CSE573S: Protocols for Computer Networks – Spring 2005

Impact of Connection Splitting on TCP Performance

Arpith Chacko Jacob
Rahul Pratap Maddimsetty

Introduction:

In this project, we investigate the effect of splitting a single TCP connection into several shorter connections taking the same route. The main motivation for these experiments is to study the inherent tendency of TCP to favour (i.e. facilitate achievement of higher throughput on) connections with lower RTT. We observe that TCP indeed performs significantly better on connections with low RTT, and hence are able to speedup a file transfer between two nodes by adding one or more intermediate nodes.

The performance metrics used to compare the TCP performance of the direct connection to that of the the split connections are the Total Transfer time, Average RTT and Throughput. We study this effect for various file sizes, link qualities and number of intermediate nodes. Planet-Lab **[1]** was used to run a large number of experiments for this analysis.

Implementation:

The programs were written in C using the Unix sockets interface. TCP **[2]** with congestion control **[3]** was implemented over UDP. A maximum segment size of 1024 bytes was used with byte counting. Buffering out of order data and cumulative acknowledgements was performed at the receiver end. ACKs triggered the increase of the congestion window at the sender.

Exponential Weighted Moving Average (EWMA) Retransmission Timeout was implemented according to the following formula:

avg_rtt = 0.875 * avg_rtt + 0.125 * per_pair_rtt
rtt_var = 0.75 * rtt_var + 0.25 * | avg_rtt – per_pair_rtt |
retr_timeout = avg_rtt + 4 * rtt_var

Each of the intermediate nodes acts as a receiver for an upstream connection and a sender for a downstream connection with both TCP connections operating in parallel. Packets received from the upstream connection are placed in a (potentially) infinite queue in memory, which is then read by the thread servicing the downstream connection.

Exploratory Experiments:

In order to gather an informal idea of how the various parameters affect the TCP relative performances of the direct connection and the split connections, several exploratory experiments were conducted over Planet-Lab. The graphs for these experiments have been included in Appendix A. The results of and conclusions from these experiments are summarised below.

1. **HK-MO-MD (3MB)**
   In this preliminary experiment, there are 2 links with great disparity in latencies. It is expected that the one bottleneck may control the entire transmission. Therefore, splitting may not improve performance. It is observed that the transfer time does not improve as a result of splitting and average RTT and throughput graphs show effect of bottleneck.
2. **HK-MO-MD-NJ-NY (0.5MB)**
   In this experiment, the number of intermediate nodes is varied. There are 4 links, of which one is very slow compared to the other 3 that follow. It is possible to effectively combine the 3 fast links so that they behave like one fast link and reproduce the behavior seen in the previous experiment. Once again, the bottleneck will dominate the connection and prevent a benefit from splitting. It is observed that the transfer time does not improve as a result of splitting and average RTT and throughput graphs show effect of bottleneck.
3. **NY-IS-DE (3MB)**
   In this experiment, the effect of the route is studied. An indirect path is deliberately chosen, and with the chosen links of poorer quality than those on the direct path. As expected, splitting does not improve performance.
4. **NY-CA-IN (3MB)**
   In this preliminary experiment, unlike the 1st experiment, there isn't as much disparity between the quality of the links, though one is still distinctly faster than the other. It is observed that the Transfer time as well as Average RTT and Throughput are better for the split connection than for the direct connection.
5. **NY-CA-HK-IN (0.5MB)**
   In this experiment, again, the number of intermediate nodes is varied, and this time, again, one know poor quality link is added to force the transfer to take another path. There are two bottleneck links in succession that are expected to constrain the others. The transfer time does not improve from splitting. This experiment, however, is inconclusive in terms of Average RTT and Throughput. A possible explanation is the small file size used that masks the possible benefits of splitting.
6. **SG-CA-NY-DE-IT (0.5MB)**
   In this experiment, a large number of intermediate hosts is used. Between this

experiment and the next, the file size is varied in order to understand how much better splitting performs for big files than for small files. Also, the links are such that the two slower trans-oceanic links are interspersed with the faster links. Since no links can effectively be combined as in the 2nd experiment, the effect of increasing the number of intermediate hosts is expected to be more pronounced in these two experiments. It is observed that the transfer time, Average RTT as well as Throughput improve clearly as a result of splitting.

7. **SG-CA-NY-DE-IT (3MB)**

   This is the same as the previous experiment except for the file size. As expected, the larger file size sees the greater benefits of splitting, in terms of transfer time, and particularly Average RTT and Throughput.

From the exploratory experiments, the following informal conclusions are made, which are more rigorously tested in the controlled experiments and analyses that are reported in the next section.

1. TCP does appear to favour lower RTT connections.
2. Increasing the number of intermediate hosts results in better performance only if
   i. the relative quality of the links is reasonably balanced
   ii. the relatively high quality links are not clustered with other high quality links
   iii. and the chosen route follows the direct path reasonably closely.
3. Increasing the file size will reveal more significant and consistent performance improvement for split connections. Small files will behave unpredictably.

Analysis:

Dependence of Throughput on RTT:

The first experiment conducted was a file transfer between Singapore and Lebanon, a connection traversing the Pacific and Atlantic oceans and passing through the continental U.S. Two intermediate nodes were added, one in California and the other in Maryland. The bottleneck link was determined to be the Maryland – Lebanon link which is a satellite link.

	TIME (s)	**RTT (ms)**	**RTO (ms)**	**LOSS RATE**	**THROUGHPUT (bytes/sec)**
DIRECT	*348*	*620*	*699*	*0.057*	*12,053*
SINGAPORE - CALIFORNIA	*142*	*241*	*384*	*0.111*	*29,537*
CALIFORNIA - MARYLAND	*143*	*226*	*482*	*0.093*	*29,331*
MARYLAND - LEBANON	*219*	*398*	*456*	*0.056*	*19,152*

Table 1: Dependence of TCP throughput on RTT

The results of the experiment are tabulated in Table 1. The split transfer completed in 219 seconds, a speed increase of 129 seconds over the direct transfer. An inspection of figure 1 shows a clear difference in the average RTT of the various connections. The direct connection has an average RTT of 620 ms while the split connections have average RTTs ranging from 226 ms to 398 ms. The Singapore – California link, which has an average RTT of 241 ms has a throughput of 29,537 bytes / second, almost 2.5 times that of the direct connection. Further, the bottleneck link has an average RTT of more than 200 ms less than the direct connection. Hence, the throughput of the split transfer increases to 19,152 bytes / second as is evident from figure 2.

The experiment clearly shows the effect low RTT has on the throughput of a connection. Though both the direct and split connections transfer the same file over the same path, the latter performs better because it divides the transfer into multiple connection each with an average RTT lower than that of the direct connection.

Effect of File Size on TCP Throughput:

The previous experiment was repeated for various file sizes over the same two nodes. The results of the experiments are displayed in table 2.

	DIRECT TRANSFER					SPLIT TRANSFER				
FILE SIZE (MB)	TIME (s)	RTT (ms)	RTO (ms)	LOSS RATE	THR. (bytes/sec)	TIME (s)	RTT (ms)	RTO (ms)	LOSS RATE	THR. (bytes/sec)
1	*102*	*634*	*827*	*0.060*	*10,280*	*59*	*433*	*556*	*0.066*	*17,772*
2	*195*	*635*	*831*	*0.050*	*10,754*	*120*	*419*	*525*	*0.069*	*17,476*
4	*348*	*620*	*699*	*0.057*	*12,053*	*219*	*398*	*456*	*0.056*	*19,152*

Table 2: Effect of file size on TCP throughput

In all cases, the split transfer outperformed the direct connection with an increase in throughput of over 7,000 bytes / second. The RTT of the bottleneck link in the case of the split transfer remains at around 400 ms, far less than the average RTT of the direct connection. The dependence of TCP throughput on average RTT holds for various file sizes. With an increase in file size the speedup is more significant while extremely small files would benefit less from connection splitting.

Effect of Link Qualities on TCP Throughput:

Next, we determine if the dependence of TCP throughput on average RTT holds on various link qualities. Experiments were done on high RTT and low RTT connections, as well as with the bottleneck link upstream and downstream.

The first experiment was conducted between Brazil and India, with intermediate nodes in the USA and Hong Kong. The links between USA – Hong Kong and Hong Kong – India were determined to be the bottleneck links. The experiment was repeated between the two nodes, but with the transfer taking place in the opposite direction: India – Brazil, where the bottleneck links are now upstream. Figures 3-6 show the variation of RTT and Throughput with time for both directions of transfer. Table 3 summarizes these results.

	BRAZIL – INDIA					INDIA – BRAZIL			
	TIME (s)	RTT (ms)	LOSS RATE	THR. (bytes/ sec)		TIME (s)	RTT (ms)	LOSS RATE	THR. (bytes/ sec)
DIRECT	*115*	*849*	*0.089*	*7,157*	**DIRECT**	*111*	*674*	*0.083*	*7,415*
BR-USA	*64*	*476*	*0.097*	*12,861*	**INDIA-HK**	*50*	*425*	*0.093*	*16,462*
USA-HK	*66*	*502*	*0.094*	*12,471*	**HK-USA**	*51*	*389*	*0.112*	*16,139*
HK-INDIA	*69*	*491*	*0.122*	*11,930*	**USA-BR**	*52*	*404*	*0.107*	*15,828*

Table 3: Effect of link qualities on TCP throughput

In both cases, the average RTT of the split connection is less than the average RTT of the direct connection. Hence, irrespective of the location of the bottleneck links, the split transfer performs better than the direct transfer. Having a bottleneck link upstream possibly constrains the maximum throughput possible on a split transfer. However, since splitting the connection reduces the RTT for each connection, the overall throughput remains higher than that of the direct connection over the same path.

Next, the experiment was performed over two high capacity, low RTT connections between UCSB and NYU in the continental U.S. Two intermediate nodes at UUTAH and WUSTL were selected. The results of the experiment are summarized in table 4.

	TIME (s)	RTT (ms)	RTO (ms)	LOSS RATE	THROUGHPUT (bytes/sec)
DIRECT	*31*	*135*	*270*	*0.163*	*67,650*
UCSB - UUTAH	*24*	*89*	*158*	*0.135*	*87,381*
UUTAH - WUSTL	*26*	*75*	*158*	*0.124*	*80,660*
WUSTL - NYU	*26*	*83*	*207*	*0.139*	*80,660*

Table 4: Dependence of TCP throughput on RTT on a high capacity link

Due to the low RTT of the links, the effect of splitting the connection is less pronounced than on longer links. However, splitting the connection does speed up the transfer by 5 seconds, with an increase in throughput of more than 13,000 bytes / second. Again, the effect of splitting the direct connection reduces the average RTT from 135 ms to between 75 ms and 89 ms with a corresponding improvement in throughput.

Effect of the Number of Intermediate Nodes on TCP Throughput:

To measure the effect of the number of intermediate nodes on TCP throughput, the experiment was conducted between two nodes in Singapore and Iceland, with intermediate nodes in California, Missouri, Maryland and the United Kingdom.

NODES	TIME (s)	RTT (ms)	THROUGHPUT (bytes/sec)
DIRECT (SP-IS)	*312*	*604*	*6,722*
1 (MD)	*84*	*275*	*24,966*
2 (CA-MD)	*105*	*321*	*19,973*
3 (CA-MD-UK)	*108*	*364*	*19,418*
4 (CA-MO-MD-UK)	*135*	*385*	*15,534*

Table 5: Effect of the number of intermediate nodes on TCP throughput

Table 5 shows the effect of adding between 1 – 4 nodes in the split transfer. Adding a single node in Maryland drastically decreases the file transfer time from 312 seconds to 84 seconds, a speedup of almost 4. The average RTT of the bottleneck link is 275 ms, less than half the average RTT of the direct connection. TCP bias toward low RTT connections is evident from the fact that the throughput increases to 24,966 bytes / second from 6,722 bytes / second.

Adding additional nodes however does not improve performance, increasing the transfer time to 135 seconds for the split transfer with 4 intermediate nodes. The optimum number of nodes to split a direct connection depends on the placement of the nodes. It is reasonable to assume that splitting the bottleneck link between two nodes would decrease the file transfer time, while adding nodes on higher throughput links could actually increase the transfer time.

Buffering on intermediate nodes:

Next, we study the variation of queue size at the intermediate node. Every intermediate node receives data packets from an upstream connection, and places them in a packet queue. The packets are then read off of the queue and sent to the downstream connection.

The size of the queue at any instant of time depends on the type of the upstream and downstream links. If the upstream link is the bottleneck, packets can be sent on the downstream link as soon as they are received at the intermediate node. The interesting case arises when the downstream link is the bottleneck, which causes the queue at the intermediate node to build up at rate much larger than what the downstream link is capable of handling.

To study the variation of the queue size, the split transfer was performed between Missouri and Lebanon using a single intermediate node at Maryland. The upstream link between Missouri – Maryland is of higher capacity than between Maryland – Lebanon which is a low latency satellite link. On a file size of 2MB, the Missouri – Maryland transfer completed in just 17 seconds, while the Maryland – Lebanon link took 156 seconds.

Figure 7 shows the variation of the queue size at the intermediate node with time. The queue size increases rapidly to a maximum of over 1800 packets at which point the file transfer over the Missouri – Maryland link completes. At this instant, the intermediate node holds 88% of the file in its queue. The queue is emptied slowly over the next 139 seconds at a much slower rate over the Maryland – Lebanon link.

With an increase in the size of the file transferred, the problem of large queue sizes becomes a limiting factor in the usefulness of this approach. However, it is to be noted that since the overall throughput of the split transfer is limited by the throughput of the bottleneck link, all other links can be artificially constrained to operate at this rate.

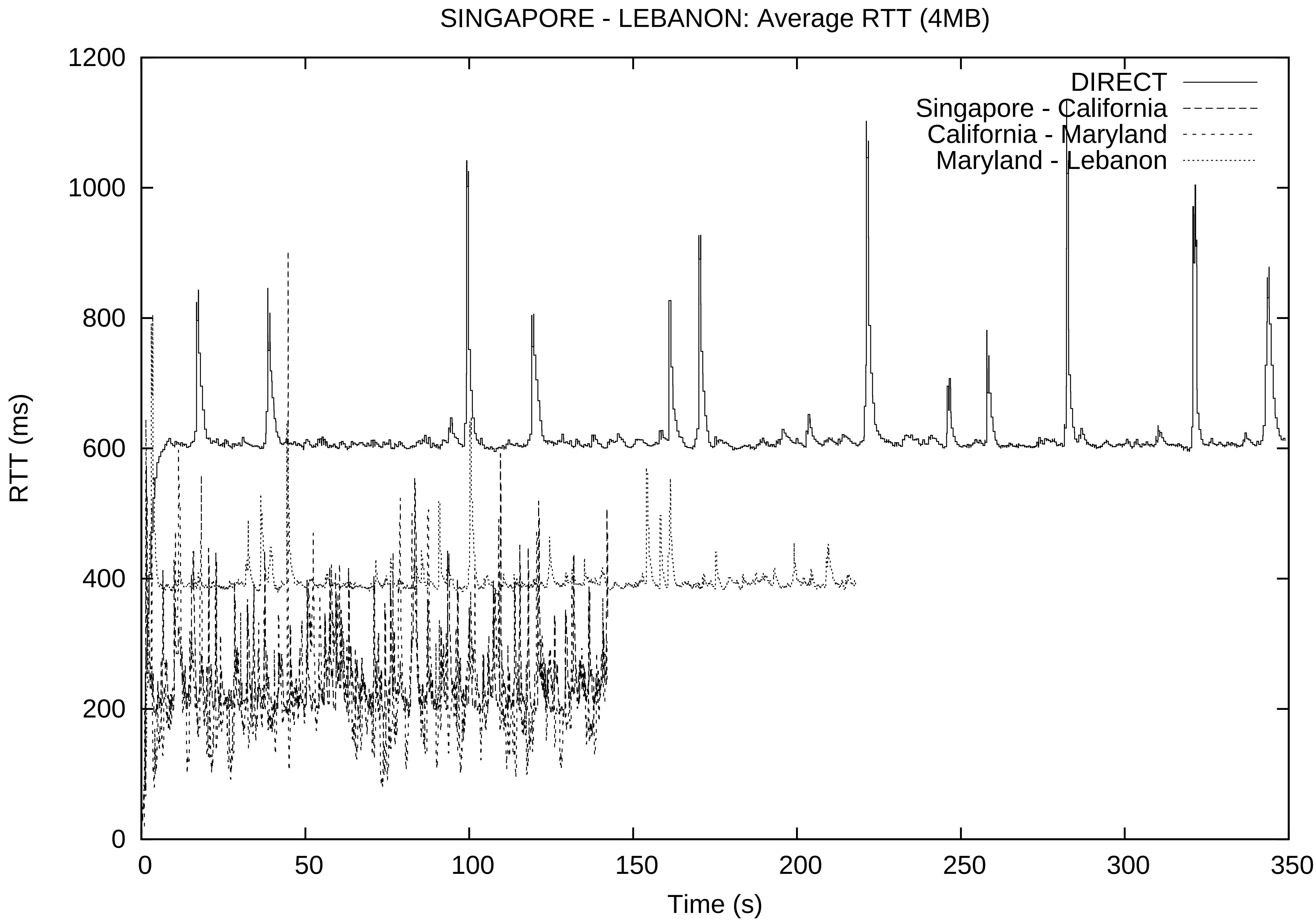
Conclusion:

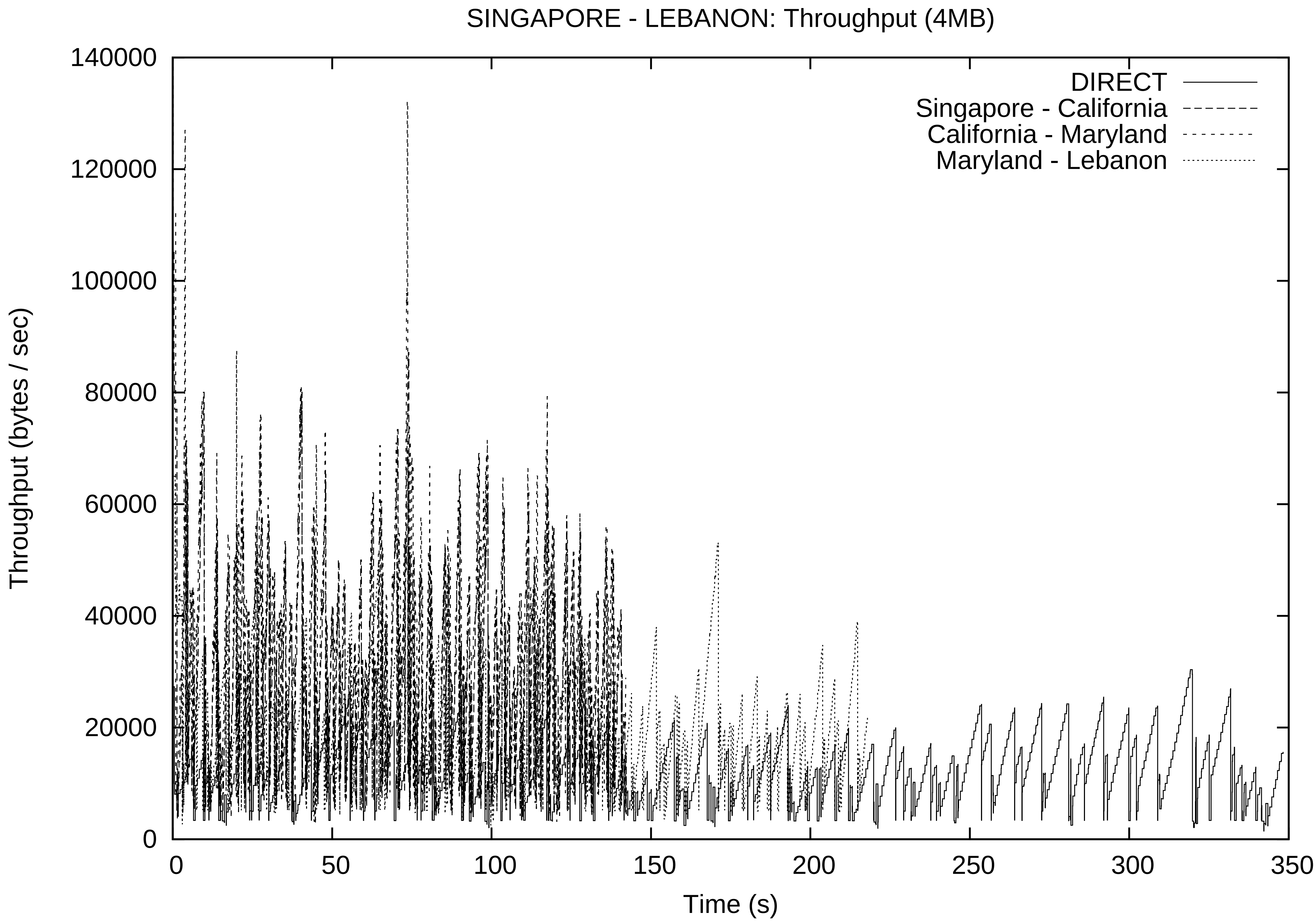
In this project, we conducted a number of experiments over Planet-Lab to study the dependence of TCP throughput on RTT. We compared two types of connections: a direct connection between two nodes, and a split transfer involving intermediate nodes between two end nodes. Even though both connections traversed the same path, and transferred the same file, the split transfer was found to perform better. This was determined to be due to the splitting of a single TCP connection which reduced the average RTT over the bottleneck link, and hence caused an increase in throughput. The dependence of TCP throughput on RTT was observed in low and high quality links, connections where the bottleneck was upstream or downstream, and with various numbers of intermediate nodes.

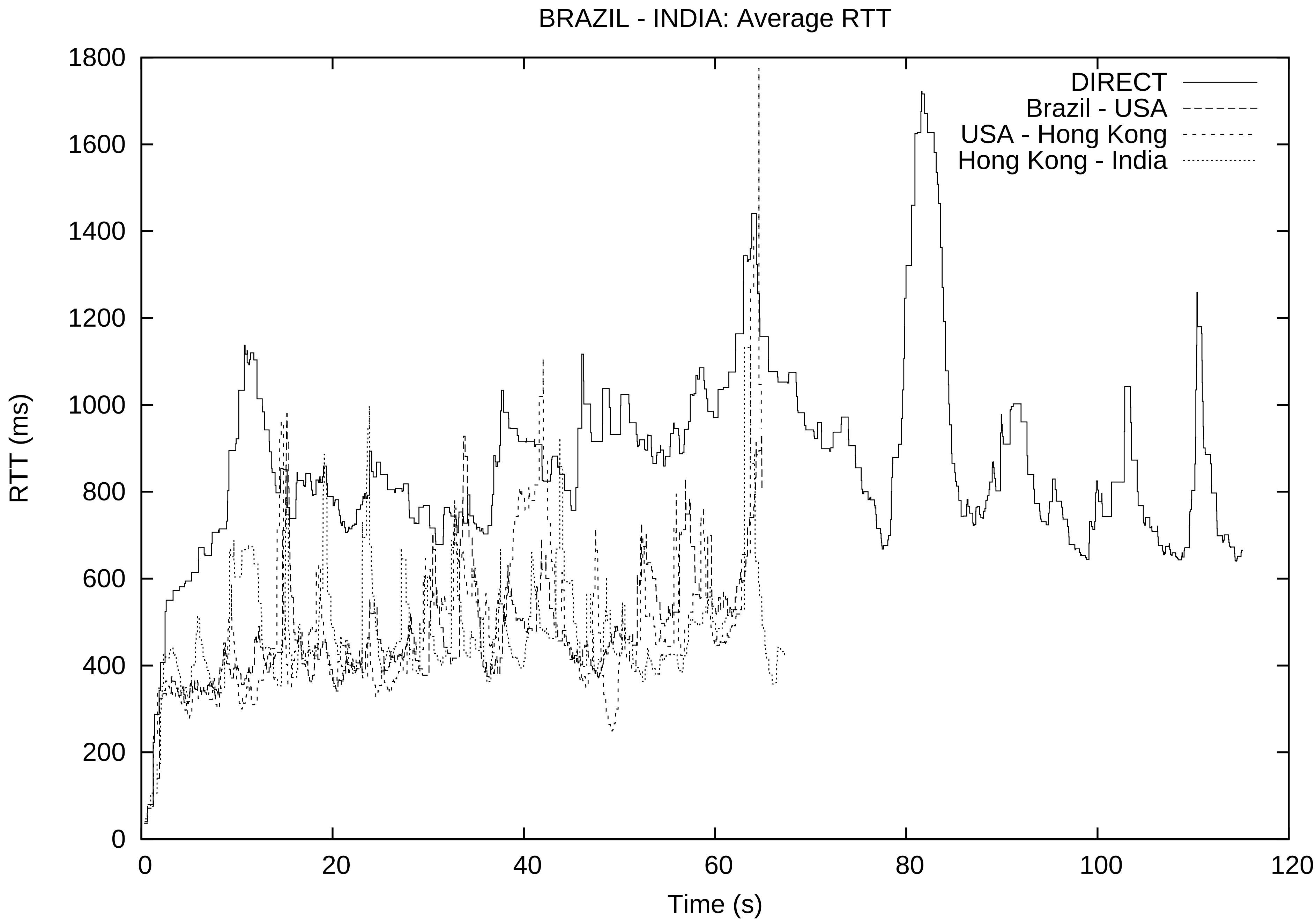
This characteristic of TCP could be used in real world applications such as P2P File Sharing, where dissemination of data can be done via intermediate nodes to help reduce the average RTT and hence increase the overall throughput.

References:

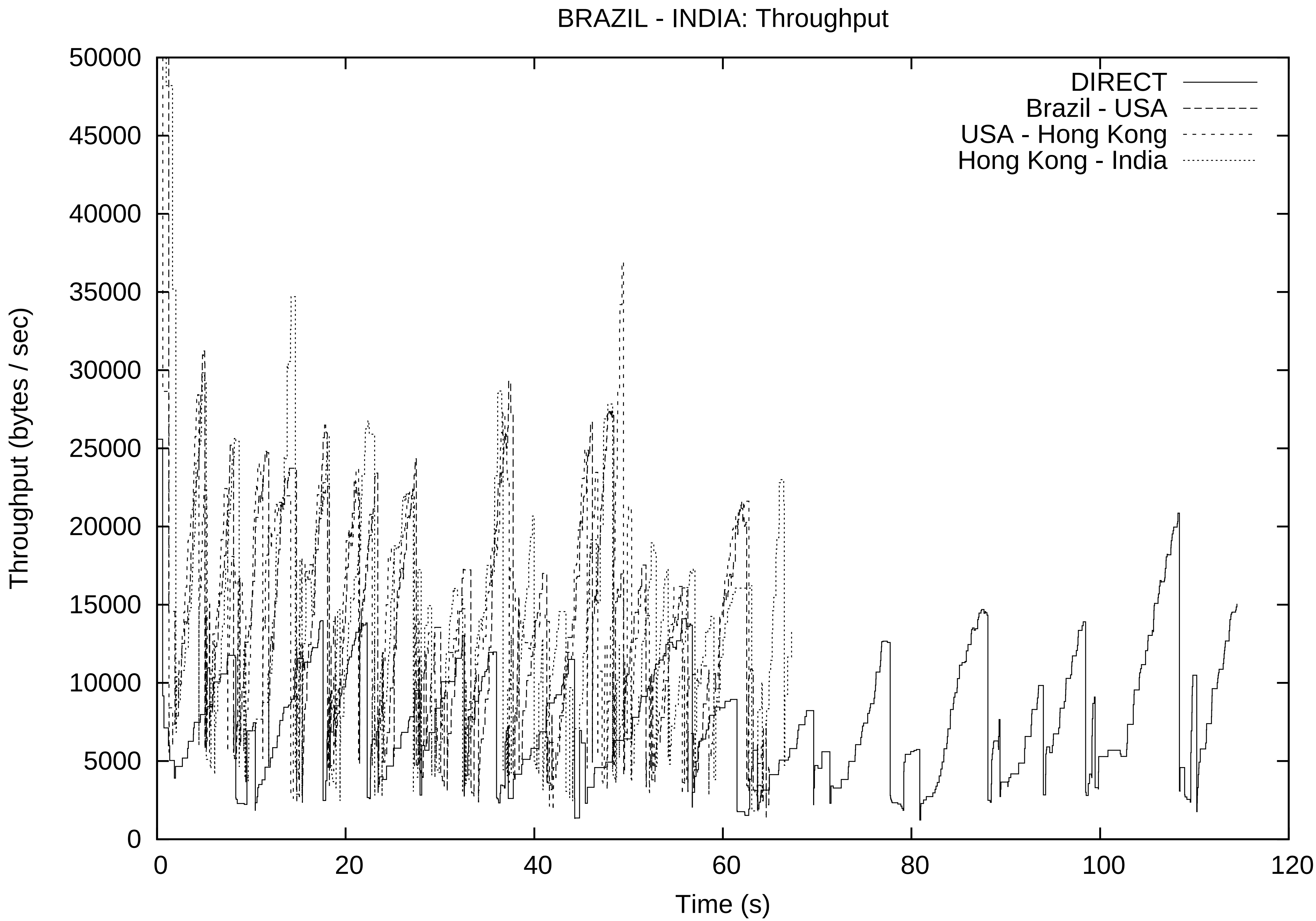
[1] http://www.planet-lab.org/
[2] RFC 793, “Transmission Control Protocol”
[3] RFC 2581, “TCP Congestion Control”

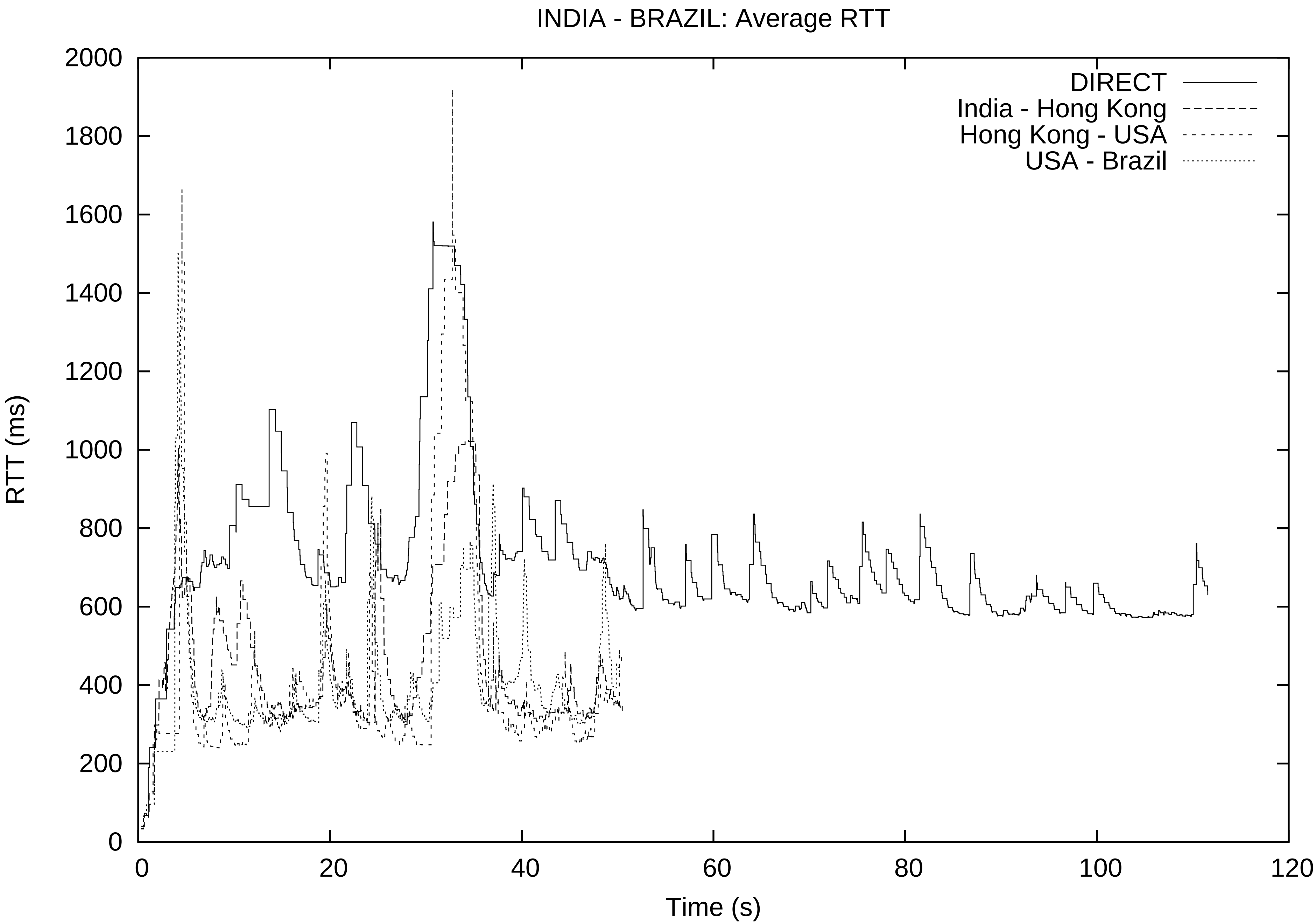

SINGAPORE - LEBANON: Average RTT (4MB)
DIRECT
Singapore - California
California - Maryland
Maryland - Lebanon
RTT (ms)
Time (s)
0
200
400
600
800
1000
1200
0
50
100
150
200
250
300
350

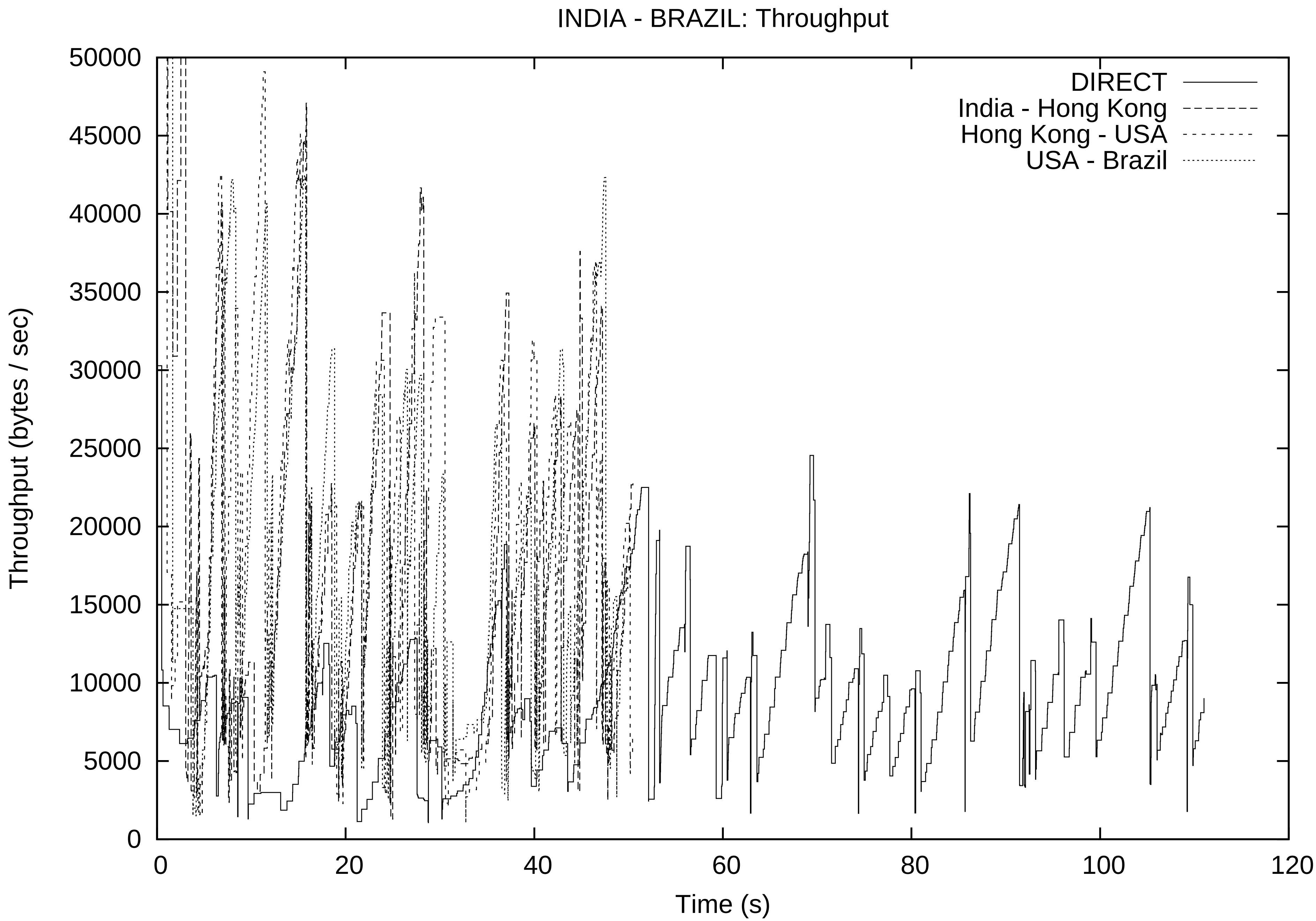

SINGAPORE - LEBANON: Throughput (4MB)
Throughput (bytes / sec)
Time (s)
DIRECT
Singapore - California
California - Maryland
Maryland - Lebanon
0
20000
40000
60000
80000
100000
120000
140000
0
50
100
150
200
250
300
350



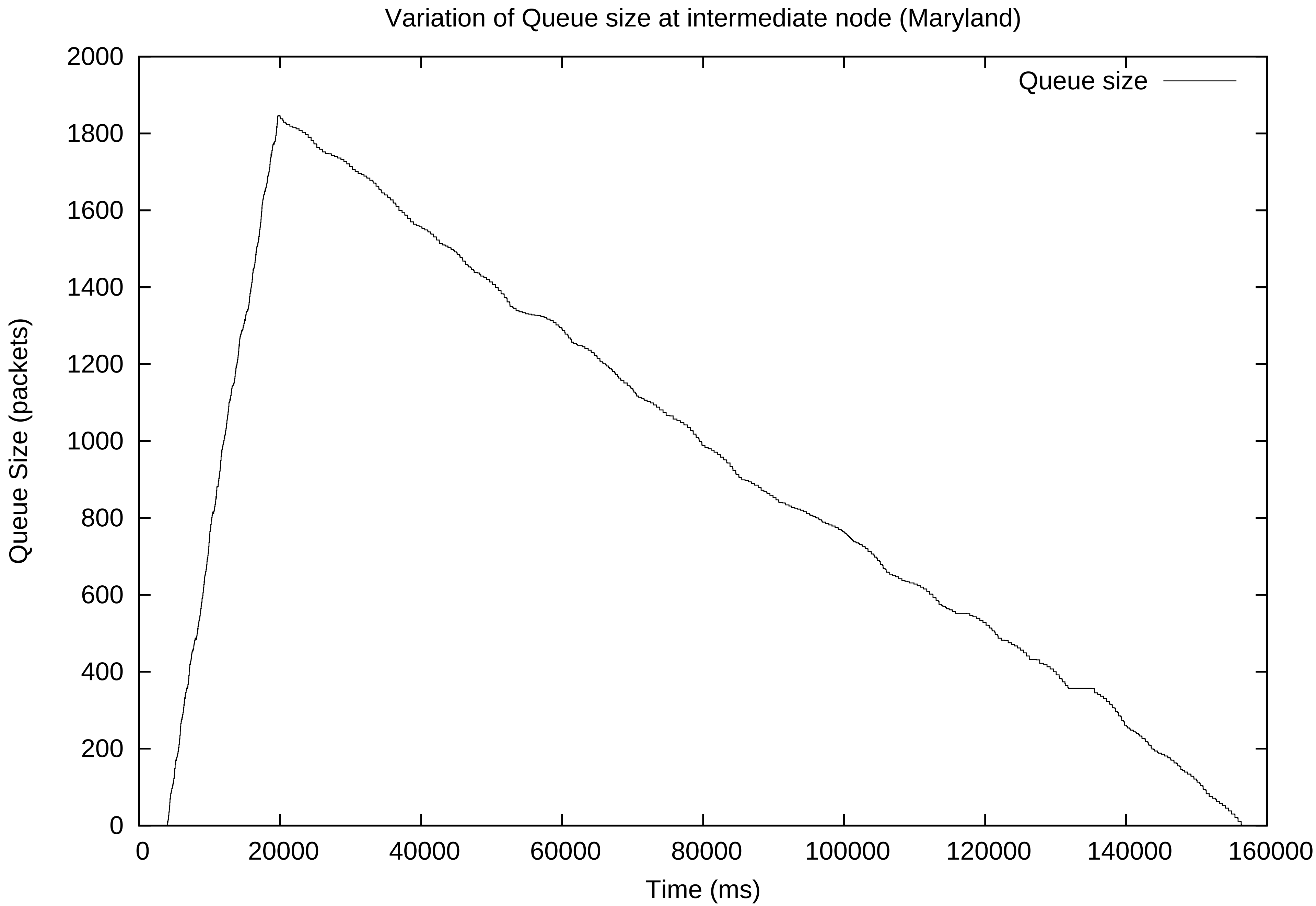
BRAZIL - INDIA: Average RTT
RTT (ms)
Time (s)
DIRECT
Brazil - USA
USA - Hong Kong
Hong Kong - India
0
200
400
600
800
1000
1200
1400
1600
1800
0
20
40
60
80
100
120


BRAZIL - INDIA: Throughput
Throughput (bytes / sec)
Time (s)
DIRECT
Brazil - USA
USA - Hong Kong
Hong Kong - India
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
0
20
40
60
80
100
120


INDIA - BRAZIL: Average RTT
RTT (ms)
Time (s)
DIRECT
India - Hong Kong
Hong Kong - USA
USA - Brazil
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
20
40
60
80
100
120



INDIA - BRAZIL: Throughput
Throughput (bytes / sec)
Time (s)
DIRECT
India - Hong Kong
Hong Kong - USA
USA - Brazil
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
0
20
40
60
80
100
120



Variation of Queue size at intermediate node (Maryland)
Queue size
Queue Size (packets)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
20000
40000
60000
80000
100000
120000
140000
160000
Time (ms)