Survey on Transport Layer Protocols: TCP & UDP

Santosh Kumar, Sonam Rai
Graphic Era University, Dehradun (India)

ABSTRACT

This paper presents a comprehensive survey on the two most popular transport layer protocols; TCP and UDP. The aim is to understand the key terminologies of TCP and UDP and to understand the key differences between them. This paper lists out the functions and basic operation of both protocols (TCP and UDP) and finally a comparison between the two protocols is presented. In, addition, this paper provides a brief survey on the performance of TCP over wireless networks.

General Terms

Transport protocol, Congestion control, Flow control.

Keywords

TCP, UDP, QoS, ACK, Segment, Datagram.

1. INTRODUCTION

Due to the day-to-day increasing popularity and high range of adoption of wireless technologies like WiMAX, WiFi, GPRS etc., there is an increase in demand for mobile multimedia applications. The most important requirement of these multimedia applications is to fulfill the quality requirements as well as optimal utilization of network resources, and one of the important factor to fulfill all these requirements is the transport layer protocol[2]. In order to define the detail architecture of the network, several transport layer protocols have been studied and analysed from the QoS (Quality of Service) point of view. However, this paper is focusing only on the two main protocols of transport layer (TCP and UDP). TCP provides a reliable mechanism to transfer the data which results in increase of delay. UDP does not provide any congestion control mechanism but provides less delay as compared to TCP. Now a days , Quality of Service (QoS) assurance is one of the issues as the behavior of transport layer protocols affects QoS. And, most of the multimedia applications use UDP as their main transport protocol. However, for the varied QoS of distinct multimedia applications, the UDP fails to provide satisfactory results[2]. The paper is organized as follows; section 2, describes the transport layer and its services. Section 3, describes the transport layer protocols (TCP and UDP) in detail, while section 4, provides a comparison between the two protocols (TCP and UDP). Finally section 5, describes conclusion and future work with reference to this review work.

2. TRANSPORT LAYER

The basic role of the transport layer is to support end-to-end communication between the two hosts in the network. It is an important part of the protocol hierarchy which provides reliable, cost-effective data transmission services from the source to the destination. The services provided by transport layer are: connection-oriented service, reliable delivery, flow control, and multiplexing[4]. Figure 1 illustrates the popular protocol stack of internet[8].

Application (DHCP,FTP,HTTP,SMTP,SNMP,TELNET)
Presentation (MIME,XDR,TLS,SS)
Session (NETBIOS,SAP,SOCKS,PPTP)
Transport **(TCP, UDP,** DCCP, SCTP, RSVP, RIP)
Network (IP,IPsec,IGMP,IPX,AppleTalk)
LLC Sub layer (HDLC,PPP,SDLC)
MAC Sub layer (CSMA/CA,CSMA/CD,OFDMA)
Physical (SONET, BLUETOOTH, POTS, PON, OTN)

Fig 1: Protocol Stack[8]

From the Figure 1, it has been seen that the lowest layer at the bottom of the protocol stack is the physical layer which is only concerned with putting bits on the medium. The layer above the physical layer, which is called as data link layer provides a link between two or more nodes so that the data can be delivered from source to destination and it also provides the services such as framing of bits, error detection, reliable delivery of data and media access control[15]. The layer above the data link layer, is the network layer which provides routing of packets over the network. The next layer above the network layer is the transport layer which provides the data transport services between two nodes in the network. It also provides the demultiplexing service for applications. The transport layer does uses the network layer addresses to identify a connection because it does not have its own address [15]. The uppermost layer of the protocol stack is called as application layer, which is identified by a name, is the only user-visible identity of a node. There are seven layers working in the protocol stack. Each of them has a specific function and a specific set of protocols that are working in each layer. Each layer provides a specific service to the layer above it. In this paper, only the transport layer is considered and focus is only on the detailed working of this layer which is provided in the next section. Further, the protocols that are working in this layer are taken into consideration. The transport layer has several protocols, but, this paper presents the review of the two basic protocols of the transport layer which are TCP and UDP.

2.1 Services of Transport Layer

The various services provided by a transport-layer protocol are listed below[8]:

- **Connection-oriented services-** To interpret the connection as a data stream provides several benefits to applications. It is normally easier to deal with the connection-oriented model instead of dealing with the connection less models.

- **Byte-oriented processing-** Instead of processing the messages in the existing communication system format, it becomes easier for an application to process the data as a sequence of bytes. This helps the applications to work easily with various underlying message formats.
- **In order delivery-** To guarantee that packets of data are received in the same order as they were sent is a desirable feature which is made possible through the use of segment numbering, with the receiver passing them to the application in the same order.
- **Reliable delivery-** Due to the network congestion and errors, the packets may be lost during the transmission. With the help of an error detection code, such as a checksum, checks that whether the data is corrupted or not, it also verifies the correct receipt by sending an ACK or NACK message to the sender.
- **Flow control-** The rate of data transmission between two nodes must be managed, in order to prevent a fast sender from transmitting more data that cannot be supported by the receiver, causing the receivers buffer overrun.
- **Congestion control-** This mechanism controls traffic entry into a network, in order to avoid the congestion in network by avoiding oversubscription of any of the processing or link capabilities of the intermediate nodes.
- **Multiplexing services-** A port provides multiple endpoints on a single node. Each computer applications will be listening for information on their own ports, which will enable the use of more than one network service at the same time.

3. TRANSPORT LAYER PROTOCOLS

Transport layer protocol acts as an essential part of the protocol hierarchy that provides end-to-end communication between two hosts in the network[5]. Transport layer protocols provides various features such as congestion control, reliability, in-sequence delivery, and flow control that contribute to performance and quality offered by the communication network. In case of multimedia applications, the correct choice of a transport layer protocol affects the quality requirements such as low delay, jitter, packet loss and throughput etc[2]. Some of the transport layer protocols are listed below [19]:

- **TCP(Transmission control protocol)-** TCP is one of the basic protocol of the Internet protocol suite. It provides various functions such as reliable, In-sequence delivery of a stream of bytes from a program on one computer to another program on the other computer.
- **UDP(User datagram protocol)-** UDP is also one of the basic protocol of internet protocol suite. With the help of UDP a host sends the message in the form of datagram, to other hosts located on a network, without requiring to set up a transmission channel before the actual communication is started.
- **DCCP(Datagram congestion control protocol)-** DCCP is a transport layer protocol which is message oriented. It offers certain features like congestion control mechanism, reliable connection setup, feature negotiation, and Explicit Congestion Notification.
- **SCTP(Stream control transmission protocol)-** It is a transport layer protocol, which is similar in functions as of TCP and UDP. It provides few features similar to both the protocols. Like UDP, it is message oriented, and like TCP it provides ordered delivery of messages.
- **RSVP(Resource reservation protocol)-** This is a transport layer protocol which has the function to reserve resources across the network for an integrated services Internet. RSVP protocol provides setup of resource reservations which is initiated at the receivers side for multicasting or unicasting.
- **RIP(Routing information protocol)-** This is a routing protocol, which uses the hop count as a metric for the routing. RIP implements a limit on the number of hops allowed in a path from the source to a destination to prevent routing loops[19].

TCP acts as a standard transport protocol for Internet and is suitable for wired network and gives a poor performance in case of wireless networks[5]. Most of the multimedia applications which are based on internet employ the user datagram protocol (UDP) as their transport protocol[18]. UDP performs better and is suitable for applications where time is a constraint irrespective of ordered delivery, whereas TCP is suitable where time is not a constraint and reliable delivery is important. This is why, this review takes into account only two protocols TCP and UDP, because TCP and UDP are the core protocols of the transport layer and each of them have a certain advantage over each other but when considered for different cases.

3.1 User datagram protocol

UDP is a simple, transport layer protocol that does not guarantee any reliability and in order delivery of the packets. It supports both multicasting and broadcasting. It is very much suitable for applications that prefers packet loss to jitter or we can say which have high delay requirements[1]. UDP is considered where the in time delivery of data is important rather than reliable delivery. A UDP packet which is called as datagram, is divided into two parts: a header and a payload. UDP employs a cyclic redundancy check (CRC) so as to check for the integrity of packets; therefore, it can detect any error in the packet. If it detects an error in the packet, it is then declared lost and discarded. This protocol is defined to make the datagram available for the packet-switched communication in an interconnected set of computer networks[11]. UDP does not provide any congestion control mechanism. Congestion control is needed in order to prevent the network from entering into a congested state in which very less useful work can be done. Thus the absence of congestion control mechanism in UDP has become a serious problem today. Many researchers have proposed new mechanism to perform adaptive congestion control in UDP, however the problem still exists[13-14].

3.1.1 UDP datagram

The UDP transmits the packets in the form of datagram. The datagram format is shown below in Figure2.

Source port address (16 bits)	Destination port address (16 bits)
Total length (16 bits)	Checksum (16 bits)

32 bits

Fig 2: UDP Datagram Format[7]

The description of each field in detail is as follows [11]:

- **Source port address-** This field indicates the port of the sending process which sends the datagram.
- **Destination port address-** It indicates the port of the destination process to which the datagram is to be sent.
- **Length-** This field specifies the length (in bytes) of datagram which includes the header also.
- **Checksum-** This field is an optional 16-bit one's complement of the one's complement sum of a pseudo-IP header, UDP header, and UDP data, where the pseudo-IP header contains source IP address and destination IP address, protocol, and UDP length.

3.1.2 Basic operation of UDP

The operation of UDP protocol is very simple. When UDP is invoked by the application layer, the following operations are performed by UDP[16]:

- UDP encapsulates the data of users into datagrams.
- Finally forwards these datagrams to the IP layer for the transmission.

On the other side, these datagrams are then forwarded to UDP from the IP layer. UDP then removes the data from the datagram and forwards to the upper application layer. In UDP, a port is a number that specifies the application which is using the UDP service. It can be assumed as an address of the applications. The port number is also been used by the UDP client on the receivers end so that it can know that to which application the user data has to be forwarded[16].

3.1.3 Applications of UDP

There are various applications that use UDP as their transport protocol, like Routing information protocol, Simple network management protocol, Dynamic host configuration protocol etc [17]. Voice traffic and video traffic is generally transmitted over the network by using UDP protocol. As real-time video and audio streaming protocols are designed to handle the loss of packets, so there is only a slight decrease in the quality of audio/video. Since, TCP and UDP, both the protocols run simultaneously over the same network, many businesses have observed that increasing UDP traffic from the real-time applications is affecting the performance of applications that are using TCP protocol such as accounting, database systems etc.[17].

3.2 Transmission control protocol

TCP is a transport layer protocol which is connection-oriented and it provides a reliable byte stream to the upper layer, called as the application layer. TCP has a mechanism based on positive acknowledgments and also provides a congestion avoidance mechanism to reduce the transmission rate when the network becomes overloaded[1]. We can also say that TCP is a time-tested transport layer protocol that provides several features like reliability, flow control and congestion control, being a heavy protocol[6]. TCP is a popular protocol which supports reliable delivery irrespective of the form of the underlying network. TCP is called as robust protocol because it can adapt to different network conditions. In principle, the TCP should be able to operate over a wide range of communication systems such as from hard-wired connections to packet-switched or circuit-switched networks[10]. To ensure the reliable delivery of the data over the network, the TCP employs a mechanism where the sender maintains a buffer, called a sliding window, of data that has been sent to the receiver. A receiver acknowledges received data by sending acknowledgement (ACK) packets. When a sender receives an ACK packet from the receiver for the data in its window, it removes that data from the window, since it has been successfully transmitted to the receiver. This mechanism of TCP is called as window based transmission. TCP employs this mechanism for flow control, so that a receiver can tell the sender, when it cannot process the data at the rate in which it is arriving. This mechanism also informs the sender that how much buffering space is available at the receiver's end, in order to avoid the overfill of receiver's buffer window[3].

3.2.1 Functions of TCP

The various functions of TCP are listed below[10]:

- **Data transfer-** The TCP has the ability to transfer a continuous stream of data between the users in the form of segments for transmission through the network.
- **Reliable delivery-** The TCP must have the ability to recover from data that may be damaged, lost or may be duplicated over the network. This is done by assigning a sequence number to each segment being transmitted over the network and receiving a positive acknowledgment (ACK) on successful delivery. If the ACK is not received within a specific time interval, the data is retransmitted. With the use of sequence numbers, the receivers end order segments in correct sequence, that may be received out of order and to avoid duplicate packets. Damage is handled in TCP by adding a checksum to each segment which is being transmitted, finally the checking is done at the receiver, and the damaged segments are then finally discarded.
- **Flow control-** TCP provides a mechanism that helps the receiver to control the amount of data sent by the sender. This is done by returning a "window" with every ACK packet indicating the acceptable range of sequence numbers beyond the last segment successfully received.
- **Multiplexing-** TCP provides a set of ports within each host so that many processes within a single host can use TCP communication facilities simultaneously. When it is concatenated with the network and host addresses, this forms a socket. The pair of sockets uniquely identifies each connection.

Thus, a socket is simultaneously used for multiple connections.

- **Connections-** A Connection is combination of sockets, sequence numbers, and window sizes. Every connection is uniquely specified by a pair of sockets identifying its both sides. Whenever the two processes wants to communicate, their TCP's has to first establish a connection (initialize the status information on both sides). Once the communication is complete, the connection has to be terminated or closed.

3.2.2 TCP segment

When comparing a TCP segment format with the UDP datagram, we can observe the key difference between the two protocols. Since UDP has smaller header frame size, UDP is faster than TCP but it lacks reliability[7]. Fig 3, shows the format of the TCP segment format. A brief description of each field is as follows:

- **Source port address-** The 16-bit source port number, which specifies the sender of the segment[7].
- **Destination port address-** The 16-bit destination port number used by sender to send the segment to the receiver[7].
- **Sequence number-** It is the 32 bits sequence number of the first data octet in the segment (except when SYN is present)[7].
- **Acknowledgment number-** It is of 32 bits, If ACK control bit is set, this field indicates the value of the next sequence number of the segment to be received. This is always required to be sent once a connection is established [10].
- **Reserved-** It is of 6 bits, which is reserved for future use and must be kept zero[10].
- **Control bits-** It is of 6 bits, there are 6 control bits listed below(from left to right)[10]:
  URG: Urgent Pointer field
  ACK: Acknowledgment field
  PSH: Push Function
  RST: Reset the connection
  SYN: Synchronize sequence numbers
  FIN: No more data from sender
- **Window-** Its size is 16 bits. It specifies the number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is expecting to receive[10].
- **Checksum-** It is of 16 bits in size, This field is the 16 bit one's complement of the one's complement sum of all the 16 bit words contained in the header and text[10].
- **Urgent pointer-** Its size is also of 16 bits, this field communicates the current value of the urgent pointer. It indicates the sequence number of the octet following the urgent data[10].
- **Options-** Options are multiple of 8 bits in length and occupy the space at the end of the header. Options are included in the checksum also [10].

<table>
<tr><td colspan="5">Source port address
(16bits)</td><td colspan="4">Destination port
address (16 bits)</td></tr>
<tr><td colspan="9">Sequence number
(32 bits)</td></tr>
<tr><td colspan="9">Acknowledgement number
(32 bits)</td></tr>
<tr><td>Hlen
(4 bits)</td><td>Reserved
(6 bits)</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td>Window size
(16 bits)</td></tr>
<tr><td colspan="9">Options and Padding
F (32 bits)</td></tr>
</table>

Fig 3 : TCP Segment Format[7]

3.2.3 Applications of TCP

The following is a list of applications that make use of TCP [16]:

- **File transfer protocol (FTP)-** Providing a mechanism for transferring data files between systems. All the FTP client-server programs and Web browsers, contain an implementation of the TCP protocol.
- **Hypertext transfer protocol (HTTP)-** Most widely used protocol over the internet to move Web pages across an internet connection.
- **Interactive mail access protocol (IMAP)-** Allows the clients to access e-mail messages and mailboxes over the network.
- **Post office protocol (POP)-** It provide the clients to read and remove e-mails which are residing on a remote server. It is also used in e-mail applications.
- **Remote login (Rlogin)-** Allows remote login capability of the network.
- **Simple mail transfer protocol (SMTP)-** The protocol main function is to deliver email from one system to another.
- **Secure shell (SSH)-** Allows the remote access to computers as well as encryption of the data

3.2.4 TCP in Wireless Networks

The TCP protocol was originally designed for the wired networks made up of links with low bit-error rates and stationary hosts. According to the TCP, a dropped packet is always treated as the loss due to network congestion. Although, this assumption may be true for a wired network, but it may not be always true in the case of a wireless networks. There are several factors that contribute to the dropping of packets in a wireless network[20]. Few of the major factors are high error rates and frequent disconnections in the case of wireless networks. In order to distinguish between the packets dropped due to network congestion or some other reasons in case of TCP, different measures are required to be taken for improving its performance over the wireless networks. Many schemes have been proposed to deal with these performance issues of TCP. Some of them are listed below[20].

- **The split connection approach-** This approach splits a TCP connection into two separate connections, in which one connection is established between the fixed host and the base station, and another connection is established between the base station and the mobile host. The splitting of connections allows the use of specialized protocols which are designed specifically for wireless links between the mobile host and the base station[20].
- **The fast-retransmit approach-** A mobile host when moves from one cell to another cell, delays and packet loss occurs during the movement of mobile hosts, while cellular hand-offs are carried out by the base stations. As the TCP provides a congestion control mechanism, these losses will be treated as a losses due to network congestion. In several cases, a timeout will be necessary to restart the packet flow between the hosts. To wait for these timeouts to occur, it will give rise to long delays during cell switching. The main goal of this approach is to reduce these delays, by allowing the sender to retransmit the segments by artificially triggering the fast retransmit mechanism of TCP. This is done by sending three duplicate ACKs, when the cellular hand-off has been completed[20].
- **Explicit loss notification-** One of the main reasons for the poor performance of TCP over wireless networks is that the TCP cannot distinguish between packets lost due to network congestion or due to several other factors. This mechanism will allow TCP to distinguish between different types of packet loss and appropriately react to these losses based on the available information[20].

4. COMPARISON BETWEEN TCP AND UDP

TCP is a connection-oriented protocol, which provides end-to-end communications. When the connection is established between the sender and receiver, the data may be sent over the connection.UDP is a simple and connectionless protocol. As, it is a connectionless protocol, it does not set up a dedicated end-to-end connection between the sender and receiver before the actual communication takes place. The data is being transferred in one direction from sender to receiver without verifying the state of the receiver[17]. Here, in this paper, the TCP and UDP are compared on the basis of the data transfer features, basic operation and applications.

4.1 Differences in data transfer

TCP provides reliable and ordered delivery of data from user to server and vice versa. UDP is connection less protocol and does not ensure the reliable delivery of data. TCP and UDP are different from each other on the basis of data transfer features as listed below[9]:

- **Reliable delivery-** TCP is more reliable as compared to UDP since it make use of message acknowledgment and retransmissions in case of loss of packets. Thus, there is no missing of data in the network whereas UDP does not ensure that the data has reached to the receiver or not and there is no such concepts of acknowledgment, time out and retransmission in UDP.
- **Ordering of messages-** TCP transmits the segments in a sequence and they are received in the same order at the destination. If in case, the data segments arrives in wrong order, TCP has the ability to reorder them. In case of UDP, message sequence is not maintained during the transmission. It is not concerned with the ordered delivery of data at the receivers end.
- **Connection setup-** TCP requires three packets for a socket connection and handles congestion control and reliability, and thus becomes a heavy-weight protocol. UDP is a lightweight transport layer protocol and supports no prior connection setup or ordering of messages.
- **Transfer features-** TCP reads data as a stream of bytes and message is transmitted in the form of segments. UDP messages are sent in the form of datagrams into the network.

4.2 Differences in basic operation

There is a huge difference between TCP and UDP on the basis of their basic operation. For TCP, the connection is established via three way handshake, in which, there is a process of initiating a connection, acknowledging the connection and termination of connection. Once, the connection is established between the communicating hosts, data is transmitted between them in both directions. After the transmission is over, the connection is terminated. UDP uses a simple transmission model and does not ensure reliable delivery as and when compared to the TCP. And thus, the datagrams may arrive out of order, may be duplicated, or may be lost also due to several factors. Unlike TCP, UDP supports broadcasting and multicasting[9].

4.3 Differences in applications

The common applications that make use of TCP are Web browsing, email and file transfer etc. TCP controls the size of the segment, it sends the data in the form of segments, reads the user data as stream of bytes, controls the rate of data exchange, provides flow control and avoid network congestion. TCP employs a congestion control mechanism, which is called as window based transmission, that allows TCP to control congestion in the network. TCP is suitable where error correction facilities are required which means, where reliable delivery is required . UDP is suitable for time sensitive applications. It is suitable for applications where some loss of information is accepted, but in-time delivery of information is expected. UDP is compatible with both broadcasting (sending data to all on a network) and multicasting (sending data to all subscribers). The applications such as Domain Name System (DNS), Trivial file transfer protocol (TFTP) and online games make use of UDP protocol [9]. The Table 1 provides a summary comparison between TCP and UDP[8].

Table1. Summary comparison between TCP and UDP[8]

Properties	TCP	UDP
Reliability	Yes	No
Ordering of messages	Yes	No
Congestion Control	Yes	No
Method of transfer	Segment	Datagram
Power Consumption	High	Low
Applications	Web-browsing, E-mail,File Tansfer	DNS,Voice Over-IP, Online games

5. CONCLUSION AND FUTURE WORK

In this paper, the transport layer, its various protocols and functions were reviewed. Further, the functions of TCP and UDP were described in detail and compared on the basis of their data transfer features, their basic operation and applications. Additionally, the performance of TCP in wireless networks is briefly reviewed in this paper. In this survey, it is finally concluded that, TCP protocol should be used where reliability is required such as in internet banking and UDP should be used for broadcasting and multicasting purposes like in internet gaming, internet radio etc. where some loss of packets is accepted [12]. This is why, these two protocols TCP and UDP have been considered for the survey, because TCP and UDP are the main protocols of the transport layer and each of them have a certain advantage over each other but in different cases. There are some other cases which can be considered like, UDP performs better and is suitable for applications where time is a constraint irrespective of reliable delivery, whereas TCP is suitable where time is not a constraint and reliable delivery is important. Thus, the purpose of this survey is to study and analyze the basic operation of TCP and UDP in detail, to understand the basic functions of these protocols, to find the key differences between the two protocols and finally to make a summary comparison between them. Future work is intended to optimize the TCP and UDP protocol for understanding its performance for suitable applications in mobile ad-hoc networks.

6. REFERENCES

[1] Mehrnoush Rahmani, Andrea Pettiti, Ernst Biersack, Eckehard Steinbach, Joachim Hillebrand 2008. "A Comparative study of network transport protocols for in-vehicle media streaming", BMW Group Research and Technology, Munich, Germany.

[2] Hafiz Muhammad Omer Chughtai, Shahzad A. Malik, Muhammad Yousaf, "Performance Evaluation of Transport Layer Protocols for Video Traffic over WiMax", COMSATS Institute of Information Technology, Islamabad, Pakistan.

Available: corenet.org.pk/data/paper48.pdf

[3] Manfred Georg 2006. "A Survey of TCP Optimizations for Wireless Channels", mgeorg@arl.wustl.edu.

[4] A. S. Tanenbaum 2003. "Computer Networks", fourth edition Prentice Hall PTR.

[5] Sirisha Medidi, Jin Ding, and Murali Medidi 2002. "Performance of Transport Protocols in Wireless Networks", School of Electrical Engineering and Computer Science, Washington State University, Pullman.

[6] Saima Zafar 2011. "A Survey of Transport Layer Protocols for Wireless Sensor Networks".

[7] Behrouz A. Forouzan 2003. "Data Communications and Networking", 2nd Edition. Mc Graw hill.

[8] http://en.wikipedia.org/wiki/Transport_layer. "Transport Layer", WIKIPEDIA.

[9] http://www.diffen.com/difference/TCP_vs_UDP. "TCP vs UDP-Difference and comparison.

[10] Postel J. 1981. "Transmission control protocol", RFC 793, DARPA internet program, protocol specification.

[11] Postel J. 1980. "User datagram Protocol", 768, USC/Information Sciences Institute.

[12] Kusum Dalal, Prachi Chaudhary, Dr. Pawan Dahiya 2011. "Performance Evaluation of TCP and UDP Protocols in VANET Scenarios using NCTUns-6.0 Simulation Tool", Murthal, Haryana, Volume 36– No.6.

[13] Frank Aune 2004. "Cross-Layer Design Tutorial", Norwegian University of Science and Technology, Dept. of Electronics and Telecommunications, Trondheim, Norway.

[14] J. Mahdavi and S. Floyd 1999. "Tcp-friendly".

[15] Vikas Kawadia 2004. "Protocols and architecture for wireless Adhoc networks", Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

[16] http://www.fujitsu.com//downloads/TEL/fnc/pdfservices/TCPIPTutorial.pdf . TCP/IP protocol suite, tutorial.

[17] http://en.wikipedia.org/wiki/User_Datagram_Protocol, "User datagram protocol", WIKIPEDIA.

[18] Haitao Zheng, Member, Jill Boyce 2001. "An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks".

[19] http://en.wikipedia.org/wiki/Internet_protocol_suite . "Internet protocol suite", WIKIPEDIA.

[20] Sean Kendall Schneyer 1998. "Survey Paper on TCP", University of Karlstad.