

Recommendations for Reducing Power Consumption of Always-on Applications

Version 1.0; September 26, 2007

Optimization

NOKIA

Copyright © 2007 Nokia Corporation. All rights reserved.

Nokia and Forum Nokia are trademarks or registered trademarks of Nokia Corporation. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this document at any time, without notice.

License

A license is hereby granted to download and print a copy of this document for personal use only. No other license to any other intellectual property rights is granted herein.

Contents

1	Introduction	6
1.1	General.....	6
1.2	Structure of this document.....	6
2	NAT and firewall traversal	7
2.1	Network address translators.....	7
2.2	Stateful firewalls.....	8
2.3	NAT keep-alive mechanisms.....	8
2.4	NAT traversal features in VPNs and Mobile IP.....	8
2.5	Recovering from the expiring NAT mappings.....	9
2.6	Explicit firewall control protocols.....	9
2.7	Protocol-aware firewalls.....	9
2.8	IETF-specified NAT traversal solutions.....	10
2.9	Recommendations.....	10
2.9.1	NAT and firewall configuration.....	10
2.9.2	VPN and Mobile IP configuration.....	11
3	Application-level optimizations and keep-alives	12
3.1	Always-on e-mail applications.....	12
3.1.1	General.....	12
3.1.2	POP, IMAP, and SMTP.....	12
3.2	Session Initiation Protocol.....	12
3.3	Recommendations.....	13
4	Power consumption of always-on WLAN	14
4.1	Introduction to IEEE 802.11 power save.....	14
4.2	Scanning.....	15
4.3	Impact of broadcast and multicast load on idle terminals.....	16
4.4	Cost of keep-alives in WLAN.....	16
4.5	Conclusions and recommendations.....	16
4.5.1	WLAN configuration in Nokia WLAN devices.....	16
4.5.2	WLAN access point configuration.....	16
4.5.3	Address Resolution Protocol (ARP) cache configuration.....	17
4.5.4	Local multicast and broadcast load.....	17
5	Power consumption of always-on WCDMA packet data and GPRS	18
5.1	WCDMA RRC state machine.....	18
5.2	Inactivity timers of the RRC state machine.....	19
5.3	Discontinuous reception (DRX) cycle length.....	19
5.4	Periodic location update.....	19

5.5	Current consumption measurements.....	19
5.6	Conclusions and recommendations.....	20
6	Conclusions and recommendations.....	21
6.1	General conclusions.....	21
6.2	NAT/Firewall recommendations	21
6.2.1	VPN and Mobile IP configuration	21
6.3	Application-level recommendations.....	22
6.4	Wireless LAN recommendations	22
6.4.1	WLAN configuration in Nokia WLAN devices	22
6.4.2	WLAN access point configuration.....	22
6.4.3	Address Resolution Protocol (ARP) cache configuration	23
6.4.4	Local multicast and broadcast load	23
6.5	WCDMA recommendations.....	23
7	Terms and abbreviations.....	24
8	References	26
9	Evaluate this resource	28

Change history

September 26, 2007	Version 1.0	Initial document release

1 Introduction

1.1 General

This document discusses power consumption and battery life of “always-on” type of applications. The target audience of this document includes mobile operators, Wireless LAN network administrators, and application developers.

Always-on applications require the terminal to be constantly attached to a radio network and to be reachable over the current radio technology. Examples of always-on applications include Push e-mail, instant messaging, and voice and video telephony. Since IPsec virtual private network (VPN) and Mobile IP sessions may be used with always-on applications to provide security and mobility, the power efficiency of these protocols needs to be discussed, too.

This document discusses how to use WCDMA and wireless LAN technologies in a power-efficient way and gives recommendations on the configuration of WCDMA and wireless LAN radio networks. Many always-on applications need to transmit or receive frequent keep-alive messages during the idle times in order to refresh the soft state in the application servers or intermediate firewalls and network address translation (NAT) devices. The keep-alive procedures may consume energy so much that the battery lifetime will no longer be acceptable. This document describes the problem and recommends some solutions to the problem.

1.2 Structure of this document

In Chapter 2, firewalls, NATs, and the keep-alive mechanisms required to traverse these devices are discussed. It provides good background information for application developers, service providers, and operators. Chapter 3 considers the application-level keep-alive mechanisms, and is probably the most relevant chapter for application developers. Chapter 4 covers the operation of IEEE 802.11 wireless LANs in the always-on use cases. Chapter 5 discusses the power efficiency of WCDMA packet data; the configuration of the WCDMA radio resource control (RRC) protocol has a significant impact on the power consumption of terminals. Chapter 4 is especially relevant for WLAN network administrators, and Chapter 5 is relevant for mobile operators. Chapter 6 contains conclusions and recommendations.

2 NAT and firewall traversal

This chapter introduces the problem space and the current NAT traversal solutions for VPN and Mobile IP. Information on NAT traversal is good background information for application developers, service providers, and operators.

2.1 Network address translators

Network address translation (NAT) is commonly used in the IPv4 Internet. In a typical configuration, a local network uses IP addresses from one of the private IP address ranges (for example, 192.168.x.x and 10.x.x.x). A network address translator router 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 router translates the local IP address to the public address(es). Usually, the router also modifies the UDP and TCP ports.

The NAT router keeps track of active connections, so that it is able to translate “downlink” packets to the correct local addresses. The information about the active TCP connections or the internal and external UDP addresses and ports is called NAT mappings. The NAT mappings are automatically created and they are associated with a lifetime — for example, 30–60 seconds for UDP mappings and an hour for TCP. The expiry timers are reset whenever there is traffic in the active connection, so the mappings will only expire if the connection is idle for the expiry time.

Because the NAT mappings are created based on uplink packets from the terminals, servers or external correspondents are not able to initiate connections to the terminal. When NATs are used, the terminal behind the NAT must always be the initiating party.

NATs are very commonly used in residential (broadband) access networks and public access networks such as in GPRS Internet and WLAN hotspots for IPv4 traffic. NATs are not expected and recommended to be used for IPv6, since the available public IPv6 address space (globally routable IPv6 addresses) is very large. However, see Section 2.2 for a discussion of stateful firewalls. Global deployment of IPv6 is proceeding, and a large number of Nokia devices support IPv6 (IPv4/IPv6 dual stack) already.

Table 1 lists default NAT expiry timers for some products.

Product	TCP timeout	UDP timeout
Check Point NG FP2 [3]	60 min	40 s
Cisco IOS NAT [6]	1440 min	300 s
Cisco PIX [1]	60 min	120 s
Juniper Netscreen [8]	30 min	60 s
Nokia IP VPN gateway [7]	60 min	120 s
ZyXEL Prestige 660W/HW [25]	60 min	60 s
ZyXEL ZyWALL 70 [27]	150 min	180 s

Table 1: Default NAT expiry timers

2.2 Stateful firewalls

Sometimes the inability of outside hosts to initiate connections may be seen as a security benefit of NATs. External connection attempts can potentially be malicious activity. Therefore, a stateful firewall is sometimes used even when there is no need to perform address translation. For example, a certain GSM operator's Internet service, which provides clients with public IP addresses, employs stateful firewalls with a 40 second timer for UDP and an hour for TCP. Downlink packets from outside are discarded if there is no active state for the connection in the firewall, created based on a terminal-initiated packet.

It is expected that numerous public IPv6 networks employ stateful firewalls, especially if the access is charged based on the amount of traffic transferred. Hence, the deployment of IPv6 will not entirely solve the battery lifetime problems of firewalls and NATs.

2.3 NAT keep-alive mechanisms

If the terminal needs to be reachable from outside the local network, the expiration of NAT mappings during idle periods becomes a problem. To refresh the NAT mappings, many protocols include a keep-alive mechanism by which the terminal can send "dummy" packets to an outside host. The dummy packets will reset the expiry timers in the intermediate NATs and firewalls. If there is regular traffic, the mappings will be kept alive anyway, so the keep-alive packets are only needed when the terminal is idle.

2.4 NAT traversal features in VPNs and Mobile IP

RFC 3519 specifies the NAT traversal extensions for Mobile IPv4. There is UDP encapsulation to traverse NATs, and also a keep-alive mechanism. The keep-alive mechanism in Mobile IPv4 NAT traversal specifies how the terminal can send "dummy" echo requests to the home agent. The home agent acknowledges the keep-alive by responding to the terminal. The Mobile IPv4 NAT traversal protocol also specifies a mechanism by which the home agent can suggest a certain keep-alive frequency.

NAT traversal extensions for IPsec are specified in RFC 3947 and RFC 3948. Just as in Mobile IPv4 NAT traversal, the client can send dummy packets to the IPsec gateway when there is no regular traffic to send. In IPsec, the gateway is not required to acknowledge the dummy packets. The frequency is a preconfigured parameter with a default value of 20 seconds, and there is no dynamic mechanism to try to change the frequency.

The standard-based NAT traversal protocols use UDP encapsulation, since Mobile IP and IPsec tunnels can be used for both UDP and TCP based protocols, and running an UDP-based protocol over a TCP-based tunnel could result in performance problems.

Figure 1 illustrates NAT traversal with VPN. The access network uses local IP addresses, so the mobile device's lowest IP layer uses a local IP address. The NAT router translates the local address to a global Internet address. The VPN gateway sees the connection coming from the NAT device's public IP address and the UDP port, which the NAT router has allocated for this connection. The IPsec gateway has allocated another IP address for the terminal from the address space used in the intranet. TCP and application protocols run end-to-end between the mobile device and the application server.

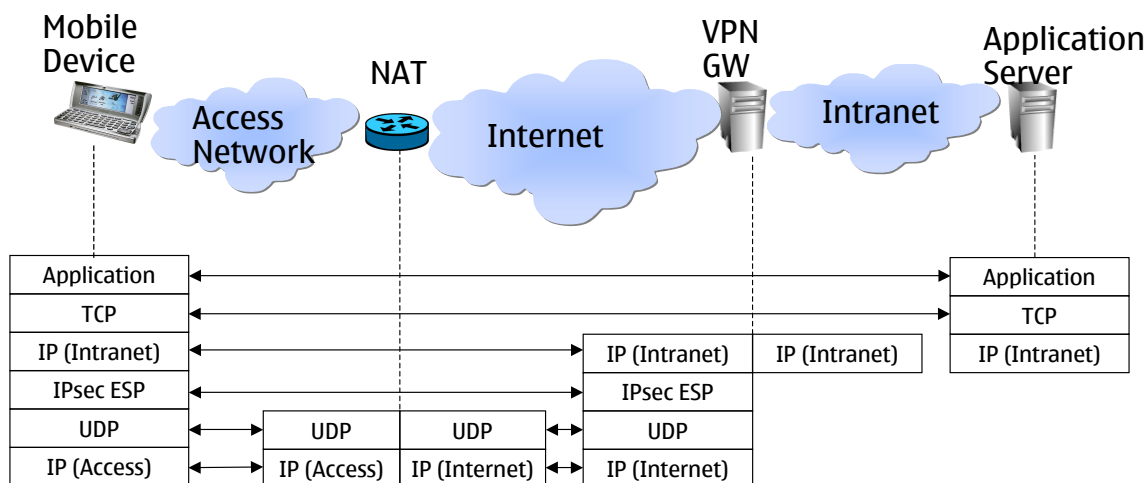


Figure 1: VPN NAT Traversal

2.5 Recovering from the expiring NAT mappings

There are cases where a NAT router decides to remove a mapping that is still alive. This can happen, for example, when the keep-alive interval is too long, or when the NAT router is rebooted. After the mapping has expired, the downlink packets sent by an outside host will no longer be able to traverse the NAT router. If the terminal sends packets to the outside host, the NAT router will create a new mapping and possibly allocate a different global IP address or port to the terminal. In this case, the (outermost) IP source address of the uplink packet, as received by the outside host, will change.

RFC 3947 recommends that the VPN gateway should recover from the change of NAT mapping by updating the address information of the connection. For Mobile IP NAT traversal, RFC 3519 specifies that if the mobile node does not receive an echo response to its NAT keep-alive, the mobile node should reregister to compensate for the loss of NAT mapping.

To sum up, the loss of a NAT mapping will always cause a temporary unreachability of the terminal from outside, but it does not need to result in the loss of the VPN or Mobile IP “session”.

2.6 Explicit firewall control protocols

Using an explicit NAT/firewall control protocol might be one way to reduce the keep-alive transmissions. In such protocols, the client uses signaling to request a mapping with a longer lifetime from the NAT device. Several such protocols exist: some existing home products support UPnP Internet Gateway Device protocol [5] or Apple’s NAT Port Mapping Protocol [10]; other similar protocols include NSIS NAT/firewall NSLP [9], and Simple Middlebox Configuration protocol [22]. However, most NATs do not yet support any such protocol.

The main reason for introducing an explicit firewall control protocol is security — the terminal has the best knowledge about the pinholes that are needed for the currently running applications. These protocols may be relevant in terms of the security in some environments, but at the moment they are not expected to become a general solution to the battery efficiency problem.

2.7 Protocol-aware firewalls

Alternatively to explicit terminal-driven firewall control protocols, it is possible to build application awareness in firewalls and have an application protocol implicitly control the firewall. For example, SIP-aware firewalls are used in IP multimedia subsystem (IMS) and other SIP networks. The firewall understands SIP signaling and opens pinholes for the negotiated media ports dynamically.

These mechanisms do not require any participation from the terminal. As long as the implicit control works correctly, there are no implications to the battery performance, other than the possibility of using long application-level keep-alive periods.

Conceivably, a firewall/NAT could be aware of IKE or Mobile IP signaling, and provide extended lifetime for its dynamic state based on preceding Mobile IP registration or IKE signaling. (Since dynamic ports are not used in Mobile IP or IKE, nearly the same effect can be achieved by configuring port-specific, longer lifetimes for MIP and IPsec UDP ports.) From the terminal's point of view, this would be transparent — the terminal would of course need to know that it is safe to apply a longer keep-alive period.

2.8 IETF-specified NAT traversal solutions

NATs present great challenges to many applications in which clients need to communicate with other clients, such as on-line gaming, peer-to-peer applications, or multimedia applications. The Internet Engineering Task Force (IETF) is working on specifying new NAT detection and NAT traversal mechanisms. Some relevant protocols and specifications are listed below:

- NAT Unicast UDP Behavioral Requirements [19] defines basic terminology for describing different types of NAT behaviors when handling Unicast UDP. It also defines a set of requirements that would allow many applications, such as multimedia communications or online gaming, to work consistently.
- Simple Traversal of UDP through NATs (STUN), [13], [23], allows a computer behind a NAT to learn its public IP address for UDP traffic and to discover the type of NAT.
- Obtaining Relay Addresses from Simple Traversal Underneath NAT [21] is a usage of STUN, called the relay usage, that allows a client to request an address on the STUN server itself, so that the STUN server acts as a relay. In other words, it allows a computer behind a NAT to be reachable over TCP and UDP via a relay. The mechanism defined in [21] was previously a standalone protocol called Traversal Using Relay NAT (TURN).
- Interactive Connectivity Establishment (ICE) [20] is a protocol for NAT traversal for multimedia sessions established with the offer/answer model. ICE makes use of the STUN protocol, applying its binding discovery and relay usages, in addition to defining a new usage for checking connectivity between peers.
- Teredo [18] enables nodes located behind one or more IPv4 network address translations (NATs) to obtain IPv6 connectivity by tunneling packets over UDP.

The use of these protocols usually involves frequent UDP keep-alives. As noted in Chapter 5, "Power consumption of always-on WCDMA packet data and GPRS," it is not feasible to send very frequent keep-alives over WCDMA and yet maintain acceptable battery performance, so the deployment of these protocols in the WCDMA environment would require special attention.

2.9 Recommendations

2.9.1 NAT and firewall configuration

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

- The longer the timers, the more memory the firewall or NAT will need for the dynamic address mappings or state information.
- The longer the timers, the longer the client's ports in question will be exposed to security attacks from the Internet.
- The shorter the timers, the shorter the keep-alive interval required by always-on applications.

In many NAT and firewall implementations, it is possible to configure the NAT expiry timers separately for different protocols, based on UDP or TCP ports. For example, the checkpoint default of 40 seconds for all UDP traffic might be unnecessarily long for DNS queries, so the timer for DNS ports could be reduced to save space in the mappings table.

For TCP ports, it is recommended to use expiry timers of at least 20 minutes, to allow application-level keep-alive intervals up to 15 minutes and account for retransmissions.

For the UDP ports that are used in IPsec and Mobile IP NAT traversal, it is recommended to use significantly longer expiry timers than for general UDP traffic. The IPsec policy and Mobile IP client's NAT keep-alive interval need to be configured accordingly. In the best case, the UDP expiry timer for IPsec should be set to over 30 minutes to allow a keep-alive interval of 15 minutes and to account for the loss of unacknowledged keep-alive packets of IPsec. For Mobile IP ports, the UDP expiry timer of circa 20 minutes should be sufficient to allow a keep-alive interval of 15 minutes, as the keep-alives are acknowledged. For IPsec NAT traversal, the UDP ports used by IPsec gateways are 500 and 4500. For Mobile IPv4 NAT traversal, the UDP port used by home agents is 434. Note that there may be several NATs in the path between the terminal and the gateway/home agent, so the NAT should not assume that the terminal would be using any specific IPsec or Mobile IP port.

2.9.2 VPN and Mobile IP configuration

This section gives recommendations for IPsec and Mobile IP clients that use the IETF standard UDP based keep-alives. In the general case, a compromise must be made between always-on battery performance and general reliability of the VPN or Mobile IP tunnel.

For IPsec VPN clients, the keep-alive interval is a configurable parameter of the client's VPN policy. For Mobile IP, the parameter can be configured in the home agent, which communicates the interval to the terminal.

If the tunnel must be as robust as possible in generic networks, without making any assumptions about special NAT or firewall configurations, then a keep-alive interval of 20 seconds should be used. This setting will not provide sufficient performance for always-on use cases, but it can be used for sessions of limited duration. During the time when the tunnel is active, the terminal will be reachable and the tunnel should be reasonably robust in general access networks with typical NAT and firewall configurations.

If VPN or Mobile IP is applied for an always-on use case, special attention must be paid to the NAT and firewall configuration. As recommended in Section 2.9.1, "NAT and firewall configuration," the expiry timers for the UDP ports used by VPN should be extended to over 30 minutes and the timers for Mobile IP ports should be extended to circa 20 minutes. The UDP keep-alive intervals of the Mobile IP and VPN clients could then be set to 15 minutes. Acceptable battery performance can be achieved, but the tunnel will be robust only in networks where the firewalls and NATs are configured accordingly.

3 Application-level optimizations and keep-alives

Many always-on applications, such as e-mail, telephony, or instant messaging applications, employ keep-alive messages in the application layer. Some always-on applications are designed to be independent from any underlying security or mobility layers. Especially when the application is run without VPN or Mobile IP, it is important to use sufficiently frequent keep-alives to keep the application client reachable by the application server.

This chapter discusses some application-level optimizations. It also gives recommendations on the configuration of application-level keep-alives.

3.1 Always-on e-mail applications

3.1.1 General

Push e-mail solutions typically use TCP as their transport protocol. When new e-mail arrives at the server, the server immediately notifies the client over the TCP connection. There is also an application-level keep-alive mechanism, typically in every 4–10 minutes, to make sure that the TCP connection is still alive and to refresh any NAT and firewall mappings.

3.1.2 POP, IMAP, and SMTP

It is also possible to implement a “stand-alone” e-mail service with the IETF standard protocols such as Internet Message Access Protocol (IMAP) [14], Post Office Protocol (POP) [11], and Simple Mail Transfer Protocol (SMTP). These protocols can be protected with Transport Layer Security (TLS), or the protocols can be run over an IPsec VPN. Switching between GPRS, WLAN, and other connections can alternatively be done at the application level (application-level roaming), or a network-level mobility solution such as Mobile IP can be used.

When an always-on e-mail is implemented with the standard protocols, the client needs to poll the server periodically. With the Nokia IMAP client, the polling period can be one of the following: 5 minutes, 15 minutes, 30 minutes, 2 hours, 4 hours, or 6 hours. The Post Office Protocol (POP) can be used alternatively to IMAP. With POP, the polling period can vary from 30 minutes to 6 hours.

The IMAP IDLE command, specified in [12], allows the client to request the server to notify when new e-mail arrives. Even when the IMAP IDLE command is used, the e-mail client needs to poll the server. IMAP IDLE is used only if the polling interval is 30 minutes or less.

If the polling interval is 30 minutes or less, the device does not close the TCP session or PDP context between the polls.

3.2 Session Initiation Protocol

The current version of Nokia’s SIP client uses the UDP transport. SIP registrations or STUN dummy packets (RFC 3489) are periodically transmitted, typically every 20 seconds. When SIP is used over WLAN, these reregistrations or keep-alives are not expected to be significant to the battery performance. However, when SIP services are to be used over WCDMA, frequent refreshing will be a problem.

IETF work relevant to NAT traversal of SIP is discussed in Section 2.8, “IETF-specified NAT traversal solutions.”

3.3 Recommendations

Generally speaking, using TCP saves power compared to UDP usage, minimizing the need for keep-alive messages; that is, TCP should be used for always-on applications if there is no specific reason for using UDP.

For always-on applications, keep-alive intervals of 10–15 minutes are recommended. Such keep-alives are expected to sufficiently refresh firewall and NAT mappings for TCP as well as access technology-specific inactivity timers when the application is run without VPNs or Mobile IP. It would be beneficial to configure the timers to even longer values — however, this requires paying special attention to the inactivity timers of PDP contexts and NATs and firewalls. For TCP-based applications, the application-level keep-alive interval should be slightly less than these inactivity timers.

UDP-based always-on applications are problematic and they always require special arrangements in the NATs and firewalls. If unacknowledged keep-alives are used, the keep-alive intervals of the application should be slightly less than half of the inactivity timers of PDP contexts and NATs and firewalls. If the protocol implements acknowledgements and retransmissions for the keep-alives, the inactivity timers only need to be slightly longer than the application-level keep-alive period.

An always-on application may be run over an always-on VPN or Mobile IP tunnel. In this case, there would typically have to be arrangements by which the keep-alive interval of Mobile IP or VPN has been extended to 10–15 minutes, in order to save power in WCDMA and 2G cellular networks. In these cases, it would be beneficial to use a slightly shorter keep-alive interval in the application layer than in the networking layer. For example, if the VPN or Mobile IP client uses a 15-minute interval, the application could use an interval of 14 minutes. The reason for this recommendation is that the keep-alive messaging of the application will usually reset the keep-alive timer of the networking layer, so that only one keep-alive will actually be sent. If the timers were configured the other way around, the terminal might end up sending two keep-alives in every ~15 minutes, since the keep-alive sent by the networking layer would not reset the keep-alive timer of the application.

4 Power consumption of always-on WLAN

While wireless LAN in general consumes less battery power per transmitted bit than GPRS or WCDMA, the power consumption when idle has not been the main focus of the WLAN standards or practical WLAN deployments. This chapter discusses the WLAN power save technologies and how to apply them for always-on applications.

4.1 Introduction to IEEE 802.11 power save

The IEEE 802.11 standards include support for a power save mode. A terminal can use an 802.11 frame to indicate to its current access point that the terminal wishes to enter the “sleep” state. The access point will record the terminal’s new state and start buffering downlink packets that are destined to the terminal. When asleep, the WLAN implementation of the terminal shuts off the transceiver and wakes up periodically to receive beacon messages from the access point. The beacons identify whether the access point has buffered incoming packets for the terminal. If there are packets, the WLAN terminal asks the access point to send the buffered packets.

Access points transmit beacons in regular intervals. The interval is usually called “beacon period” or “beacon interval” in the product documentation. The beacon period is measured in Time Units (TU) of 1024 μ s; a typical value is 100 Time Units. The Delivery Traffic Indicator Maps (DTIM) period specifies how often a terminal in power save mode should wake up to check for buffered multicast and broadcast transmissions. The DTIM period is measured in beacon intervals. For example, if the DTIM period is 2, the terminal will wake up to check for buffered packets on every other beacon.

Almost all access points use the beacon period of 100 TUs by default. Typically, the DTIM period ranges from 1 to 3.

The state diagram shown in Figure 2 illustrates IEEE 802.11 power save. When the terminal is not associated with an access point, it is obviously not able to send or receive packets. There is no paging in IEEE 802.11. While associated, the terminal can be either in the constantly awake mode (CAM) or in the power save mode (PSM). When the terminal is in the constantly awake mode, it will keep its receiver active all the time so it will be able to receive transmissions at any time.

Unlike in WCDMA, the power save states do not have any impact on the radio channels or radio resource allocations. The state transitions are fast and cheap to perform, so it is not so critical to avoid transitions to and from the power save mode. It is also acceptable to send and receive small packets while in power save, so the application’s or VPN client’s keep-alive messages are not necessarily very harmful in WLAN.

Note: See also Technical Solution [TSS000650](#) about UAPSD power save mode for VoIP applications [24].

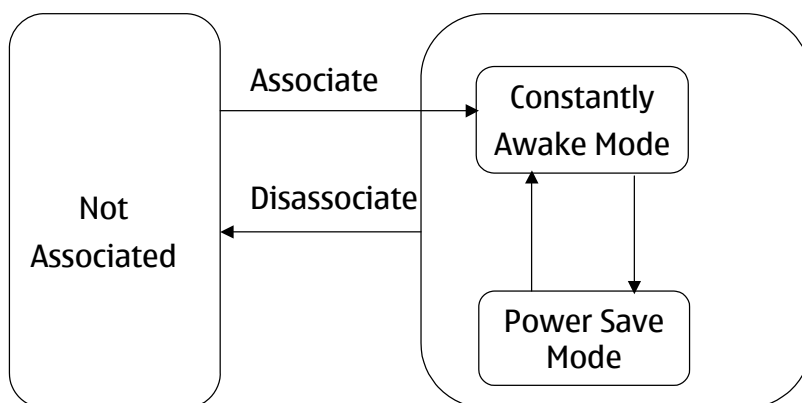


Figure 2: A high-level state diagram of IEEE 802.11 power save

The use of power save obviously increases the delays, and while in power save, the throughput may be reduced. Some older access points do not even buffer broadcast and multicast traffic, and the mobile device might not wake up on every DTIM period, so the WLAN terminal will not necessarily receive broadcast and multicast transmissions while on power save. Typically, multicast and broadcast are not used/needed for always-on applications when the terminal is idle, so this is generally not a problem. However, ARP requests and DHCP packets are broadcast. See Section 4.5.3, “Address Resolution Protocol (ARP) cache configuration,” for more discussion on ARP. The terminal can transmit packets while in the power save mode. Many implementations try to select between active mode and power save based on the amount of traffic that there has been recently.

The state transitions between CAM and PSM in the Nokia WLAN client implementation depend on the activation of the user interface. The implementation is in one of the following modes:

1. **Power save disabled** in the settings: In this case, the device stays in the Constantly Awake Mode.
2. **PSM_dynamic**: When power save is enabled, and when the user interface is active, the device is in the dynamic power save mode (PSM_dynamic). In PSM_dynamic, the device can constantly change its 802.11 state between CAM and PSM depending on the traffic. The algorithm varies a bit between Nokia device models, so the simplest version is described here. The CAM time-out parameter (in Time Units), which is not configurable in the UI and defaults to 100 TUs, specifies the inactivity timer for downlink packets, after which the device moves from CAM to PSM. In other words, the device sleeps during data transfer if there are breaks of 100 TUs or longer in downlink direction. The device sends uplink packets immediately, and listens to beacons and uses the DTIM parameter as usual in 802.11.
3. **PSM_powersave**: When power save is enabled and the user interface is inactive, the device stays constantly in the power save mode. With DTIM periods of 3 or higher, the device wakes up, as usual, at every DTIM period. However, if the access point uses a DTIM period 1 or 2, the device skips beacons and does not wake up at every DTIM period to save power. This may result in lost multicast and broadcast frames.

Power save is supported in Nokia wireless terminals. In general, WLAN implementations do not support power save in the ad-hoc mode.

4.2 Scanning

Before the terminal can join a WLAN network, the terminal needs to discover the network by scanning. Nokia devices can be configured to perform background scanning also when there is no active WLAN connection. In Nokia’s implementation, the background scanning interval ranges from 1 minute to 10 minutes. The terminal will also scan for better access points during an active WLAN association.

Background scanning consumes battery power. The scanning interval has a direct impact on battery efficiency. In addition, the availability of WLAN networks and the use of hidden SSIDs may influence the battery performance of WLAN scanning.

4.3 Impact of broadcast and multicast load on idle terminals

Unicast transmissions to other WLAN devices will not wake up an idle device so having a busy network with many unicast streams should not have any impact on the idle power consumption.

However, the idle power consumption can change dramatically if there are a lot of broadcast or multicast transmissions in the local network. As explained above, all WLAN devices, including idle devices, receive all the multicast and broadcast transmissions. Microsoft NetBIOS, DHCP, ARP, UPnP, and the control protocols of local bridges, switches, and access points are examples of protocols that use broadcast or multicast.

The amount of broadcast and multicast traffic depends on the configuration of local nodes and on the size of the subnetwork. In some cases, the terminal can receive several megabytes of small broadcast/multicast packets per hour. Even though the terminal would eventually ignore all the transmissions, the need to wake up frequently will have a significant effect on the battery performance.

Therefore, it is important to pay attention to the broadcast and multicast transmissions in your wireless LAN network. The local sources of broadcast and multicast transmissions should be analyzed and the local devices should be configured to avoid any unnecessary broadcast and multicast “noise.” For example, configuring the NetBIOS implementations of local hosts to use unicast (for example, through the node-type value from local DHCP servers) can reduce the broadcast traffic significantly.

4.4 Cost of keep-alives in WLAN

According to measurements, the current consumption of a single sent unacknowledged “keep-alive” message in WLAN is approximately 0.003–0.005 mAh. The cost of a single keep-alive is so low that having a keep-alive interval of 20 seconds will increase the idle current less than 1 mA. Hence, (short) higher-layer keep-alive packets that are sent no more frequently than every 20 seconds are not expected to be significant to the power consumption.

4.5 Conclusions and recommendations

It is possible to get an acceptable WLAN power consumption even with always-on use. It should still be noted that with small DTIM values or with excessive background scanning, the WLAN idle consumption is significant and it can compromise the battery performance of the device. Recommendations for the WLAN device, access point, and local routers are discussed in this section.

4.5.1 WLAN configuration in Nokia WLAN devices

WLAN power save should be enabled in the terminal. In order to use automatic roaming to WLAN, background scanning needs to be enabled. For many always-on applications, it is sufficient to use a long background scanning interval such as 5 or 10 minutes.

4.5.2 WLAN access point configuration

The product of beacon period and DTIM period defines the wake-up frequency of the terminals in the power-save mode. Thereby these values have a major impact on the current consumption performance of always-on WLAN connections.

To get good battery performance, it is recommended that the DTIM period should be at least 3 or more preferably 5 with the typical beacon period of 100 Time Units.

Since the use of hidden SSIDs makes it more expensive for the terminal to discover the network by scanning, using hidden SSIDs is not recommended. If hidden SSIDs cannot be completely avoided, the number of different hidden SSIDs needed by the terminal should be as low as possible.

4.5.3 Address Resolution Protocol (ARP) cache configuration

As discussed above, multicast and broadcast packets are not always reliably delivered to the terminal when the terminal is in the power save mode. This limitation may be relevant in cases when long keep-alive periods are used, which might cause the Address Resolution Protocol (ARP) cache entries or IPv6 neighbor discovery cache entries in the local routers to expire. In this case, a downlink packet arriving at the local WLAN network will first trigger the router to broadcast an ARP request or multicast a neighbor solicitation for the terminal's IP address. If this transmission is lost, the terminal will not be reached. This can be avoided by making sure that the ARP cache and IPv6 neighbor discovery cache expiry timers in the local routers are longer than the keep-alive intervals used by the terminal. The default values of the ARP cache expiry timer vary a lot depending on the vendor of the networking device.

In many cases, the keep-alive periods used by applications or networking layer protocols can be up to 15 minutes, and in the worst case the keep-alive message is UDP-based and unacknowledged. Therefore, ARP cache and IPv6 neighbor cache expiry timers of over 30 minutes are recommended for both local routers and the WLAN client.

Some Wireless LAN access points implement a proxy ARP feature and reply to ARP requests on behalf of associated wireless clients. When such a feature is used, it is not important to extend the ARP expiry timers of local routers.

4.5.4 Local multicast and broadcast load

Network administrators are advised to analyze the multicast and broadcast transmissions in the local WLAN network, since the multicast and broadcast traffic will wake up idle terminals and cause an increase in the idle power consumption. Any multicast or broadcast transmissions that can be avoided by means of configuration — such as the NetBIOS node-type — should be disabled in the local network.

5 Power consumption of always-on WCDMA packet data and GPRS

This chapter discusses the power consumption in WCDMA packet data networks and compares it to 2G networks. The power consumption of idle terminals depends greatly on how the radio resource control (RRC) state machine is configured in the network.

5.1 WCDMA RRC state machine

Figure 3 illustrates the radio resource control states in WCDMA packet data. The device is in one of the following states:

- **CELL_DCH** (dedicated channel): In this state, the current consumption is at its highest, comparable to the consumption during circuit-switched voice calls. The device has a dedicated channel, which it does not share with other devices, so maximum throughput and minimum delay are achieved.
- **CELL_FACH** (forward access channel): In this state, the device shares the channel with other devices. 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 of around 1–2 percent of the consumption in CELL_DCH state. If there are downlink packets for the terminal, the terminal will be paged. In this state, the terminal is not able to send or receive packets, but the terminal will have to enter either the CELL_DCH or CELL_FACH state to send or receive. Not all network implementations currently use the CELL_PCH state. In the future, a new state called **URA_PCH** will be introduced, which provides the same benefits as CELL_PCH and further enhances the battery performance when there is mobility.
- **Idle mode**: In this state, the device does not have an RRC connection, so it is not possible to send or receive packets in this state. The terminal can still have a PDP context and it can be reached by paging procedures, after which the terminal can leave the idle mode and receive downlink packets. However an RRC connection will have to be established before the downlink packets can be received. Usually, the terminal will use the CELL_DCH state to send keep-alives. Hence, although the current consumption is similar to the CELL_PCH state in the idle mode, the keep-alives and incoming e-mails will consume a lot more power. Some networks support the state transition from idle mode directly to CELL_FACH.

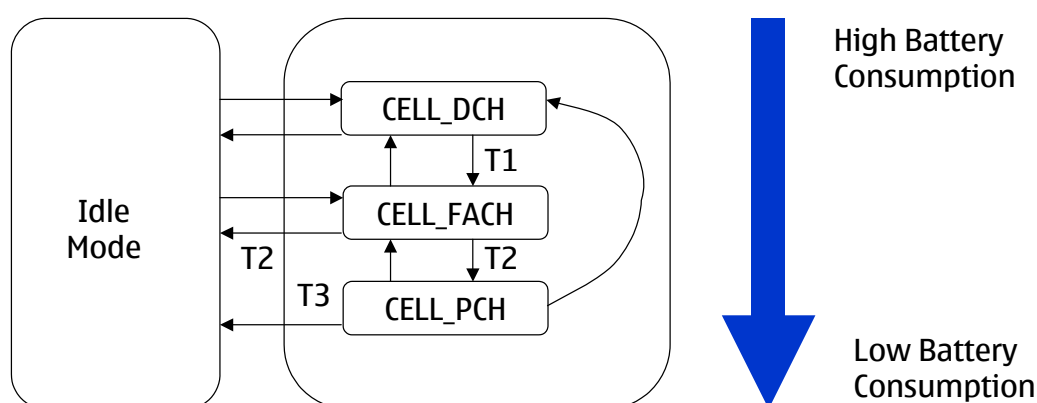


Figure 3: Overview of WCDMA radio resource control (RRC) state machine

5.2 Inactivity timers of the RRC state machine

State transitions are based either on explicit signaling or inactivity timers. The timers T1, T2, and T3 are shown in Figure 3. The names T1, T2, and T3 are not officially used in 3GPP specifications but they have been established in WCDMA parlance. The timers are network controlled and managed by the radio network controller (RNC). The timers are discussed below:

- **T1** is an inactivity timer that is used in the CELL_DCH state. This timer is reset whenever there is traffic. The timer will expire only after an inactive period of T1, and the terminal will enter the CELL_FACH state. The shorter the T1 timer, the worse the user experience will be, for example, in Web browsing. The T1 value may depend on the DCH data rate. The default values used in the Nokia RNC implementation are 5 seconds for 8–32 kbit/s, 3 seconds for 128 kbit/s, and 2 seconds for data rates greater than 128 kbit/s. In some networks, significantly longer timers than the Nokia defaults may be used.
- **T2** is an inactivity timer in the CELL_FACH state. If CELL_PCH is used, the state machine will enter the CELL_PCH state after an inactivity period of T2. If CELL_PCH is not used, the state machine will enter the idle state. The default value in Nokia's implementation is 2 seconds, but often significantly longer T2 values are used.
- **T3** is a timer used in CELL_PCH (and in the URA_PCH state that may be introduced in the future). After staying in the CELL_PCH or URA_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 device transitions from the more power-consuming states to less consuming states. The sum T1+T2 defines the general power consumption behavior of the device, and the value of T1 has a significant effect on the perceived performance of several applications. Since short keep-alive packets can often be transmitted in the CELL_FACH state, the value of T2 accounts for most of the idle battery performance in those networks where CELL_PCH or idle-to-CELL_FACH transitions are available.

5.3 Discontinuous reception (DRX) cycle length

When the terminal is in the idle mode, CELL_PCH state, or URA_PCH state, its main task is to monitor the paging channel and respond to any pages received. During those periods when the terminal is not required to monitor any channels, the terminal powers off its receiver to conserve power. This mechanism is referred to as the discontinuous reception (DRX) mode. The time between two successive times when the terminal needs to power on its receiver is called the DRX cycle length. The DRX cycle length has an impact on the battery performance of the terminal.

The DRX cycle length is generally a second or a couple of seconds. Increasing the DRX cycle length improves the stand-by battery performance, but at the same time it increases the paging delays and thereby increases the packet roundtrip delays and call setup delays.

5.4 Periodic location update

The time interval between periodic location updates has an impact on the battery performance. A very frequent update period would result in a significant increase in the power consumption.

5.5 Current consumption measurements

Refer to [4] for information on WCDMA current consumption measurements.

5.6 Conclusions and recommendations

To get an acceptable battery performance, the target average current for always-on applications should be 8 mA or less. The following conclusions can be drawn:

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

The power performance in 2G networks is generally tolerable, especially if the keep-alive interval is 150 seconds or longer. It is recommended to ensure that the radio network and PDP contexts tolerate at least a period of 10–15 minutes without disconnecting the terminal.

For 3G networks, the general conclusion is that with short application-level or networking-level keep-alive intervals, the battery performance is often not acceptable. Tolerable performance can be achieved if the CELL_PCH state is enabled. The operator is recommended to enable the CELL_PCH state or the URA_PCH state whenever provided by the RNC vendor. Alternatively, enabling the state transitions from the idle mode to CELL_FACH is recommended whenever provided by the RNC vendor.

Generally speaking, the shorter the T1 and T2 timers, the better the battery performance of always-on applications will be. In non-CELL_PCH networks, the sum of T1+T2 determines much of the idle power consumption, and, in networks where either CELL_PCH or idle-to-CELL_FACH transitions are supported, 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 then degrade other performance metrics, such as the perceived browsing performance. The default values used in the Nokia RNC — with T1 ranging from 2 to 5 seconds depending on data rate, and T2 = 2 seconds — provide adequate battery performance for always-on applications. The timer T3 should be long enough, preferably more than 10–15 minutes, to allow for long keep-alive periods.

Using DRX cycle lengths of at least 1.28 s or 2.56 s is recommended to reduce the standby power consumption in WCDMA.

Periodic location updates (timer T3212) should be configured to a period of at least 30 minutes.

6 Conclusions and recommendations

6.1 General conclusions

Acceptable power consumption for TCP-based always-on applications can be achieved. Generally speaking, using TCP saves power significantly compared to UDP usage, minimizing the need for keep-alive messages; that is, TCP should be used for always-on applications if there is no specific reason for using UDP. For the VPN and Mobile IP protocols, acceptable operation in the general case, especially over WCDMA, cannot be guaranteed, since these protocols use UDP encapsulation. If firewalls and NATs are in the control of the organization that is deploying the VPN or Mobile IP solutions, it may be possible to get acceptable results by paying special attention to the configuration of the NATs, firewalls, and the VPN policy.

For running always-on applications over WLAN, the main factors influencing the current consumption are the access point DTIM period configuration and the background scanning settings of the terminal. The frequency of higher-layer keep-alives does not play a significant role in WLAN.

For running always-on applications over WCDMA, the main factors influencing the current consumption are the WCDMA RRC parameters (inactivity timers, whether keep-alives can be sent in CELL_FACH state, and DRX cycle length), and the frequency of application or networking-level keep-alives.

6.2 NAT/Firewall recommendations

In many NAT and firewall implementations, it is possible to configure the NAT expiry timers separately for different protocols, based on UDP or TCP ports.

For TCP ports, it is recommended to use expiry timers of at least 20 minutes, to allow application-level keep-alive intervals up to 15 minutes and account for retransmissions.

For the UDP ports that are used in IPsec and Mobile IP NAT traversal, it is recommended to use significantly longer expiry timers than for general UDP traffic. The IPsec policy and Mobile IP client's NAT keep-alive interval need to be configured accordingly. In the best case, the UDP expiry timer for IPsec should be set to over 30 minutes to allow a keep-alive interval of 15 minutes and to account for the loss of unacknowledged keep-alive packets of IPsec. For Mobile IP ports, the UDP expiry timer of circa 20 minutes should be sufficient to allow a keep-alive interval of 15 minutes, as the keep-alives are acknowledged. For IPsec NAT traversal, the UDP ports used by IPsec gateways are 500 and 4500. For Mobile IPv4 NAT traversal, the UDP port used by home agents is 434. Note that there may be several NATs in the path between the terminal and the gateway/home agent, so the NAT should not assume that the terminal would be using any specific IPsec or Mobile IP port.

6.2.1 VPN and Mobile IP configuration

In the general case, a compromise must be made between always-on battery performance and general reliability of the VPN or Mobile IP tunnel. For IPsec VPN clients, the keep-alive interval is a configurable parameter of the client's VPN policy. For Mobile IP, the parameter can be configured in the home agent, which communicates the interval to the terminal.

If the tunnel must be as robust as possible in generic networks, without making any assumptions about special NAT or firewall configurations, then a keep-alive interval of 20 seconds should be used. This setting will not provide sufficient performance for always-on use cases, but it can be used for sessions of limited duration. During the time when the tunnel is active, the terminal will be reachable

and the tunnel should be reasonably robust in general access networks with typical NAT and firewall configurations.

If VPN or Mobile IP is applied for an always-on use case, special attention must be paid to the NAT and firewall configuration. The expiry timers for the UDP ports used by VPN should be extended to over 30 minutes and the timers for Mobile IP ports should be extended to circa 20 minutes. The UDP keep-alive intervals of the Mobile IP and VPN clients could then be set to 15 minutes. Acceptable battery performance can be achieved, but the tunnel will be robust only in networks where the firewalls and NATs are configured accordingly.

6.3 Application-level recommendations

For always-on applications, keep-alive intervals of 10–15 minutes are recommended. Such keep-alives are expected to sufficiently refresh firewall and NAT mappings for TCP as well as access technology-specific inactivity timers, when the application is run without VPNs or Mobile IP. It would be beneficial to configure the timers to even longer values — however, this requires paying special attention to the inactivity timers of PDP contexts and NATs and firewalls. For TCP-based applications, the application-level keep-alive interval should be slightly less than these inactivity timers.

UDP-based always-on applications are problematic and they always require special arrangements in the NATs and firewalls. If unacknowledged keep-alives are used, the keep-alive intervals of the application should be slightly less than half of the inactivity timers of PDP contexts and NATs and firewalls. If the protocol implements acknowledgements and retransmissions for the keep-alives, the inactivity timers only need to be slightly longer than the application-level keep-alive period.

An always-on application may be run over an always-on VPN or Mobile IP tunnel. In this case, there would typically have to be arrangements by which the keep-alive interval of Mobile IP or VPN has been extended to 10–15 minutes, in order to save power in WCDMA and 2G cellular networks. In these cases, it would be beneficial to use a slightly shorter keep-alive interval in the application layer than in the networking layer. For example, if the VPN or Mobile IP client uses a 15-minute interval, the application could use an interval of 14 minutes. The reason for this recommendation is that the keep-alive messaging of the application will usually reset the keep-alive timer of the networking layer, so that only one keep-alive will actually be sent. If the timers were configured the other way around, the terminal might end up sending two keep-alives in every ~15 minutes, since the keep-alive sent by the networking layer would not reset the keep-alive timer of the application.

6.4 Wireless LAN recommendations

It is possible to get an acceptable WLAN power consumption even with always-on use. However, with small DTIM values or with excessive background scanning, the WLAN idle consumption is significant and it can compromise the battery performance of the device.

6.4.1 WLAN configuration in Nokia WLAN devices

WLAN power save should be enabled in the terminal. In order to use automatic roaming to WLAN, background scanning needs to be enabled. For many always-on applications, it is sufficient to use a long background scanning interval such as 5 or 10 minutes.

6.4.2 WLAN access point configuration

The product of beacon period and DTIM period defines the wake-up frequency of the terminals in the power-save mode. Thereby these values have a major impact on the current consumption performance of always-on WLAN connections.

To get good battery performance, it is recommended that the DTIM period should be at least 3 or more preferably 5 with the typical beacon period of 100 Time Units.

Since the use of hidden SSIDs makes it more expensive for the terminal to discover the network by scanning, using hidden SSIDs is not recommended. If hidden SSIDs cannot be completely avoided, the number of different hidden SSIDs needed by the terminal should be as low as possible.

6.4.3 Address Resolution Protocol (ARP) cache configuration

ARP cache and IPv6 neighbor cache expiry timers of over 30 minutes are recommended for both local routers and the WLAN client. Some Wireless LAN access points implement a proxy ARP feature and reply to ARP requests on behalf of associated wireless clients. When such a feature is used, it is not important to extend the ARP expiry timers of local routers.

6.4.4 Local multicast and broadcast load

Network administrators are advised to analyze the multicast and broadcast transmissions in the local WLAN network, since the multicast and broadcast traffic will wake up idle terminals and cause an increase in the idle power consumption. Any multicast or broadcast transmissions that can be avoided by means of configuration — such as the NetBIOS node-type — should be disabled in the local network.

6.5 WCDMA recommendations

The power performance in 2G networks is generally tolerable, especially if the keep-alive interval is 150 seconds or longer. It is recommended to ensure that the radio network and PDP contexts tolerate at least a period of 10–15 minutes without disconnecting the terminal.

For 3G networks, the general conclusion is that with short application-level or networking-level keep-alive intervals, the battery performance is often not acceptable. Tolerable performance can be achieved if the CELL_PCH state is enabled. Enabling the CELL_PCH state or the URA_PCH state is recommended whenever provided by the RNC vendor. Alternatively, enabling the state transitions from the idle mode to CELL_FACH is recommended whenever provided by the RNC vendor.

Generally speaking, the shorter T1 and T2 timers are, the better the battery performance of always-on applications will be. In non-CELL_PCH networks, the sum of T1+T2 determines much of the idle power consumption, and, in networks where either CELL_PCH or idle-to-CELL_FACH transitions are supported, 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 then degrade other performance metrics, such as the perceived browsing performance. The default values used in the Nokia RNC — with T1 ranging from 2 to 5 seconds depending on data rate, and T2 = 2 seconds — provide adequate battery performance for always-on applications. The timer T3 should be long enough, preferably more than 10–15 minutes, to allow for long keep-alive periods.

Using DRX cycle lengths of at least 1.28 s or 2.56 s is recommended to reduce the standby power consumption in WCDMA.

Periodic location updates (timer T3212) should be configured to a period of at least 30 minutes.

7 Terms and abbreviations

Term or abbreviation	Meaning
AP	Access point
ARP	Address Resolution Protocol; a protocol for resolving link-layer addresses based on IPv4 addresses.
CAM	Constantly Awake Mode; in IEEE 802.11 wireless LANs, the receiver is kept constantly powered on in this mode.
DCH	Dedicated channel
DHCP	Dynamic Host Configuration Protocol
DNS	Domain name service, a protocol for resolving IP addresses based on host names.
DRX	Discontinuous reception
DTIM	Delivery Traffic Indicator Maps, used to indicate WLAN terminals about buffered downlink packets.
FACH	Forward access channel
IKE	Internet key exchange, a signaling protocol used by IPsec virtual private network solutions.
IMAP	Internet Message Access Protocol, one of the Internet protocols for receiving e-mail.
IMS	IP multimedia subsystem
IPsec	Internet protocol security; a security extension to Internet Protocol (IP), used in many virtual private network solutions.
MIP	Mobile IP, a mobility extension to Internet Protocol (IP).
NAT	Network address translation
PCH	Paging channel
POP	Post Office Protocol, one of the Internet protocols for receiving e-mail.
PSM	Power Save Mode; an operating mode in IEEE 802.11 wireless LANs where the client can keep its receiver powered off for certain periods of time.
RNC	Radio network controller; in WCDMA, the RNC is responsible for radio resource control in its service area.
RRC	Radio resource control
SMTP	Simple Mail Transfer Protocol, an Internet protocol for sending e-mail.
SSID	Service Set Identifier; WLAN network name string used in WLAN network discovery to identify the network.
TCP	Transmission Control Protocol; a transport protocol in the Internet protocol suite.
TLS	Transport Layer Security

Term or abbreviation	Meaning
TU	Time Unit (1024 μ s)
UDP	User Datagram Protocol; a transport protocol in the Internet protocol suite.
VPN	Virtual private network

8 References

- [1] Cisco Systems, "ADSM Online Help, Release 5.0(1)," May 2005.
- [2] Jeff Doyle, Jennifer DeHaven Carroll, "Network Address Translation," Feb 8, 2002, Cisco web pages, <http://www.ciscopress.com/articles/article.asp?p=25273&seqNum=5&rl=1>
- [3] Check Point Software Technologies, "Check Point Management Guide, NG FP2," Part No. 700348, March 2002.
- [4] Henry Haverinen, Jonne Siren, Pasi Eronen: "Energy Consumption of Always-On Applications in WCDMA Networks," IEEE Vehicular Technology Conference, April 2007.
- [5] UPnP Forum, "Internet Gateway Device (IGD) Standardized Device Control Protocol V 1.0," November 2001.
- [6] Cisco Systems, "Cisco IOS IP Addressing Services Command Reference, Release 12.4," June 2005.
- [7] Nokia Corporation, "Nokia IP VPN Gateway Configuration Guide, Version 6.3," Part No. N451439003 Rev A, June 2005.
- [8] Juniper Networks, "NetScreen CLI Reference Guide," Version 5.2.0, P/N 093-1592-000 Rev. A, May 2005.
- [9] Stiernerling, M., Tschofenig, H., Aoun, C., and Davies, E.: "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)," work in progress, Internet-Draft draft-ietf-nsis-nslp-natfw-14, March 2007.
- [10] Stuart Cheshire, Marc Krochmal, Kiren Sekar, "NAT Port Mapping Protocol (NAT-PMP)," work in progress, Internet-Draft draft-cheshire-nat-pmp-00, June 2005.
- [11] J. Myers, "Post Office Protocol," IETF RFC 1939, May 1996.
- [12] B. Leiba, "IMAP4 IDLE Command," IETF RFC 2177, June 1997.
- [13] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN- Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.
- [14] M. Crispin, "Internet Message Access Protocol – Version 4rev1," IETF RFC 3501, March 2003.
- [15] H. Levkowitz, S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices," RFC 3519, April 2003.
- [16] T. Kivinen et al., "Negotiation of NAT-Traversal in the IKE," IETF RFC 3947, January 2005.
- [17] A. Huttunen et al., "UDP Encapsulation of IPsec ESP Packets," IETF RFC 3948, January 2005.
- [18] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," IETF RFC 4380, February 2006.
- [19] F. Audet, C. Jennings, "NAT Behavioral Requirements for Unicast UDP," IETF RFC 4787, January 2007.
- [20] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," draft-ietf-mmusic-ice-16, June 2007, work in progress.

- [21] Rosenberg, J., Mahy, R., Huitema, C., "Obtaining Relay Addresses from Simple Traversal Underneath NAT (STUN)," draft-ietf-behave-turn-03 (work in progress), March 2007.
- [22] Martin Stiemerling, Juergen Quittek and Christian Cadar, "Simple Middlebox Configuration (SIMCO) Protocol Version 3.0," work in progress, Internet-Draft draft-stiemerling-midcom-simco-07, May 2005.
- [23] Rosenberg, J., Huitema, C., Mahy, R., Wing, D., "Session Traversal Utilities for (NAT) (STUN)," draft-ietf-behave-rtc3489bis-06 (work in progress), March 2007.
- [24] Technical Solution TSS000650 - Increasing WLAN power efficiency for full-duplex VoIP and Video applications, [http://wiki.forum.nokia.com/index.php/TSS000650 -
Increasing WLAN power efficiency for full-duplex VoIP and Video applications](http://wiki.forum.nokia.com/index.php/TSS000650_-_Increasing_WLAN_power_efficiency_for_full-duplex_VoIP_and_Video_applications)
- [25] ZyXEL Communications Corporation, "Prestige 660W/HW Series ADSL Gateway with 802.11g Wireless User's Guide," Version 3.04, June 2004.
- [26] ZyXEL CI Command reference, http://www.zyxel.com/support/supportnote/p314/ci_cmd/p314_ci.htm
- [27] ZyXEL Communications Corporation, "ZyWALL 3/35/70 Series Internet Security Appliance User's Guide," Version 4.00, August 2005.

9 Evaluate this resource

Please spare a moment to help us improve documentation quality and recognize the resources you find most valuable, by [rating this resource](#).