PROPOSED TRANSPORT PROTOCOLS SUITE FOR WIRELESS MEDICAL BODY AREA NETWORKS

Sotirios Kontogiannis[1], George Kokkonis[2] and Stavros Valsamidis[3]

[1]Dept. of Informatics and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece
[2]Dept. of Applied Informatics, University of Macedonia, Thessaloniki, Greece
[3]Dept. of Accounting and Finance, TEI of East Macedonia and Thrace, Kavala, Greece

ABSTRACT

Wireless Body Area Networks (WBAN) is the recent trend for localized and personalized medical services. For that purpose, several tele-medicine architectures and WBAN frameworks for patient monitoring have been proposed.

This paper presents a new medical protocol MESETP (Medical Services Transport Protocol), that underlies data delivery for both real-time and non real-time medical services. Furthermore, performance tests of MESETP with existing real time protocols and non real-time protocols used by existing medical services are examined and comparison results are presented.

KEYWORDS

Network Protocols, Wireless Body Area Networks, Mobile Networks, M-Health services, Web 2.0 and NGN

1. SYSTEM ARCHITECTURE OF A WIRELESS BODY AREA NETWORK

Wearable health monitoring systems (WHMS) are a commodity due to the recent technological advances in bio-sensing devices, microelectronics and wireless communication protocols [1, 2]. More specifically a category of WHMS systems designed for out of hospital use are wearable body area systems. Wearable WHMS communication medium and protocols used for non-interruptible medical data transmission of patient body measurements along with the participating nodes are part of wireless body area networks called WBANs. A subset of WHMS systems. Apart from expensive live supporting or critical systems, wearable medical systems are consisted of low-cost certified sensors and computing components, with custom-made software included. A wearable WHMS system transfers medical data wirelessly from a patient to a central station. Such systems are also designed with patient mobility provisions [3, 4, 12, 13, 14].

Wearable Health Monitoring Systems are composed by three distinctive parts: The Central Station (CS), where the collection of data takes place, the Sensors Data Controller (SDC), where sensors instrumentation, communication protocols and logic reside and the wearable medical sensors. Wearable health monitoring system architecture is illustrated at Figure 1 [1, 2, 3, 4, 12, 13, 14].

Since the need for best satisfaction at reduced deployment cost is the future in Tele-medical and body health systems [16], cheap open source SDC controller such as the e-Health v.2.0 wearable board [5] is commonly used. The e-Health data collector is consisted of an AVR microcontroller

(Arduino-based) sensors interface board (E-health board) connected to a Raspberry Pi embedded ARM microprocessor (RPi) via a SPI (Serial Peripheral Interface) shield. Different types of medical sensors can interface to the e-Health board and collect patient medical data. Commonly used sensor types for the V2.0 e-Health board are the following: Pulse and blood oxygenation (SpO2), airflow (breathing), body temperature, Electrocardiogram (ECG), blood sugar, galvanic skin response, blood pressure (sphygmomanometer), patient position (accelerometer) and electromyography (EMG) sensors. If required, a real time diagnostic image camera can be connected to the RPi in order to capture patient photos and videos for medical imagery analysis [15]. Similarly, to e-Health platform an approach called wrenching for the creation of cheap home services for the elderly and medical bedridden is proposed at [16, 17].

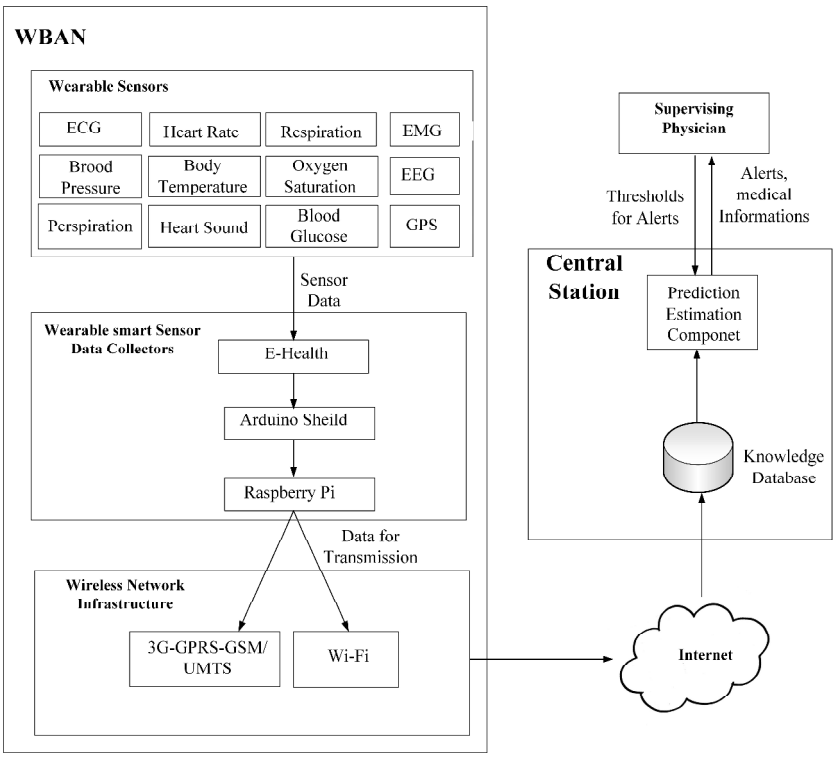

Figure 1. Open e-Health Wearable health monitoring system architecture (WHMS).

The medical sensors collected data are then pushed via SPI connectivity to the RPi microprocessor. The RPi can either store data into SD-card or transmit data wirelessly using any of the five available connectivity options (Wi-Fi, 3G, GPRS, Bluetooth and ZigBee). Medical data are then transmitted to the hospital central station that performs permanent measurement database storage and applies the medical monitoring service logic [12, 13, 14].

Focusing on the different types of medical measurements and medical acts requirements, it is evident that one protocol cannot support adequately and efficiently all kind of medical services. Therefore, classification of services is imminent [14].

2. MEDICAL SERVICES DIFFERENTIATION BASED ON DATA ACQUISITION

Medical sensors' data are classified by authors into four distinct categories, depending on resolution, data update intervals and priority:

Real-time Services: In this category real-time traffic is being generated by sensors that are performing measurements continuously. Examples of such sensors are the electrocardiogram, the accelerometer; the electromyography sensors et al. Real-time services follow the store and forward data approach at SDC controller operating system scheduler intervals. This means that aggregation, differentiation, quantization or other types of correlation function transformation is applied into sensor measurements before they are packed into packets for transmission. Then packet transmission is performed at constant time intervals of 50msec-500msec. The receiver end awaits packet reception at the same steady rate. Real-time services are sensitive to jitter and packet drops. However, for real-time services that the measurement repeatability period is small, dropped packets can be ignored since the measurement can be repeated at a following interval. On the other hand, maintaining constant minimum jitter among measurements is a more significant factor for measurement precision and validity.

Periodic Services*:* In this category, sensor data are sent to the Central Station at specified by a physician time intervals periodically. Examples of such services are the notification service, the reminding service, the nutrition proposal service and the health cognitive services. Cognitive services provide a health model based on previous time series of sensor measurements and use complex decision algorithms whether a patient's attributes fit into a descriptive model [18]. Furthermore, periodic services can also be divided into periodic stream and trend services. Periodic stream services utilize non-expiring or late expiring sessions with the central monitoring station, while periodic trend services perform short session lifetime periodic data update requests.

Close to Real-time Services*:* In this category, sensor measurements are triggered on demand either by a physician or from a health cognitive service. That is, data are transmitted to the medical centre when such necessity is encountered. For example: Immediate sensor data transmission when indication of abrupt medical status is set (rapid temperature increase, gyroscopic sensor; fast posture change, accelerometer; rapid acceleration change). Another example is when a sensor provides results that are not within medical thresholds. Then on demand messages called alerts are populated and sent (thresholds can be set remotely by medical personnel who may also provide prescription services). Close to real-time data transmission needs to be treated with respect to the content of the message transmitted. An acknowledgment of reception needs to be sent on every message received and a persistent retransmission mechanism needs to be initiated for packet loss.

Interactive medical services: This category includes multimedia and Haptic services. For example, services which the doctor interacts with a patient at the surgery. Such devices are sophisticated robotic systems that give or even get a sense of touch, strength or precise position accuracy interactively. Interactive voice and video services require constant delays with respect to the buffering mechanism used at the receiver end and intermediate routers, while Haptic services require both minimum delay and jitter. Concluding, video services have high throughput requirements and are more tolerant to jitter. The commonly used protocol for video and voice data delivery is RTP/RTCP [6, 7]. Haptic services are services of low throughput, less jitter tolerance and high data priority [9].

3. TECHNICAL REQUIREMENTS FOR MEDICAL SERVICES PROTOCOLS

Based on authors' differentiation upon medical services and prior to presenting authors proposed protocol, per service technical characteristics are outlined:

A Real-Time Medical protocol (RTM) that supports medical services shall maintain the following capabilities:

- An RTM protocol shall authenticate itself with the receiver using information carried out on each packet header.
- An RTM protocol shall have a header no bigger than of the data it carries.
- An RTM protocol shall include data recovery mechanism based on cumulative negative acknowledgements issued on constant or selective intervals only. The missing packets need then be retransmitted through another flow (recovery flow) in order to maintain data resolution and integrity for post processing operations only.
- An RTM protocol shall include sequence numbering for the reordering out-of-order reception at the receiver (minimal flow control at the receiver).
- An RTM protocol at the receiver end shall maintain a buffer of small as a normal real-time reception queue and a huge buffer for the recovery flow.
- An RTM protocol shall measure and monitor network conditions per flow at the receiver end (simplicity approach).
- An RTM protocol shall have rate adaptive or scaling capabilities.
- An RTM protocol shall maintain signals for on demand flow rate scaling by the application layer and shall not be TCP friendly or shall include QoS provisions.
- An RTM protocol shall maintain a level of authentication and data integrity.
- An RTM protocol usually shall not include timestamp measurements because the sensors used do not include accurate timing or have no timing at all.

A Close to Real-time Medical protocol (CRM) or on-demand protocol for medical services shall maintain the following capabilities:

- A CRM protocol shall be of small size, with a rate no more than the MTU.
- A CRM protocol shall include no authentication handshake mechanism.
- A CRM protocol shall maintain flow control (per packet acknowledgment).
- A CRM protocol shall aggressively retransmit if no acknowledgment of reception occurs.
- A CRM protocol packet can be potentially marked with a maximum priority flag in order to maintain flow prioritization mechanism.

Periodic Medical protocols (PM) can be classified into periodic streaming that their non-transmission interval is no more than TCP sockets timeout and non-periodic streaming of transmission intervals of more than sockets timeout.

- All types of Periodic services require session and authentication provisions.
- Streaming periodical services may binary streams for data delivery and maintain an open session for a significant time interval (up until a high timeout value). This is persistent connection behaviour. Non-streaming periodic services may use connection-disconnection mechanism for every period interval with low timeout values.

Interactive Medical protocols (IM): This category of protocols includes multimedia medical services which flows requirements are covered by the characteristics of the RTP protocol [6, 7, 8]. Moreover, Haptic interactive services that are also part of the Interactive services, require a different protocol approach than of RTP, discussed in detail at [9].

In the following section, taking into account both the aforementioned functional and technical characteristics of medical services differentiation, authors proposed a protocols' suite called MESETP.

4. PROPOSED MEDICAL SERVICES PROTOCOLS SUITE

In order to accomplish long-range transmission of medical data, from the SDC controller to the remote central station, a suite of protocols were designed and implemented, named MESETP (Medical Services Transport Protocol). This protocol suite includes three sub-protocols, where each one tries to meet the requirements of a different medical service class (periodic, close to real-time and real-time), described at previous sections. The sections that follow describe each sub-protocol of MESETP suite.

4.1 MESETP protocol for periodic medical services

For periodic services MESETP uses an application layer Restful protocol [10] with different HTTP header options per service. If it is a non-streaming periodic service, then HTTP v.1.1 POST requests are used or HTTP 1.0 requests with keep-alive timeouts. The main problem of HTTP v.1.x requests is that only one request at a time can be served by a connection, thus leading to degradation of concurrent capabilities of a threaded web service. Since HTTP v.1.x is a blocking request response process, HTTP clients that send streaming requests (requests back to back), seemingly result to a numerous of blocked requests.

The solution to the HTTP requests blocking problem is the use of multiplexing HTTP v.2 requests in order to allow multiple requests and responses data to be in flight at the same time. If it is a streaming service, then an HTTP v.2 [11] connection is created, that allows multiple concurrent requests on the same connection. Furthermore, HTTP v.2 is a binary protocol with optimal header compression and thus is less error prone than HTTP v.1x textual stream protocols.

4.2 MESETP protocol for close to real-time medical services

MESETP protocol for close to real-time services is based on a custom TCP half-open connection mechanism to transmit sensor data. Data acquisition is accomplished with the exchange of only two data packets (two packets exchange protocol). The first packet is used for a connection/data initiation transmission (TCP connection initiation SYN packet). In the TCP SYN initiation packet a payload exists where sensor data are packed. The packet second is the acknowledgement packet (TCP acknowledgement packet reception) send by the receiver upon reception of a TCP|SYN packet.

CRM packet exchange is terminated with the use of a TCP RST packet sent by the sender, indicating CTM connection termination. Then a reply RST packet is sent by the receiver indicating acknowledgement of connection termination. In order to avoid FIN_WAIT state delays, FIN packets were not used as TCP connection termination dictates. Data collection of this arbitrary data exchange is performed and buffered at kernel level thus reducing unnecessary application processing delays during data exchange.

Close to real-time data transmit need to be treated with respect to the content of the message transmitted. An acknowledgment of reception is sent on every message received and persistent retransmission mechanism is set for every packet loss within TCP RTO/5 timeout interval (no more than 100ms).

Prioritization of CRM packets is performed with the use of PUSH and URG flags:

- The URG|SYN flag is used to inform the Central station that certain data within this SYN segment is urgent and should be prioritized. If the URG flag is set then the central station uses the urgent pointer, that indicates how much of the data in the segment, counting from the first byte, is urgent. This indicates that inside a TCP|SYN packet the payload may contain both urgent and non urgent measurements.
- CRM packet retransmission is performed using the SYN|PUSH flags set. When a packet acknowledgment of a TCP SYN CRM packet is not received for RTO/5 timeout, then a PUSH|SYN retransmission packet is send immediately. Having the PUSH flag on, indicates the receiver end to flush data immediately from the receiver buffer. This operation can be repeated at least N times (network parameter). Then if all re-transmission attempts fail the packet is considered lost. If no connection is established, the sender-receiver communication continues to transmit the next packet in the sender buffer.

4.3 MESETP protocol for real-time services

For real-time services MESETP protocol for real-time named Adaptive Medical Sensor Transmission Protocol (AMESETP) is proposed. In general, the AMESETP senses the network status at the receiver end and depending on the detected network conditions, it adapts the sending rate of the data packets and sensor values quantization levels, in order to overcome lack of network resources.

4.3.1. AMESETP Header

The header of the AMESETP protocol is shown at Figure 2 and includes the following fields:

- The Sequence Number is used by the negative acknowledgment mechanism to reinforce protocol's reliability.
- The Checksum is used for error-checking of transmitted data
- The Sending Rate Level (SRL) field informs the receiver for the sending data rate and the grouped data values per packet (see Table 2).
- The Sensor ID (SenID) and the PatientID indicates to which sensor and patient the data correspond to.
- The Encryption (En) field declares if encryption mode is enforced. Symmetric 3DES with ECB mode encryption is used at the application layer.

Bits	*0*	*1-4*	*5-7*	*8 – 15*	*16-23*	*24-31*
0	Sequence Number				CheckSum	
32	En	SRL	SenId	PatientId		-
	Data (grouped values)					

Bits	*0-3*	*4-7*	*8 – 15*	*16-31*
0	SensorId	ACK Sequence Number		CheckSum
32	PatientId			Length
	Data (Last Received Sequence Number, Dropped Sequence Numbers)			

Figure 2. AMESETP protocol packet header and header of cumulative negative acknowledgment packets (CNAK).

Table 1. AMESETP protocol overhead compared to other protocols.

	AMESETP	***UDP***	***RTP***	***TCP***
OVERHEAD (bytes)	7	8	12+8(UDP)	20
PAYLOAD (bytes)	30-375	30-375	30-375	30-375
EFFICIENCY %	81.08-98.17	78.95-97.91	60.00-94.94	60.00-94.94

The protocol overhead results of the AMESETP compared to other common transport protocols such as the UDP, the RTP and the TCP are presented at Table 1. From Table 1, the AMESETP protocol is more efficient that the other protocol as it uses small overhead. Since the real-time measurements payload varies from 30 to 375 bytes. The maximum data rate of a real-time sensor is one value of 2 bytes per ms, which means 2 Kbytes/s per sensor. AMESETP protocol packs these values into packets.

4.3.2. AMESETP Rate Adaptive transport mechanism

When the network is heading to congestion, the AMESETP protocol lowers its throughput in order to avoid congestion. The AMESETP throughput's reduction is achieved by two factors; by lowering the sending rate with packet grouping, or by minimizing the packet size through lowering the quantization levels of the sensor's produced values. The sending rate, the packet size and the quantization levels of the sensor's values are shown at Table 2.

The combination of packet rates and the bytes per sensor value and the values per packet that the AMESETP protocol rate adaptation is shown at at Table 2 and consists of 12 distinct levels. The higher the level the more frequent and more quantized the sensor values transmitted.

AMESETP attaches a sequence number to every transmitted packet. The transmitter sends the packets with the grouped values as soon as they are created. A copy of this packet is stored in a buffer. The Cumulative Negative Acknowledgement (CNAK) mechanism at the receiver informs the transmitter which packets have been dropped. The transmitter resends the lost packets. The receiver sends a CNAK with the lost sequence numbers and the last received sequence number every k=1 sec. The packet header for the CNAK is depicted at Figure 2.

Initial Transmission Rate is Level 5 at Table 2. The packing level is 100 values per packet. The packet rate is 10 packets/sec and the quantization level is 16 bits per value, which means 2 bytes per value. The minimum overhead corresponds to Transmission Rate 2 which uses 8 bits per value and 30 values per packet. The maximum overhead is used at Transmission Rate 11, which uses 6 bits per value and 500 values per packet.

Table 2. AMESETP Protocol Rates and Compressed Rates for Real-time Services.

Transmission Rate	Sending Rate Level (SRL)	Sending Rate (Packets/s)	Values Grouping (Values/ Packet)	Packet Size (Bytes/ packet)	Quantization Levels (Bits/value)
High data rate	1	33	30	60	16
Q-High data rate	2	33	30	30	8
Medium data rate	3	20	50	100	16
Q-Medium Data rate	4	20	50	50	8
Normal data rate	5	10	100	200	16

Q-Normal data rate	6	10	100	100	8
Q-Slow data rate	7	5	200	300	12
Q-very slow data rate	8	5	200	250	10
Q-very very slow data rate	9	5	200	200	8
Extra slow data rate	10	4	250	250	8
Compressed 1 data rate	11	2	500	375	6
Compressed 2 data rate	12	2	500	250	4

4.3.3. AMESETP Congestion control mechanism

The sending rate level is controlled by the receiver. The AMESETP monitors the packet loss, and the Inter-Package Gap (IPG) variation based on Eq. 1

$$Td_i=dt(R_i-R_{i-1})-Li-IPG_0 \qquad (1)$$

Tdi is the IPG for packet i, Ri is the reception time of packet i, Li is network queuing latency of packet i and i-1and IPG0 is set by medical service as the real-time transmission interval.
The sending rate of the AMESETP transmitter is periodically calculated using Eq. 2. When detected increased packet loss, the receiver notifies the transmitter to lower the sending rate via negative acknowledgements. There are 12 states of the Sending Rate Levels (SRL) at Table 3.

$$SRL = \begin{cases} l_0 = 5 \\ l_{i-1} + 1 \;, pl < \frac{pl_{max}}{2} \;\; or \frac{SJ_{i-1}}{4} > SJ_i \\ l_{i-1} \qquad\qquad , \frac{pl_{max}}{2} < pl < pl_{max} \\ l_{i-1} - 1 \qquad\qquad , pl_{max} < pl \end{cases} . \qquad (2)$$

Packet Loss (pl) is divided into three intervals: The first interval pl<plmax/2 or SJi<SJi-1/4 corresponds to perfect network conditions. The AMESETP at these conditions tries to increase its sending rate. The second interval plmax/2<pl<plmax corresponds to good network conditions and the protocol attempts to keep the SRL steady. The last interval pl>plmax corresponds to bad network conditions and the protocol lowers its throughput by lowering the SRL. The factor plmax is set 20 times less the maximum acceptable packet loss for real-time graphic images transmission (10%) and is set to 0.5 % packet loss.

Another method the AMESETP uses to lower its throughput is the network adaptive quantization. The quantization levels are changing according to the network conditions. When network conditions are good, the quantization levels increase and vice versa. The quantization levels have direct effect to the value of the sensor and the transmitted packet size.

5. AMESETP AND MESETP CRM PERFORMANCE TESTS

In the following experimental scenarios, authors put to test MESETP performance for real-time and close to real-time medical services. MESETP was tested against UDP and RTP protocols as representative protocols for real-time services (Scenario I) and against HTTP v.1.0 requests for close to real-time services (Scenario II). Both scenarios include flows initiated at a medical sensors' controller (an RPi B+ controller with 512MByte RAM and a 32bit 700MHz ARM processor- running Rasbian Jessie – Linux distribution) that sends medical data to the central station (an Intel core I7 32bit system at 3.2GHz and 4GB DDR3 RAM – running Ubuntu server 14.10 Linux distribution).

The communication channel used between the central station and the sensor data controllers is a Wi-Fi IEEE802.11n wireless network transmission over ADSLv2. Wi-Fi transmission provides a symmetrical radio channel of an average of 54Mbit/s from the medical data collectors to the central access point where data transmitted to the central medical station using ADSLv2 wired transmission technology. ADSL ATM used in our experimental scenario utilizes a maximum downlink capacity of 24Mbps and uplink of 1Mbit/s (Average speeds measured for the test link at 12Mbit/s Downlink and 784Kbit/s Uplink).

This scenario simulates the operation of home health monitoring platforms. In our case 802.11n USB dongle were plugged into the RPi controllers to connect to patient's Wi-Fi network (ADSLv2 Wi-Fi router) located at patient's house, in order to transmit sensor data via DSL to the hospital's central station.

5.1. Scenario I: AMESETP performance testing

In this scenario, Real-Time Medical sensor data flows (RTMs) are transmitted from multiple sensor controllers to the central station. The RTM flows used and put to test are UDP, RTP over HTTP tunneling and AMESETP. The experiment is carried out using 12, 24, 36 and 48 sensors. Since the number of RPi's at authors disposal is limited to six, sensor data are simulated and distributed evenly among the RPi distributed system in terms of flows (2 minimum flows up to 8 flows per RPi). Each sensor flow transmits a total number of no more than 10000 sensors' generated values per to the central station. The contention increase mechanism that is used to the transmitter is the insertion of one flow per second up until all flows are inserted.

The experiment is repeated for each protocol accordingly. Sensor data transmitted are taken of an ECG (Electrocardiogram data) sensor microcontroller that includes three analog electrodes. Such ECG data are collected from the RPi (SDC) every 25-100msec and the A2D sensor resolution is 16bit (2bytes of data every 1us, equivalent to real-time 2Kbps of data per second per flow). The flows are sent from the RPi sensors data collector to the central station.

Table 3 depicts the percentage of packet drops of each protocol for different number of flows. It is obvious that as the number of flows increases, the packet drop is also increased. The AMESETP protocol present by far smaller packet drops than the UDP protocol. The RTP and the AMESETP show almost the same results, as far as the packet loss is concerned. The advantage of the AMESETP protocol is that it manages to recover all the lost packets through its reliability mechanism and requires less complex setup than RTP/RTCP multicast or HTTP transmission.
Furthermore, RTP/RTCP multicast UDP flow transmission is not supported by most Internet routers and more specifically are not supported (part of the Service level agreement) of the 3G network providers. This is because 3G technology cannot carry out adequately high throughput/video resolution RTP flows.

Table 3. AMESETP, UDP, RTP flows % average packet drops.

# flows	%Packet Drops UDP flow	%Packet Drops RTP flow	% Packet Drops AMESETP flow
12	0.1	0	0
24	0.73	0.01	0.02
36	1.2	0.2	0.05
48	3.2	1.1	1.1

Figure 3 and Table 4 depict the Inter Packet Gap (IPG) of each protocol for different number of flows. It is clear that as the number of flows increases, the IPG for all protocols is also increased. The AMESETP protocol manages to outperform UDP protocol that presents the worst performance characteristics in this case study scenario. The AMESETP protocol manages also to keep the IPG lower than the other two protocols. This means that the AMESETP protocol transfers packets at the receiver end at a more constant rate than RTP.

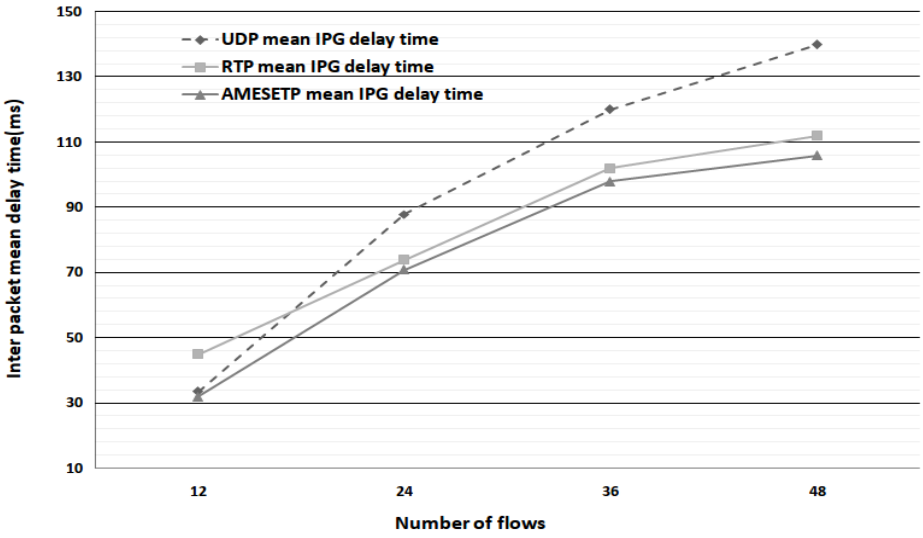

Figure 3. Average Inter-Packet Gap (ms) per flow of AMESETP, UDP, and RTP protocols over number of flows.

Table 4. AMESETP, UDP, RTP mean IPG (ms).

# flows	UDP flow mean IPG (ms)	RTP flow mean IPG (ms)	AMESETP flow mean IPG (ms)
12	33.5	45	32
24	88	74	71
36	120	102	98
48	140	112	106

5.2. Scenario II: MESETP Close to Real-time Medical protocol performance testing

In scenario II MESETP CRM protocol is tested against full HTTP medical data transmission for close to real-time medical services. In this test a body temperature sensor sends data of body temperature (thermometer), using HTTP protocol flows and CRM flows. A uniform random process sets the measurements intervals from 50ms up to 100ms. Each flow sends 10000 measurement requests before termination.

The central station receives 60 HTTP real-time flow measurements using HTTP v.1.0 and then using MESETP CRM (distributed as 10 flows per RPi controller) and packet loss and delay is measured at the receiver end. The purpose of this experiment is to test performance MESETP CRM protocol over TCP in comparison to HTTP TCP real-time flows.

The percentage request timeouts/fails and the mean delay time of the two protocols are presented at Table 4. It is clear that the MESETP CRM protocol provides by far lower request timeouts/fails than the HTTP TCP protocol. The average end-to-end delay of the MESETP CRM protocol is also significantly lower than of HTTP protocol. The improved results of the MESETP CRM protocol is due to the custom TCP half-open handshake connection mechanism that uses to transmit sensor data.

Table 4. MESETP CRM, HTTP flows % average packet drops and average mean delay time (ms).

# flows	HTTP flow % request timeouts/fails	MESETP CRM % connection timeouts/fails	HTTP flow mean delay time (ms)	MESETP CRM flow mean delay time(ms)
60	5.4	0.5	470	36

6. CONCLUSIONS

This paper presents a new protocol called MESETP for the transmission of medical data among medical sensors of different data transmission requirements. The design and implementation of authors' proposition was put to test on low-cost off the shelf hardware components, open-source software and programming environments such as the Linux (RPi) and the Arduino IDE.

Medical sensor protocols are classified by authors into real-time, close to real-time and periodic based on resolution; data update intervals and priority of the measurements transmitted to the central station. Based on derived categorization and requirements, authors proposed protocol design was also decoupled into three different protocols: AMESETP protocol for real-time medical services, MESETP CRM (Close to Real-time Medical) protocol for close to real-time and MESETP PM (Periodic Medical) stream and non stream for periodic services.

All protocols of the MESETP protocol suite are technically described in detail. Moreover, Authors MESETP protocols experimentation using open E-health system, has shown that the for real-time services AMESETP protocol is a promising candidate, since it outperforms in most of the cases other existing medical protocols, such as UDP and RTP. Authors also stress out with experimental results the increased performance characteristics of MESETP protocol for close to real-time services compared to existing HTTP protocol used for these services. Concluding, authors proposed protocols in the MESETP suite do not include a protocol implementation for interactive services. This is considered as future work.

REFERENCES

[1] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40(1), January 2010.

[2] V. Custodio, F. J. Herrera, G. Lopez, and J. I. Moreno, "A Review on Architectures and Communications Technologies for Wearable Health-Monitoring Systems", Sensors, 12(1), pp. 13907-13946, 2012.

[3] C. Doukas and I. Maglogiannis, "Bringing IoT and Cloud Computing towards Pervasive Healthcare", In Proc. of sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS, pp. 922-926, 2012.

[4] A. Bouromuis, M. Feham, and A. Bouchachia, "Ubiquitous Mobile Health Monitoring system for the elderly (UMHMSE)", International Journal of Computer Science and Information Technology, 3(3), pp. 74-82, 2011.

[5] e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi. https://www.cooking-hacks.com/-documentation/tutorials/ehealth-biometric-sensor-platform-Arduino-raspberry-pi-medical, 2013.

[6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, 2003.

[7] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[8] J. Ott, C. Perkins, "Guidelines for Extending the RTP Control Protocol (RTCP)", RFC 5968, September 2010.

[9] G. Kokkonis, K. E. Psannis, M. Roumeliotis, S. Kontogiannis, "A Survey of Transport Protocols for Haptic Applications", In proc. of Panhellenic Conference on Informatics, 1(0), IEEE Computer Society, 2012.

[10] L. Richardson, S. Ruby "Restful Web Services", ISBN 9780596529260, O'Reilly, 2007.

[11] M. Belshe, R. Peon, M. Thomson. "Hypertext Transfer Protocol version 2 (HTTP v.2)", RFC 7540, 2015.

[12] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications and approaches", Wireless Communications and Mobile Computing, 13(18), pp. 1587-1611, 2013.

[13] A. Bourouis, M. Feham, and A. Bouchachia, "A new Architecture of a Ubiquitous Health Monitoring System:A Prototype of Cloud Mobile Health Monitoring System", CoRR, abs/1205.6910, 2012.

[14] S. Ullah, P. Khan, N. Ullah, S. Saleem, H. Higgins, and K. S. Kwak, "A review of wireless body area networks for medical applications", CoRR, abs/1001.0831, 2010.

[15] I. Lee, G. Pappas, R. Cleaveland, J. Hatcliff, B. Krogh, P. Lee, H. Rubin, and L. Sha, "High-confidence medical device software and systems", Computer, 39(4), pp. 33-38, April 2006.

[16] S. Sharma, J. Wong, U. S. Timb, and S. Gadia, "Bidirectional Migration between Variability and Commonality in Product Line Engineering of Smart Homes", International Journal of Systems Assurance Engineering and Management, 4(1), pp. 1-12, Springer, 2012.

[17] S. Sharma, H Yang, J Wong, and C Chang, "Wrenching: Transient Migration from Commonality to Variability in Product Line Engineering of Smart Home System", In proc of 9th International Conference on Smart Homes and Health Telematics (ICOST), Montreal, Quebec, Canada, pp. 230-235, Springer, 2011.

[18] S. Sharma, U. S. Timb, M. Paytonc, H. Cohlyd, S. Gadiae, J. Wonge, and S. Karakalaf, "Contextual motivation in physical activity by means of association rule mining", Egyptian Informatics Journal 16(3), pp. 243-251, 2015.