Voice over TCP and UDP

Xiaotang Zhang and Henning Schulzrinne
Department of Computer Science
Columbia University
xz2104@columbia.edu hgs@columbia.edu

September 28, 2004

We compare UDP and TCP when transmitting voice data using PlanetLab where we can do experiments globally. For TCP, we also do experiments using TCP NODELAY which sends out requests immediately. We compare the performance of different protocols by their 90^{th} percentile delay and jitter. The performance of UDP is better than that of TCP NODELAY and the performance TCP NODELAY is better than that of TCP. We also explore the relation between TCP delay time minus the transmission time and the packet loss rate and find there is a linear relationship between them.

1 Introduction

We perform tests to compare transmitting voice using TCP and UDP. The test environment is PlanetLab. We compare the delay and jitter incurred by using UDP, TCP and TCP with NODELAY. According to our test results, the performance of both delay and jitter incurred by UDP is better than that incurred by TCP and TCP with NODELAY. We also compare the relation of the delay difference between delay time of TCP with NODELAY minus transmission time and the packet loss rate. The tests result shows that there is a relationship between them: the bigger the packet loss rate is, the bigger the delay difference between these two protocols.

2 Test environment

The test environment we used for this project is PlanetLab. PlanetLab is an open, globally distributed platform for developing, deploying and accessing planetary-scale network services. It is very convenient to conduct experiments at Internet scale using PlanetLab. It experiences all of the behaviors of the real Internet where the only thing predictable is unpredictability (latency, bandwidth, paths taken). A second advantage is that PlanetLab provides a diverse perspective on the Internet in terms of connection properties, network presence, and geographical location. [1]

For this experiment, in order to test the performance globally, we chose both international nodes and nodes within the US:

- Three PlanetLab international nodes:

  CAM: Cambridge University, UK

  HK: The Chinese University of Hong Kong, China

  AU: University of Technology, Sydney, Australia

- Four PlanetLab nodes within the United States:

  UV: University of Virginia

  UW: University of Washington

  CMU: Carnegie Mellon University

  UCSB: University of California at Santa Barbara

- One node outside PlanetLab:

  Home: from the first author's home, Elizabeth, NJ, using DSL

All of the above eight nodes communicate with the same node, a PlanetLab node at Columbia University.

2.1 The problem of PlanetLab and the solution

Though using PlanetLab is very convenient to do global experiments and experience the real Internet environment, it does have a fatal problem for this experiment, namely the poor performance of NTP synchronization.

Many of the PlanetLab nodes have very poor NTP synchronization. For example, the offset of Sydney varies from 200 ms to five seconds and the jitter varies from 100 ms to more than one second. If we use the local time of Sydney node, we will get a ridiculous result: the average delay from Sydney to Columbia is three seconds and the average delay from Columbia to Sydney is negative. However, if we try to ping Sydney, the round trip ping time is about 200 ms.

After observation, we found the following nodes have relatively stable and good NTP performance: CAM, HOME, UV, UW and CU.

Other nodes have poor NTP performance. They have big offset and jitter and the worse is that the offset and jitter keep on changing rapidly. The time for those nodes cannot be trusted at all.

In order to solve the NTP synchronization problem, we can choose a node which has good NTP performance as both data source and destination and calculate the round trip delay time and jitter.

3 Test plans

The goal of this experiment is to test and compare the performance of transmitting voice data using UDP, TCP and TCP NODELAY. TCP NODELAY is TCP but it sends out all requests immediately without waiting for the buffer to fill.

As we know, due to the poor NTP performance of some PlanetLab nodes, we cannot get one way delay time and jitter between all the node pairs. In order to overcome the NTP problem, we need a node, which has stable and good NTP performance. The node is planetlab2.comet.columbia.edu. We call it CU.

The following is the test plan for all eight node pairs.

The sender always sits on the Columbia University node. It sends packets out and timestamps them. The receiver sits on other nodes. It receives packets from the sender and then sends them back. The sender receives returned packets which are sent back by the receiver. It checks the packets' timestamp and calculate the round trip delay and jitter.

The following is the one way test plan for those node pairs which have good NTP synchronization.

The sender sits on a node other than CU, e.g., CAM, HOME, UV or UW. It sends packets out and timestamps them. The receiver sits on CU. It receives packets from the sender and checks the packets' timestamp to calculate the one way delay and jitter.

So, for all eight node pairs, we will do three experiments, UDP, TCP and TCP NODELAY. However for those four node pairs which have trusted NTP performance, we will do two more experiments, UDP and TCP NODELAY to measure the one-way performance.

In order to find out the stability of these node pairs and their performance in a certain time of period, we will do above tests five times on different days.

Since the goal of our experiment is to compare the performance of different protocols when transmitting voice data, we chose to send 160 bytes data per 20 milliseconds. The bit rate is 160*50*8 = 64 kb / s which is enough to ensure good voice quality. The internet is unpredictable. Even the most stable network may vary greatly if you just send several packets. In order to get the real situation of the network, we will send tens of thousands of packets for one run.

4 Test result and analysis

We analyze our data according to the following criteria:

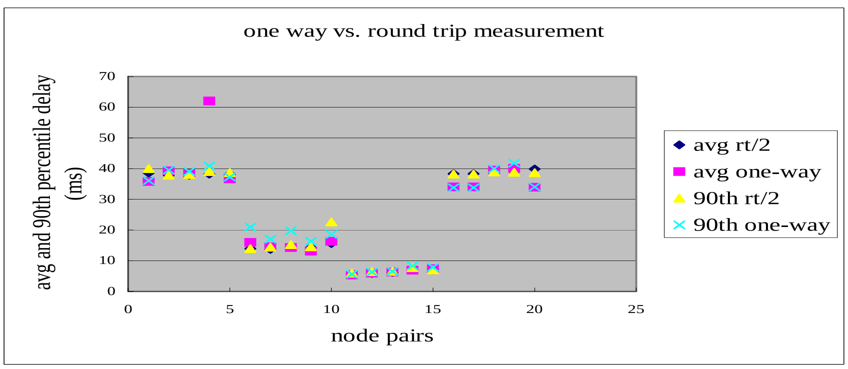
- The round trip measurement compared with the one way measurement (Section 4.1);
- The performance of all the node pairs (Section 4.2);
- The performance comparing TCP and UDP (Section 4.3);
- The relation of delay difference and packet loss rate (Section 4.4).

4.1 The round trip measurement compared with the one way measurement

First, we'd like to find out whether our round trip measurement is correct or not. This is the base of the whole set of experiments.

Following is the data for those four pairs nodes with relatively stable NTP performance. The data is average round trip delay divided by two, one-way delay, round trip 90^{th} percentile delay divided by two and one way 90^{th} percentile delay. The first column is the location of the node and the time when we did the experiments. For example, HOME2 means the result of node HOME on the second day. The protocol used is UDP.

UDP	Average value (ms)		90th percentile value (ms)	
	rt / 2	one way	rt / 2	one way
CAM1	38.6	35.8	40.1	36.1
CAM2	37.8	39.3	37.9	39.5
CAM3	37.9	38.5	38.1	39.0
CAM4	38.2	62.0	39.1	40.8
CAM5	38.3	36.7	38.8	37.3
HOME1	14.0	15.9	13.9	21.0
HOME2	13.8	14.5	14.4	17.0
HOME3	14.6	14.3	15.2	19.8
HOME4	14.3	13.1	14.5	16.3
HOME5	15.7	16.3	22.6	18.6
UV1	5.9	5.3	6.0	5.6
UV2	5.8	5.9	6.5	6.3
UV3	6.2	6.3	6.6	6.5
UV4	7.1	6.9	7.8	8.3
UV5	7.0	7.4	7.0	7.8
UW1	38.4	34.0	38.3	33.8
UW2	38.4	34.0	38.3	33.8
UW3	39.2	39.5	38.9	40.0
UW4	39.1	40.1	38.8	41.8
UW5	39.9	33.9	38.7	34.1



From both the chart and the data table, we can see that the results from these two measurements are almost the same. This proves that using the round trip measurement is as accurate as one-way measurement.

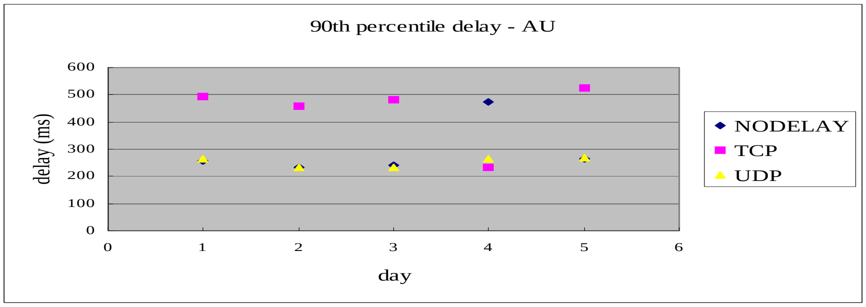
4.2 The performance of all eight pairs of nodes

For VoIP, we are mostly concerned with the percentile value of delay and jitter, so we record all the 90^{th} percentile delay and jitter for all pairs of nodes and draw charts for the 90^{th}

percentile delay on every node pair. Following are the data tables and charts for all eight pairs of nodes:

4.2.1 AU (Sydney)

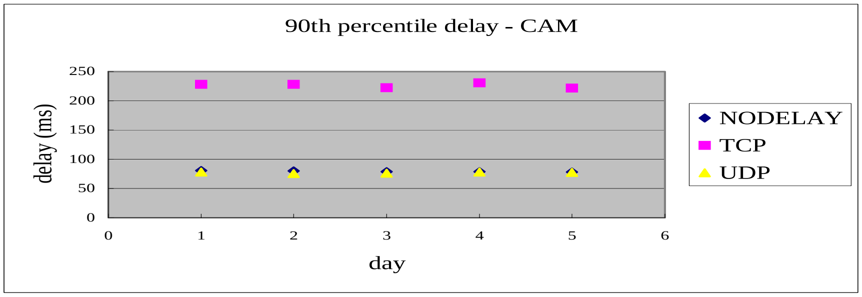
	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	254.8	493.0	266.0	10.2	91.8	13.0
2	233.0	458.7	233.8	8.7	91.1	14.2
3	240.0	482.3	232.9	14.9	94.0	3.8
4	472.7	234.2	262.7	14.3	96.5	4.4
5	262.7	524.1	269.1	12.1	103.4	14.9



We can see that the delay of UDP and TCP NODELAY are very close. Both of them are much lower than that of TCP. The delay is around half of TCP's, and the jitter is only 15% of TCP's. However, on the fourth day, there is an exception. The delay of TCP NODELAY is much higher than that of TCP. However, the jitter is normal. This may be due to the heavy congestion of the network when we did the TCP NODELAY test. This shows that the network between Sydney and Columiba is not stable.

4.2.2 CAM (Cambridge)

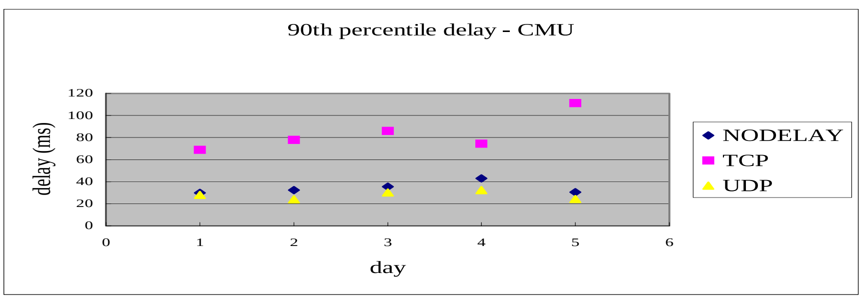
	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	81.0	228.0	78.1	8.0	38.4	4.5
2	80.1	227.9	75.8	3.8	36.3	2.3
3	78.7	222.4	76.1	3.4	36.4	3.0
4	78.9	231.0	78.2	3.8	45.8	3.7
5	77.9	221.7	77.5	4.8	28.7	5.5



The network from Cambridge to Columbia is very stable. In all five days of experiments, both delay and jitter are very stable without obvious variation. The results for UDP and TCP NODELAY are very close. The difference is less than 5%, on the fifth day, they were even almost the same. However, both delay and jitter of TCP are more than three times those of UDP and TCP NODELAY.

4.2.3 CMU

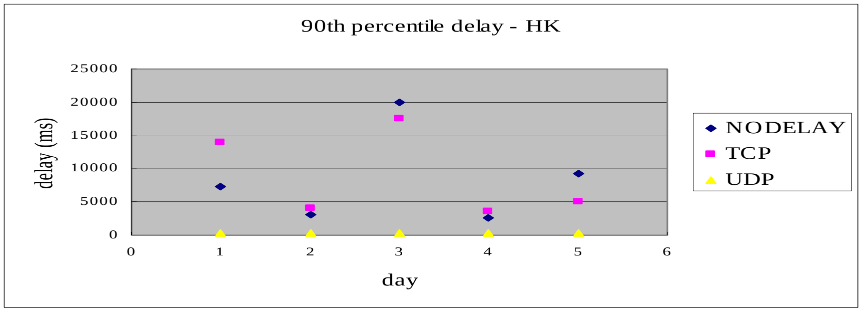
	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	30.0	69.0	28.0	8.4	28.3	9.8
2	32.5	77.9	24.1	20.8	34.5	8.1
3	35.6	86.0	30.3	18.6	64.4	13.6
4	43.1	74.5	32.5	31.5	31.9	9.5
5	30.6	111.3	24.3	15.6	53.1	6.8



The delay for of CMU is worse than that for CAM, but better than that for Sydney. There is no exception for these three protocols. UDP and TCP NODELAY are very close. The difference is about 10% to 20% for delay. TCP has a much worse performance. Both delay and jitter are more than twice those of the other two protocols.

4.2.4 HK (Hong Kong)

	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	7329.7	13952.9	266.3	4035.6	7438.4	8.5
2	3132.1	4077.4	263.5	2096.2	2253.1	5.9
3	19891.5	17517.9	262.8	9465.7	2914.7	7.9
4	2661.5	3640.9	274.3	1784.0	1975.1	7.0
5	9281.6	4989.7	267.0	5039.2	2466.5	11.9

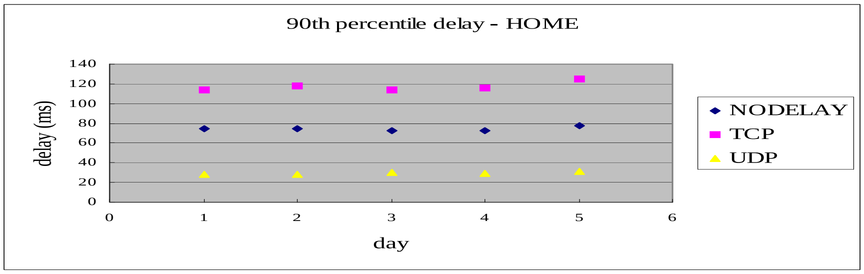


Hong Kong's performance for both TCP and TCP NODELAY are unacceptable for VoIP. The delay of TCP varies from 3.6 seconds to more than 17 seconds and jitter varies from two seconds to 7.4 seconds. TCP NODELAY doesn't help under such situation. The delay of TCP NODELAY varies from 2.7 seconds to almost 20 seconds and jitter varies from 1.7 seconds to nine seconds. This shows that when network has heavy congestion, using TCP NODELAY won't help to improve the performance because using NODELAY will increase the traffic of the network and compensate the benefit brought by sending out requests immediately.

On the contrary, the performance of UDP is very stable. There is no big variation and the delay and jitter are adequate for transmitting voice.

4.2.5 HOME

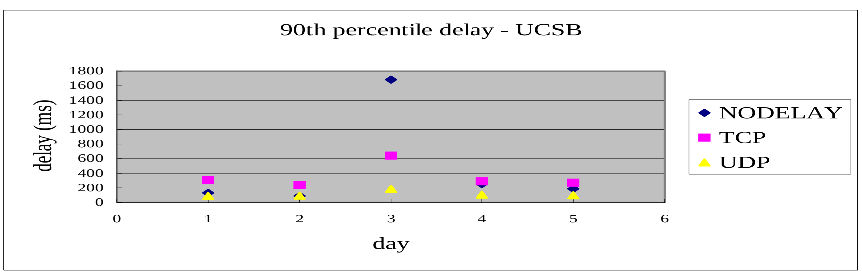
	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	74.5	114.3	27.8	16.4	19.7	3.8
2	74.8	117.7	28.7	19.2	34.1	4.2
3	72.8	114.3	30.4	18.5	26.2	4.4
4	72.9	115.9	29.0	18.3	30.4	4.6
5	78.0	125.2	31.0	22.2	33.7	10.5



The situation of HOME is similar to that of CAM except that the performance of TCP NODELAY is worse than that of UDP. Both delay and jitter of TCP NODELAY are almost triple those of UDP. This is due to the fact that the network between HOME and CU uses DSL which has a much smaller upload bandwidth compared with the download bandwidth. A sender using TCP sends out more packet than that using UDP because it needs to send acknowledgments other than data packets. The smaller upload bandwidth will cause more delay. However, the delay for all those protocols are stable across the five days of measurements.

4.2.6 UCSB

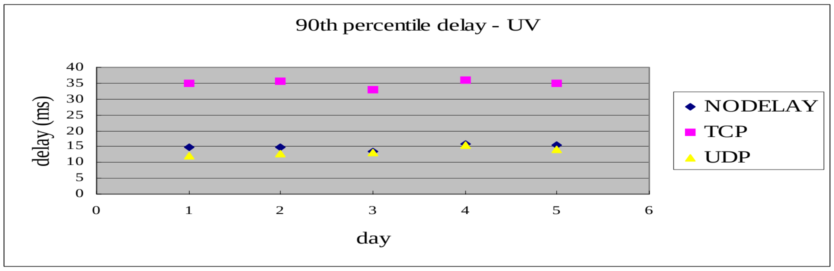
	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	128.4	307.3	89.9	31.2	89.0	6.5
2	88.4	238.1	91.4	16.4	42.3	17.6
3	1683.1	642.6	183.8	440.0	337.9	533.7
4	250.0	287.0	107.2	111.7	78.3	24.3
5	185.7	269.9	96.8	107	121.1	40.8



The performance of UCSB varies across measurements. Though most of the time, the performance of UDP is better than that of TCP NODELAY, and the performance of TCP NODELAY is better than that of TCP, there is an exception. On the third day, the delay of TCP NODELAY is abnormally high. It is 1.7 seconds, almost ten times of other days' delay.

4.2.7 UV

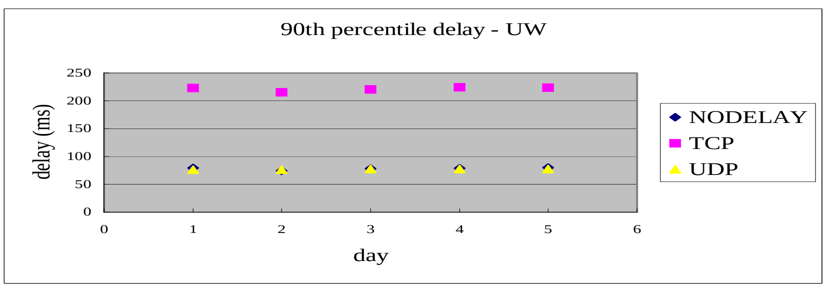
	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	14.7	34.9	12.0	5.9	15.7	3.5
2	14.9	35.6	12.9	4.9	17.6	3.0
3	13.6	33.0	13.2	4.6	18.3	4.6
4	15.9	35.8	15.5	8.3	18.9	5.8
5	15.4	34.8	14.0	4.3	19.9	6.4



The performance of UV is just like that of CAM. All three protocols are very stable. The performance of UDP and TCP NODELAY are very close. The difference is less than 15%. The delay of TCP is about two to three times that of the other two protocols.

4.2.8 UW

	90th percentile delay (ms)			90th percentile jitter (ms)		
day	NODELAY	TCP	UDP	NODELAY	TCP	UDP
1	79.2	223.1	76.5	8.1	36.7	4.9
2	74.5	215.1	76.5	2.5	36.0	4.9
3	78.2	220.7	77.8	8.2	41.2	6.2
4	78.8	224.4	77.6	7.6	44.2	9.8
5	79.9	223.9	77.4	7.4	40.9	10.5



The performance of UW is also very stable and just like that of UV and CAM.

4.2.9 Summary

From the above, we find that the performances of CAM, UV and UW are very stable. The 90^{th} percentile delay using UDP is 10% to 20% lower than that using TCP NODELAY. The delay using TCP is much higher. It is more than twice of that using UDP and TCP NODELAY.

For HOME, though the performances of three protocols are very stable, the 90^{th} percentile delay using UDP is less than half of TCP NODELAY, around 40%. This is because that the node pair between HOME and CU uses DSL whose upload bandwidth is much smaller than the download bandwidth. TCP needs to send acknowledgments besides data packets. So the traffic of using TCP is almost doubled than that of using UDP, thus cause a much longer delay.

For the rest of the four pairs of nodes, we can see that their performances are very unstable. They change rapidly and greatly. Especially for HK, the delay time of using TCP not only varies greatly, but far from tolerable. Its delay varies from two seconds to almost 20 seconds. The acceptable 95^{th} percentile delay time for VoIP is at most 500 ms. So, the result of HK is not acceptable for VoIP. In the following analysis, we exclude this pair of nodes.

For AU, UCSB and CMU, though their performance is not stable, we still can see that generally the performance of UDP is better than that of TCP NODELAY, and the performance of TCP NODELAY is better than that of TCP. We will compare the performance of these protocols later.

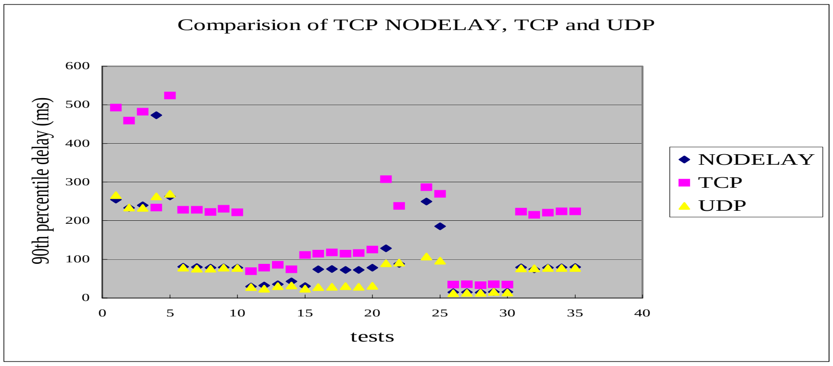
Besides the 90^{th} percentile values, we also got the 95^{th} percentile delay and jitter values. The results from these values are very similar to the result from 90^{th} percentile values. The difference is that the result from 95^{th} percentile values is less stable than that from 90^{th} percentile values. We still can draw the same conclusion. However the difference between the performance of TCP NODELAY, UDP and that of TCP is more obvious and for those unstable node pairs, their performance changes much more greatly than the result from the 90^{th} percentile values.

4.3 The performance of TCP NODELAY, TCP and UDP

After analyzing all the eight pairs of nodes, we excluded the HK node pair because of its intolerable delay time using TCP. Now we are ready to compare the performance of TCP NODELAY, TCP and UDP.

For the rest of seven pair of nodes, we have 35 experiments. Most of the result are good for VoIP application. However, for UCSB on the third day, the delay of TCP NODELAY is 1.68 s which is intolerable for VoIP. So we excluded this result also.

Following is the chart of the delay of TCP NODELAY, TCP and UDP for the remaining 34 experiments:



From this chart, we can see that the performance of TCP is the worst. TCP NODELAY and UDP are very close, especially on those stable node pairs. However, on those unstable node pairs, the performance of UDP is much better than that of TCP NODELAY.

The reason that UDP performs better than TCP is because when there is congestion, TCP will resend packets till all the previous ones have been received. This will increase the delay greatly especially for those heavily congested nodes, such as HK, which was excluded because the delay time is intolerable. However, when using UDP, the sender sends packets out no matter they are received or not. So the delay time of using UDP is very stable. This is most obvious for the HK node. The 90^{th} percentile of TCP varies from two seconds to almost 20 seconds, however, the values for UDP are less than 300 ms and very stable.

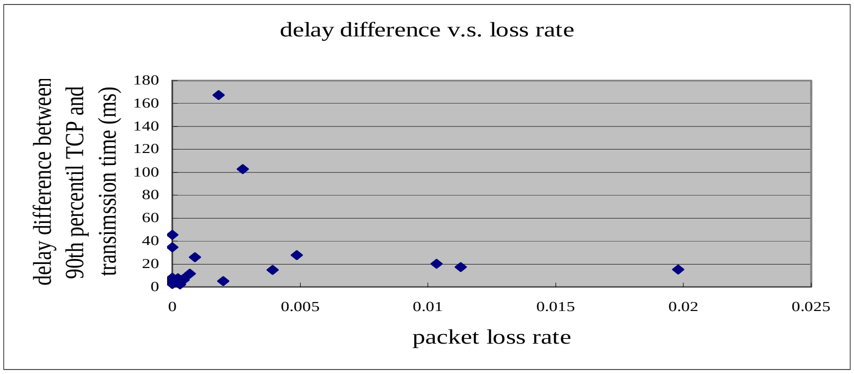
For TCP and TCP NODELAY, we can see clearly that TCP NODELAY performs much better than TCP. This is because for TCP NODELAY, it sends acknowledges back immediately after the receiver receives a packet. So the sender can save time on waiting for acknowledgments and send more packets out earlier than that using TCP.

Though UDP drops packets, the maximum packet loss rate we observed across all experiments was 3%, which is tolerable for sending voice data. When sending voice data, it is more important to reduce the delay and jitter than to reduce the packet loss rate. So UDP is more suitable for sending voice data especially for unstable network. If we need to insure zero packet loss, we should choose TCP NODELAY.

4.4 The relation of delay difference and packet loss rate

After comparing the performance of TCP and UDP, we know that UDP performs better than TCP. The worse the network is, the more UDP performs better than TCP. As we analyzed above, the reason is that TCP has to resend packets when the previous packets are lost or are not acknowledged in time. What is the relation between the packet loss rate and the performance difference of TCP and UDP? Do they have linear relationship?

Based on the data we get from the above 34 experiments, the chart below summarizes the packet loss rate and 90^{th} percentile delay minus the minimum UDP transmission time.

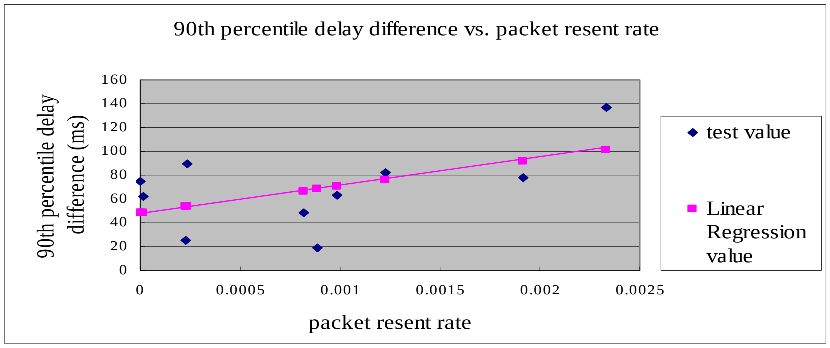


From this chart, we can see that when packet loss rate is small, the delay difference is also small. However, there are some exceptions and there is no obvious linear relationship between the delay difference and loss rate. This might be because of two reasons. First, the loss rate measurement for those unstable nodes is not accurate because although we run the TCP and UDP within a relatively short period of time, it is still at different times and the network performance of running UDP won't be the exact same as that of running TCP. Second, the minimum transmission time for different node pairs varies greatly. For example, the UDP time to AU is around 200 ms, which is 30 times longer than that of UV. So, even the packet loss rates are both zero, the TCP sender needs to wait longer to get acknowledged from AU than UV.

In order to find more accurate relation between delay difference and loss rate, we need to overcome the above two drawbacks we mentioned, namely unreliable loss rate measurement and physical transmission difference between those node pairs.

So, we choose a node which is not that stable but has acceptable VoIP performance and use the tcpdump command to measure the packet resending rate.

The node we chose is UCSB because it has acceptable VoIP performance when using TCP and the loss rate varies over time. Following is the chart of the delay difference vs. the packet resent rate. There are two kinds of values, the test value and the expected values by linear regression. Though the test value are not exactly linearly related to the packet resent rate, we can see an approximately linear relationship between them. In general, the bigger the packet resent rate, the bigger the difference between the 90^{th} percentile TCP and the transmission time.



5 Conclusion

In this set of experiments, we found that the NTP performance of many PlanetLab nodes is very poor. The clock for those nodes cannot be trusted. So we use round trip measurement to solve the NTP synchronization problem. We also do the one-way measurement on those node pairs which have good NTP performance and compare them with the results of round trip measurement. From the comparison, we know that the round trip measurement is accurate.

For every node pair, we have three kinds of sender and receivers. They use UDP, TCP NODELAY and TCP respectively. After analyzing all the node pairs, we found the performance of Hong Kong is intolerable, so we excluded it. For all the other node pairs, we can see that the average and 90^{th} percentile of both delay and jitter for UDP are better than those for TCP. This is very reasonable, because TCP is a connection-based protocol. When there is congestion on the network, the sender cannot send packets before all previous packets in the buffer are received so it causes longer delay. However, UDP just drops the packets, so the average delay time is much smaller for those network with high congestion, such as from Hong Kong to Columbia.

When sending packets using UDP, there is packet loss. However the loss rate is small, it won't affect the voice quality much. It is more important to have smaller delay and jitter, so using UDP is better than using TCP. TCP NODELAY performs better than TCP because with NODELAY, the sender gets acknowledgements earlier, so it can send more packets out earlier and decrease the overall delay.

In order to find out the relation of delay difference and packet loss rate, we used two methods. One is to use the data for all the node pairs when we compare the performance of these three protocols. Though we can see the trend of bigger loss rate results longer delay in TCP, the measurement is not accurate because the packet loss measured by using UDP cannot exactly represent the network performance when we send TCP packets. So we adopted another measurement to solve this problem. We get the packet resend rate of TCP sender by using the tcpdump command. We can see an approximately linear relationship between the delay difference and the packet resent rate.

In summary, UDP is more suitable for sending voice data than TCP, especially when the network loads are significant.

6 Reference:

1. www.planet-lab.org

2. Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture by Douglas E. Comer, Prentice Hall, third edition, 1998.

3. UNIX network programming Networking APIs: Sockets and XTI by W. Richard Stevens, Prentice Hall, second edition, 1997.