Evaluation of UDP-based Reliable Transport Protocols in Wireless Network

Madzirin Masirap
Faculty of Engineering, UiTM, Malaysia
madzirin@gmail.com

Abstract—**This paper is mainly interested in near real-time wireless application with regards to Internet of Things application. Wireless network speed had been improved significantly since its introduction. We had seen its wide implementations in various sectors. This however doesn't dismiss the fact that wireless network is unreliable. Its unreliability comes from various sources such as wireless signal collision and hardware issue. The idea of wireless networking is interesting mainly because it introduce portability. In near real-time wireless application, the traditional TCP protocol is not suitable because of its packet header size and its conservative Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm and slow start strategy. This paper mainly evaluates three kinds of typical UDP-based high speed transport protocols (RUBDP, UDT, and PA-UDP). The evaluation was done on a test bed that consist of two machine connected by wireless ad-hoc network. This paper focused on the effect of data size by monitoring throughput and the CPU behaviour. PA-UDP protocol has the best transport performance but it require high CPU utilization during the link negotiation. UDT protocol is the most convenient, is CPU friendly but it has less throughput. In our scope, UDT protocol is most likely suitable because it has consistent performance and easy to implement. RBUDP sender shows high CPU usage whereas the receiver is not strained.**

Keywords- UDP-based transport protocol; wireless ad-hoc network; performance evaluation

I. INTRODUCTION

Network technology had been improving at rapid growth. In engineering sector, the idea of interconnecting various equipment and make them talk to each other open endless possibilities which why the special term Internet of Things was introduced to define this idea. In internet of Things, most applications would require reliable and fast communication. Luckily, at some location we are able to achieve this by using high-speed wired network along with any reliable transport protocol. However this is not viable when portability is a must. Wireless network is the most obvious answer for portability but we already know how unreliable it is. In order to be able to use wireless network for this kind of applications, there are several concern need to be addressed. Wireless network speed is much slower than wired network so a high-speed communication protocols is needed such as UDP but, UDP is not reliable. TCP on the other hand has its AIMD which would perform worse in wireless network. This is where the reliable UDP-based transport protocol came into light. Apart from UDP-based transport protocol, there are also TCP-based implementations which usually strip away the excess baggage in the TCP header. There are other novel prototypes but UDP-based or TCP-based transport protocol is the easiest to implement.

The objective of this paper is not developing another novel transport prototype but is mainly to evaluate existing UDP-based reliable transport protocol for use in near real-time application in wireless network. There are several notable prototypes and some of them are already being used in real high-speed network computing. This paper had chosen three famous protocols for comparison. RBUDP, PA-UDP and UDT.

II. RELATED WORK

There are several evaluation work done before. The most dedicated work that evaluates UDP-based protocol is the work done by Zhaojuan Yue et. al [5]. Their work focused on high-bandwidth delay application and they have selected RBUDP, PA-UDP, UDT and Tsunami UDT as their subject of evaluation. Their finding was PA-UDP performed the best followed by RBUDP. UDT came in last but they had given credit to if for being user friendly. Another notable work [7] is investigating the effect of dynamic buffer in wireless network.

The work also evaluates other related prototypes. The result shows that reliable dynamic buffer show significant improvement over the typical reliable UDP-base protocol.

III. PROTOCOLS

A. RBUDP:

The main transport protocol used by RBUDP [2] is UDP but it also uses TCP for signalling. To use the RBUDP, user need to specify the transmission speed at the sender side. The sender then transmits all the data packages according the specified transmission speed. At the RBUDP receiver, the received data will be tallied to determine which data packages have been received. At the end of transmission, the sender send DONE signal using the TCP protocol. When the receiver receives the DONE signal, it will send an ACK packet along with the tally. If there are lost packets, the receiver will retransmit the lost packets. This process is repeated until the receiver receives all the data successfully. The user set initial speed is crucial. Users are usually able to get the optimum initial speed by using Iperf.

One thing to note about RBUDP is that it is not in its design to consider fairness in network. This makes RBUDP behaviour in public network unpredictable.

B. PA-UDP:

PA-UDP [3] is implemented on top of UDP protocol. The sender initiates the transfer using three-way handshake. The three-way handshake is sent using TCP protocol. The transfer rate will be initiated during the three-way handshake. This receiver will keep updating the sender with newer transfer rate based on the packet losses, receiving rate, disk processing rate and buffer size. The sender than adjust the inter-packet delay to adjust the sending rate.

PA-UDP implement complex disk processing algorithm which very useful when the disk writing speed becomes the bottleneck. In our test, the highest network speed is around 12Mbit/sec which is much lesser than the average consumer disk writing speed.

B. UDT:

UDT is UDP-based protocols but it implements the congestion control found in TCP protocols. UDT [4] integrates rate based congestion control and window-based flow control mechanisms. Rate control periodically updates packet-sending period, and flow control updates flow window size based on acknowledgement for the received data.

The congestion control includes slow start and congestion avoidance phases. In congestion avoidance phase, UDT adopts DAIMD (AIMD with decreasing increases) congestion control scheme: at the beginning of sending, bandwidth is relatively sufficient and the probability of occurring congestion is low; therefore, the transmission rate increases faster; when sending rate approaches the available bandwidth, the sender decreases the additive sending rate to avoid the congestion. DAIMD congestion control algorithm fully considers the effectiveness, fairness and stability.

III. PROTOCOL EVALUATION

A. Methodology

There several methods to evaluate the performance of transport protocols. The first one is theory model method that models and analyses protocols. The second one is network model and simulation method which usually use NS2 for the simulation. The third one is based on real network transmission test. The fourth is the method to build test bed. This paper evaluates the candidate transport protocols by building a test bed. Table 1 shows the test bed components.

The test bed used in this experiment is constituted by two terminal servers with quad core processors and 8GB of RAM. Both servers are running with the latest Ubuntu i.e. Ubuntu 14.04 LTS. Both servers connected through ad-hoc network. These servers separated at consistent distance during the evaluation of the transport protocols. In this test, we used the PA-UDP prototype, RBUDPv1.0 and UDT 4.6. In our experiments, we use memory-to-memory mode transfer and calculate the mean value of several test result to represent the final result. We monitored the throughput for all three protocols.

Server	CPU	Memory	OS
A	Intel quad core	8GB	Ubuntu 14.04
B	Intel quad core	8GB	Ubuntu 14.04

Table 1. Test bed components

B. Throughput Measurements

For near real-time communication, data may vary from several kilo bytes to several mega bytes. For this measurement, we had varied the size from 1kB to 100MB. In our case, we had considered the network speed which is around 12Mbit/s. We had confirmed this by using Iperf. Figure 1 shows the throughput in Mbit/s. PA-UDP able to reach the maximum transmission speed but due to some packet loss in wireless network, the transmission speed drop when sending file between 100KB-10MB. The RBUDP is able to maintain the transmission speed even though we noticed some packet loss occurrence when transmitting data above 10KB. Our test using the UDT protocol shows that it can only reach half of the network speed. However, UDT sending rate is very stable, it able to maintain the transmission speed during the test especially when sending data bigger than 100KB.

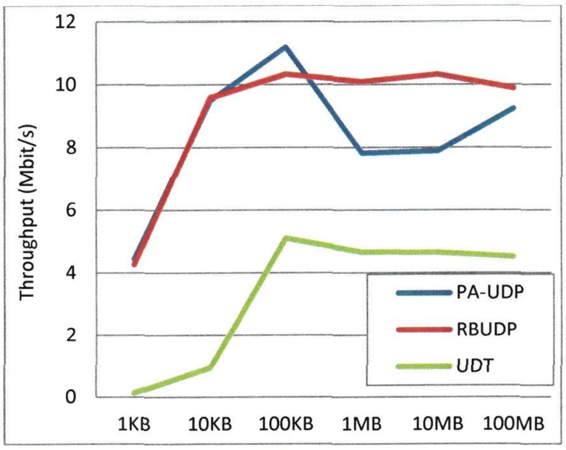

Figure 1. The effect of data size to the throughput

C. CPU Usage

During the experiment, we had monitored the CPU and network usage for each protocol. For convenience, the CPU usage data shown here is collected while transmitting 1MB of data. We use the pre-installed System Monitor on Ubuntu for this purpose. Figure 2 shows the CPU usage at receiver side that implements PA-UDP protocol. We can see the CPU usage spike during the transmission start and at the end of transmission. Figure 3 shows the CPU usage at the sender side that implements PA-UDP protocol.

Figure 4 shows the CPU usage at receiver side that implements RBUDP protocol. Figure 5 shows the CPU usage at sender side that implements RBUDP protocol. When using RBUDP, we could see that the receiver ends CPU usage is low but the sender side is using too much CPU resources. It keeps the CPU usage to the maximum throughout the transmission process.

Figure 6 shows the CPU usage at sender side that implements UDT protocol. Figure 7 shows the CPU usage at receiver side that implements UDT protocol. The sender and receiver end that implements UDT show no excessive CPU usage.

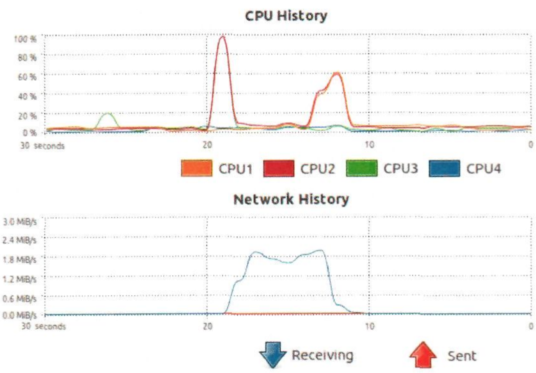

Figure 2. CPU and network usage of the receiver using PA-UDP protocol

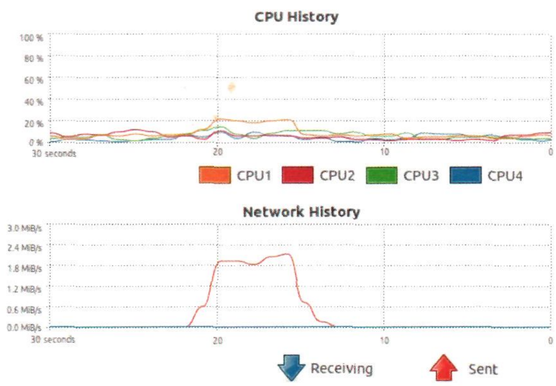

Figure 3. CPU and network usage of the sender using PA-UDP protocol

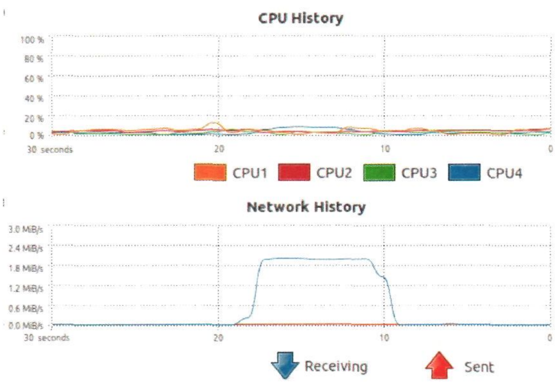

Figure 4. CPU and network usage of the receiver using RBUDP protocol

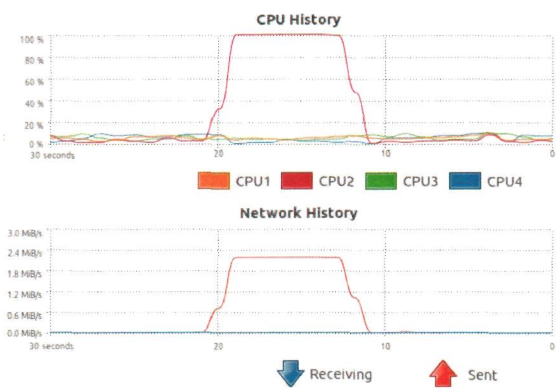

Figure 5. CPU and network usage of the sender using RBUDP protocol

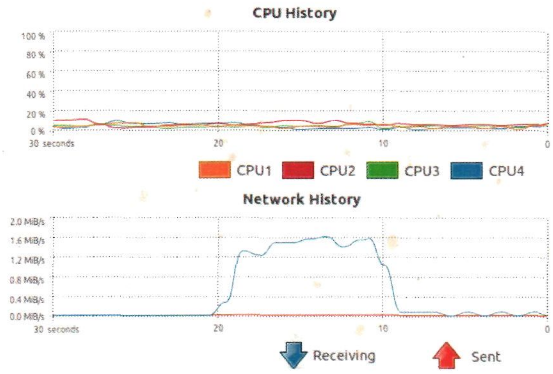

Figure 6. CPU and network usage of the receiver using UDT protocol

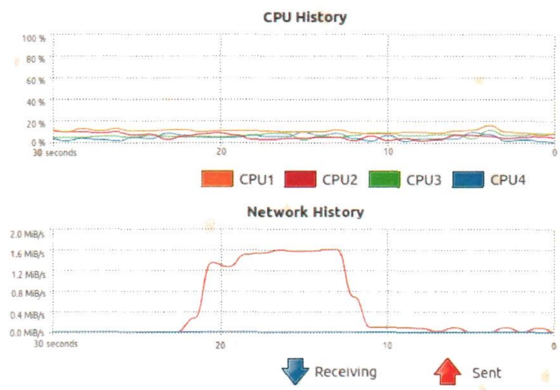

Figure 7. CPU and network usage of the sender using UDT protocol

IV. DISCUSSION

We had seen how PA-UDP out-performs other UDP-based reliable protocols in one of the previous work [5]. Our experiments on the same protocols also show the same result. In terms of maximum throughput, PA-UDP definitely out-performs the other two protocols. However we observed that PA-UDP would perform below its capability when it encounters transmission error or packet loss. RBUDP is able to perform its best even though it encounters packet losses. As we had expected UDT come out with the lowest throughput. This is because of the DAIMD in UDT which adopted the AIMD concept in TCP.

Based on the CPU snapshot, the UDT come out as the most resource friendly. It consume lesser CPU usage which translate to power saving. RBUDP is the most resource intensive protocol.

Based on the measurements result, if power usage is tolerable, PA-UDP can be considered as a candidate. Although RBUDP's speed is very stable, it comes with several minus feature for example the network fairness. The CPU usage for RBUDP also is very high. Although UDT protocol came out last in throughput showdown, UDT protocol offers the most interesting package. UDT protocol is matured and slowly gaining popularity because its ease of use. UDT also is firewall friendly and works well with other transport protocol. UDT is suitable for low-powered Ethernet devices because it low processing footprint. UDT protocol could easily make on the top of the list of the candidates for Ethernet real-time application.

Due to limited resources and time, this paper omitted one crucial part that is performing these protocols evaluation on low-powered device. The evaluation on low-powered device will provide better conclusion on which protocols will perform best under limited resources.

REFERENCES

[1] Bova, T. & Krivoruchka, T. "Reliable UDP Protocol," IETF Internet-Draft, 25 February 1999.

[2] E. He, J. Leigh, O. Yu, and T. DeFanti. Reliable blast UDP: Predictable high performance bulk data transfer. In Proceedings of IEEE Cluster Computing, pages 317-324, September 2002.

[3] Eckart, B., Xubin He, Qishi Wu, "Performance Adaptive UDP for High-Speed Bulk Data Transfer over Dedicated Links", Parallel and Distributed Processing (IPDPS), 2008.

[4] Y. H. Gu and R. L. Grossman, "UDT: UDP-based Data Transfer for High-Speed Wide Area Networks," Computer Networks, vol. 51, pp. 1777-1799, 2007.

[5] Zhaojuan Yue, Yongmao Ren, Jun Li, "Performance Evaluation of UDP-based High-speed Transport Protocols ", Software Engineering and Service Science (ICSESS), 2011.