

## International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

*IJCSMC, Vol. 3, Issue. 6, June 2014, pg.825 – 829*

### **RESEARCH ARTICLE**

# Proxy Server for Hybrid TCP/IP and UDP

Sangeeta Yadav, Sangeeta Yogi, Dr. Rajkumar Yadav

<sup>1</sup>Student, <sup>2</sup>Associate Professor

<sup>1,2</sup> Department of Computer Science and Engineering, G.I.T.M, Gurgaon, India

<sup>3</sup>Department of Computer Science and Engineering, UIET, MDU, Rohtak

<sup>1</sup> [Sangitayadav91@gmail.com](mailto:Sangitayadav91@gmail.com), <sup>3</sup> [rajyadav76@rediffmail.com](mailto:rajyadav76@rediffmail.com)

---

*Abstract: In this paper we discussed about the hybrid approach of HTTP-TCP/UDP protocols. The HTTP use UDP for the short conversation, connectionless host-to-host communication, minimal overhead, and it is fast. On the other hand HTTP used TCP for sequenced, in-order, error-free packet delivery by using a sliding window flow control protocol with retransmission facility. In this paper a proxy server is used between the client and the server for fast and efficient access of data.*

*Keywords: HTTP, UDP, TCP, IP, Proxy Server*

---

## I. INTRODUCTION

Today the entire internet based application uses HTTP protocol for data transmission over web. According to Wikipedia, "The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web". It is a request response protocol used in the client server computing model. The HTTP is an application layer protocol designed within the framework of IP suit. It presumes Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

The TCP is main protocol in TCP/IP network. In this IP deals with packets and TCP enables the hosts to establish a connection to transfer data. It is a connection-oriented protocol which is responsible for reliable communication between two hosts. Sometime due to network congestion, or traffic load data packets not reach to destination then it detects these problem and request retransmission of lost packets in their proper order.

The UDP is User Datagram Protocol. It provides a connectionless host-to-host communication. It is a unreliable protocol i.e. there is no guaranty that data will delivered at destination host. There is no assurance that data will get in the same order as it was sent as you can receive a packet before another one, even if the second has been sent before the first you just received. You can also receive the same packet twice.

TCP/IP and UDP both are used by IP to transport them from one network to another. TCP and UDP are both transport protocols which are used by protocols such as HTTP, FTP, and SMTP. Where TCP is mainly used for sequenced, in-order, error-free packet delivery by using a sliding window flow control protocol with retransmission facility and UDP is used for connectionless host-to-host communication, minimal overhead, datagram boundaries are respected, can broadcast, and it is fast.

## II. HYBRID APPROACH OF TCP/IP AND UDP

If a hybrid approach is developed so that TCP and UDP both can be used by HTTP at a time then we can overcome from both disadvantage. This can be done by making a proxy server, which can be connected to a browser. This browser is connected to server. The connection between the proxy and the browser is by the HTTP which use TCP and the connection between the browser and the server which have all file that a client seeks, can be by HTTP- TCP protocol or HTTP-TCP/UDP protocol the building up a proxy server improves the performance as the data accessed by one user and after that request by another one, can be used from cache, thus give a very fast access. A proxy server can support hundreds to thousands of users. The other main advantage is, a proxy server can be used to filter requests (by preventing accessing a websites).



## III. PROPOSED WORK

In the hybrid approach, UDP is used for the short connections and TCP is used for the large connections as their transport protocol. The client always first tries to establish connection through UDP as their transport protocol, but if UDP is wrong choice then they use TCP. This fall back is due to reasons:

- If any UDP packets are lost, because the loss is easily handled by TCP. As in UDP data can't be retransmit.
- This scheme works correctly if some host do not implement hybrid scheme the connection seems to them as normal HTTP-TCP connection.

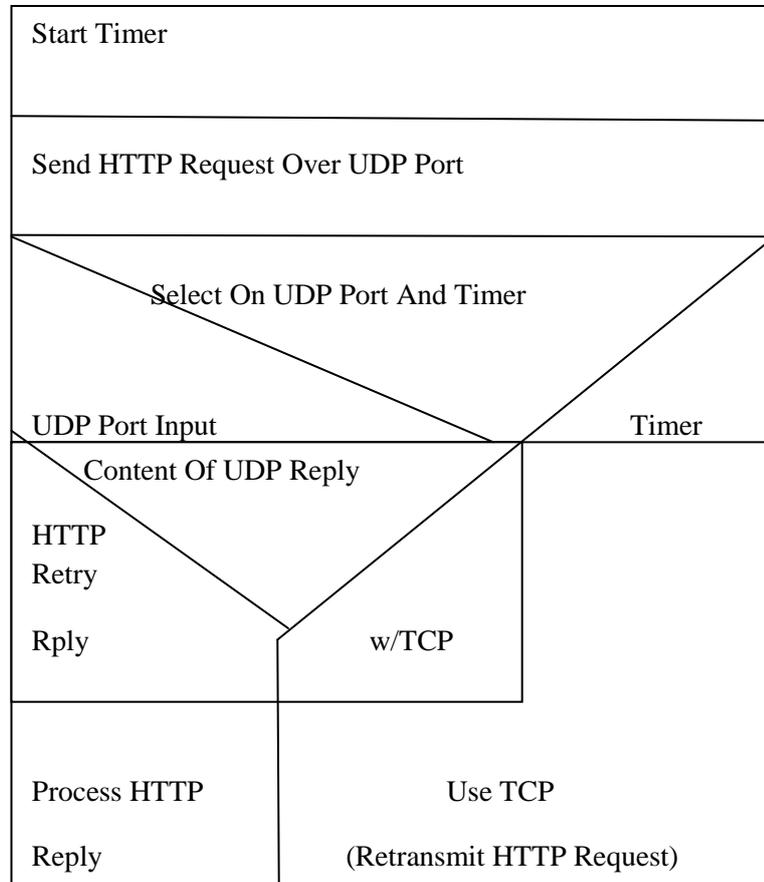


Figure 1: Algorithm execute at HTTP client application to receive HTTP service over either TCP/UDP

The hybrid-capable HTTP clients send the HTTP GET request using UDP as the underlying transport protocol; a timer is start by the client at the time the request is sent. When the client's request is processed by server, it can choose from one of the following:

- a) If the server's response is small enough that it can fit in one packet, then the server should return it by using UDP; for example: small web pages and most cache validation and HTTP REDIRECT responses.
- b) If the server's response is large that, then the server ask client to re-try by using TCP; for example: large web pages. In this case the server can also ask the client to try a different URL.

At the client side.

- The client gets a response from the server: if the reply contains the desired HTTP reply, the client processes the data; if the server asks the client to re-try (a different URL and/or using TCP, as the case may be), the client does so.
- If the sever does not handle HTTP packets sent over UDP, the client may get an ICMP (Internet Control Message Protocol) error message (destination unreachable/protocol unreachable); in this case, the client should re-try using TCP.
- If the timer expires, the client should re-try with TCP.

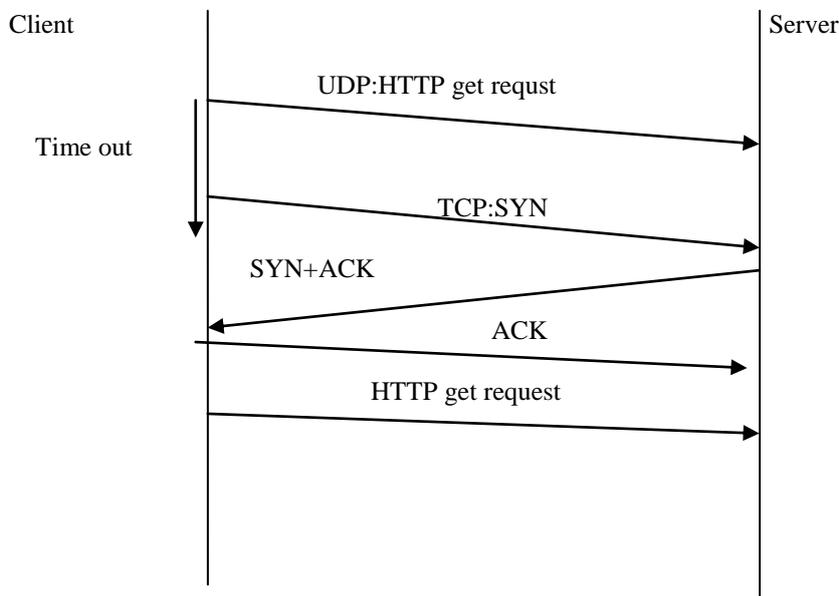


Figure 2: Timeout of UDP and transfer of request to TCP

Figure 2 explains the packet exchange between UDP and TCP when the timer expires. This use *of* a timer provides reliability and backwards compatibility with Servers that do not use the hybrid TCP-UDP scheme; the value of the timer should be chosen to be equal to TCP timer. As explain earlier, this fall back mechanism allows clients to interoperate with servers that do not implement the hybrid scheme.

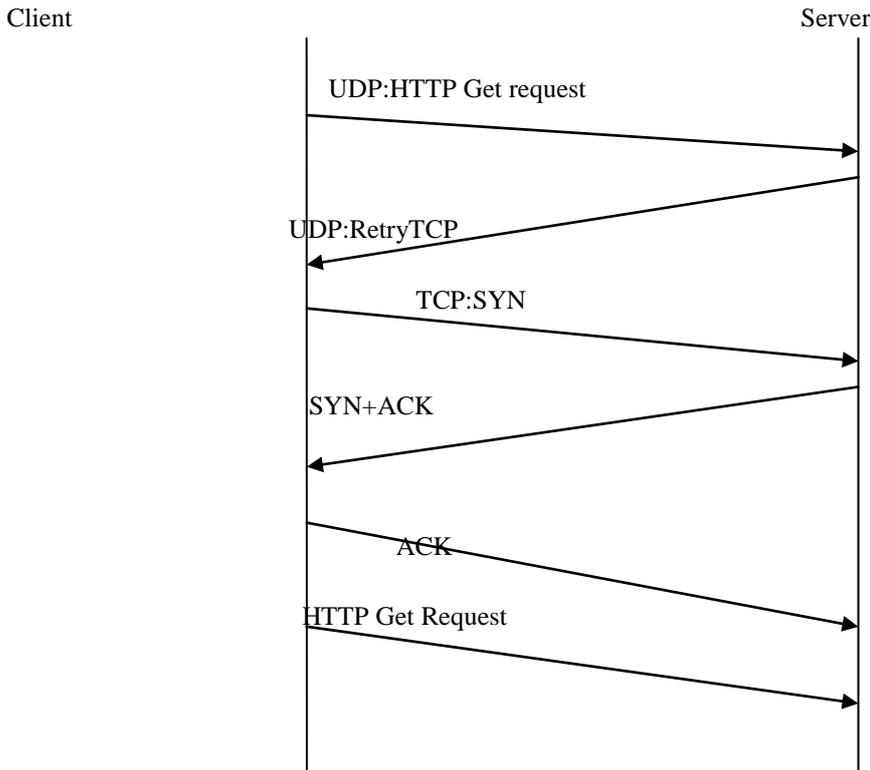


Figure3: Explicit fall back from UDP to TCP.

Figure 3. demonstrates the packet exchange where the server requests the client to resend the HTTP request over TCP.

#### IV. PROPOSED ALGORITHM

The developed proxy server receives a normal HTTP-TCP GET request from a client. The UDP request is prepared which holds the URL, and two port numbers, one for the UDP connection and one for the TCP connection. The proxy server forwards the client's request to the desired server. Then proxy server waits for the reply from the server, by the UDP connection or by the TCP connection. The decision is taken by the target server whether to send a UDP reply or a TCP reply. If the client server is according to the hybrid HTTP-TCP/UDP protocol then it will send the reply the reply can send through the TCP connection or the UDP connection, by comparing the size of the file. If the client server is a normal HTTP-TCP based server then it will not recognize the UDP request packet, and it will not respond to it. So, the proxy server will receive a TIMEOUT event. Hence the proxy server will try to connect to the target server again, only this time it will use a normal HTTP-TCP request. As the proxy server will receive the reply from the target server it will send the reply back to the client by using an HTTP-TCP reply. If a HTTP-TCP POST request is received from a client and then it prepares a TCP POST request and then sends it to server.

#### Algorithm

- 1) Select port address either for TCP or UDP depending upon the message size
- 2) Start timer
- Check timer limit
- If timer limit=expired
- Then
- Msg"UDP contents expired and switch to TCP"
- 3) If range>65535
- Then
- Msg"switc to TCP"
- 4) If range<65535
- Then
- Msg"packet transfer by UDP"
- Else
- Goto TCP

```
5) Check packet loss, duplication of packet
If packet=lost
Then
Msg"no data is received"
6)If packet =duplicate
Then
Check the condition of flow control
If control bit=1
Then
Msg"packet is duplicate"
Otherwise
Msg"packet is original"
//if packet is duplicate then it passes over to the TCP bcoz UDP has no provision of floe control//
7) Check chksum bit
If CHKSUM=1
Then
Goto TCP segment
8) Verify data packet through all these steps and sends data to the destination
9) End()
```

## V. FUTURE WORK

As this hybrid HTTP- UDP/TCP scheme by using proxy server is very effective, but by adding some sort of data base to the proxy server will tell it, that the sever that it is connected to, works according to the new protocol or not. The reason for adding this data base is to avoid the time wasted in the timeout of UDP request. So, we need a cache data base when we connect to a server that works according to the normal HTTP-TCP protocol. The proxy server will no longer have to wait for the UDP TIMEOUT period

## VI. CONCLUSION

In this paper we discuss about the hybrid TCP-UDP transport layer scheme for HTTP. This hybrid scheme offers the low cost by using UDP which gives high reliability and the congestion control features of TCP. It use of UDP benefits from the low overhead when used for short transfers, and for large transfers, TCP is used for reliable delivery and good congestion behaviour. The reason for developing a proxy server and not to implement a browser with the TCP-UDP protocol is a practical one. As we can't force someone to use our browser when there are two dominant Browsers in the market (Explorer & Communicator). Due to the fact, that most browsers are usually connected to a proxy server, so it would be so much easier to insert the new protocol into existing proxy server.

## REFERENCES

- [1] Bob Braden. RFC-1644 T/TCP - TCP Extensions for Transactions. Network Working Group, July 1994.
- [2] CERT. IP Spoofing Attacks and Hijacked Terminal Connections, CA-95:01. Computer Emergency Response Team, Carnegie Mellon University, Pittsburgh, Pennsylvania, January 1995.
- [3] Roy T. Fielding, Jim Gettys, Jeffrey C. Mogul, HenrikFrystyk Nielsen, and Tim Berners-Lee.RFC-2068
- [4] Hypertext Transfer Protocol – HTTP/1.1. Network Working Group, January 1997.
- [5] "HYBRID TCP-UDP Transport for Web Traffic" article by. Geoffrey Raehr, Amit
- [6]Steven D. Gribble. UC Berkeley Home IP HTTP traces. July 1997. Available at <http://www.acm.org/sigcomm/ITA/>
- [7] Je\_rey Mogul and Steve Deering. Path MTU Discovery. Arpanet Working Group Requests for Comment, DDN Network Information Center, SRI International, Menlo Park, CA, November 1990.RFC-1191.
- [8] B. Cli\_ord Neuman. The Virtual System Model: A Scalable Approach to Organizing Large Systems. PhD thesis, University of Seattle, Washington, 1992.
- [9] Venkata N. Padmanabhan. Private communication, 1997.
- [10] K. Poduri and K. Nichols. Simulation studies of increased initial tcp window size, February 1999. Internet Draft expires 8/98.
- [11] Kedarnath Poduri and Kathleen Nichols. RFC-2415 Simulation studies of increased initial TCPwindow size. Network Working Group, September 1998.