Performance of UDP-Lite for IoT network

Lucas Chai† and Regina Reine∗

†Bachelor of Technology Student, Electrical and Computing department, Curtin University Malaysia
∗Lecturer, Electrical and Computing department, Curtin University Malaysia
E-mail: 7e4a3278@student.curtin.edu.my and reginareine@curtin.edu.my

Abstract. In recent years, the Internet of Things (IoT) has gained popularity due to greater data accessibility and the ability to incorporate prediction algorithm in the IoT devices. However, IoT devices have limited energy resources as the devices are mostly light in weight. By improving the efficiency of IoT communications systems, energy efficiency of the IoT devices can be increased. One way is to utilise the suitable transport protocol for the IoT networks as transport protocol indirectly influences the energy efficiency and the quality of service of the IoT communication systems. Conventional transport protocol, such as Transmission Control Protocol (TCP), introduces latency due to its three way handshakes, large header size and strict reliability rule which can lead to unnecessary energy usage for some applications. In this paper, we propose to use User Datagram Protocol Lite (UDP-Lite) for the IoT networks. The performance of the total packet loss for these UDP-Lite configurations in IoT networks is investigated and the numerical analysis illustrates the effectiveness of the proposed method.

1. Introduction

Wireless sensor networks (WSNs) are commonly used for Internet of Things (IoT) communications. However, WSNs have problems such as low processing speeds, low-power usage and limited for short ranged wireless transceivers [1]. Moreover, the connection between IoT and WSN devices are commonly bridged by different and multiple proprietary solutions which usually have incompatibility issues with each other [2]. Thus, rendering a smart device ecosystem is hard to manage.

The recent trend is to use the Internet Protocol (IP) to establish the Internet connection among the WSNs [3]. However, this type of connection is not optimized for the low-power WSNs. The transport protocol TCP is unable to distinguish between the packets are lost or the packets are dropped because of the congestion in the wireless links. When the nodes fail, energy exhaustion and sleep duty cycles may cause further performance degradation [1]. Due to the energy limitation constraint, the IoT networks only use low-power Layer-2 technologies such as low-power WiFi technologies, Bluetooth Low Energy, and most notably IEEE 802.15.4 [4].

In the TCP/IP network, strict reliability and in-order delivery check are employed for every byte of the payload transmission. Implementing this strict reliability TCP rule in IoT network is inefficient as the IoT devices may be in sleep mode to save power. The IoT devices mostly transmit burst data and a long-lived connection will deplete the limited power source. IoT applications that require low-latency service are unable to use TCP as the three way handshakes will introduce delays every time the communication is established [5]. Furthermore, in the lossy
WSN, if there are missing packets, head-of-line blocking may occur and further delay will be introduced due to the re-transmission process [6].

The connectionless User Datagram Protocol (UDP) [7] is viewed as a good alternative for the IoT devices communications. Unlike TCP, it does not have strict reliability rule and it is suitable for low-latency and low-power applications. Traditionally, UDP is mostly used in Voice over IP (VoIP) and streaming applications. VoIP is a method of transmitting voice communications and media content through Internet Protocol, it is also an example of a real-time application. In IoT network, UDP can be beneficial for IoT communications that focus on the low-latency communications rather than reliability. UDP header size is light-weight compared to the TCP header-size which will indirectly contribute to the energy efficiency in payload transmission. In order to reduce the header size further, UDP-Lite was introduced. One of the disadvantages of connectionless protocol is the lack of congestion control may results in higher packet drops [8]. However, delivery with an acceptable level of packets lost are preferred for multimedia transmission to maintain good level of throughput [9].

Currently, there is lack of study in using light header connectionless protocol in the IoT networks. Current conducted research [10] [11] on UDP-Lite is mostly done without taking IoT in mind and their low-power behaviors, such as UDP-Lite using wired or satellite networks. This paper will investigate the novelty of using UDP-lite as the transport protocol in place of UDP for the IoT communications.

This remainder of this paper is structured as follows. Section II presents the introduction of the UDP-Lite. In Section III, the hardware and software setup for the simulation is presented. In Section IV, the simulation results of the performance of the the modified sensors are illustrated and discussed. Finally, the concluding remarks are drawn in Section V.

2. Light header for Internet of Things
2.1. UDP and UDP-Lite
In the high traffic wireless channels, one of major disadvantages of UDP is that the corrupted packets are discarded instantly. This is detrimental to the network system where it is desirable to maximize the total send and receive rate in average to the energy used within constraints of specified allocated time and quality of service. Due to the lack of flow control in UDP [7], Real-Time Transport Protocol (RTP) is commonly used. Most common example is the real time streaming applications such as VoIP in order to measure packet loss, delay and jitter as these measurements provide the information related to payload type and packet delivery monitoring. [12] [13] [14].

RTP may help with flow control on the receive. However, UDP will still drop damaged payloads as the checksum covers the whole payloads as illustrated in the Fig.1. The Bit Error Rate (BER) of the protocol cannot be taken into account as measurement or used for other protocols in the TCP/IP stack as UDP already discarded the damaged packets [7]. Hence, it will reduce the performance of the entire network as significant numbers of packets being dropped unnecessarily.

UDP-Lite allows applications to deliver partially damaged payloads through the use of partial checksums [15]. The UDP-Lite packet is divided into two parts; sensitive part and insensitive part. The sensitive part ranges from the first octet of the IP header and the last octet of the checksum coverage field as illustrated in Fig.2. The insensitive part would cover the payload. This allows packets that have been partially corrupted to be received. If the checksum covers the whole packet, UDP-Lite is the same as UDP and is semantically identical to UDP. Length field has been replaced by the checksum coverage field. It is important for the upper link layers to have the knowledge of the packet size and this information is provided by the IP header. Checksum coverage field refers parts of the UDP-Lite payloads that are covered by the checksum. If the field is 0, this indicates that the entire UDP-Lite packet will be covered by the checksum. If
the field is 8, this means that there is no protection over the payload, but this is beneficial to applications that use UDP-Lite and wish to avoid protection of the packet payloads [15].

**Figure 1.** Standard UDP Header (8) bytes) [7]

**Figure 2.** Standard UDP-Lite Header (8) bytes) [15]

### 2.2. Lower Layer Considerations

UDP-Lite packets will not be discarded by the lower layer protocols when errors occurred in the insensitive part. For the lower layer links that support partial error detection, the UDP-Lite header can be used to hint at where errors do not need to be detected using the checksum coverage field in the header. These lower layers must use a strong error detection mechanism so that the sensitive parts of the packets can be checked for errors and be discarded [16].

UDP-Lite allows the physical layer to implement unequal Forward Error Correction (FEC) to reduce the probability of corruption before passing through the transport layer, as error-prone links can benefit from being aware of error sensitive packets [15]. Furthermore, UDP-Lite improves performance for applications such as multimedia streaming by delivering damaged payloads and allowing low BER. Using supports from RTP profiles will detect the errors and give the attempt to recover the damage for playback. In practical implementation, these profiles are what is commonly known as audio/video codecs. Examples of these error tolerant codecs like G.729, AMR utilize damaged data payload streams and overall reduce final packet loss and improve the performance of the application under wireless links that commonly experience lossy performances [17].

### 3. Simulation Setup

Simulation was performed with the use of NodeMCU modules integrated with an Espressif ESP8266 chip. The chip is embedded in the ESP8266 Arduino core which contains libraries to communicate over WiFi using TCP and UDP protocols through the light weight Internet Protocol (lwIP) stack.

The studies and the simulation were conducted in a constant environment in the network laboratory in Curtin university. There were two different setups for the transceivers. First of
all, sender and receiver had communication with each other in 5 meter distance between each other with complete line of sight (LOS) and without obstructions. For the second setup, each device communicate between each other in 10 meter distance with an obstruction to simulate the non line-of-sight (NLOS) condition. These setups are labelled as 5 Meter Unobstructed and 10 Meter Obstructed that are easily represented in the Figure 3.

Figure 3. Network Laboratory Layout and Module Setup

The configuration of the protocols for the simulation were UDP, UDP-Lite, UDP-Lite v2 and UDP-Lite v3. UDP-Lite provides partial checksum to payloads. UDP-Lite v2 differs from UDP-Lite as UDP-Lite v2 has a value of 8 for the checksum coverage field which means that it will not perform checksum on the payloads. UDP-Lite v3 disables checksum functionality at both transceivers when receiving or sending packets, the payloads for these packets will not be protected nor be checked for checksums. Each of these cases are summarized into Table 1 and will be simulated to obtain the throughput of each case. The performance can be obtained through comparing the amount of packets sent and received, which will be different for each test as packet count and send rate are increased to test more payloads sent and test for the need of flow control respectively.

Table 1. Protocol Configurations and Setups

| Case | Protocol      | Checksum |  |
|------|---------------|----------|---|
| 1    | UDP           | FULL     |  |
| 2    | UDP-Lite      | PARTIAL  |  |
| 3    | UDP-Lite v2   | NONE     |  |
| 4    | UDP-Lite v3   | DISABLED |  |

4. Simulation Results
4.1. Protocols Performance
The data were analysed during the simulation period and compiled into tables for the UDP, UDP-Lite, UDP-Lite v2 and UDP-Lite v3 transport protocols. Parameters that will changed for each case are send rate, packet count and the setups (5 Meter Unobstructed & 10 Meter Obstructed). Numerical results were observed and studied for all setups and configurations by numerous times and the average packet counts received were calculated.
4.1.1. 5 Meter Unobstructed Communication  In the first module setup, both sender and receiver modules were placed within a range of 5 meter distances between each other and have no obstruction in between them. This means that each module has line of sight of each other. In this setup, it is expected to have relatively low delay time and low numbers of packets dropped or lost.

| Protocol     | Transmission Rate (ms) | Avg. Packets Received | Lost Packets | Avg. Delay (ms) |
|--------------|------------------------|-----------------------|--------------|-----------------|
| UDP          | 2                      | 9988                  | 12           | 2.154           |
| UDP-Lite     | 2                      | 9983                  | 17           | 2.157           |
| UDP-Lite v2  | 2                      | 9978                  | 22           | 2.162           |
| UDP-Lite v3  | 2                      | 9968                  | 32           | 2.158           |
| UDP          | 5                      | 9996                  | 4            | 5.162           |
| UDP-Lite     | 5                      | 9989                  | 11           | 5.162           |
| UDP-Lite v2  | 5                      | 9999                  | 1            | 5.166           |
| UDP-Lite v3  | 5                      | 9997                  | 3            | 5.171           |

Table 3. Performance for 25,000 Packets Distance 5 Meters

| Protocol     | Transmission Rate (ms) | Avg. Packets Received | Lost Packets | Avg. Delay (ms) |
|--------------|------------------------|-----------------------|--------------|-----------------|
| UDP          | 2                      | 24969                 | 31           | 2.163           |
| UDP-Lite     | 2                      | 24979                 | 21           | 2.159           |
| UDP-Lite v2  | 2                      | 24551                 | 449          | 2.196           |
| UDP-Lite v3  | 2                      | 24754                 | 246          | 2.177           |
| UDP          | 5                      | 25000                 | 0            | 5.166           |
| UDP-Lite     | 5                      | 24984                 | 16           | 5.159           |
| UDP-Lite v2  | 5                      | 24990                 | 10           | 5.160           |
| UDP-Lite v3  | 5                      | 24996                 | 4            | 5.159           |

Table 4. Performance for 50,000 Packets Distance 5 Meters

| Protocol     | Transmission Rate (ms) | Avg. Packets Received | Lost Packets | Avg. Delay (ms) |
|--------------|------------------------|-----------------------|--------------|-----------------|
| UDP          | 2                      | 49935                 | 65           | 2.151           |
| UDP-Lite     | 2                      | 49978                 | 22           | 2.159           |
| UDP-Lite v2  | 2                      | 48171                 | 1829         | 2.158           |
| UDP-Lite v3  | 2                      | 49568                 | 432          | 2.156           |
| UDP          | 5                      | 49999                 | 1            | 5.161           |
| UDP-Lite     | 5                      | 49999                 | 1            | 5.161           |
| UDP-Lite v2  | 5                      | 49991                 | 9            | 5.163           |
| UDP-Lite v3  | 5                      | 49995                 | 5            | 5.164           |

4.1.2. 10 Meter Obstructed Communication  In the second module setup, both sender and receiver modules were located within range of 10 meter distances between each other. Obstruction was placed between the sender and the receiver to simulate the NLOS characteristic.
### Table 5. Performance for 10,000 Packets Distance 10 Meters

| Protocol   | Transmission Rate (ms) | Avg. Packets Received | Lost Packets | Avg. Delay (ms) |
|------------|------------------------|-----------------------|--------------|-----------------|
| UDP        | 2                      | 9921                  | 79           | 2.170           |
| UDP-Lite   | 2                      | 9821                  | 179          | 2.200           |
| UDP-Lite v2| 2                      | 9913                  | 87           | 2.185           |
| UDP-Lite v3| 2                      | 9768                  | 232          | 2.212           |
| UDP        | 5                      | 9997                  | 3            | 5.162           |
| UDP-Lite   | 5                      | 9998                  | 2            | 5.163           |
| UDP-Lite v2| 5                      | 9998                  | 2            | 5.161           |
| UDP-Lite v3| 5                      | 9992                  | 8            | 5.164           |

### Table 6. Performance for 25,000 Packets Distance 10 Meters

| Protocol   | Transmission Rate (ms) | Avg. Packets Received | Lost Packets | Avg. Delay (ms) |
|------------|------------------------|-----------------------|--------------|-----------------|
| UDP        | 2                      | 24788                 | 212          | 2.167           |
| UDP-Lite   | 2                      | 23831                 | 1169         | 2.192           |
| UDP-Lite v2| 2                      | 24552                 | 448          | 2.160           |
| UDP-Lite v3| 2                      | 22884                 | 2116         | 2.256           |
| UDP        | 5                      | 24980                 | 20           | 5.164           |
| UDP-Lite   | 5                      | 24799                 | 201          | 5.163           |
| UDP-Lite v2| 5                      | 24963                 | 37           | 5.161           |
| UDP-Lite v3| 5                      | 24741                 | 259          | 5.167           |

### Table 7. Performance for 50,000 Packets Distance 10 Meters

| Protocol   | Transmission Rate (ms) | Avg. Packets Received | Lost Packets | Avg. Delay (ms) |
|------------|------------------------|-----------------------|--------------|-----------------|
| UDP        | 2                      | 49163                 | 837          | 2.185           |
| UDP-Lite   | 2                      | 47978                 | 2022         | 2.192           |
| UDP-Lite v2| 2                      | 49164                 | 835          | 2.227           |
| UDP-Lite v3| 2                      | 45223                 | 4430         | 2.307           |
| UDP        | 5                      | 49845                 | 155          | 5.218           |
| UDP-Lite   | 5                      | 49916                 | 84           | 5.169           |
| UDP-Lite v2| 5                      | 49798                 | 202          | 5.178           |
| UDP-Lite v3| 5                      | 45570                 | 4777         | 5.509           |

### 4.2. Results and Analysis

Performances of the three protocol configurations UDP, UDP-Lite, UDP-Lite v2 and UDP-Lite v3 were evaluated based on the transmission rate, total packets dropped, the range of the transmitter distance and the obstructions type. The results have been compiled within the Table 8 and Table 9. It can be observed that packet loss occurs in the different transceiver setups, protocols and configurations. As the distance between transceivers and amount of packets are increased, the packet loss percent increases. The percentage of loss differs between each protocol and transmission rate.

The UDP-Lite v2 does not compute checksum for the payload where as normal UDP-Lite will compute checksum over the entire packet. The benefit to this comes from the computation of checksum may not be needed if the wireless link was already error prone or
Table 8. Performance for protocols with 5 Meter Unobstructed

| Protocol    | Transmission Rate (ms) | Packets Sent | Packets Lost | Packet Loss(%) |
|-------------|------------------------|--------------|--------------|----------------|
| UDP         | 2                      | 10000        | 12           | 0.12           |
| UDP-Lite    | 2                      | 10000        | 17           | 0.17           |
| UDP-Lite v2 | 2                      | 10000        | 22           | 0.22           |
| UDP-Lite v3 | 2                      | 10000        | 32           | 0.32           |
| UDP         | 5                      | 10000        | 4            | 0.04           |
| UDP-Lite    | 5                      | 10000        | 11           | 0.11           |
| UDP-Lite v2 | 5                      | 10000        | 1            | 0.01           |
| UDP-Lite v3 | 5                      | 10000        | 3            | 0.03           |
| UDP         | 2                      | 25000        | 31           | 0.12           |
| UDP-Lite    | 2                      | 25000        | 21           | 0.08           |
| UDP-Lite v2 | 2                      | 25000        | 449          | 1.80           |
| UDP-Lite v3 | 2                      | 25000        | 246          | 0.98           |
| UDP         | 5                      | 25000        | 0            | 0.00           |
| UDP-Lite    | 5                      | 25000        | 16           | 0.06           |
| UDP-Lite v2 | 5                      | 25000        | 10           | 0.04           |
| UDP-Lite v3 | 5                      | 25000        | 4            | 0.01           |
| UDP         | 2                      | 50000        | 65           | 0.13           |
| UDP-Lite    | 2                      | 50000        | 22           | 0.04           |
| UDP-Lite v2 | 2                      | 50000        | 1829         | 3.66           |
| UDP-Lite v3 | 2                      | 50000        | 432          | 0.86           |
| UDP         | 5                      | 50000        | 1            | 0.00           |
| UDP-Lite    | 5                      | 50000        | 1            | 0.00           |
| UDP-Lite v2 | 5                      | 50000        | 9            | 0.02           |
| UDP-Lite v3 | 5                      | 50000        | 5            | 0.01           |

if the multimedia protocol was already expecting damaged payloads [15]. This method will rely on the implemented error detection or correction schemes such as RTP and FEC to provide an overall higher performance. Overall, while performance does not take a larger hit compared to normal UDP and UDP-Lite when packets are dropped or lost, it is still beneficial to receive as much undamaged packets as possible while still allowing damaged frames to be processed by other error correcting link layer protocols.

In UDP-Lite v3, the protocol now does not compute checksum for the entire packet, this renders the header of the UDP-Lite packet completely unprotected compared to UDP-Lite v2 which disables checksum only on the payload contents. The benefit to UDP-Lite v3 is to lessen computation time by disabling all checksum functionality at both sender and receiver. However, this protocol might not be beneficial to our objective as packets may be lost due to damage before the payload is even brought into the protocol stack. The authentication of the packet would fail in this transport protocol as IP header is within the sensitive part of a UDP-Lite packet, damages to the sensitive part will cause the packet to be dropped [15].

Without taking distance and the obstruction into account, the 2 millisecond send rate for each packet from the sender causes many packets to be completely lost as there is no flow control between the transceivers while still losing some of the packets damage to the UDP-Lite header or UDP payload. For the 5 millisecond send rate, flow control is not as necessary as packets start to arrive at receiver at a computable and sustainable pace that means any further packet drops or losses could be inferred to as damage to the packet header or payload. However, this is not the only reason why packets are dropped but this simulation aims to determine the
performs of the protocol configurations. Even accounting for the possibility of overflow and damage to packet, from the results we can see that packet loss was much more apparent in the 10 meter obstructed setup. For 10 meter setup, 25,000 and 50,000 packets sent had the most percentage change of packet lost/dropped reaching minimum 200% to maximum around 2000% for all protocol configurations compared to their 5 meter counterparts. From all the results shown, higher amounts of packets sent will increase the probability that more and more packets will be lost.

It can be inferred that unobstructed wireless channels would bring better performances to the NodeMCU modules and specifically using normal UDP for transport as it had the best result out of all configurations. However, for lossy wireless links which stem from large distances between transceivers and have obstruction, UDP and UDP-Lite v2 prove to have 25% to 46% of difference with packets lost with UDP-Lite v2 losing out on the amount of packets received. While it is possible to conclude UDP as the undisputed choice for this type of wireless channel, RTP and FEC along the link layers are able to improve the performance of packets received by UDP-Lite v2 and especially when the application is real-time and handles with audio and video payloads. In a wireless environment where the link is comprised mostly of media transmissions, the benefit of a protocol like UDP-Lite will increase the effectiveness of the transmission rates. UDP-Lite v3 protocol performance within the wireless links show that there is no trade off for disabling checksum computation as it does not alleviate packets being dropped in any setup, the IP headers information should always be protected as it contains the information needed to process packets.

| Protocol       | Transmission Rate (ms) | Packets Sent | Packets Lost | Packet Loss (%) |
|----------------|------------------------|--------------|--------------|-----------------|
| UDP            | 2                      | 10000        | 79           | 0.79            |
| UDP-Lite       | 2                      | 10000        | 179          | 1.79            |
| UDP-Lite v2    | 2                      | 10000        | 87           | 0.87            |
| UDP-Lite v3    | 2                      | 10000        | 232          | 2.32            |
| UDP            | 5                      | 10000        | 3            | 0.03            |
| UDP-Lite       | 5                      | 10000        | 2            | 0.02            |
| UDP-Lite v2    | 5                      | 10000        | 2            | 0.02            |
| UDP-Lite v3    | 5                      | 10000        | 8            | 0.08            |
| UDP            | 2                      | 25000        | 212          | 0.85            |
| UDP-Lite       | 2                      | 25000        | 1169         | 4.68            |
| UDP-Lite v2    | 2                      | 25000        | 448          | 1.79            |
| UDP-Lite v3    | 2                      | 25000        | 2116         | 8.46            |
| UDP            | 5                      | 25000        | 20           | 0.08            |
| UDP-Lite       | 5                      | 25000        | 201          | 0.80            |
| UDP-Lite v2    | 5                      | 25000        | 37           | 0.148           |
| UDP-Lite v3    | 5                      | 25000        | 259          | 1.04            |
| UDP            | 2                      | 50000        | 837          | 1.68            |
| UDP-Lite       | 2                      | 50000        | 2022         | 4.04            |
| UDP-Lite v2    | 2                      | 50000        | 835          | 1.67            |
| UDP-Lite v3    | 2                      | 50000        | 4430         | 8.86            |
| UDP            | 5                      | 50000        | 155          | 0.31            |
| UDP-Lite       | 5                      | 50000        | 84           | 0.17            |
| UDP-Lite v2    | 5                      | 50000        | 202          | 0.40            |
| UDP-Lite v3    | 5                      | 50000        | 4777         | 8.86            |
5. Conclusion
The simulation results illustrated that UDP-lite have less dropped packets compared to the UDP due to the flexibility of the checksum. This will be beneficial for the type of payload transmission where receiving corrupted payloads is preferred than none at all. If IoT applications require checksum, the link layer protocols such as RTP and FEC can be used to support UDP-Lite.

References
[1] Culler D, Estrin D and Srivastava M 2004 Computer 37 41–49 ISSN 0018-9162
[2] Fysarakis K, Askoxylakis I, Soultaotos O, Papaeftathiou I, Manifavas C and Katos V 2016 1–7
[3] Vasseur J P and Dunkels A 2010 Interconnecting Smart Objects with IP: The Next Internet (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.) ISBN 0123751659, 9780123751652
[4] Sethi P and Sarangi S R 2017 J. Electrical and Computer Engineering 2017 9324035:1–9324035:25
[5] Fairhurst G and Wood L 2002 Advice to link designers on link automatic repeat request (ARQ) BCP 62 RFC Editor
[6] McQuistin S, Perkins C and Fayed M 2016 TCP goes to hollywood Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video NOSSDAV ’16 (New York, NY, USA: ACM) pp 5:1–5:6 ISBN 978-1-4503-4356-5
[7] Postel J 1980 User datagram protocol STD 6 RFC Editor
[8] Al-Kashoash H A A, Al-Nidawi Y and Kemp A H 2016 Congestion analysis for low power and lossy networks 2016 Wireless Telecommunications Symposium (WTS) pp 1–6
[9] Stanislaus W, Fairhurst G and Radzik J 2005 Cross layer techniques for flexible transport protocol using UDP-Lite over a satellite network 2005 2nd International Symposium on Wireless Communication Systems pp 706–710 ISSN 2154-0217
[10] Reine R and Fairhurst G 2003 MPEG-4 and UDP-Lite for multimedia transmission Proceedings of Post-Graduate Networking Conference
[11] Singh A, Konrad A and Joseph A D 2001 Performance evaluation of UDP Lite for cellular video Proceedings of the 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video NOSSDAV ’01 (New York, NY, USA: ACM) pp 117–124 ISBN 1-58113-370-7
[12] Schulzrinne H, Casner S, Frederick R and Jacobson V 2003 RTP: A transport protocol for real-time applications STD 64 RFC Editor
[13] Perkins C 2003 RTP: Audio and Video for the Internet 1st ed (Addison-Wesley Professional) ISBN 0-672-32249-8
[14] Peterson L L 2007 Computer Networks 4th ed (Morgan Kaufmann) ISBN 1-55860-832-X
[15] Larzon L A, Degermark M, Pink S, Jonsson L E and Fairhurst G 2004 The lightweight user datagram protocol (UDP-Lite) RFC 3828 RFC Editor
[16] Karn P, Bormann C, Fairhurst G, Grossman D, Ludvig R, Mahdavi J, Montenegro G, Touch J and Wood L 2004 Advice for internet subnetwork designers BCP 89 RFC Editor
[17] Schulzrinne H and Casner S 2003 RTP profile for audio and video conferences with minimal control STD 65 RFC Editor