Agent-Based Mobile Multimedia Service Quality Monitoring

Man Li

Nokia Research Center
5 Wayside Road, Burlington MA 01803
man.m.li@nokia.com

Abstract. An agent-based mobile multimedia service quality monitoring architecture is proposed where agent software is installed on end user mobile phones to monitor service performance, to report measurements to a centralized management server, and to raise alarms if service quality falls below a pre-set threshold. The proposed framework effectively turns thousands of mobile phones into service quality probing stations and provides a cost effective way to obtain true end user experience. Key service performance indicators for service performance evaluation are proposed. Parameters that should be monitored by agents for each service instance are discussed. A procedure to derive the key service performance indicators from service instance measurements is also described. Future researches are then outlined.

1 Introduction

Monitoring end user perceived service quality is a critical step in managing multimedia services over mobile networks. Unlike fixed networks, the air interface in a mobile network has low bandwidth and is subject to higher packet loss and error. As a result, offering multimedia services over mobile wireless networks presents many challenges. On the other hand, multimedia services when offered commercially must meet certain performance standards in order to attract and sustain subscribers. For example, video-streaming services should not have many display interruptions or fuzzy images so that users can enjoy the movies. Furthermore, driven by fierce competitions, mobile operators strive to make sure that the quality of their services is superior to those provided by competitors. As a result, mobile operators have a keen interest in monitoring service quality perceived by end users. In addition, through continuous monitoring, operators can also detect potential performance problems early and make corrective actions in time before large-scale performance problems occur.

Previous studies have produced a good understanding of service performance targets for different services, e.g., web browsing [1]. Impacts of packet loss, delay, and jitter on service performance have also been studied extensively [2], [3], [4], [5], [6]. A current approach for service quality monitoring is to install monitoring tool or software on a dedicated monitoring device such as a laptop. Equipped with a wireless card such as a GSM card, this device can be used for drive-through tests. The monitoring tool would initiate various mobile services from the device and then monitor different aspects of the services, e.g., response time, throughput, etc. The monitoring results are then uploaded to a central management server. Dedicated monitoring devices are reliable but their monitoring results are approximations of the performance

J. Vicente and D. Hutchison (Eds.): MMNS 2004, LNCS 3271, pp. 189–199, 2004.

perceived by actual customers. The geographic and temporal distribution of the measurements may be quite different from the real user experience. In addition, the application performance on the dedicated devices may be quite different from that on a mobile phone. Further more, deploying thousands of drive-through tests with dedicated monitoring devices can be costly in terms of both hardware and human resources required.

In this paper, we propose an agent-based architecture for monitoring mobile multimedia service quality. It effectively turns thousands of mobile phones into service quality probing stations and provides a cost effective way to obtain true end user experience. The organization of the paper is the following. Section 2 proposes an agent based mobile service quality monitoring architecture. Section 3 discusses the key service performance indicators. Section 4 details what mobile agents shall monitor for each service instance. Section 5 describes the procedure of deriving key performance indicators. Section 6 provides an example for streaming service performance monitoring. Section 7 discusses future researches and Section 8 concludes the paper.

2 Agent-Based Mobile Service Quality Monitoring Architecture

We propose an agent-based architecture for monitoring mobile multimedia services as shown in Figure 1. It is based on the Open Mobile Alliance (OMA) Device Management framework [7]. It consists of mobile agents installed on phones and device management (DM) servers in network operation centers (NOC). The communications between the mobile agents and the device management servers are through the OMA SyncML for device management protocols [9], [8].

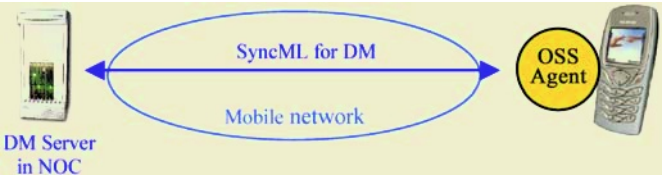

Fig. 1. Mobile agent based monitoring architecture.

Mobile agents are responsible for capturing performance information for each monitored service instance whereas the DM server has the responsibility of collecting measurements from mobile agents and of developing key performance indicators out of the collected measurements. To reduce memory consumption and measurement reporting traffic, a mobile agent can also produce and report performance indicators based on its own measurements. Details of the agent based monitoring architecture are described in the following sections.

2.1 SIPAs and KPIs

We define a *service instance* as an occurrence of a service. For example, the download of a single web page is an instance of a browsing service. A *Service Instance Performance vAlue (SIPA)* contains information that captures a performance aspect of a single service instance. For example, a “status” (success or failure) that is associated with each service instance is a SIPA.

A *Key performance indicator* (KPIs) is a metric that is crucial for evaluating the performance of a service. KPIs are derived from a population of SIPAs. For example, a "service success ratio" is a KPI that is derived from many "status" SIPAs collected by mobile agents.

KPIs provide a high level indication on service performance whereas SIPAs support the derivation of KPIs and are also useful in root cause analysis.

2.2 Mobile Agent

A mobile agent is a piece of software running on a standard mobile terminal. One of the responsibilities of an agent is to collect and report SIPAs. As shown in Figure 1, an agent communicates with a device management server via the SyncML DM protocol. The server controls the agent behavior by loading a monitoring profile. The profile contains the following information:

- Monitoring style. Indicates if the monitoring should be active or passive. For active measurement, a mobile agent executes scheduled tests, e.g., downloads a predefined web page, after checking that enough resources on the mobile phone are available for performing the test. This test activity would have lower priority and can be interrupted if necessary so that real user services are not affected. For passive measurement, the mobile agent starts monitoring calls and service sessions as instructed by this profile. Passive measurement is likely to have little disturbance to real user applications. In both active and passive measurements, an agent records SIPAs associated with each monitored service.
- Schedules. Indicate the start and end of monitoring periods
- User locations. Monitoring can be triggered by user location, i.e., start monitoring when a user is in these cells.
- Bearer types. Instructs the agent to monitor a service only when it uses the specified bearer types. A service may be provided over different bearers, e.g., GSM, Blue tooth, WLAN.
- Service names. The type of services to be monitored
- Servers. For passive monitoring, only when these servers are involved, shall the performance be recorded. For active monitoring, these are the servers to be contacted in order to start services
- Reporting schedule. This dictates when and how agents report results to a management server. Possible choices are: at the end of each service, periodically (e.g., once every hour or every day), or when asked by the server.
- Others, e.g., thresholds for different services. When quality falls below the thresholds, alarms should be raised.

A mobile agent can also pre-process the SIPAs it collects and produce performance indicators (PI) out of these SIPAs. Then the PIs can be reported to the server. For example, instead of reporting the "status" of many web page browsing service instances, the agent can report a single success ratio derived from the SIPAs. We call the indicators derived by a single mobile agent "performance indicators (PIs) ", not KPIs, because they represent only a single end user experience. To further build on this idea, the monitoring profile can also include thresholds for the PIs so that if a threshold is exceeded, the agent will send alarms to the management server. Note, however, that this method needs to be implemented with caution. When many agents

in the same cell detect the same performance problem, e.g., service response time is too long, they may all send alarms and introduce a flood of additional traffic into the network that further worsen the situation.

The actual monitoring and recording of SIPAs can be performed in at least two ways. The first approach is to have service applications conduct the measurements. Hence as shown in Figure 2(a) the mobile agent must interface with the service applications. Through the interfaces and based on the monitoring profile, an agent can configure the applications on what, when and how to conduct monitoring of services. When it is time to report measurements, the agent will query the applications for SIPAs and transport them to the server via SyncML. If the DM server instructs the agent to report PI measurements, the agent will query the applications for SIPAs, derive PIs out of the SIPAs, and then deliver the PIs to the server as requested. Periodically, the agent shall query the applications for SIPAs, derive PIs from them, and raise alarms if thresholds are exceeded. This approach employs a relatively simple agent whose responsibility is to collect and report monitoring results. The agent does not need to understand the messages, traffic flows, or logics involved in providing services. On the other hand, this approach does require service applications to implement monitoring interfaces so that they can communicate with an agent. If a third party application does not support a monitoring interface, then that service cannot be monitored.

A second approach is for an agent to snoop the transport layer (e.g., IP packets) as shown in Figure 2(b). The agent then parses the packets to and from applications, and infers SIPAs from them. For example, when the agent sees a HTTP GET request, it records a web service request accordingly. With this approach, the agent is responsible for all the monitoring, recording, and reporting. This approach does not require service applications to support extra interfaces. As a result, any third party developed applications can be monitored. On the other hand, this approach demands a very intelligent and hence complicated agent that understands all the applications. With the limited processing power on current mobile devices, this may not be feasible for the time being. But it may be an option after a few years.

If the speed of growth for memory and processing power of desktop computers could be any predication for the growth of memory and processing power of mobile devices, the consumption of processing power and memory for service performance monitoring may not be a major concern after a few years. On the other hand, the extra reporting traffic introduced into the network may be a concern, given the limited air interface bandwidth and the fact that there may potentially be thousands of mobile agents in a network. There are at least two approaches to reduce reporting traffic. First, compressing the data before transportation in order to save bandwidth. Second, reporting PIs instead of SIPAs.

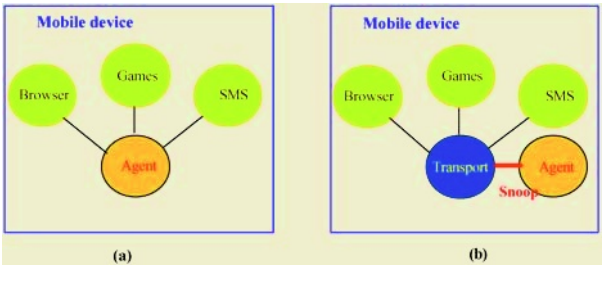

Fig. 2. An agent inside a mobile device.

2.3 Device Management Server

Within the OMA device management framework, a Device Management (DM) server remotely performs mobile device configuration, inventory, diagnostics, software installation, upgrading and configuration. Service quality monitoring can be seen as part of remote diagnostics. The DM server remotely configures monitoring profiles to be used by mobile agents. It also collects or receives SIPA reports from mobile agents and derives KPIs from the SIPAs. Very valuable KPIs can be derived. For example, when location information (e.g., Cell ID) is associated with the measurements, an operator can produce a "service weather report" – how each service is doing in different area. Potential problems may be detected early and diagnosed with the help of the measurements. The measurements can also serve as real time feed back for network optimization.

Specific to the agent based monitoring architecture, the DM server is also responsible for managing the mobile agents, for example, remotely installs and updates agent software, activates or deactivates an agent, pings an agent, if necessary, when the agent has missed multiple scheduled reports.

2.4 SyncML DM Protocols

The protocols used for communications between mobile agents and DM servers are the OMA SyncML Representation Protocol [8] and the SyncML DM protocol [9]. The SyncML Representation Protocol is an XML-based representation protocol that specifies the representation or format of all the information required to perform synchronization or device management. To reduce the data size, a binary coding of SyncML based on the WAP Forum's WBXML is defined. In [9], the use of the representation protocol for device management is specified. It describes how SyncML messages are exchanged in order to allow a device management client and server to exchange additions, deletes, updates and other status information. The SyncML Representation and DM protocols are transport-independent.

2.5 Management Tree

In the OMA DM framework, each device that supports SyncML DM must contain a management tree [10]. The management tree organizes all available management objects in the device as a hierarchical tree structure where all nodes can be uniquely addressed with a URI. Nodes are the entities that can be manipulated by management actions carried over the SyncML DM protocol. The actions include ADD, GET, REPLACE and DELETE. For service performance monitoring, the leaf nodes specify the SIPAs and their properties.

All managed objects or nodes are defined using the SyncML Device Description Framework [11]. OMA DM has been standardizing management objects [12]. Proprietary device functions can also be described using SyncML Device Description Framework. This allows new proprietary device functions be managed before they are standardized. Specifically, device manufacturers will publish descriptions of their devices as they enter the market. When the descriptions are fed into DM servers, the servers will be able to recognize and manage the new functions in the devices.

The above-proposed agent-based service monitoring architecture effectively turns thousands of mobile phones into service quality probing stations. Mobile agents are

close to end users and hence can faithfully monitor real end user experience. Drive-through tests are conducted as end users move in mobile networks. Since no additional hardware and manpower are required, it is a cost effective approach for monitoring mobile multimedia service performance.

3 Mobile Service Key Performance Indicators (KPI)

We have described an agent-based monitoring architecture. The next step is to determine a set of key performance indicators (KPI) used by the framework to assess service quality. Mobile multimedia service KPIs shall be able to measure service success ratio, service response time and the quality of the media involved in a service. The recommended service success ratio and response time KPIs are:

- *Service success ratio*: The percentage of service instances that are complete successfully
- *Service response time*: Measures the speed of response to an end user's request for service
- *Bearer Failure Ratio*: The percentage of bearers (e.g., GPRS) that either cannot be established or are established but prematurely disconnected before the end of a service. It indicates how much the network contributes to service failures – a KPI of great interest to mobile operators.

In terms of media quality, there is no single metric that can completely characterize media quality. Instead, we recommend the use of multiple KPIs to evaluate different aspects of media quality. If a service involves multiple types of media (e.g., voice, video), the quality of each media type shall be evaluated separately.

As packets travel through a mobile network, they experience different end-to-end delays. For real time applications, a display buffer is implemented at the receiving end to smooth out the delay jitters so that the media can be displayed continuously. It may happen, for example due to low throughput of the network, that the buffer becomes empty temporally. This causes a display break – a period during which the rendering of streaming data stops. A very short break may not be visible to an end user. However, when a break lasts longer than a threshold, it becomes noticeable performance degradation. Therefore, we propose the following two media KPIs:

- *Display break ratio*: The ratio of the sum of display break durations that are longer than a pre-defined threshold over the service time.
- *Number of display breaks*: The number of media display breaks that last longer than a pre-defined threshold

In addition, the following two KPIs are also necessary for assessing media performance:

- *Packet Loss & Error Ratio: The percentage of packets that are either lost or detected as erroneous. Applicable to media content carried by RTP protocol only*
- *Throughput: Packet throughput of the media involved in the service.*

Many previous studies indicate that packet loss and error ratio has a direct impact on real time media quality [2], [3], [4].

End to end packet delay is also an important performance indicator for media quality. Unfortunately, it is difficult for mobile agents to measure this type of delays because estimating one-way end-to-end delay requires clock synchronization between

mobile phones or between mobile phones and servers. Which is difficult to achieve. In addition, monitoring packet delay requires mobile agents to capture and time stamp packets received. Which could potentially consume too much memory and processing power.

4 Mobile Service Instance Performance Values (SIPA)

With the KPIs being specified, we now define the information that must be captured for each service instance, i.e., SIPA, in order to produce KPIs and to help in root cause analysis.

For mobile multimedia services, a service may involve multiple types of media. Therefore, we separate service SIPAs from media SIPAs. Service SIPAs capture the different aspects of a service instance whereas media SIPAs capture the information on media delivery and display. The service and media SIPAs are specified in Table 1 and Table 2. In formal definitions, these SIPAs are arranged in a tree structure and are specified in XML, following the OMA device management framework.

Table 1. Service SIPA.

SIPA	Descriptions
Service Name	The name of a service, e.g., browsing
Application Name	The name of the application used for the service. It identifies the vendor of the application software.
Session ID	A random number that uniquely identifies a service instance on the mobile phone
Setup Duration	The duration to setup a service. Indicate the time required to gain resources that are necessary for the service.
Function Duration	The duration of the time when the service is delivered.
Status	The outcome of a service
Date and Time	The date and time when a user starts to use the service
Location	The location at the end of a completed service or an unsuccessful service attempt.

5 Deriving KPI from SIPA

An agent-based service monitoring architecture shall be able to derive KPIs for a specific service from a pool of SIPA reports. The steps to do so are described below:

1. Filter out the service instances by "Service Name" SIPA. Apply the following calculations on the resulted service instances.
2. *Service success* ratio KPI can be computed as

$$Service\,Success\,Ratio = \frac{Number\,of\ Services\ With\ Status = success}{Number\,of\ recorded\ Services} \quad (1)$$

3. *Bearer failure ratio* KPI for a specific type of media of a service can be computed as

$$Bearer\,Failure\,Ratio = \frac{Number\,of\ Media\ With\ Bearer\,Failure}{Number\,of\ recorded\ Media\,Streams} \quad (2)$$

If a service has bi-directional media delivery, the Bearer Failure Ratio should be calculated for each direction separately.

Table 2. Media SIPA.

SIPA	Descriptions
Session ID	The Session ID of the service to which this media stream belongs
Media ID	A random number that uniquely identifies a media stream within the service instance. This SIPA is applicable when there are multiple media involved in a service.
Server	The address of the server that provides the media delivery. If there is no server involved, this SIPA shall be ignored
Media Type	The type of the media whose quality is being measured.
Direction	The direction (incoming or outgoing) of the measured traffic
Media Setup Duration	The duration to setup this media. Indicate the time required to gain resources that are necessary for the delivery of the media content.
Media Function Duration	The duration of the time when the media content is delivered.
Status	The outcome of a media content delivery
Bearer	Bearer type and if the requested bearer, e.g., UMTS bearer, is established and maintained for the service.
Data Size	The total amount of error free data (in bytes) received (or sent, if monitoring outgoing media stream). This SIPA only applies when the status of the media delivery is success
Packet Counts	The total amount of error free packets received (or sent, if monitoring outgoing media stream). This SIPA only applies when the status of the media is success
Loss & Erroneous Packets	Number of lost or erroneous media packets of the incoming Media. Only applicable for monitoring incoming media
Display Breaks	Number of display breaks that last longer than a pre-set threshold. Only applicable for monitoring incoming real time media
Total Break Length	The sum of all the display breaks (in milliseconds) that are longer than a pre-set threshold. Only applicable for monitoring incoming real time media
Date and Time	The date and time when a user starts to use the media
Location	The location at the end of a completed media delivery or an unsuccessful media delivery.

4. Further filter out the successful service instances by Status = success.
   1. For EACH successful service instance:
      - Service response time = Setup Duration
      - For EACH media stream associated with the service instance:
        - Number of Display Breaks is directly taken from the Display Breaks SIPA.
        - Display Break Ratio for a media stream instance is calculated as

$$Display\ Break\ Ratio = \frac{Total\ Break\ Length}{MediaFunctionDuration} \tag{3}$$

        - Packet Loss & Error ratio for the media stream instance is computed as

$$Packet\ Loss\ \&\ Error\ Ratio = \frac{Loss\ \&\ ErrorPackets}{PacketCounts + Loss\ \&\ ErrorPackets} \tag{4}$$

– Throughput for the media stream instance is calculated as

$$Throughput = \frac{DataSize}{MediaFunctionDuration} \quad (5)$$

Note that this estimates the average "good" throughput since the Data Size excludes erroneous and retransmitted data.

2. The *Service Response Time, Number of Display Breaks, Display Break Ratio, Packet Loss & Error ratio, Packet Delay and Throughput* KPIs are obtained as statistics of service instances whose individual performance metrics are computed as described above. For example, 95 percentile of service response time is a Service Response Time KPI.

One may notice that not all SIPAs are used for computing KPIs. This is because that the SIPAs are designed not only for KPI calculations but also for root cause analysis when problems arise.

6 An Example: Streaming Service Quality Monitoring

We use a streaming service to illustrate the monitoring of service quality. Streaming refers to the ability of an application client to play synchronized media streams like audio and video streams in a continuous way while those streams are being transmitted to the client over a data network. As specified in the 3rd Generation Partnership Project (3GPP) [13], [14], [15], a streaming service contains a set of one or more streams presented to a user as a complete media feed. The content is transported with RTP over UDP. The control for the session set up and for the playing of media (PLAY, PAUSE) is via the RTSP protocol [16]. Figure 3 shows streaming service message flows.

When a user starts a streaming service by either clicking a link in a web page or entering a URI of a streaming server and content address, the streaming client on the mobile phone must first obtain a presentation description that contains information about one or more media streams within a presentation, such as encoding, network addresses and information about the content. This presentation description may be obtained in a number of ways, for example, via MMS or RTSP signaling. 3GPP mandates that the description be in the form of a Session Description Protocol (SDP) file [16].

Once the presentation description is obtained, the streaming client goes through a session establishment for each media stream. Specifically, it tries to establish a secondary PDP context for each streaming media and also sends a SETUP request message to the media server in order for the server to allocate resources for the stream. The SETUP reply message contains a session identifier, server port for displaying the media and other information required by a client for playback the media stream.

After all media stream sessions and their required PDP contexts are established, the user may click the play button to start playing the synchronized media. The user can also pause, resume or cancel the streaming service at any time. The RTSP PLAY, PAUSE and TEARDOWN messages are sent to the server for the corresponding action.

A streaming service instance is the setup, delivery and tear down of a streaming service. Applying the SIPA descriptions in Table 1 and Table 2 to a streaming service instance is relatively straightforward. In terms of service SIPA, The Setup Duration is

the duration to setup the service. The starting point is when a user clicks a URI from a web page or clicks the return key after entering a URI. The end point is when the user is informed that the streaming service is ready, e.g., the play button is visible. The Function Duration is the duration for the service. The starting point is when the first byte of a media is received. And the end point is when the service is complete, i.e., when all media streams are disconnected. In terms of KPI, service response time in this case is the Setup Duration as shown in the figure.

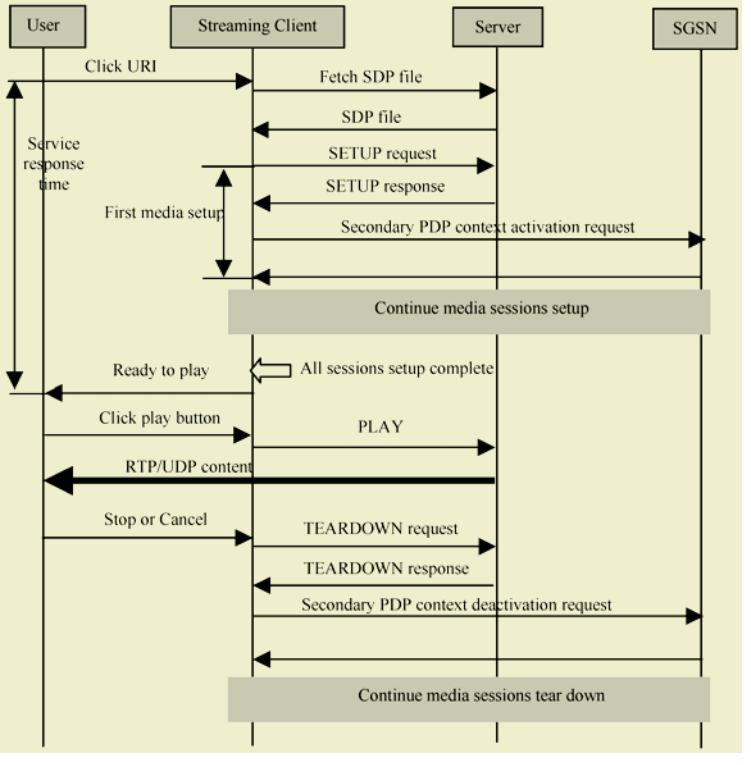

Fig. 3. Streaming Service.

7 Future Investigations

The mobile agents described in this paper can be enhanced with increased intelligence. For example, an agent may also conduct diagnosis. When a user fails to access a service for several times, the agent may be triggered to analyze the situation to decide, for example, if the settings on the phone are correct, if the network is available or if the server is functioning. It can then inform the user on what the problem is and recommend corresponding solutions. Further research is needed on efficient logics to be used by agents for diagnosis, keeping in mind that the memory and processing power are limited on mobile devices.

In order for the agent based monitoring architecture to work with mobile devices from different vendors, the SIPAs need to be standardized.

8 Conclusions

We have proposed an agent-based architecture for monitoring mobile multimedia service quality where agent software is installed on end user mobile phones to monitor service performance, to report measurements to a centralized management server, and to raise alarms if service quality falls below a pre-set threshold. For evaluating service performance, we have also recommended a set of KPIs and SIPAs as well as a procedure of deriving KPIs from SIPAs. The proposed architecture provides a cost effective way to monitor real end user experience. Future research on enhancing the intelligence of mobile agents is also discussed.

References

1. Chakravorty, R., Pratt, I.: WWW performance over GPRS. 4th international workshop on Mobile and Wireless Communications Network (2002)
2. Schlaerth, J., Hoe, B.: A technique for monitoring quality of service in tactical data networks. IEEE MILCOM (1996)
3. Verscheure, O. et al: User-Oriented QoS in Packet Video Delivery. IEEE Network Magazine, November/December (1998)
4. Jassen, J. et al: Assessing Voice Quality in Packet Based Telephony. IEEE Internet Computing, May/June (2002)
5. Kostas, T. et al: Real-Time Voice Over Packet-Switched Networks. IEEE Network Magazine, January/February (1998)
6. Sun, L. et al: Impact of Packet Loss Location on Perceived Speech Quality. Internet Telephony Workshop (2001)
7. ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs (2001)
8. Open Mobile Alliance™: Enabler Release Definition for OMA Device Management, version 1.1.2 (2003)
9. Open Mobile Alliance™: SyncML Representation Protocol, Device Management Usage, v1.1.2 (2003)
10. Open Mobile Alliance™: SyncML Device Management Protocol, v1.1.2 (2003)
11. Open Mobile Alliance™: SyncML Device Management Tree and Description, v1.1.2 (2003)
12. Open Mobile Alliance™: SyncML DM Device Description Framework, v1.1.2 (2003)
13. Open Mobile Alliance™: SyncML Device Management Standardized Objects, v1.1.2 (2003).
14. 3GPP TS 23.233: Transparent end-to-end packet-switched streaming service, stage 1, v6.3.0 (2003)
15. 3GPP TS 26.233: Transparent end-to-end packet-switched streaming service, general description, v5.0.0 (2002)
16. 3GPP TS 26.234: Transparent end-to-end packet-switched streaming service, protocol and codecs, v5.0.0 (2003)
17. M. Handle et al: SDP: Session Description Protocol, IETF RFC 2327 (1998)