Medical Sensor Application Framework Based on IMS/SIP Platform

I. Markota, I. Ćubić
Research & Development Centre, Ericsson Nikola Tesla d.d.
Poljička cesta 39, 21000 Split, Croatia

Phone: +38521 305 656, E-mail: ivan.markota@ericsson.com

Abstract: Development of miniature measuring equipment and sensors led to increasing number of applications that use them. Huge number of such applications can be found in medicine, such as distance monitoring of vital medical parameters. It is very important to provide communication without human control (machine–to-machine) and real-time medical data transfer (telemedicine).
Telecommunication network can be adequate infrastructure for these applications. This paper proposes framework for sensor application development based on Internet Multimedia System (IMS) and Session Initiation Protocol (SIP) Platform. This framework is analysed on medical observing aid prototype that uses data obtained from Bluetooth (TM) Oximeter Sensor.

I. INTRODUCTION

Sensor devices, due to their miniature dimensions and capability of measuring various parameters at place of origin in a non-invasive manner, are suitable for different applications. The connecting glue between diverse sensors/Wireless Sensor Networks (WSNs) and applications is IP-based. In order to move from existed wild, private, vertical sensor-based applications toward wide-available, trustable, secure, horizontal sensor-based application we investigate the IP Multimedia Subsystem (IMS) platform. Since advanced sensors can collect real-time data (video, audio), the platform should support real time data transfer.

The IMS, solution for Next Generation Networks, can provide machine-to-machine communication (M2M) as well as real-time data transfer, necessary for advanced sensor-based applications. Telemedicine is one of those application areas, where medical data is transferred via network in order to enable distance medical examinations. Above mentioned principles have been analysed through prototyping of medical vital signs monitoring in real time.

This paper is organized as follows. Second section describes IMS; its assignment and architecture. Second section deals with Session Initiation Protocol (SIP) which is main protocol in IMS architecture for session control. Since we need to enable real time data transfer it is necessary to use Real-time transport protocol (RTP) which is described in fourth section. Fifth section describes our prototype and measurements. Finally, sixth section presents conclusion.

II. IP MULTIMEDIA SUBYSTEM

IP Multimedia Subsystem (IMS) is network architecture, specified by Third Generation Partnership (3GPP), that provides greater flexibility to operators for the development and launch of multimedia applications. It is based on the Session Initiation Protocol (SIP), which has become the main protocol for session controlling in IP-based next-generation networks [1]. IMS is aimed to merge cellular and Internet networks and in that way to provide cellular access to all Internet services. In that way users can access to all services anytime and anywhere whether they are at home or at move. Even though users with 3G terminals can access Internet without IMS, the main advantages of this architecture are: acceptable Quality of Service (QoS), support for flexible charging and integration of different services. Also, IMS provides all the services using packet-switched technology which is more efficient than circuit-switched technology.

A. IMS Architecture

The IMS has horizontally layered architecture and consists of: Service Layer, Control Layer, Transport Layer and Device layer.

The Service layer comprises application and content servers to execute value-added services for the user.

The Control layer comprises network servers for managing call and session set up, modification and release. The key feature of IMS in this layer, Call/Session Control Function (CSCF) is responsible for session control and processing of SIP signaling. There are three types of CSCF, depending on the functionality they provide: Proxy-CSCF, Serving-CSCF, and Interrogating-CSCF.

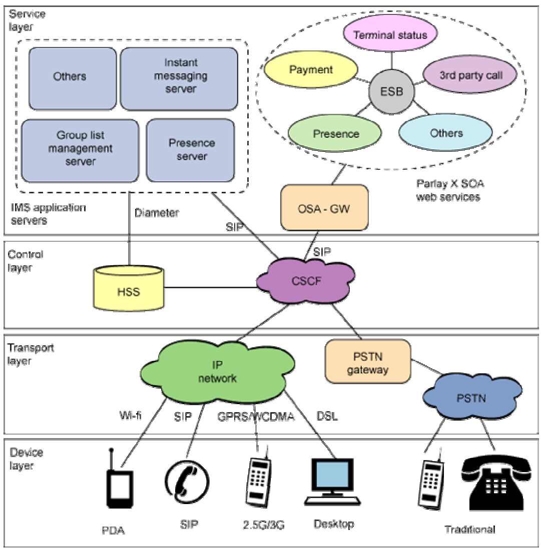

Fig.1 IMS architecture.

The Proxy-CSCF is the first contact point of the system and acts as SIP proxy. The Serving-CSCF is the central node in signaling plane and is responsible for session establishment, modification and release. The Interrogating-CSCF forwards SIP requests and responses to the Serving-CSCF. The Home Subscriber Server (HSS) is the main data storage of user related information that is used to set up sessions.

The Transport Layer provides the mechanism for transporting voice, data and multimedia information.

The Device Layer comprises various devices such as computers, PDAs, mobile phones which can connect to IMS architecture via network.

Classically division of circuit-switched (mature) networks in signaling and media plane is also completely applicable to packet-switched networks. The IMS goes far in separation between signaling and media paths: the IMS terminals are the only nodes that handle both signaling and media. Signaling traverses a set of CSCFs, while media is sent end-to-end (from an IMS terminal to another IMS terminal) traversing only IP routers and in case of mobile access a Gateway GPRS Support Node (GGSN). Besides the central protocol, SIP, signaling plane includes other protocols such as Session Description Protocol (SDP) and Diameter. The media plane converts real-time content (audio and video) into a digital form and transports it using protocols such as Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP).

III. SESSION INITIATION PROTOCOL

The *Session Initiation Protocol* (SIP) is an Internet Engineering Task Force (IETF) standard protocol for establishing, manipulating, and tearing down sessions. SIP can also invite participants to already existing sessions, such as multicast conferences.

SIP is request-response application-layer protocol that provides the capability to: determine the location of the target end point, determine the media capabilities of the target end point via Session Description Protocol (SDP), determine the availability of the target end point, establish a session between the originating and target end point, handle the transfer and termination of calls [2].

A. SIP Architecture

SIP architecture has two basic components SIP User Agent (UA) and SIP Network Server.

The SIP UA is an endpoint component used by user to initiate and answer a call. User Agents are often called User Agent Client (UAC), which initiates calls and represents calling side of a SIP transaction and User Server Client (UAS), which receives calls and represents receiving side.

The SIP Network Server handles signaling associated with multiple calls and its main function is to provide name resolution and user location. It consists of three main groups:

- SIP Register Server that receives registration messages from endpoints regarding current user location and maps the SIP addresses with the physical location in the domain where the endpoint is located.
- SIP Proxy Server that forwards the SIP messages necessary for session establishment regarding current location of invited user. There are two different operating modes for these servers: stateless (the server forgets all the information once the request is sent because information needed to route massage is contained in message itself) and stateful (the server stores previous routing information and use it for routing new messages).
- SIP Redirect Server that helps endpoints to find the SIP UA by providing alternative locations where it can be reachable.

B. SIP Request/Response Syntax

SIP communication, i.e. signalization, consists of various messages. There are two types of such messages: requests and response. Requests are often used to initiate some action and responses are used to confirm that request is received and processed. SIP requests are forwarded from client to server and content of these requests is shown on Fig. 2. Each request consists of a message header and a message body. The message header contains various header fields that serve for routing requests. The message body is used to convey information regarding particular application. An application could be media streaming in which case is Session Description Protocol (SDP) in message body. SDP is protocol used to provide information about media stream such as number, type of media, port number, and payload type. This information is textual feedback to users.

SIP defines following types of requests:
INVITE – initiates session.
ACK – confirms session establishment and can be used only with *INVITE*.
BYE – terminates session. This message can be sent by any of parties.
CANCEL – cancels a pending *INVITE*.
OPTIONS – capability inquiry.
REGISTER – binds a permanent address to current location and it may convey user data

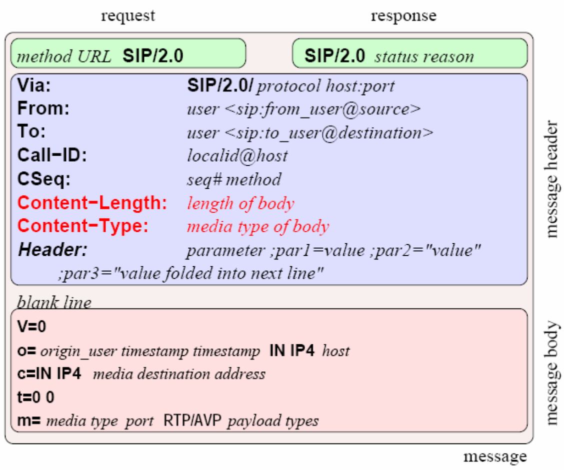

Fig. 2. SIP message syntax.

IV. REAL-TIME TRANSPORT PROTOCOL

Real-time Transport Protocol is an application level protocol that provides transfer of real-time data, such as audio and video, over Internet. RTP does not provide resource reservation or provide other quality-of-service guarantees, but relies on lower-layer services to do so. The data transport is augmented by a control protocol (RTCP) which provides minimal control and identification functionality. RTP and RTCP are designed to be independent of the underlying transport and network layers. RTP does not ensure time delivery data or prevent out-of-order delivery and also does not assume that lower layers are reliable. It just provides necessary information for media to be sent correctly between endpoints. RTP is framework protocol that is intentionally incomplete [3].
Nevertheless RTP is primarily designed for multi-participant multimedia conferences, it can also be applicable for continuous data storage, and control and measurement applications.
RTP usually runs on top of another transport layer protocol - most often the User Datagram Protocol (UDP).

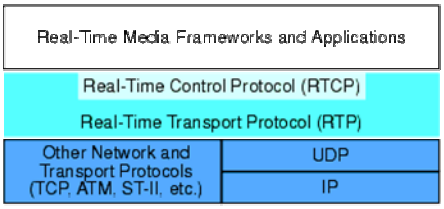

Fig. 3. RTP architecture.

A. Data Packets

The media data for a session is transmitted as a series of packets (RTP stream). Each RTP data packet in a stream has the following header format, illustrated in Fig. 4. The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer. The most important fields in the packet header are timestamp and sequence number.

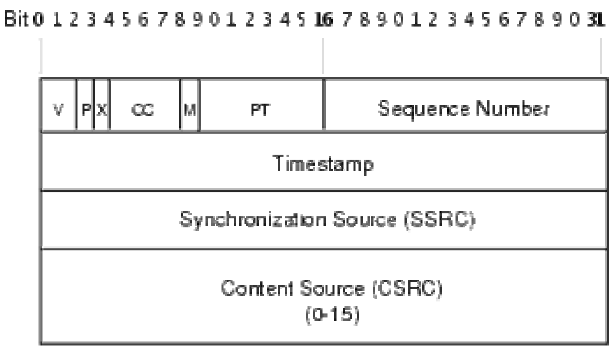

Fig. 4. RTP packet header.

With this two values receiver can reconstruct the timing produced by source, and reorganize arrived packets. It is necessary to have timing information and sequence numbers in packet header because Internet, like other packet networks, occasionally losses and reorders some packets and delays them for amount of time.
The timestamp is used to place incoming packets in the correct timing order and for each packet transmission timestamp increases by the duration time of packet. According to timestamp value receiver can estimate playout delay and in that way compensate jitter (variation in delay). Sequence number is used to detect losses and to restore packet sequence and increments by one for each packet sent.

B. Control Packets

Like we already mentioned RTP protocol is augmented by RTP control protocol in order to monitor the quality of service and to convey information about session participants. Therefore control data (RTCP) packets are sent periodically to all of the participants in the session.

There are several types of RTCP packets: Sender Report, Receiver Report, Source Description, Bye, Application-specific. These reports are used to report the quality of connection between senders and receivers.

The Sender Report (SR) is issued from participant that has sent data packets and includes events since last report. It contains the total number of packets and bytes sent as well as information that can be used to synchronize media streams from different sessions. Session participants that receive data packet issue a Receiver Report. The Receiver Report (RR) contains information about the number of packets lost, the highest sequence number received, and a timestamp and is feedback to sender about what has been received. According to these two reports sender may modify its transmission rates and receiver can detect problems. All compound RTCP packets must include a source description (SDES) element that contains the canonical name (CNAME) that identifies the source. When participant wants to quit session, it sends an RTCP BYE packet. The BYE notice can include the reason that the source is leaving the session.

V. PROTOTYPE APPLICATION AND MEASUREMENTS

The proposed framework is analyzed through a medical observing aid prototype. The prototype consists of monitored client, observer client, and observer on duty IMS application (Fig. 5).

Behind monitored client is a person equipped with medical sensors (in our case oximeter) that looking for medical opinion on measured data. Observer client is tied with a qualified medical observer e.g. physician. The IMS application matches monitored client with observer client based on pool of available and trustable observers.

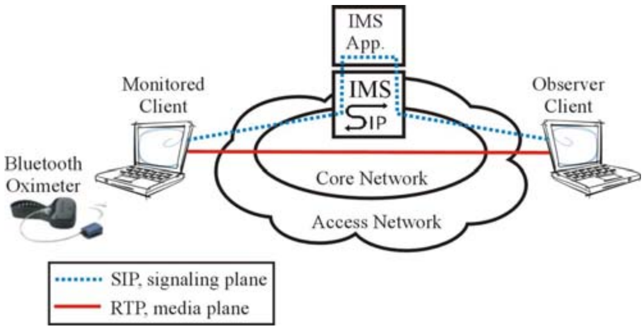

Fig. 5. The medical observing aid prototype

Monitored user uses Nonin Bluetooth 5100 Oximeter to measure medical parameters. The Oximeter is a sensor device that measures human heart rate, oxygen saturation and plethysmograph values. Plethsmograph values represent changes in volume within an organ or whole body. This device offers multiple serial data formats for output data and we are using Serial Data Format 2 (Fig. 6).

FRAME

PACKET

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
1	01	STATUS	PLETH	HR MSB	CHK
2	01	STATUS	PLETH	HR LSB	CHK
3	01	STATUS	PLETH	SpO2	CHK
4	01	STATUS	PLETH	SREV	CHK
5	01	STATUS	PLETH	reserved	CHK
6	01	STATUS	PLETH	reserved	CHK
7	01	STATUS	PLETH	reserved	CHK
8	01	STATUS	PLETH	BTS	CHK
9	01	STATUS	PLETH	SpO2-D	CHK
10	01	STATUS	PLETH	SpO2 Fast	CHK
11	01	STATUS	PLETH	SpO2 B-B	CHK
12	01	STATUS	PLETH	reserved	CHK
13	01	STATUS	PLETH	reserved	CHK
14	01	STATUS	PLETH	E-HR MSB	CHK
15	01	STATUS	PLETH	E-HR LSB	CHK
16	01	STATUS	PLETH	E-SpO2	CHK
17	01	STATUS	PLETH	E-SpO2-D	CHK
18	01	STATUS	PLETH	reserved	CHK
19	01	STATUS	PLETH	reserved	CHK
20	01	STATUS	PLETH	HR-D MSB	CHK
21	01	STATUS	PLETH	HR-D LSB	CHK
22	01	STATUS	PLETH	E-HR-D MSB	CHK
23	01	STATUS	PLETH	E-HR-D LSB	CHK
24	01	STATUS	PLETH	reserved	CHK
25	01	STATUS	PLETH	reserved	CHK

Fig. 6. Nonin Serial Data Format 2.

The Nonin Serial Data Format number 2 has following: a frame consists of 5 bytes; a packet consists of 25 frames; three packets are transmitted each second and each packet value is 8 bit encoded [4].

Nonin is connected with the monitored client device by Bluetooth wireless connection and communicates with it through serial port. Nonin collects data in real time so it's not possible to transmit such data within SIP messages. SIP is used only to enable signalization and to provide (via Session Description Protocol) additional information for RTP stream set up.

Fig. 7 depicts SIP signalization. Both clients register themselves and then monitored client invites observer client. Observer on duty is an IMS application that takes care about available and trustable observer.

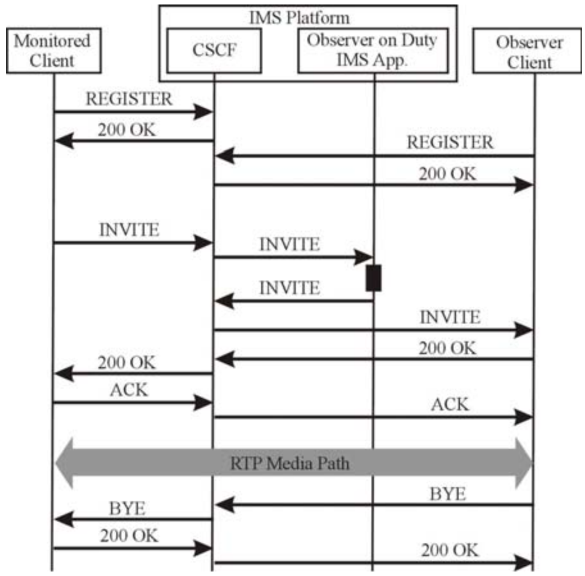

Fig. 7. Simplified SIP sequence diagram.

Since data is transferred over packet network, it is necessary to compensate network impact. Network introduces variable delays (jitter), reorders packets, and even losses packets. A playout buffer on receiver side reproduces data in correct way (Fig. 8).

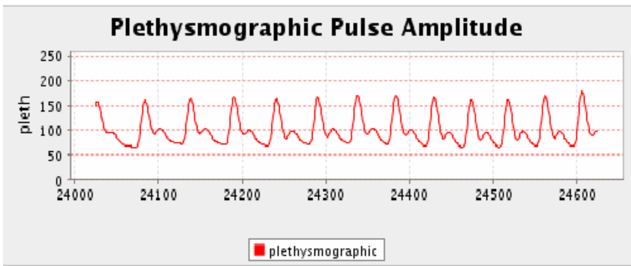

Fig. 8. Plethysmograph.

The prototype is used for measurement of time dependency of sent, received and played packets. Prior playing, receiver needs to store packets in buffer, and after playout delay reproduces them according to timestamps. Packets arrived after playout delay are considered lost and are not played out. Since we are using custom data consisted of 333 ms packets, our playout delay is set up to 1 second. During this time packets are stored in the buffer and reorganized according to their sequence numbers. Measurement results are depicted by Fig. 9. Time dependency of aggregate number of packets on source side (sent packets) is, expectably, linear since it is determined by sampling frequency and data format. Slightly differences between delays of received packets on destination side results in nonlinear received packet line.

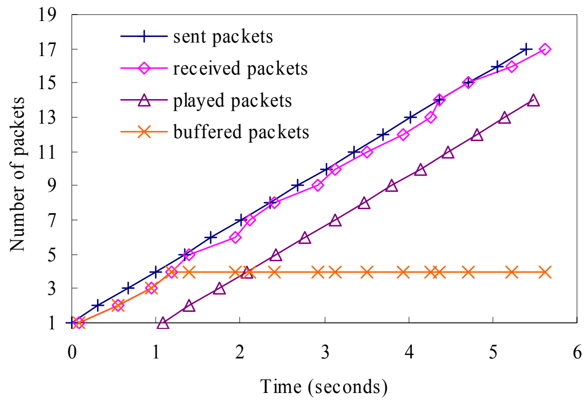

Fig. 9. Number of packets time dependency.

The playout buffer introduces playout delay in order to compensate jitter. Hence, time dependency of aggregate number of played packets on destination side is linear and parallel to sent packets line, but shifted in time for playout delay. Buffer length is constant in time since reaching its value after playout delay. That means maximal packet delay is smaller than a packet time duration.

VI.CONCLUSION

The aim of this paper was to demonstrate suitability of IP Multimedia Subsystem for sensor-based applications. This particular prototype deals with personal sensors, but similar principle could be deployed with WSN comprises number of sensors. In that case a WSN will be part of infrastructure queried by sensor-based IMS applications. A WSN answer doesn't have to be limited to simple numerical value, but could be also a real-time stream as shown in this paper.

REFERENCES

[1] G. Camarillo, M. Garcia-Martin: The 3G IP Multimedia Subsystem: Merging the Internet and the Cellular Worlds. John Wiley and Sons, Ltd., England, 2004

[2] J. Rosenberg et al.: SIP: Session Initiation Protocol. IETF RFC 3261, June 2002

[3] G. Camarillo: SIP Demystified. McGraw-Hill, 2002

[4] H. Schulzrinne, et al.: RTP: A Transport Protocol for Real-Time Applications. IETF RFC 3550, July 2003

[5] Nonin Model 4100 containing Bluetooth® Technology Specifications