

## Chapter 2

# ALWAYS ON SERVICE INTELLIGENT NETWORK

SARIT MUKHERJEE, SANJOY PAUL, KRISHAN SABNANI

*Bell Laboratories Research, 101 Crawfords Corner Road, Holmdel, NJ 07733, USA*

**Abstract:** With the popularity of services like Push-to-Talk, the need for “always on” services is becoming important for service providers. This paper addresses the problem of supporting always on services in existing and new network architectures. It defines the requirements of always on services, identifies the problems in supporting such services, and proposes an overlay network based solution to make always on services real. Some results from initial prototyping and experimentation are also presented to demonstrate the feasibility of deploying such services.

## 1. INTRODUCTION

Overwhelming success of Instant Messaging (IM) made it clear that there is an inherent appeal in being able to communicate “instantly” at the click of a mouse. This was further confirmed by the huge success of Nextel’s Push-to-Talk (PTT) feature which enabled Nextel subscribers to communicate using a nation-wide walkie-talkie at the push of a button. Note that PTT succeeded even when people could call anyone anytime from their cell-phones. There are several reasons why PTT is preferred over a phone call:

- PTT connection time is less than a second while the connect time for a cell-phone call can be as much as 5 seconds, if not more.
- PTT does not require the user to do anything more than pushing a button as opposed to dialing a phone number.
- PTT uses Voice over IP (VoIP) instead of circuit-switched voice and hence is cheaper and feature rich.

The critical observation is that both in IM and in PTT, the user is “always on” [1] in the sense that the user is always connected to the network; network knows the presence of the user and hence keeps the user’s state and the user knows it is connected to the network and hence does not need to explicitly dial the network to set up a call.

While the term “always on” has become popular, there is no formal definition of “always on” in the technical community. We define a service to be always on if:

User:

- (i) does not feel disconnected anytime, anywhere
- (ii) can access the service instantly regardless of location and time
- (iii) once connected receives desired quality of service and improved experience
- (iv) is always in control; is always protected and is always engaged [2].

Network:

- (i) can reach an end-user instantly at any time regardless of the user’s location
- (ii) can push info to end-user regardless of the state of the mobile (active/ dormant)

## 1.1 Motivation and Problem

While the concept of being always on seems inherently appealing, it becomes increasingly difficult to provide such a perception to a user in a mobile wireless network because of the following reasons:

- (i) Choppy network connection: Wireless links are inherently error-prone because of multi-path and fading. A noisy link leads to connection drop making always on a difficult proposition.
- (ii) Inadequate coverage: All areas are not covered equally well by a wireless carrier. Therefore, a mobile user can get disconnected once it moves from an area of better to a poorer coverage.
- (iii) Mobility in dormant state: Locating the user becomes difficult especially when the mobile is dormant and as a result when a dormant mobile becomes active far from where it went dormant, the network has to spend some time locating the user and that makes “instant” connection almost impossible.

The challenges can then be summarized as:

- (i) Reduction of connection time: The network must maintain user's state and should not tear it down even when the user is dormant. In addition, the network must keep track of the user regardless of whether the mobile is active or dormant and keep transferring its state to the closest network element, if, need be, so that connections are never broken.
- (ii) Improving the quality: It is not enough just to have "continuous" connectivity because a poor-quality connection would leave the user dissatisfied. Thus the goal in always on service is to provide and maintain "improved" service quality. Moreover, the same quality should be maintained for the user even when the mobile comes out of dormancy.
- (iii) Richer functionality: Since the network knows the user's location and presence it would be possible to provide an "enhanced" experience to the user or to provide the user some service that is not otherwise possible.
- (iv) Pushing information: In contrast to existing mechanisms in which the user initiates a connection to the network, the network has to initiate a connection to the user. This requires additional "smarts" within the network.

## **2. SHORTCOMINGS OF EXISTING NETWORK ARCHITECTURE**

Existing networks are not built to support always on services. Figure 1 shows the typical steps a client, network and a server go through before executing any transaction in a legacy network (without always on) and in an always-on network. For example, in a legacy network, the client goes through a set of device-related steps such as powering up the device, booting up the operating system; network-related steps such as setting up connection (PPP connection), authenticating the user; and application-related steps such as starting the application (such as Outlook or Internet Explorer), connecting to the server (Outlook server or eBay server for example), authenticating with the server and accessing the service. The network also goes through air-interface related steps such as selecting and acquiring the channel; device-related steps such as authenticating the device (using IMSI or device-specific characteristics) and registering the device; network-related steps such as setting up link-level connection (PPP connection), and allocating IP

addresses. The server authenticates the user, sets up connection and responds to client request. All these steps in a legacy network add up to several minutes and as a result, the user feels “disconnected” and the process of doing a transaction becomes very cumbersome. Always on network simply eliminates many of these time-consuming steps (shown by striking out itemized steps in Figure 1) resulting in a minimal number of steps (shown at the lower part of Figure 1) that are needed for connecting a client to a service. As a consequence, the process of connecting to a service is simplified and the time is reduced to a few seconds or even less than a second in some cases and hence the user does not feel disconnected.

As shown in Figure 1, to make a service always on, each one of the client, network and the server has to do something to eliminate delays. This paper focuses on the network aspects.

There are several other drawbacks of the existing networks:

- (i) Inadequate wireless coverage leads to disconnection: When a mobile enters a tunnel or a building where the wireless network signal is not strong, the data connection drops. The user has to go through the entire connection establishment phase once he gets out of the low-coverage area.
- (ii) Connection state maintenance: In order to provide always on connection, the network has to maintain state of the mobile’s connections even when the mobile is not active. This implies that a carrier’s network has to potentially maintain states of as many mobiles as there are subscribers. This number can potentially go into tens of millions and the current network elements are not equipped to handle that.

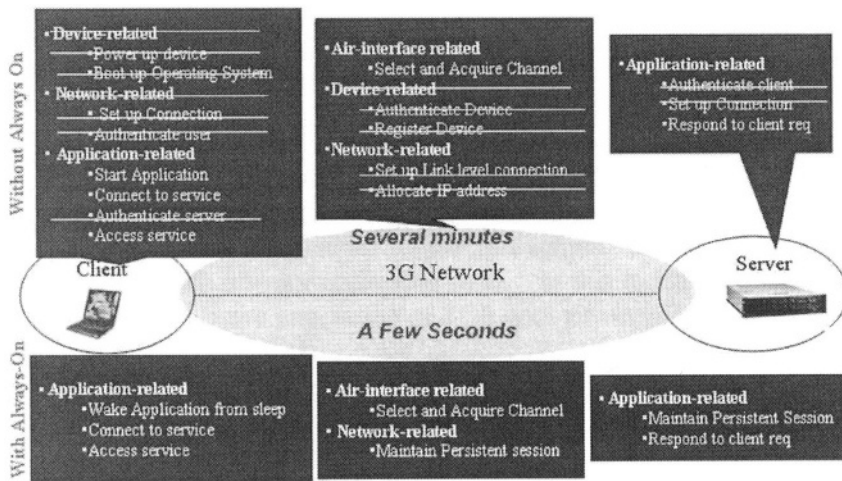


Figure 1 Always On eliminates time-consuming operations of a Legacy Network

- (iii) Making applications “always on” one at a time: In order to enable always on service, the carriers are putting intelligence in the servers. For example, Qualcomm’s QChat server which is used to provide PTT service maintains user’s state and location information. Instead of incorporating the same features in each server independently, it makes more sense to incorporate them once and for all in a gateway that every server (potentially belonging to different services) can leverage.
- (iv) Making applications “always on” one Network at a time: While carriers are making applications always on, the scope of always on is limited to a single carrier’s network and also it is limited to one “type” of network. For example, Nextel’s PTT service works only within Nextel’s network and it does not work when the mobile moves into an area covered by WiFi hotspots but not covered by Nextel’s network. Ideally, the always on service should be provided across different types of networks (3G, WiFi, 4G) and across different carriers’ networks. Open Mobile Alliance (OMA) consortium is pushing Push to Talk over Cellular (PoC) [10] which is an open standard using SIP signaling

protocol aimed at making Push to Talk interoperable across carriers. However, PoC is not without limitations either. First, the participants in the PoC effort have different views. For example, Nextel wants proprietary technology within a carrier's network and wants to standardize across the carriers' networks at the edge. Ericsson, Nortel and Siemens are pushing for a completely open solution: open within the carrier's network as well as open across the carriers' networks. As a result of opening up the architecture, and enabling multiple types of applications to share a common set of IMS resources, it becomes virtually impossible to meet the stringent performance requirements of Push to Talk service in the PoC architecture.

- (v) IMS infra-structure has Go/Gq interface between PDSN/GGSN (on the bearer path) and PDF (on the signaling path) but there is no standardized way of communication between the two entities: SIP is used to set up connections and once the connection is established, mobiles exchange data using the bearer path. However, if the data connection breaks, there is no feedback between the bearer path and the signaling path and as a result, the connection remains on by default until an administrator removes those connections via a manual operation. Ideally, there should be an explicit feedback between the bearer plane and the signaling plane such that the network element on the bearer plane, on behalf of the mobile, can keep the SIP connection on even when the data connection is broken because of poorer coverage, for example.
- (vi) Support for Push: In the current network, a client has to initiate a PPP connection with the network and once the connection is established, the client is able to pull content from the network or from servers outside the network. However, if there is no explicit mechanisms for the network to initiate a connection with the client and push content. Recently, 3GPP working group has proposed mechanisms such as Network Requested PDP Context Activation (NRPCA) [7] and Multimedia Broadcast / Multicast Service (MBMS) [8, 9] to enable network-initiated push functions.

### 3. PROPOSED SOLUTION

There are various ways in which an “always on” service can be provided in a carrier’s network. However, there are two important points to keep in mind:

- 1. If the goal is to provide always on service across multiple networks, then we need a “network” independent architecture.
- 2. If the goal is to provide always on service for a particular carrier’s network, then various network elements in the traditional carrier’s network need to be augmented and that may delay deployment of such a service.

In order to facilitate (1) and avoid the drawbacks of (2), we propose to use an “overlay” network which is independent of the underlying network technology (3G, 4G, WiFi) and the administrative boundaries (Verizon Wireless, Sprint PCS, Cingular, etc.). Naturally, the overlay network can provide always on service either to the customers of a specific wireless carrier, such as, Verizon Wireless across disparate link layer technologies like 3G and WiFi networks; or to the customers of a Mobile Virtual Network operator (MVNO) across multiple 3G wireless carriers’ networks.

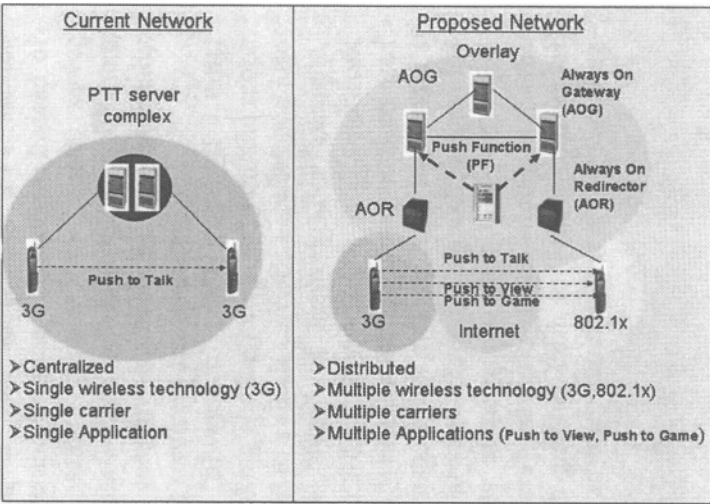


Figure 2 Components of an Always On Network

The overlay network is created using two network elements, namely, Always on Gateway (AOG) and Always on Redirector (AOR). The goal is to bundle “all the always on specific functionalities” into the AOG so that the existing network elements in a carrier’s network do not need to be changed. In addition, the purpose of AOR is to haul traffic from various access networks into the overlay network transparent to the end user. Once the traffic is hauled into the overlay network, AOG can provide all the necessary always on functionalities. In addition to that, a Push Function (PF) is introduced in the architecture to support network-initiated “push” functions on the overlay network. A typical always on overlay network is shown in Figure 2. Here is a description of the functions provided by the network elements in the overlay.

**Always on Gateway (AOG):** The AOG is the core of the overlay. One or more AOGs are used to build the overlay. Each AOG performs the following functions.

1. Maintain state information: AOG keeps multi-layer state information of a large number of mobiles that it services.
  - Data link-layer state: The mobile establishes a point-to-point protocol (PPP) [3] session with the AOG. PPP sessions are state-full in the sense that the AOG needs to keep the following session information corresponding to every session with a mobile: characteristics of the link state, compression protocol and its state, IP address of the mobile, etc.
  - Network-layer state: AOG also keeps information about the mobile at the network layer. For example, if the mobile uses Mobile IP [4] at the network layer, the AOG acts as the Foreign Agent (FA) for the mobile and keeps the following state information: states for the FA to Home Agent (HA) tunneling, IPSec parameters, if any, data link layer reachability information to the mobile, etc.
  - Transport-layer state: AOG keeps persistent TCP sessions with the mobile. This helps in reducing TCP startup delay in certain applications. Note that TCP startup delay can be quite large in a cellular environment where over the air transmission delay could be large and highly variable [5].
  - Application-layer state: For certain networked applications, AOG monitors the application state so that it can mimic a subset of mobile’s behavior to a server in case the mobile becomes temporarily unreachable due to unavoidable network conditions (for example, no coverage within a tunnel).



2. Migration of state information: A mobile's state is initially created in an AOG when the device is powered up. As the mobile roams in the underlying network, and goes far away from the AOG that has its state information and comes closer to another AOG, then for better service, the whole state information can be transferred from the previous AOG to the new AOG. This eliminates the unnecessary session setup with the new AOG but still keeps the mobile always on.
3. One time authentication: In today's network, when a mobile roams from one network to a visiting network (where network boundary may be defined across different carriers or across different physical-/data-link-layers), usually the mobile and/or user must be authenticated by the visiting network, even if the mobile and/or user was authenticated by the previous network. In the overlay network architecture, AOG can authenticate the mobile and/or user once and make roaming across different networks seamless and fast.
4. Enhance services for the mobile: Since "over the air" communication is expensive compared to communication over the wired network, AOG cuts down on "over the air" communication by proxying on behalf of the client and performing the client functions in the wired network. This reduces the session setup time, eliminates or reduces round trip time to execute DNS queries, etc.
5. Multimodal operation: The overlay network can manage multiple overlapping networks. For example the underlying networks could be a WiFi hotspot and a wide area 3G network. In the hotspot region, both networks are accessible by a mobile equipped with multiple interfaces (e.g., built-in WiFi interface and 3G PCMCIA card). In such a scenario, the common wisdom is to have intelligence in the mobile to switch to the high bandwidth network (i.e., WiFi hotspot) and stick to it as long as the quality of reception is good. In the overlay network environment, we let AOG control which network interface should be used by the mobile. There are several advantages to this approach. First, since the intelligence is moved from the mobile to the network, the end device can be made simple. Second, the AOG can seamlessly use the high bandwidth network in the middle of an application session between the mobile and a server. Third, even if the hotspot may provide higher data rate, it could be temporarily overloaded. In such a situation the mobile would be better off using the wide area network. AOG can determine this very easily and efficiently and direct the mobile accordingly.
6. Context notification: AOG monitors roaming of a mobile within a network or across different networks, and dynamically prepares context information about the mobile. The context information includes parameters like network identification, available bandwidth, IP addresses of the intermediaries (e.g.,

local web object cache, local SIP proxy, local DNS server, etc). AOG sends the context information to the mobile and the servers that subscribe to such messaging. The mobile can use the context message to configure itself. For example, the mobile can reconfigure its web browser to explicitly use a proxy cache in the AOG. This improves web browsing experience by keeping persistent session with the local cache and reducing “over the air” DNS operations. Similarly the SIP proxy address can be configured into the mobile to reduce DNS access over the air. The server may use the information to provide personalized, location-dependent, bandwidth-sensitive services.

7. Overlay network setup: AOG sets up an overlay network with other AOGs. The details of overlay setup are omitted from this paper, but the interaction between AOGs is shown in an example application later in the paper.
8. Location and paging of mobiles: AOG keeps track of the mobile as it moves across multiple networks. Note that a mobile can either move across networks with the *same* physical-/data-link-layer (e.g., 3G) owned and operated by *different* carriers or move across networks with *different* physical-/data-link-layer (e.g., 3G and WiFi). AOG keeps up-to-date location information in both the above scenarios. In some networks, to reduce network load, paging may also be initiated by AOG to find a dormant mobile.

**Always on Redirector (AOR):** AOR is used to redirect traffic from the mobile to the “best” AOG, based on location of the mobile, QoS requirement of the session, home network of the mobile, etc. An AOR interfaces with a particular access network (e.g., 3G, 4G, WiFi, etc.). While there may be multiple AORs interfacing with a particular access network, there cannot be a single AOR interfacing with more than one access network.

An AOR has two interfaces, one facing the overlay network (AOG in particular), and the other facing the access network from which traffic needs to be hauled into the overlay network. The interface facing AOG implements signaling and bearer sessions following the 3GPP2 standards (A10-A11) [6]. In other words, a PPP connection originating from a mobile must terminate at AOG thereby giving AOG the full control of the connection including assignment of IP address. In order to achieve this in a uniform manner across different networks, the functionalities of the interface of the AOR facing the access network are customized based on the characteristics of the access network. We briefly describe this feature of AOR for two types of access networks: CDMA2000 network and a WiFi network.

In order to interface with a CDMA2000 network, AOR acts as a PDSN to the PCF of the CDMA2000 network, and acts as a PCF to the AOG. AOR takes a mobile’s session from the PCF and redirects it to an AOG depending on the

service attributes of the mobile. In other words, if the mobile has subscribed to the always on service, then the connection is redirected to an AOG, otherwise the connection is passed on to the PDSN. Therefore, it is possible to offer always on service only to subscribers that sign up for it, and leave aside the remaining subscribers. To achieve this, AOR proxies as a PDSN and terminates A11 signaling from PCF, and then terminates Link Control Protocol (LCP) phase of the PPP negotiation with mobile. Then AOR initiates user authentication with the mobile during which the user identification is sent. AOR uses the user identification to determine if always on service is to be provided to the user. If yes, the AOR opens a new session with the AOG; otherwise it opens a new session with the PDSN in the CDMA2000 network. In both cases, AOG or the PDSN completes the data call with the mobile and AOR splices the two PPP sessions: one between the mobile and the AOR and the other between the AOR and the AOG/PDSN.

When AOR interfaces with a WiFi network, it sits upstream of an Access Point (from the mobile's point of view) and implements a PPTP proxy (or L2TP proxy). All the mobiles in the network get configured with the IP address assigned by the Access Point. The mobiles that subscribe to the always on service initiate a PPTP session with AOR as the PPTP server. This forces the mobile to tunnel PPP frames to the AOR using the IP address provided by the access point as the source address. AOR, instead of terminating the PPP session, tunnels the PPP frames all the way to the AOG responsible for terminating the session and assigning IP addresses. In this case the mobile is assigned two IP addresses: the address assigned by the Access Point is used for tunneling PPP frames while the address assigned by the AOG is used for all data communications. Again, in this case, always on service is provided only to a mobile that subscribes to the service and the rest are left untouched.

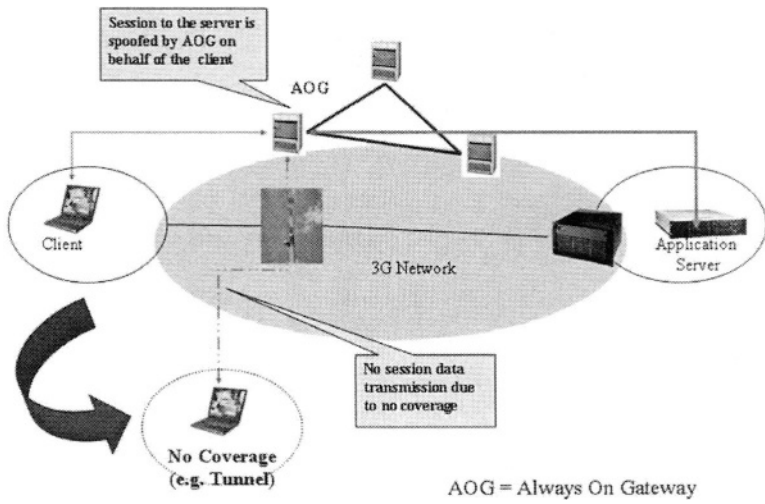


Figure 3 Architectural view of Network-initiated Smart Push

**Push Function (PF):** Push Function is used by the network to facilitate network-initiated content push to the user. PF interfaces with the Application Server on one side and with the AOG on the other side. Push Function is triggered by an Application server to initiate a push operation. However, PF determines how the content will be pushed. For example, content may be pushed using SIP in the IMS architecture [7] or using Multimedia Broadcast Multicast Service (MBMS) [8] or using WAP [7]. PF together with AOG can provide interesting capabilities, such as “smart” push to a wireless network. The idea behind “smart” push is to prevent a network from pushing content to a mobile end user who may not be temporarily reachable and hold the content in the network until either the mobile becomes reachable when the content is delivered or the content loses relevance. Architectural view of “smart” push is shown in Figure 3.

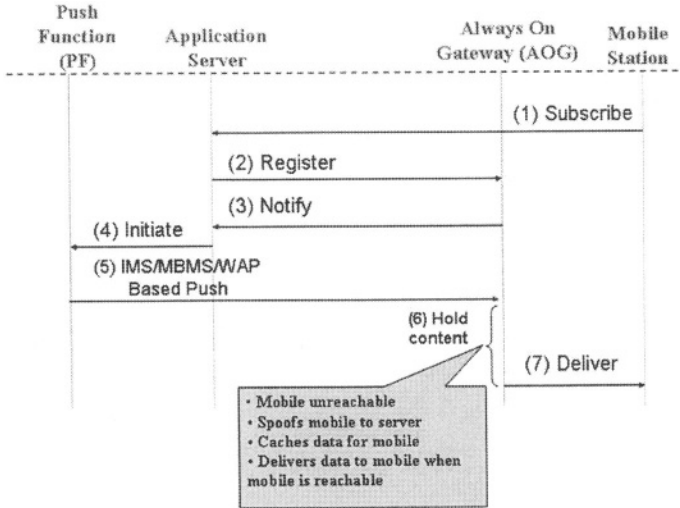


Figure 4 Call Flow for Network-initiated Smart Push

Call flow corresponding to smart push is shown in Figure 4. First the mobile subscribes to a service, such as “location-based smart push”. Application server registers with AOG (Step 2) by providing the conditions under which the AOG should send it a notification. For example, in a location-based service, the application server could be a “coupon” pusher for Macys and the condition could be the mobile being in the vicinity of a Macys store. When the mobile is in the vicinity of a Macys store, AOG notifies (Step 3) the application server. The application server sends a message to PF to initiate the push. PF decides the best technique for pushing the content based on the content itself, the capability of the mobile station and the available network bandwidth (Step 5). AOG intercepts the content and holds it (Step 6) if the mobile is not reachable due to poor coverage while spoofing on behalf of the mobile client to the Application server. When the mobile becomes reachable (before the content loses relevance), AOG delivers the content to the mobile (Step 7). If the mobile is unreachable for a long duration such that the content loses relevance, AOG simply drops the content thereby saving network bandwidth.

### 3.1 Example: Always On Multi-party Chat

Using always on multi-party multi-network chat as an example application, we show the high level call flow for the overlay network. The overlay network is

built on top of multiple access networks as shown in Figure 5. We show both wireline and wireless networks in the access to show that AOR can indeed interface with networks with different physical-/data-link layer technologies. The application is a multi-party chat where users join the session on their own or are invited to the session by a participant. The session establishment uses SIP-like protocol. The steps in the example are described below:

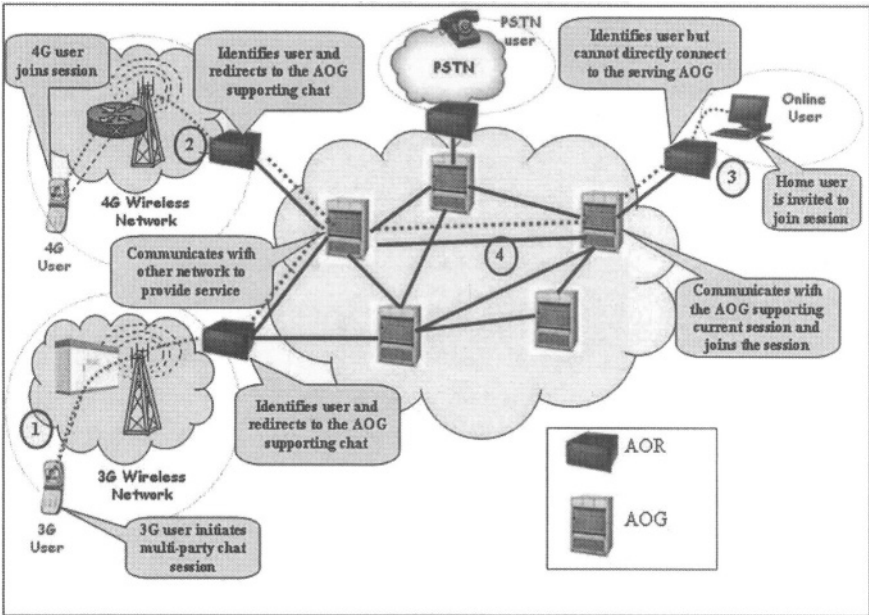


Figure 5 Call Flow in an Always On Network supporting Multi-party Chat

1. A 3G user initiates a chat. The AOR associated with the 3G access network recognizes this and redirects the call to the nearest AOG supporting the multi-party chat application.
2. A 4G user joins the session. The AOR associated with the 4G access network recognizes this and connects the user to the nearest AOG supporting the multi-party chat application. In this example, the AOG selected by the AORs happens to be the same.
3. A user at home using a cable modem is invited to join the session. The AOR associated with the cable modem access network recognizes this and redirects the user's connection to the nearest AOG. In this case the AOG selected is not the same as the one above.

4. Now the selected AOGs communicate with one another on the overlay network to bring all chat participants to the same chat session.

Note that the components of the overlay network participate both in signaling and bearer path (unlike SIP architecture where they are separate). The advantages of having them in both paths result in better management of state information and bearer path quality provisioning.

## 3.2 Prototype Implementation

In this section we describe results from two prototypical implementations and experimentations. First one shows how keeping persistent session with AOG reduces service time for web browsing giving the user always on experience. The second one describes implementation of AOR on Linux and shows how fast it redirects connection even though it functions in the overlay network.

### 3.2.1 Response Time Reduction with Session Persistence

We implemented prototypes of session level optimization techniques in Linux and conducted some controlled experiments to measure the quantitative benefits of session persistence. In this section we present the experimental setup and the summary of the results obtained from the experiments we conducted.

The experiments are conducted for mobile web browsing. The browser of the mobile can be set to point to an (explicit) proxy cache collocated with the AOG. A persistent TCP session is maintained between the browser and the AOG. To perform the experiments with specific web pages in a controlled environment, the top level pages and the embedded objects in them from the web sites were copied to a local apache web server. For this experiment, the top level pages from Yahoo, CNN and Britannica were copied. The statistics for these sites are:

- Yahoo ([www.yahoo.com](http://www.yahoo.com)): It has 16 embedded objects hosted in 3 different domains. The size of the page is 74 KB. This constitutes a typical web site with small number of domains.
- CNN ([www.cnn.com](http://www.cnn.com)): It has 58 embedded objects hosted in 6 different domains. The size of the page is 197 KB. This constitutes a typical web site with medium number of domains.
- Britannica ([www.britannica.com](http://www.britannica.com)): It has 32 embedded objects hosted in 15 different domains. The size of the page is 178 KB. This constitutes a typical web site with large number of domains.

The browser at the mobile was instrumented to compute the time between the sending of the request for the top level page and the complete display of the page (including all embedded objects). We refer to this as the *user*

*perceived* response time for the page. We measured this response time at the browser to download three popular top level pages (Yahoo, CNN and Britannica) and the embedded objects contained in them. The results are shown in Figure 6.

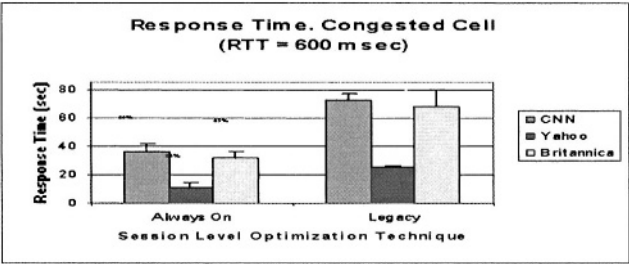


Figure 6 Improvement in Response Time with Always On

The response times were measured and averaged over 20 downloads of each of the top level pages and the corresponding embedded objects. The legacy case is where the mobile sends all the DNS requests over the air. Moreover, no persistent session is maintained which results in TCP startup delay for every TCP connection established by the mobile. In the always on case, the mobile maintains a persistent session with AOG and all the DNS resolutions are done by AOG on the wired network.

As is evident from Figure 6, eliminating over the air DNS queries significantly reduces the response time. In addition, response time is improved by eliminating the need for re-establishing TCP connections for every web session. Response times for CNN and Britannica are much higher than that for Yahoo because Yahoo has fewer domain names and embedded objects than the other two. Between CNN and Britannica, CNN has more embedded objects but fewer domains. The time required for making DNS requests balances out the time required to download the embedded objects and as a result, they have similar response times. In all cases, the response time for always on sessions is much smaller than that for legacy sessions. In fact, we observe a mean response time improvement of 50% for congested cells.

**3.2.2 Price of Overlay**

In order to estimate the overhead for using an overlay network, we built an AOR and measured the additional latency introduced due to overlay. AOR prototype for CDMA2000 access network was built on Linux 2.4.18 kernel running on a Pentium III 450MHz workstation. Different modules of the prototype are shown in Figure 7. The Ethernet Frame Processor and the IP Packet Processor are



already provided in the standard Linux distribution. We implemented the rest of the modules as a patch to the kernel. Brief description of each of these modules is given below:

- **UDP Processor:** This module is responsible for sending and receiving UDP packets.
- **R-P Signal Processor:** This module is responsible for R-P session termination between PCF and AOR, R-P session establishment between AOR and AOG and for splicing them.
- **AAA Client:** This module implements a RADIUS client to access a AAA server for authenticating a user and to get the IP address of the destination AOG.
- **GRE Processor:** This module takes a PPP frame from the PPP Control Processor, encapsulates it and sends the frame in a GRE session. It also decapsulates a PPP frame from a GRE packet and delivers the frame to PPP Control Processor. It is also responsible for splicing the PCF-AOR and AOR-AOG sessions using the GRE keys.
- **PPP Control Processor:** This module is responsible for terminating a mobile initiated link control phase of a PPP session, identifying the user from PAP/CHAP, authenticating the user, getting AOG IP address from a AAA server using the AAA client, and for initiating link control phase of a PPP session with the selected AOG.

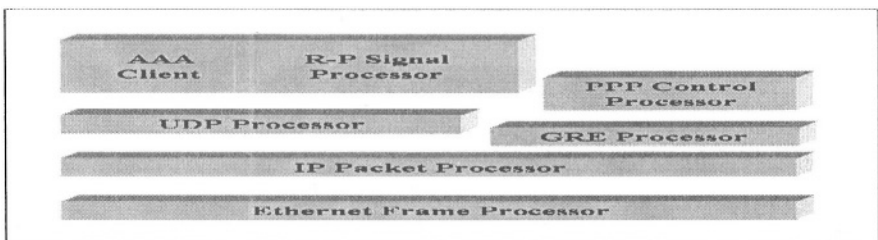


Figure 7 Prototype Implementation of AOR

We conducted a set of experiments to measure the additional overhead AOR introduces in setting up a data session between a mobile and the AOG. We measure the delay introduced in the data or bearer path (i.e., actual packet transfer).

Figure 8 shows the transfer latency of a data packet (averaged over multiple runs) with and without AOR. A packet size of 1300 bytes (i.e., the MRU of the PPP

session) was used. In order to create the background load on AOR, a number of PPP sessions each sending a constant bit rate traffic to the AOG were generated. By changing the number of these sessions background load was varied. Packet transfer latency was computed by tagging the reception of every packet. Note that the packet transfer time increases with load, however the overhead introduced by AOR is negligibly small.

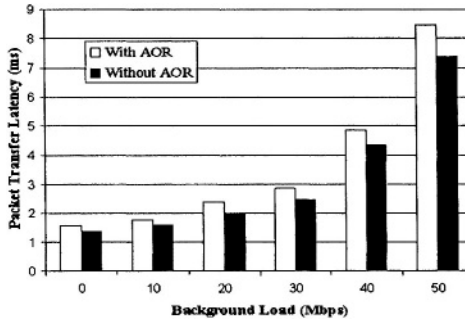


Figure 8 Packet Transfer Latency with and without AOR

In addition, Figure 8 shows that in the worst case, the connection setup time and the packet transfer delay with AOR is only a few milliseconds larger than in the case without AOR. This shows even though AOR redirects packets in the overlay always on network, the overhead is minimal. This overhead will become even smaller with product quality hardware and software and hence should not be a cause for concern.

## 4. CONCLUSIONS

In this paper we have defined the requirements of always on services, described the major problems in supporting such services in today's network, proposed a network independent solution that provides the benefits of an always on service without changing existing networks, and have also presented some preliminary results from our prototype implementations to show the benefits of always on services. In addition to that, we have shown how novel services, such as "smart" push can be provided a service provider by combining the control-plane intelligence, such as user-profile and policies with data-plane intelligence, such as reachability of the mobile user. Currently we are optimizing the implementation of AOR and AOG for the basic function and are enhancing the

functionality of these elements to provide an “enhanced” always on service experience to the end users.

## **REFERENCES**

- [1] 3rd Generation Partnership Project 2. Interoperability Specification (IOS) for cdma2000 Access Network Interfaces --- Part 3 Features. 3GPP2 A.S0013-A, Version 2.0.1, July 2003.
- [2] America On Line. <http://www.wave-report.com/other-html-files/alwayson2003.htm>
- [3] W. Simpson et. al. The Point to Point Protocol (PPP). Internet Engineering Task Force (IETF) Request for Comments (RFC) 1661, July 1994.
- [4] C. Perkins et.al. IP Mobility Support for IPv4. Internet Engineering Task Force (IETF) Request for Comments (RFC) 3220, August 2002.
- [5] P. Rodriguez, S. Mukherjee and S. Rangarajan. Session Level Techniques for Improving Web Browsing Performance on Wireless Links. In Proceedings of the Thirteen International World Wide Web Conference, New York, May 2004.
- [6] 3rd Generation Partnership Project 2. Interoperability Specification (IOS) for CDMA2000 Access Network Interfaces --- Part 7 (A10 and A11 Interfaces). 3GPP2 A.S0017-0 v2.0, May 2002.
- [7] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects Push Architecture (Release 6). 3GPP TR 23.976 v2.0.0 (2004-03).
- [8] 3rd Generation Partnership Project; Technical Specification Group 22: TSG 22.146 “Requirements for MBMS”.
- [9] 3rd Generation Partnership Project; Technical Specification Group 23: TSG 23.246 “Architecture and Functional Description”.
- [10] Open Mobile Alliance (OMA). Push to Talk over Cellular Requirements; Draft Version 1.0 - January 31, 2004. OMA-RD\_PoC-V1\_0-20040131-D

Emerging Location Aware Broadband Wireless Ad Hoc  
Networks

Ganesh, R.; Kota, S.L.; Pahlavan, K.; Agustí, R. (Eds.)

2005, XIII, 329 p., Hardcover

ISBN: 978-0-387-23070-2