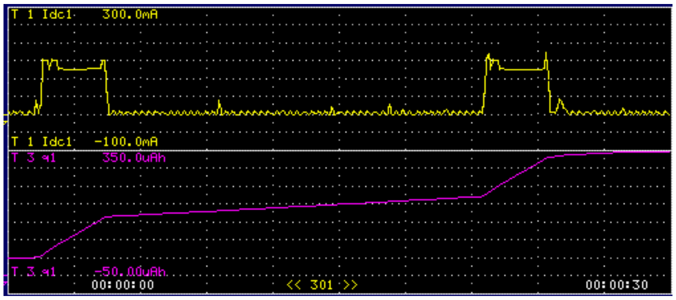

Fig. 2. Current consumption of a VPN keep-alive messages in WCDMA, with CELL_PCH enabled and T2=2s. The x-axis is divided into 3-second intervals. The upper graph shows the transient current in mA, and the lower graph shows the cumulative consumption in mAh.

IV. Measurements

To understand the impacts of WCDMA RRC configuration and keep-alive messages on battery performance, we performed a series of measurements. We measured the current consumption of a 3G phone using different RRC parameters and application keep-alive intervals.

A. Measurement Setup

The measurements were done by connecting a Nokia 6630 phone to a power supply in series with a Yokogawa digital power meter, which can plot the transient current [mA] as a function of time and also measure the cumulative electric charge [mAh] during the measurement.

Two different 3G test networks were used for the measurements. In both cases, there was very little other traffic and no mobility; the effects of these parameters are left for further study.

B. Impact of the Keep-alive Interval

The first set of measurements focused on the effect of the keep-alive interval. We measured current consumption for both 2G GPRS and 3G WCDMA; for 3G we used the default RRC configuration for T1 and T2. CELL_PCH was enabled in the first set of measurements. We varied the keep-alive interval from 20 seconds to 300 seconds. The results are shown in Table 2.

Fig. 2 illustrates the current consumption of a single keep-alive message when the state sequence CELL_FACH → CELL_PCH → CELL_FACH etc. is used. From this figure, we can also see that the current consumption is 120 mA in CELL_FACH, and that the current consumption of the actual transmission is not very significant.

From these measurements, we can calculate the average energy consumption of a single keep-alive event.The consumption of a single keep-alive transaction varied between 0.15 mAh and 0.6 mAh in 3G, and between 0.11 mAh and 0.13 mAh in 2G.

C. Impact of the RRC Timers

We performed another set of experiments in order to estimate how the cost of a single keep-alive depends on the T2 parameter and the use of CELL_PCH. In these measurements, the T1 parameter was left to the default value (described above), and a keep-alive interval of 40 seconds was used. As a different test network was used, there may have been some variation in the radio conditions to the first set of measurements. The results of the second measurements are summarized in Table 3. In the last column, we calculated the cost of a single keep-alive transaction, by subtracting the idle current 6.1 mA and noting that with an interval of 40 seconds there are 90 keep-alive transactions in an hour.

Fig. 3 shows the transient current consumption of a VPN keep-alive message when the state sequence CELL_DCH → CELL_FACH → Idle → CELL_DCH etc. is used. The state transition from the idle mode to CELL_FACH was not enabled, so the terminal needs to transition to the CELL_DCH channel in order to transmit the keep-alive message. The consumption is around 250 mA in CELL_DCH. The state transition and the transmission take some time, which causes significant power consumption. However, the inactivity timers again cause most of the consumption. After T1, the state machine enters the CELL_FACH state, where it stays until the T2 timer expires.

Based on the results in Table 3, and on observing the transient current consumption of Fig. 2, we conclude that when keep-alives can be sent in the CELL_FACH state, the cost of a single keep-alive event is roughly linearly dependent on T2. When keep-alives are sent in the CELL_DCH state, we would expect the cost to be a function of first degree of T1 and T2.

V. Analysis and Recommendations

A. Target Current Consumption

Battery capacities in current mobile phones typically range between 700 and 1500 mAh. Most users would like to use their

T2 [s]	CELL_PCH	Avg. current 3G [mA]	Cost of a single keep-alive [mAh]
2	Enabled	20	0.15
5	Enabled	30	0.27
10	Enabled	45	0.43
2	Disabled	61	0.61
5	Disabled	74	0.75
10	Disabled	98	1.0

Table 3. Current consumption for different T2 values and CELL_PCH configurations, with keep-alive interval of 40 s.

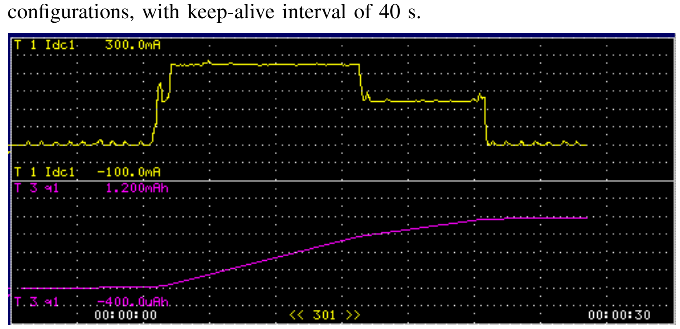

Fig. 3. Current consumption of VPN keep-alive messages in WCDMA, without CELL_PCH and T2=5s.

phones for several days without charging. If we assume that half of the capacity of a 1000 mAh battery can be used for the keep-alives and other idle tasks, and if we further assume that the idle time should be 2–3 days, then the average idle current should be 7–10 mA. Hence, let us use an average current of 8 mA as a target in order to get acceptable battery performance for always-on applications.

B. WCDMA Configuration

To summarize the results of the WCDMA measurements in the light of the current consumption target of 8 mA, we can conclude that

- Keep-alive intervals of less than a couple of minutes will not yield acceptable performance.
- In order to use always-on applications in 3G and meet these requirements, the WCDMA RRC must support either CELL_PCH or Idle → CELL_FACH transitions.

Let us further examine the impact of T2 to power consumption. Let us assume that CELL_PCH or Idle → CELL_FACH transitions are supported, and a keep-alive interval of 5 minutes is used. Assume that the cost of a single keep-alive is the same as in the arrangement of Table 3. Assume further that the idle current between keep-alives is 6.1 mA (based on our measurements). This would give us the following current consumptions depending on T2:

- 8.0 mA with T2=2 seconds
- 9.3 mA with T2=5 seconds
- 11 mA with T2=10 seconds

We can note that T2 of 2 seconds would barely meet the target of 8 mA, while longer T2 values such as 10 seconds would yield unacceptable idle currents.

For 3G networks, the general conclusion is that with short keep-alive intervals, the battery performance is often not acceptable. Tolerable performance can be achieved if keep-alives can be sent in the CELL_FACH state. Generally speaking, shorter the T1 and T2 timers, the better the battery performance of always-on applications will be. If keep-alives need to be transmitted in CELL_DCH, the sum of T1+T2 determines much of the idle power consumption, and, in networks where keep-alives can be transmitted in CELL_FACH, the T2 defines the idle power consumption. In all networks, T1 has an impact on power consumption when there are frequent data events, such as received e-mails. Obviously, making the timers too short will degrade other performance metrics, such as the perceived browsing performance [3].

The timer T3 should be long enough, preferably more than 10–15 minutes, to allow for long keep-alive periods.

C. NAT and Firewall Configuration

The configuration of NAT and firewall expiry timers is a trade-off between the following:

- Long timeout means that more memory is needed to store the state for sessions that have already ended, especially in UDP where the end of session cannot be recognized and short request/reply transactions are common.
- Short timeouts allow the NAT to re-use the TCP/UDP port numbers faster (a NAT with a single public IP address cannot have more than 2^{16}–1 concurrent connections).
- With long timeouts, the client's port is exposed to unwanted traffic longer.
- Short timeouts require frequent keep-alive messages.

IETF BEHAVE working group currently requires a timeout of at least 124 minutes for TCP and 120 seconds for UDP (with 300 seconds recommended) [2], [6].

Our measurements show that using a UDP timeout longer than 120 seconds can increase battery lifetime considerably. However, even 120 seconds is significantly better than what many products currently use.

In some implementations, it is possible to configure the expiry timers separately for different protocols and port numbers. For example, a shorter timeout could be used for DNS queries (port 53) that do not need long-lived sessions.

D. Client Configuration

In general, the client does not know the timeout values used by the middleboxes. Thus, a conservative approach is to use an interval short enough for most commonly deployed products. In IPsec NAT Traversal, keep-alive messages are not acknowledged so the possibility of packet loss must also be taken into account. Given this, [7] recommends a default keep-alive interval of 20 seconds.

Mobile IPv4 NAT Traversal [11] and the MOBIKE extension to IKEv2 [5] allow the client to discover when a NAT mapping (public address/port number) has changed. This makes it possible to probe for the correct keep-alive interval. However, this probing works only with NATs but not firewalls.

E. Summary of Conclusions

For running always-on applications over WCDMA, the main factors influencing energy consumption are the RRC parameters (inactivity timers, whether keep-alives can be sent in CELL_FACH state), and the frequency of keep-alive messages.

Based on the measurements, we conclude that acceptable power consumption for TCP-based applications can be achieved if the RRC is configured as discussed in this paper.

For the UDP-based protocols, acceptable operation in the general case over WCDMA cannot be guaranteed. If the middleboxes are in the control of the organization that is providing the service (e.g., VPN), then it may be possible to get acceptable results by paying special attention to the configuration of the NAT/firewall timeouts and client configuration.

F. Alternate Solutions

So far, this paper has assumed that the middleboxes are necessary network elements that are difficult to change in ways beyond simple configuration. However, in some environments, alternate solutions may be possible.

When the service, such as push e-mail, is provided by the 3G operator, it may be possible to arrange the network so that there are no middleboxes between the client and the server.

However, this approach is not generally applicable to services provided by other parties, such as corporate IT departments.

It may also be possible to reduce the number of keep-alive messages by explicitly negotiating the connection (and its timeout) with the middlebox using a signaling protocol such as NSIS or SIMCO [13], [14]. This would allow the client to request a longer timeout for always-on applications. However, current NATs and firewalls used by 3G operators do not usually support any such protocols.

Always-on applications can also use non-TCP/IP protocols that bypass the middlebox completely. For example, in Multimedia Messaging Service (MMS) the messages are delivered over TCP/IP, but notification of new messages is done using Short Message Service (SMS).

VI. Related Work

The work most closely related to ours is by Yeh, Lee, and Chen [10], [17] who analyze the impact of RRC timers on energy consumption. Their work is based on analytical models and simulations instead of measurements, and the traffic models considered are web browsing and streaming video. Consequently, the recommendations about timer lengths are an order of magnitude larger than ours—even the shortest value used in the simulations (10 seconds) is longer than our recommendations. This may be partly due to the assumption that all non-real-time traffic, such as web browsing, is sent in the CELL_FACH state. In most networks, however, downloading a web page triggers transition to the CELL_DCH state.

Another closely related paper is by Bruynseels [3] who also uses simulations to analyze the impact of RRC timers on power consumption and user-experienced performance. While the traffic model used is web and WAP browsing, the results show that varying the timer length in the 2–10 second range can already have significant impact on power consumption, but this represents a trade-off with user browsing experience.

The only work that has explicitly considered the impact of keep-alive messages on power consumption is by Johnson et al. [9], who analyzed how the RRC timers impact power consumption with a TCP-based push e-mail where keep-alives are sent every 4 minutes. However, the work did not consider how changing the keep-alive interval affects performance, and was based on calculations using assumed power consumption figures rather than measurements.

Several other authors have used simulations to analyze how RRC timer values impact system capacity and performance in UMTS and cdma2000; for example [4], [12], and [15].

VII. Conclusions and Future Work

In this paper we have shown that keep-alive messages required by stateful middleboxes can have a considerable impact on energy consumption in WCDMA networks. Especially UDP-based protocols with long sessions can lead to unacceptably short battery lifetimes. The most important factors influencing energy consumption are the RRC parameters and frequency of keep-alive messages, and thus 3G network operators can, to some degree, influence the situation by RRC and NAT/firewall configuration.

This research suggests several questions that are in need of further investigation. For example, it would be interesting to assess the effect of different phone models, general network load, and terminal mobility. Future studies should also look into the effects of new radio improvements, such as HSDPA/HSUPA.

The results also suggest that the maintenance of connection-specific state in the network should be explicitly considered in network architecture. The current official 3GPP architecture does not include NATs or firewalls, so their effect was not taken into account in RRC, and solutions such as middlebox signaling protocols cannot be included in the architecture.

Acknowledgment

The authors would like to thank N. Asokan, Sandro Grech, Jyri Karhu, Markku Niemi, Vesa Saarinen, Mikko Siltanen, Risto Timisjärvi, and Kai Vehmanen for their valuable comments and suggestions.

References

[1] 3rd Generation Partnership Project (3GPP), "Radio Resource Control (RRC) Protocol Specification", Tech. Spec. 25.331, 2006.

[2] F. Audet and C. Jennings, "NAT Behavioral Requirements for Unicast UDP", work in progress (draft-ietf-behave-nat-udp-07), 2006.

[3] L. de Bruynseels, "Tuning Inactivity Timer Settings in UMTS", white paper, Commsquare Ltd., 2005.

[4] M. Chuah, Wei Luo, and X. Zhang, "Impacts of inactivity timer values on UMTS system capacity", in *Proc. IEEE Wireless Communications and Networking Conference*, 2002.

[5] P. Eronen, "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, 2006.

[6] S. Guha et al., "NAT Behavioral Requirements for TCP", work in progress (draft-ietf-behave-tcp-01), 2006.

[7] A. Huttunen, B. Swander, V. Volpe, L. DiBurro, and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, 2005

[8] C. Jennings and R. Mahy, "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)", work in progress (draft-ietf-sip-outbound-04), 2006.

[9] C. Johnson, S. Grech, H. Holma, and M. Kristensson, "Application and end-to-end performance", in H. Holma and A. Taskala (eds.), *HSDPA/HSUPA for UMTS*, John Wiley & Sons, 2006.

[10] Chi-Chen Lee, Jul-Hung Yeh, and Jyh-Cheng Chen, "Impact of inactivity timer on energy consumption in WCDMA and cdma2000", in *Proc. Wireless Telecommunications Symposium*, 2004.

[11] H. Levkowetz and S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices", RFC 3519, 2003.

[12] Jae-Woo So and Dong-Ho Cho, "On effect of timer object for sleep mode operation in cdma2000 system", in *Proc. IEEE International Conference on Communications*, 2000.

[13] M. Stiemerling, H. Tschofenig, C. Aoun, and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", work in progress (draft-ietf-nsis-nslp-natfw-12), 2006.

[14] M. Stiemerling, J. Quittek, and C. Cadar, "NEC's Simple Middlebox Configuration (SIMCO) Protocol Version 3.0", RFC 4540, 2006.

[15] A. Talukdar and M. Cudak, "Radio Resource Control Protocol Configuration for Optimum Web Browsing", in *Proc. IEEE 56th Vehicular Technology Conference*, 2002.

[16] J. Wigard et al., "Packet scheduling", in H. Holma and A. Taskala (eds.), *WCDMA for UMTS* (3rd ed.), John Wiley & Sons, 2004.

[17] Jui-Hung Yeh, Chi-Chen Lee, and Jyh-Cheng Chen, "Performance Analysis of Energy Consumption in 3GPP Networks", in *Proc. Wireless Telecommunications Symposium*, 2004.