UiO : Department of Informatics
University of Oslo

Understanding Network Performance Bottlenecks

Pratik Timalsena

Master's Thesis Autumn 2016

Understanding Network Performance Bottlenecks

Pratik Timalsena

November 15, 2016

Abstract

Over the past decade, the rapid growth of the Internet has challenged its performance. In spite of the significant improvement in speed, capacity, and technology, the performance of the Internet in many cases remains suboptimal. The fundamental problem is congested links that cause bottleneck leading to poor network performance. Apart from that, It is widely accepted that most congestion lies in the last mile. However, the performance of a network is also deteriorated in the core networks nowadays as the peering links have been affected severely due to the overburden of packets resulting in packet loss and poor performance. In the thesis, we investigated the presence and location of congested links in the core networks and the edge networks on the Internet. We measured end to end latency between over 200 node pairs from all over the world in PlanetLab and identified congested node pairs among them. The congested links between two end nodes were identified using traceroute analysis. By locating congested links in a network, we examined congestion in the edge networks and the core networks. We observed congestion both in the edge networks and the core networks, however, we detected around 58% congestion in the core networks and around 42% in the edge networks.

Contents

List of Figures

List of Tables

List of Algorithms

Preface

This thesis is submitted in a partial fulfillment of the requirements for a Master's Degree in Programming and Networks at the University of Oslo. My supervisors on this project have been Ahmed Elmokashfi, Andreas Petlund, and Pål Halvorsen. This thesis has been made solely by the author; a lot of the contents, however, is based on the research of others, the references to these sources have been provided as far as possible. I would like to thank Ahmed Elmokashfi and Andreas Petlund for their most valuable supervision and worthy guidelines during whole master thesis. I am thankful to Pål Halvorsen for the participation in the thesis.

Finally, I would like to thank everyone who has been helpful and supportive during my master thesis.

Part I

Introduction

Chapter 1

Introduction

Network performance has been a central research topic during the last decade. In reality, a network is designed in conjunction with its performance in mind. The performance is the service delivered by networks to its users. For example, the core business of a content delivery network hinges on its ability to deliver content at a predictable, consistent, and acceptable performance.For the sake to achieve the high performance, a significant effort has been made by improving speed, capacity, and technology. Despite spending a lot of money on upgrading technologies and resources, the network performance remains suboptimal[41]. The underlying problem is the congested links which cause bottlenecks and plague the network performance. In addition, a data packet can travel with a speed of the light in a theory [40]. Then a serious question arises why it takes so long time to cross short distances if the network is not congested. In this project, we investigate the prevalence of congestion in the wide area .

Although the Internet appears to be a single entity, it is a collection of thousands of different networks each providing connectivity to certain groups of end users . From the economic point of view, a network can be viewed as a first mile (ie, web hosting), a middle mile and the last mile (ie,end users). A middle mile is the part of the network between the network core and last mile providers, which comprises heterogeneous networks owned by multiple highly competing entities often peering with each other or providing transit service [35] .

It is generally accepted that most congestion lies in the last mile. This convention urged us to improve and speed up the last miles capacity. Under this circumstance, the last miles capacity has increased 50 folds over the last decade. The first miles in the network has also acquired attention and increased the speed by 20 folds over last 5 to 10 years. However, the middle part of a network or the core network has not enjoyed a similar growth. The peering links has been affected severely due to the overburden of packets resulting in packet loss and poor performance. Hence, the myth of last mile congestion has been outdated as the network performance has deteriorated in the middle part of the network as well [30] . In this thesis, we have made a small attempt to find the congested links in transit networks and the last mile networks that are affecting the performance of the network.

In this project, We performed active end to end measurement with more than 200 pairs that are part of the PlanetLab testbed. The nodes comprising the links are distributed all over the world. We selected node pairs such that they are located in different cities and belong to different networks, in order to maximize the inter-peer network distance. We probed each link for three weeks by sending packets from one end to another end and calculated RTT for all the links. We first identified the congested node pair links with the help of latency trend analysis. After that, we dug more into these congested links using traceroute. We performed correlation analysis between RTT and Hop by Hop delay for each hop on the path between the congested node pairs and found the congested links on the path. We located the position of these links on the network and identified whether the links are inter-domain links or intra-domain links. On basis of that information, we found that there are more congested links in transit networks than in last mile networks. In addition, we detected more congested intra-domain links than congested inter-domain links.

1.1 Motivation

1.1.1 Continuous and rapid growth of the Internet

The evolution of the broadband Internet has facilitated video and audio streaming on the Internet due to the availability of more bandwidth. At the same time, the Internet is growing rapidly in terms of the number of users and data traffic. Nowadays, there are more than 3 billion Internet users, generating a large amount of the data traffic. In the context of streaming data on the internet , the video traffic has surpassed all other traffic such as text, image, and audio, within a short time frame. In addition, various multimedia and cloud applications have emerged to utilize the available bandwidth on the Internet. Content providers like Netflix and YouTube generate enormous traffic volumes which is causing troubles for access providers by creating overloaded link due to congestion [42] .

The Introduction of the Smart Mobile phone and mobile broadband service has also contributed to the growth of the Internet traffic.Tthe mobile data has surpassed the fixed broadband data nowadays and is still growing significantly[26] .

Hence, we can predict that increasing the capacity of the network will not be sufficient for improving network performance. Since the network capacity will always be filled by data from new users and the applications, we need to dig more into identifying the actual problems within a network such as congestion, bottleneck, delay, and loss. Thereafter, we can solve the problems using some novel techniques.

1.1.2 Slow Internet speed

Because of congestion on the Internet, the end users are not receiving the quality of service they have expect. Users are complaining about the speed of the Internet and are not happy with a quality of the service. Nowadays they have reported that the broadband speed is not consistent and is slow and thus frustrates users as they did not get what they paid for. In the US only 30% of online users received the advertised speed [10]. Furthermore, user expectation is very high especially when video streaming, VOIP, online gaming. Thus, when there is a delay and buffering while online streaming or playing games, it might be frustrating for users. The main point is that performance of the network is not satisfactory in terms of users perspective because of congestion [] .

1.1.3 High Internet delay

The bufferbloats term has been coined to represent the large queuing delays on the internet. The use of very large buffers often lead to high queuing delay and thus contributes to network performance degradation and packet loss. As a result, the one-way trip delay can sometimes be around one second and two-way delay can be few seconds. This much of delay is comparable to time for communication from earth to the moon and back to earth[11]. Hence, the delay is one of the performance degrading factors, we need to investigate.

1.1.4 Problems in the core Network

The content provider routes their content via access providers to end consumers. In this process, they send excessive traffic causing congestion in the link between content providers and access provider or transit provider. The recent peering dispute between Netflix and Comcast reflects the scenario better which is explained in [16]. Netflix and Cogent suggested that Comcast made congestion on the route between Netflix and Cogent and forced for the direct interconnection.

1.2 Problem Statement

In the thesis, our goal is to examine congestion in the edge networks and core networks. In order to address this problem, we will look through following questions.
1) Which links are congested ?
2) Where in a network are congested links located ?
3) Whether congested links are in Intra-domain networks or Inter-domain networks ?
4) Where is more congestion (in the edge networks or in the core networks) ?

Chapter 2

Background

2.1 Internet

A computer Network is a set of computing devices, which communicate via a communication channel and share information, resources and data. The Internet is a giant network, which is a network of the networks that connects computers worldwide[33]. The internet might appear to be a single big network but the Internet is not merely a single network. It is formed by collecting various small network with a complex architecture beneath the surface of each. The group of networks under a single administration (Internet service provider or any large Institute) with a defined routing policy of its own is referred to as Autonomous system(AS). Moreover, Internet consists of about 50k Autonomous Systems controlled by ISPs (Internet Service providers), routers connecting them and protocols which facilitate the communication among them. We will discuss more on this topic later [15]. In this section, we will discuss on the Internet architecture, history of the Internet, protocols and other topics central to Internet bottleneck measurements.

2.1.1 A Brief History of Internet

The history of the Internet began with the formation of the Advanced Research Projects Agency (ARPA) in 1958 in the US. The history of the Internet can be explained as evolution from ARPANET to NFSNET and to the commercial Internet that we have nowadays.

After the establishment of ARPA, it was changed to DARPA (Defense Advanced Research Project Agency) and later changed back to ARPA. Thereafter, there was an ongoing research on packet switching both in academia and industry with the US government being the intertwined partner. The feasibility of using packets instead of circuits was studied and the concept of a computer network was realized. The first ARPANET plan was began as a design paper in 1967 meanwhile,the National Physical Laboratory (NPL) in England deployed an experimental network called the NPL using packet switching [28]. The world's first packet-switching computer network was established in 1969 by connecting computers at the University of California Los Angeles (UCLA), the Stanford Research Institute (SRI), the University of Utah and University of California Santa Barbara (UCSB) using separate mini computer which worked as a gateway for packets and called as Interface Message Processors (IMPs). The ARPANET gradually expanded as thirty academic, military and other research networks joined ARPANET by 1973. Due to the expansion of the ARPANET, there was a demand for an agreed set of rules for handling the packets. Thus, computer scientists Bob Kahn and Vint Cerf proposed a new method of sending packets in the network in 1974 by using technique packet within the digital envelope. The packet can be transferred to any computer in the network but can only be opened from the digital envelope at the final destination. This technique was referred to as the TCP/IP protocol. After the introduction of the TCP/IP communication among networks were through a common ARPANET language and the network grew significantly giving rise to a global interconnected network of networks, or Internet [1].

2.1.2 Growth in the Internet

In 1969, the first Internet node was installed aiming to connect 15 computers. After ongoing experiment for 4 years, 52 computers were connected. For 18 year the Internet hosts doubled every 15 months meanwhile the network traffic were doubled every 12 months. The trend changed drastically after 1997 after the introduction of Dense Wavelength Division Multiplexing (DWDM), which lowered the communication costs by a half every 12 months, and hence doubling the network traffic every six months. At the same time, the emergence of e-commerce also fuelled the increasing trend of Internet traffic in a such a way that the pace of the growth was four times a year. Because of this reason, there was strong demand for the improvement of the routers performance at a rate faster than 18 months doubling of semiconductor performance that Moore had predicted in 1975. The author [37] predicted that the same trend will continue until 2008 and after that as long as other methods to decrease costs of bandwidth is not introduced, the internet traffic growth will slow down as predicted in 1975. Figure 2.1 shows growth trends of Internet traffic, voice traffic, maximum trunk speed, and maximum switch speed required for large cities.

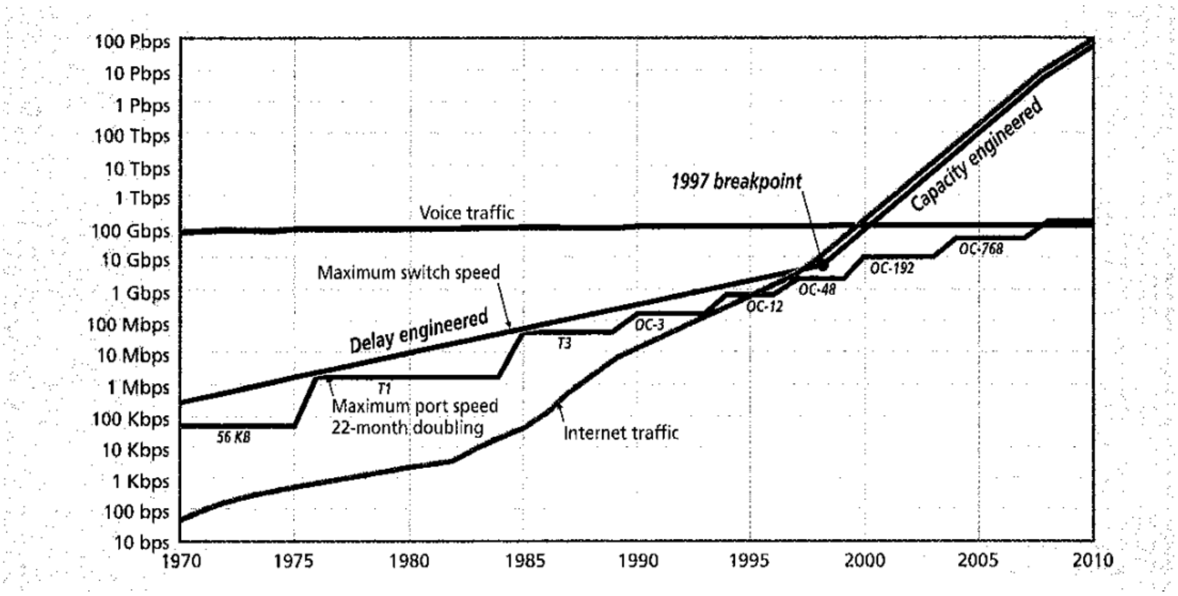


Figure 2.1: Growth trends of lnternet traffic, voice traffic, maximum trunk speed, and maximum switch speed required for large cities. [37]

After discussing on the history of the Internet growth, we need to take a turn towards the current trend of growth in the Internet . As shown in Figure 2.2, the Internet continued to grow. With this trend of the growth in Internet , the number of the internet user is about to cross 3 billions by 2015.

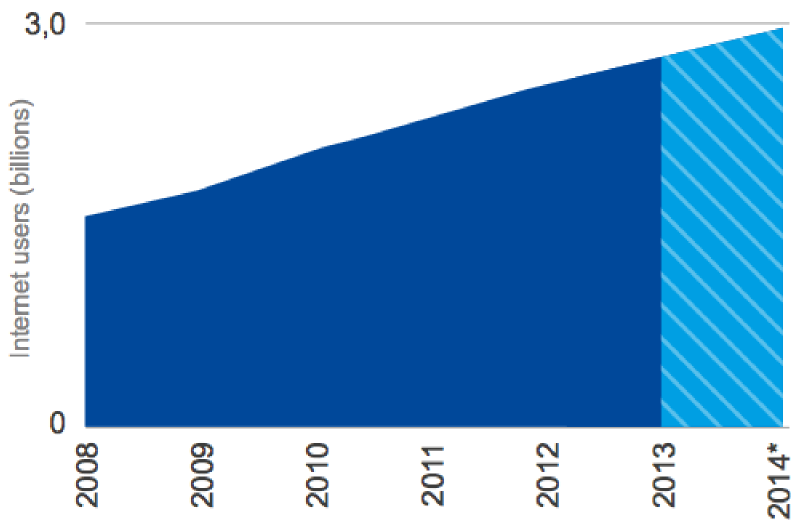


Figure 2.2: Internet users growth trend. [42]

After the broadband Internet took over dialup connection by 2004, users were to able to stream video and audio and signal. The video streaming become so popular so that today's video traffic have beaten all the traffic such as audio, image, email in terms of volume. Another turning point on the internet occurred with invention of the smartphone and Mobile broadband Internet. The number of mobile users began to grow faster as a result the mobile Internet user appeared in the significant figure among the Internet users after 2008. Then the fixed broadband Internet access and Mobile Internet access grew continuously. However, mobile Internet access grew significantly than fixed broadband Internet. In this context, the developing country exceeded the developed country on mobile Internet access. The global Internet access raised by 12% during 2008-2012.Thereafter 2012, the growth trend was slowed down from 10% annual growth to 5% for the broadband Internet access because the mobile broadband Internet acess got an importance over it. [42]The author predicted that this trend will last until 2018 and mobile Internet user and Mobile broadband Internet access is likely to flourish significantly as well. In this way, within this period,the mobile broadband Internet access will surpass fixed broadband Internet access.

[27]In the recent paper from Cisco, there is an update on the global mobile data traffic forecast for the period between 2015 and 2020. According to this report, the mobile data traffic grew 74 percent in 2015 as more than half a billion (563 million) mobile devices and connections were added. Furthermore, the smart phone has contributed the most for the growth. They also predicted that mobile data traffic will increase nearly eightfold between 2015 and 2020.

From the above information, We can predict that due to the rapid growth of the internet the link will be overloaded. Hence, the available resource might not be enough to handle those internet traffic causing degradation on the performance due to congestion.

2.1.3 Internet Architecture

In this section, we will explain more about Autonomous System because the Autonomous System is a foundation of the Internet architecture. Thereafter, we will discuss on how do they interact in the network.

Autonomous System

Autonomous System is a collection of routers and protocols which operate them and is owned by a single administrative domain. The routers exchange traffic within the AS using Interior gateway protocol such as RIP, OSPF and with other ASes using the border gateway protocol (BGP) . Thus the ISPs communicate with each other via BGP while allowing the individual ASes to implement their own policy. In addition, the interaction and relation among ISPs are governed by their policy and commercial agreement between the other ISPs as well[4].

Commercial agreements can be classified into customer-provider and peering. This also signifies what sort of relation and role do the ISPs have on the Internet. The ASes can play a role as service provider for customers. Customer pays the provider to get an internet connection.

Whereas in peering, the ASes agrees to exchange the traffic from their customer without any charge[18].

ISP Tier

Mainly, ISP can be classified to Tier1, Tier2, and Tier3 ISP. On the basis of the size and the geographic coverage, Tier 1 is further divided on regional Tier1 and global. Figure 2.3 depicts the classification of the ISPs on the basis of the size and the geographical coverage.

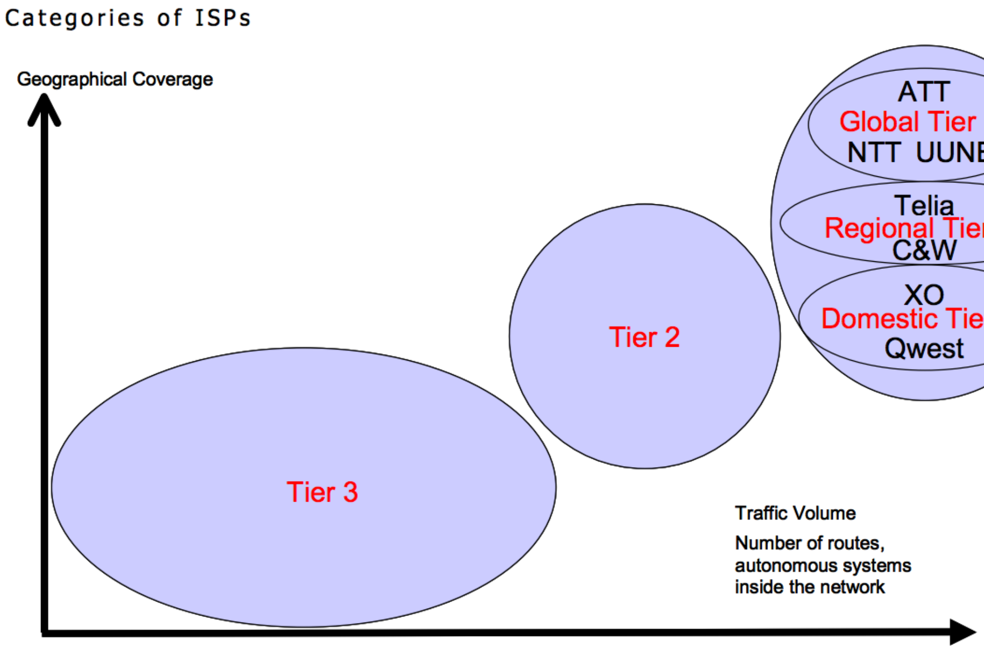


Figure 2.3: Types of ISP [44]

A Tier 1 ISP has larger network and greater geographical coverage than a Tier 2 ISP and a Tier 1 ISP. It has its own operating infrastructures including routers and other intermediate devices which constitute the backbone. The Tier 1 ISPs are connected to other Tier 1 ISPs or similar sized networks by private peering. They are interconnected at Internet Exchange points(IXPs). The global Tier 1 ISP have its own communication infrastructure or it can also use the alternative carrier communicating circuit depending upon the agreement with other ISPs. Generally, the Tier 1 ISPs are ASes that cover many continents.
The scope of Tier 2 ISPs is limited, very few of them can provide service over more than 2 continents. The important feature is that they at least one hop far from the core Internet. Tier 3 ISPs have a very limited scope as they only cover one country or metropolitan areas. Basically, they provide the Internet connection to the end users.Usually, Tier 3 ISPs are customers of the Tier 1 ISPs. They need to travel through many network and routers to access some parts of the Internet[44].

2.1.4 Routing Protocol in the Internet

Internet Routing is governed by Intra-domain Routing Protocol for routing in a single AS and Inter-domain Routing Protocol for routing in different ASes. In the Intra-domain routing protocol, all the routers are equal and announces the routing path to every router. Here, the router selects the best path on basis of a metric specified by the administrator. However, in Inter-domain Routing all the routers are not equal and do not provide transit service to all the routers. A router in an AS announces the path to the destination via another ASes on the basis of the metric set by administrator and agreement set among the ASes[36].

Broder Gateway Protocol (BGP)

BGP is a very robust and scalable routing protocol used for routing on the Internet. BGP is mainly inter-domain routing protocol as it is used to route traffic between ASes but it is also used to route traffic within the same AS. Thus BGP can be classified into EBGP (External Border Gateway Protocol) when used for communicating with different ISPs and IBGP (Interior Border Gateway Protocol) when used to interact within the same ISP. Figure 2.4 depicts basic distinction of IBGP and EBGP. BGP uses the various routing parameter to address the scalability and effective routing or to choose the best path. These routing parameters are referred to as BGP attributes. These attribute used in BGP for route selection are Weight, Local preference, Multi-exit discriminator, Origin,AS_path, Next hop, Community. The detail explanation of those attributes can be found in [13]. In order to reduce the Internet routing table, apart from BGP attributes classless inter-domain routing (CIDR) is implemented by BGP.

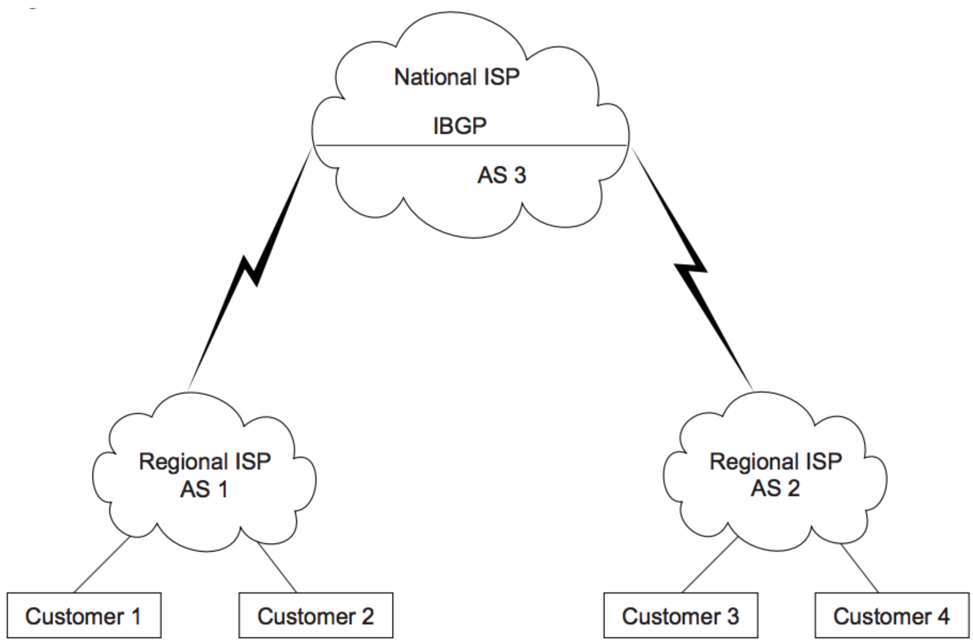


Figure 2.4: External and Internal BGP [13]

How BGP Works

BGP is a path vector protocol for routing between ASes. It carries routing information where the routing is path is a sequence of the Autonomous System Numbers which needs to be traversed to reach a certain prefix This feature contributes to enabling loop prevention. BGP uses TCP as a transport protocol and the BGP session starts with TCP connection between the BGP speakers. All the routers do not run BGP process only selected router which has to communicate with other ASes run BGP process and they are called as BGP speakers. The BGP speakers who establish a connection for exchange of routing information are neighbours or peers. Thus the routing informations are exchanged with all the candidate which are connected. There is no periodic update in BGP but neighbours are updated if the networking information is changed via the UPDATE message. The BGP routers can advertise routes via the UPDATE message and also can withdraw the invalid route i.e, the destination can not be reached through this path. To check if the connections between peers are alive BGP router periodically sends KEEPALIVE message. BGP has a graceful feature to facilitate the closing of connection with the peer in case there is a disagreement between the peers because of various circumstances. In this context, BGP sends a NOTIFICATION error before TCP connection hence saving the time and resource of the Network. BGP speaker has a full view of the Internet routing table. [39]

2.2 Congestion in the Internet

Congestion occurs when there is more demand than the available capacity. The congestion is not defined officially in such a way that the definition can be accepted universally. It is defined differently by different entities from the different perspective. We will discuss some definition of congestion from a selection of textbooks and articles. [43] According to user experience perspective, a network is said to be congested if the service quality noticed by the user decreases because of an increase in network load. According to queuing theory, there is a congestion if the arrival rate is greater than the service rate. However, the networking textbook defined building of queue of packets is not a congestion rather it is a contention. According to Networking textbook, congestion occurs if the packet is dropped when the queue is full. The Network operator definition of congestion is based upon the load on the network over a particular time. More precisely the network is congested if the load on the links has exceeded the threshold level [5].
From the above definitions, if a delay happens while transferring packet over a link from one end to another and the the performance deteriorates because of queuing, the link is said to be congested.

2.2.1 Distribution of congestion in the internet

The congestion can happen anywhere on the Internet for an instance, it might be at the core, edge of the network or somewhere in between. In this thesis, our main goal is to investigate whether the congestion is at the core or at the edge of the network. Although the congestion is an important topic nowadays, understanding of the congestion is affected by the unavailability of real data. The complexity of the Internet makes it hard to precisely simulate any larger part of the system. Models and simulation can be a very useful tool for picturing a state of system but It doesn't provide the probability distribution describing the likelihood of different states. This scenario is well explained in [19]. With in this context, they measured the distribution of congestion in DSL and cable Internet Service providers network in the US. They found the different congestion patterns in DSL and cable networks. In the DSL the most congestion was found in the last mile portion Whereas in cable networks the congestion was detected somewhere in the middle mile expect few cable ISP networks where the congestion was detected in the last mile. Indeed, the article [19]gives a good vision for measuring a distribution of congestion on the network.

2.2.2 Congestion in the core of Internet

The major part of the Internet traffic is comprised of the traffic that originates from the larger content providers and their content delivery networks (CDNs). In 2013, research showed that half of all peak period downstream consumer traffic came from Netflix or Youtube [14]. Although there should be the suitable interconnection between CDNs and ISPs to carry the traffic over the internet, it is viewed that the negotiations between them have been contentious resulting that traffic is flowing over the link with insufficient capacity,finally causing the congestion[14].

The evolution of the large content providers and their CDNs implementation has given rise to peering disputes although it existed before as well. These interconnection link between them are being congested for many hours while carrying high loads of the data.The peering disputes between Comcast and Netflix via cogent manifested the significant congestion on the path while carrying high volumes of video traffic. The similar case studies related to content providers and peering disputes between them resulting the congestion is explained in [14]. They also mentioned that when the additional link is added the congestion vanishes.

2.2.3 Internet Buffer and Congestion

The networks are suffering from the unnecessary delay and poor performance nowadays. There are several factors governing the delay in the network and one of the significant contributing factors is a poor buffer management[20] .We need a buffer to store packet when the network is busy and later on send it to destination for improving the performance by reducing packet loss . However, large-sized buffers are installed nowadays everywhere such as in routers,switches,and gateways, without proper visions and testing might affect the performance of the network. Excessive buffering of packets on the network causing a high latency and the reduced throughput is called as bufferbloat. The main issue of bufferbloat is it affects the working of the congestion control algorithm. For example, TCP congestion control algorithm works on the basis of the packet loss notification. When we are using the large buffers it takes very long time to fill the buffer and it only drops packets in a queue when the buffer is completely full. Due to this fact, the congestion avoiding mechanism does not get informed about the congestion timely by packet loss or explicit congestion notification (ECN). Therefore, it cannot take action in right time to avoid congestion on the network by controlling the sending rate. So, the buffer management should be handled very effectively in correspondence with congestion avoidance solution to get the overall good performance on the network. Besides the latency due to buffer-bloating, there are more factors that are jointly affecting latency experienced by the packets. The latency experienced by a packet is comprised of communication delay (time taken to send the packets across communication link), processing delay (time spent by each network item to handle the packet) and queuing delay (time spent for the packets being processed or transmitted) [20]. To handle the queuing delay the several solutions has been implemented one of the best methods is Active Queue Management. We will discuss more on the AQM in another section.

2.2.4 Active Queue Management (AQM)

Current Internet usage is dominated by TCP traffic thus TCP congestion control mechanism along with some packet queuing algorithms are used widely to handle congestion on the Internet. TCP uses an additive-increase-multiplicative-decrease algorithm (AIMD) to handle the congestion on the internet [45] . TCP sends the packet using window through which it controls the sending rate. After every round trip time the window size is doubled until there is no packet loss detected. When the packet is dropped, TCP assumes that there is a congestion and the window size is reduced by half. In this way, TCP controls the sending rate on the basis of the acknowledgement from the receiver[38]. But this method has a big loop hole as it cannot detect congestion before the network gets overloaded. The worst case may happen when most of the queues at routers are full leading to simultaneous packets drop on most connections. This phenomenon is referred to as global synchronization [23] . In that case, all the senders will lower the sending rate at the same time and again try to increase the sending rate to check ACK rate. In this way, the network might suffer from severe problems such as inefficient bandwidth utilization, a poor performance, and an inevitable congestion. To overcome the drawbacks of the older method we need to look for more efficient algorithm which can detect early and handle congestion better and AQM might be a good choice.

AQM is a mechanism for dropping packet from routers queues that have been proposed to support end-to-end congestion control mechanism on the Internet. In the current tail-drop (TD) method, the packets are dropped from the tail when the queue is full while in the AQM the packets are dropped before the queue is full by using RED algorithm [23]. AQM schedules the packets and it has dropping function to handle the congestion detection and control.

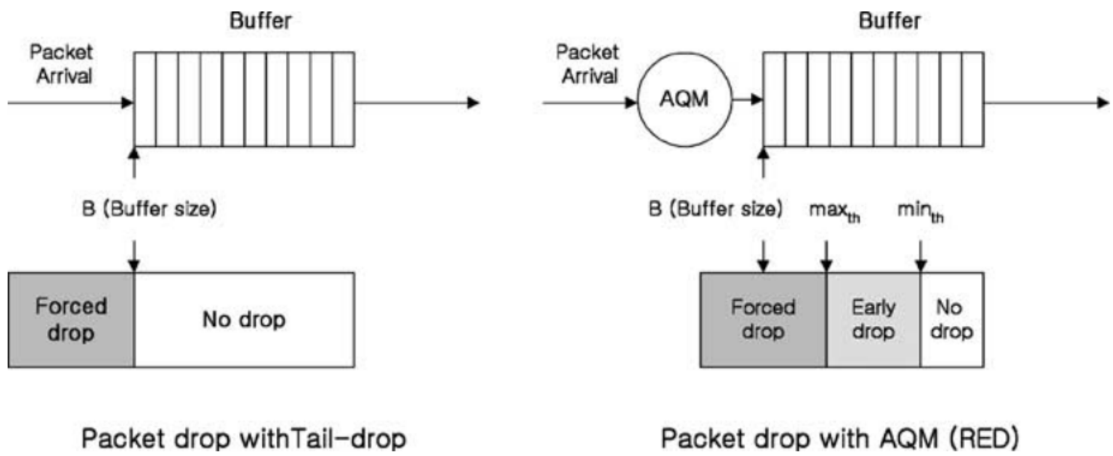


Figure 2.5: Packet drop functions with AQM and tail-drop. [38]

Figure 2.5 illustrates tail drop and AQM queues. There are two main functions which are based on FIFO mechanism to handle packets at router they are congestion indicator and congestion control function respectively [38]. The congestion Indicator detects congestion and the congestion control function avoids and controls the congestion. In the TD congestion control mechanism, current existing queue length acts as congestion indicator and controls congestion by dropping the packets when the buffer is full. In the AQM the congestion indicator is enhanced with probabilistic early dropping functionality called as RED which contributes for the early dropping of the packet before the buffer is full. In addition, it also implements exponentially weighted moving average (EWMA) queue length which boosts the congestion detection by dealing smoothly with bursty incoming traffic [38].

2.3 End to end delay measurement

The end-to-end delay is the sum of the delay occurred on each node on the way from source to destination. For example,UDP probe packet is sent at regular interval and the round trip delay is measured to analyse end-to-end delay and packet loss behaviour on the Internet. With this method, we can study the structure of Internet load with respect to various timescales by changing the time interval between probe packets[6].

Component of the end to end delay

The packet from the source has to be routed through various nodes and routers on the way to the destination. We need to categorise the delay on the basis of the delay occurred in between these intermediate node and routers. [8] The end to end delay can be categorised into four main types: processing delay, transmission delay, propagation delay and queuing delay. The time required for processing a packet at each node and also prepare for retransmission to the respective node is a processing delay. The protocol stack, computational power available and link driver are the factors deciding the processing delay. The time needed to transfer from first to last bit via a communication link is referred as a transmission delay. The transmission delay is directly affected by the speed of the communication channel. The propagation delay is the time to propagate a bit via communication channel link. It is governed by a travel time of an electromagnetic wave through a physical channel of the communication path and is independent of the actual traffic on the link. While the packet traverses the various node it has to be in the buffer of the routers before it is retransmitted. Thus the waiting time in the queue is a queuing delay[8].

Significance of end to end delay measurement

The one way or round trip delay of a UDP packet had been measured on the Internet. Apart from that, various experiments were conducted to measure TCP delays, losses, and other routing dynamics. These experiments often help researchers to study the strange behaviour of the Internet. Besides that, we can measure the delay distribution on the internet and we can figure out if the QoS on the Internet is verified or not. We can get vital ideas from experiments to re-dimension to minimise the delay. It is possible to find the bottleneck links where competing traffic leads to congestion via end to end delay measurement. The delay along with hop count measurement can support researchers while choosing the parameters for the large-scale simulation and modelling of the Internet [7, 8, 9, 17].

2.4 Performance Bottlenecks

A bottleneck refers to a phenomenon where the performance of a system is limited because of a resource or an application component. The resource can be, CPU, memory, disk ,and Network Interface Card. The bottleneck components are the prime causes of undesirable behaviour and poor performance of the system. [25].

2.4.1 Types of Bottlenecks

There are mainly two types of bottlenecks which are explained as following.

Resource Saturation Bottlenecks

When a system has fully utilised the resource or has crossed a set threshold, the situation is regarded as resource Saturation. The performance of the system is deteriorated because of the resource saturation. Different system resources are bottlenecked differently after resource saturation in the system. CPU utilisation around 100% results in a congested queue and hence contributes to growth in latency. If a system reaches the memory constrained capacity condition due to limited physical memory or memory leak in the system, there will be constant paging and swapping resulting loss in performance. Similarly, when a system faces disk saturation, the constant disk access beyond the available bandwidth will force the new IO request to be in a queue. Network saturation conditions due to fully utilised bandwidth will affect new traffic by dropping them or delaying their processing[24].

Resource Contention Bottlenecks

The system has limited resources such as CPU cycles, IO bandwidth, physical memory, buffers, semaphores, mutexes etc.,however, the application processes in a multitasking environment will contend for those limited resources and lead to a performance bottleneck. The most appropriate example is resource contention among different cloud tenants in cloud data centres. The contention for different system resources has a distinguishing impact on performance degradation of the system. The contention for CPU among multiple process results to congested queue and performance interference especially, in a virtualised system, using CPU hogging programs. Memory contention also has a severe impact on performance. In the same way, disk contention among processes will cause the performance loss especially, in IO loads because of the performance gap between Processor and IO with restricted disk payload. Network contention will also result in deterioration of the performance by demanding more communication links at the peak times and hence lowering the effective offered bandwidth [24].

2.4.2 Bottlenecks behaviours

The bottlenecks behaviour is different for the different system an application. This is governed by the interaction between components and the system. Basically, there are three kinds of bottlenecks behaviours.

Single Bottlenecks

The Bottlenecks in a system is because of the predominant saturation of resource at a single point or component of a system.

Multiple Bottlenecks

Two or more than two components of the systems get saturated and simultaneously contributes for the bottlenecks in the system.This may happen because of the interdependency of the components on the system.

Shifting Bottlenecks

This is a little bit complicated issue where the bottleneck shifts from one component to another or from one point on the system to another point. This happens because of interdependency between components. One application may cause another application to change its behaviour and thus changing the behaviour of the application shifts the bottleneck from one component to another and so on.[25]

2.5 Network Performance Metrics

In order to gain insight into network performance and know its behaviours, we need to measure it. There are several standards and non-standard metrics available for measurement. In this project, we will use some of the well-known metrics such as latency, loss etc. The brief overview of the network metrics is explained as follows.

Availability

Availability metrics evaluates the reliability of the network which means the percentage of the time the network is running without failure.

Loss

The loss metrics assess the percentage of packets lost because of the network congestion or transmission error. The loss can be measured for one-way path or two-way path depending on the requirement.

Delay

The delay is a measurement of the time that a packet takes to reach the destination from the sender. On the basis of the routing path, it can be Round trip time or just in a single path called one-way delay.

Bandwidth

Bandwidth is the amount of data which can be transferred in the network in a time unit, both dependent and independent from the current network traffic.
Apart from those performance metrics, we need to look for other non-standard metrics which are often related to the system and can contribute to the performance degradation of the network. Thus, monitoring system resources such as CPU,memory, and load in the network provides the

systems overview and resource status. In this way, we will not be misled by the result in case the system is causing the trouble for performance deterioration.[34]

2.6 PlanetLab Testbed

The experiment is carried out on the PlanetLab Testbed. This section gives a brief overview of the PlanetLab Testbed and how it operates. PlanetLab is a global research Network which consists of dedicated servers. The main goal of the PlanetLab is to support the development of new Internet services and protocols such as peer to peer systems, overlay routing distributed storage etc. PlanetLab is mainly divided into four branches based on the geographical distribution of the sites. PlanetLab Central (referred as PLC is the main authority handling nodes in the USA). The PlanetLab Europe (PLE) consist of the European nodes, PLJ (PlanetLab Japan) contains node in japan and similarly PLK (PlanetLap Korea) contains node in the Korea. The PlanetLab consist of about 1100 nodes which are associated with 500 sites being distributed over the world. For a sake of explaining how the PlanetLab operates, we took PlanetLab Europe as a example.Meanwhile, the PlanetLab Europe have more than 300 nodes distributed all over the world [29]. The distribution of the PlanetLab node Europe is illustrated in Fig 2.6.

Figure 2.6: PlanetLab European sites. [29]

PlanetLab nodes are gathered into a set called a slice. Administrator on the basis of the user's requests creates the slices. The node in the slice runs a Linux virtual machine referred to as silver. The user can login remotely to these nodes and run services for experimental purpose. Nodes from different sites can be added to a slice,therefore same nodes are added to different slices and running at the same time. The PlanetLab creates a new silver and runs on the node thus giving impression that silver as a node for users.
The PlanetLab slice indeed is a collection of the distributed resources.[12] Virtual Machine runs on a single node and allocates the certain portion of the resource of the node thus slice can be also the network of the Virtual Machines. Multiple numbers of Virtual Machines run on a PlanetLab node and thus there is a VMM to manage the resource sharing among these VMs at that node. It is interesting to know how the slices are created dynamically and resources are distributed and managed among them. There are 5 components that take control over the process of acquiring slice and resource management. The first component is node manager, which acts partly as VMM in the node. It takes tickets as inputs and checks if the request can be redeemed. If the request can be fulfilled it reserves the resource and create a VM that takes the reserved resource and finally replied with leased status. The second component is the resource monitor, which monitors resource periodically and reports to the agent about the resource availability. In figure 2.7, the steps while acquiring the slice are depicted, the first step is resource monitoring and reporting resource availability to the agent a third component which is responsible for advertising the resource availability and requirements to the tickets.

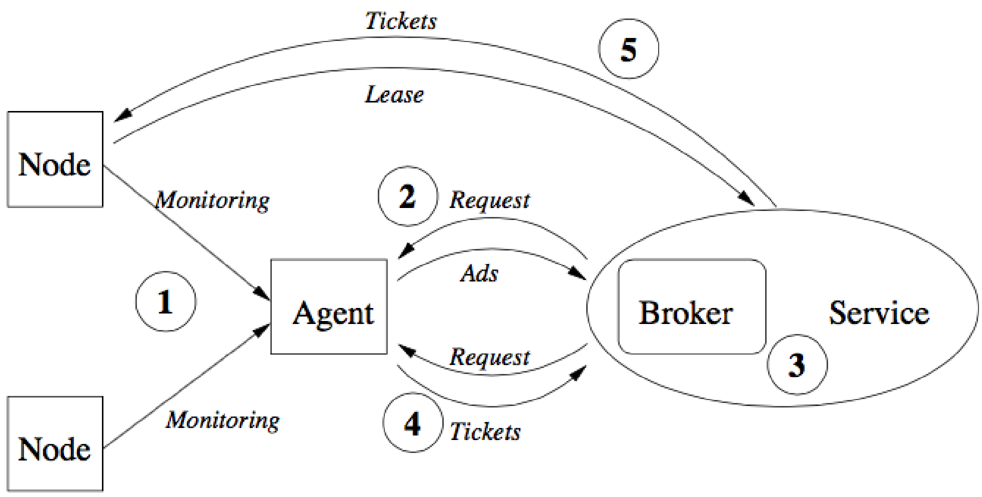


Figure 2.7: The process of acquiring the slice [12]

The fourth component is resource broker, which replies the queries of the service manager. The service manager is the fifth component that is associated with each service, and it contacts a resource broker to find slice specification and tickets to run it.The query from service manager describes the resource need to run the service and the principal behind the request for service. At step 2, the resource broker contact agent for the description of the ticket that is held by agents for a service. Then the agent responds with sets of advertisements. The broker combines the advertisements with known service requirement in order to generate the specification of the slice. Then broker requests for the ticket to instantiate the slice, and agent replies with the ticket. These phenomena are depicted as steps 3 and 4 while acquiring slice. Finally, at step 5, service provides the tickets to admission control on each node to create a network of virtual machines. When the virtual machines are created then service manager loads and starts a program in every virtual machine. The admission control returns the status lease on the slice.

Related Works

One of the latest work was done on inferring congestion on the inter-domain link. The simple and lightweight method called Time Sequence Latency Probes (TSLP). The idea behind this method is to frequently repeat the Round Trip Time (RTT) measurements from a vantage point to near and far routers of the inter-domain link where measured RTTs being a function of the queue length between two routers. The main advantage of the method is that it tries to localize the congestion from a single point, Vantage Point (VP), without a need of responding server on the other end. However, if the experiment produces broadband performance map, it is required to have many VP on the several access points. The experimental results value proves that it can localise the congestion on the inter-domain link at the edge. On the course of the experiment, there are many challenges on inferring the inter-domain link and congestion. The challenges arose because of the inconsistent numbering conventions as the router may have IP interface coming from third party ASes. More precisely the major challenges are i) identifying congestion on links with AQM and WFQ ii) proving the response from the far router returns over the targeted inter-domain links iii) ICMP queuing behavior[32].

On another related work, a lightweight single end active probing tool called pathneck was developed which is based on a probing technique known as Recursive Packet Train (RPT). This tool facilitates the end user to locate bottleneck links on the Internet efficiently. The key idea behind RPT is that it combines load packet and measurement packet in a single packet train. The load packets are queued at router interface and the trend of the packet train length is changed and the help of the measurement packets measures the change in the packet train length. In this way by measuring the packet train length, the location of congestion can be inferred. The result of the experiment suggests that more than half of the bottleneck locations were found in the Intra-AS link which is contrary to the widely believed assumption that bottlenecks often occurs at the edge of the network or at the boundary between the ASes. The stability of the Internet bottleneck was also investigated and found that intra-AS bottlenecks are more stable than inter-AS bottlenecks[22].

With the availability of pathneck to infer the bottleneck on the Internet, the detailed measurement studies were conducted on the Internet bottlenecks. The main four aspects of the Internet bottleneck were investigated. Firstly the persistence of the Internet bottleneck was checked; secondly, the sharing of the bottleneck among the destination cluster was examined.

Besides that the correlation of the bottlenecks with link loss and delay and the relationship to routing properties and link capacity including the router CPU and link capacity and traffic load were studied. The experiment revealed that 60% of the bottlenecks on loss paths could be correlated with a loss point no more than 2 hops away. There is no strong relation between bottleneck and the routing CPU, link capacity memory usage whereas the traffic load has strong relation with bottleneck occurrence on the internet.[21]
There have been done a lot of works on locating bottlenecks in the network. One of the approaches is locating last-mile downstream throughput bottlenecks. The main contribution of the paper was to identify whether the throughput bottlenecks lies inside the home networks or in their access ISPS. In order to facilitate the task, an algorithm was developed which finds out the throughput bottlenecks by monitoring traffic flows between home networks and access networks. The lightweight network metrics namely Packet Inter-arrival Time and TCP RTT were identified for the experiment. To validate the algorithm the experiment was conducted on 2652 home across the United States. The experiment revealed that wireless bottlenecks are more common than access-link bottlenecks when the downstream is greater than 20 Mbps. On the other hand, there is also access-link bottlenecks if the downstream speed is less than 10Mbps in conjunction with at least one device in a home network contributing to the throughput bottlenecks. There were some limitation of this project. The experiment is based on passive traffic analysis. It cannot detect the bottlenecks that are far away from the last-mile network. This is applicable only for finding downstream throughput bottlenecks and cannot detect the upstream throughput bottlenecks [3].
End-to-End delay is a very prominent performance metrics for studying and investigating network performance bottlenecks. Delay on the one bottleneck link can have a severe effect on the overall performance of the network. One of the research has been conducted to investigate the bottleneck delays and find the geographical distribution of the bottleneck links causing delay. The main contribution of the research is to identify the delays at the bottleneck links and study the delay feature on the internet which can be beneficial for designing the efficient distributed algorithms. In the project, the measured probing data has been deployed for conducting the statistical analysis of relationship between one-way delay and bottleneck delay. The experiment has demonstrated that bottleneck appears in the 70% of the paths on the Internet. Apart from this, for more analysis on bottleneck delay, the scheme which combines the IP centralised mapping with IP geographical mapping was proposed. In addition, that mapping scheme is handy to calculate link delay on the Internet and analyse the relationship between link delay and features of Internet links such as the structure of the internet and geographical distribution. The experiment has demonstrated that the links which had a greater number of entrances(in-degrees) but a smaller number of exit (out-degrees) or the average shallower links are the culprits for the bottleneck-delay and the two end of the bottleneck links are mainly distributed in the same country. The further more analysis

on the bottleneck links mapped in the same country has also revealed that the main cause of the delay in the bottleneck links is queuing delay.Thus, the paper has revealed how the structural properties of the Internet can make an impact on the transmission of the internet traffic and contribute to greater end-to-end delay [31].

Part II

The project

2.7 Overview of the project

The goal of this project is to examine congestion in the network which is limiting the performance of the network. The network is comprised of the core-network and edge. The general convention is that there is a problem at the edge which causes the performance degradation. So thesis will investigate if congestion usually happens in the last mile network or in the core as well. In order to locate congestion in the network, we have designed the experimental setup in the PlanetLab Testbed which is explained in detail in the coming section. The basic idea is to send the packets between nodes which lie on different domains and record Round Trip Time and also record the loss among those link. More precisely, we will form inter-domain links by picking up the nodes on the PlanetLab Testbed. We will attempt to maximize the number of the inter-domain link as far as possible and investigate if there is congestion on those links or not. We will attempt to find the reasons behind the congestion on these inter-domain links. The detailed explanation of the experimental design and relevant procedures and tools are explained in the respective sections.

Chapter 3

Experiments design and setup

Figure 3.1 represents a general overview of the experimental design where main building blocks of experiments are shown precisely. We have presented 3 components namely PlanetLab testbed, shell scripts and tools in 3 separate boxes as the main components of the experiment. The PlanetLab Testbed is used as Testbed for experiments and all available nodes of it will take part in the experiment. First of all available nodes are found out. After that, nodes are filtered such that they should belong to different cities and autonomous systems. The idea is to find the maximum number of the inter-domain links between nodes having most hops as far as possible. On this course, each node is assigned another 5 nodes that it will probe . Here the important assumption is that there should not be duplicate links just by interchanging senders and receiver role rather all links should be unique. A detailed description of selecting nodes and node pair is presented in Experiment details section below. All the scripts and tools that are devised for the experiment are supposed to run on the PlanetLab nodes. To automate operations on the PlanetLab nodes, shell scripts are required and therefore it is regarded as one of the building blocks of the experimental design. Basically, a master shell script is used to login to all nodes and prepare everything and copy the scripts and programming codes that are required to run the experiment. The other shell scripts run respectively after master scripts on respective probing and probed nodes to facilitate the automation over there. Few tools will be also used in the experiment which are shown in the box labeled as tools. One of the tool is the round-trip time calculating c programming code which sends the packets along with sequence number records the sending and receiving time of the packet and thus calculates the RTT of the packets. Traceroute is a handy tool to probe nodes and get RTT for each hop. High resource consumption such high usage of CPU and memory can sometimes result in an increased delay. So, to make sure that the larger RTT value is not the impact of the high resource consumption of the memory and CPU at the particular node. We are using the tool like top to keep track of the resource consumption at the PlanetLab node.

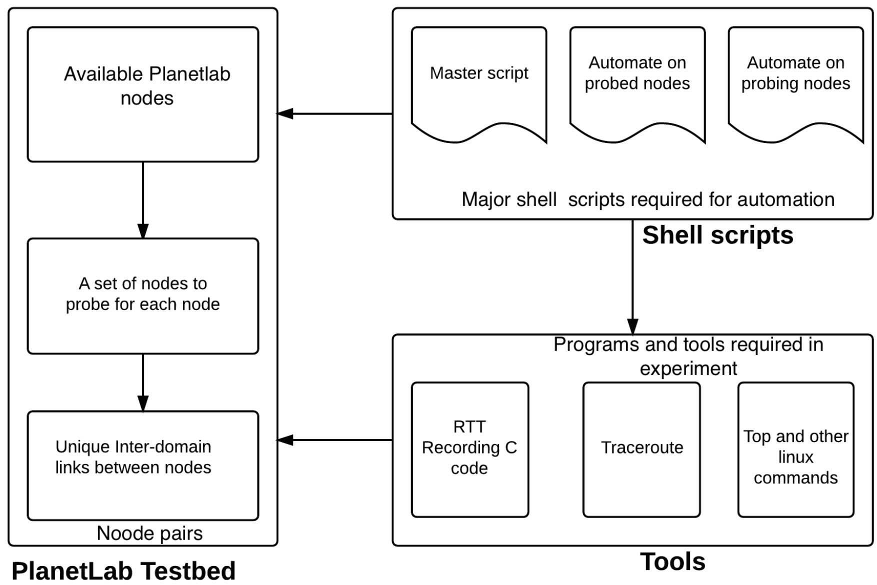


Figure 3.1: Overview of the component of experiment.

Figure 3.2 depicts the flow of the experiment more precisely. The systematic steps and the processes carried out during the experiment are displayed in the flow chart. In diagram two spots is shown separately. One is PlanetLab testbed and another one is the computer used to conduct the experiment and communicate with PlanetLab nodes and via which the automation is performed in the testbed. Moreover, in flow chart we depicted the interaction of the components mentioned in figure 3.1.

3.1 Description and Procedure of Experiment

In this section, we describe the Experiment thoroughly. A detail explanation of the entities involved in the experiment will be covered. In addition, we attempt to make the experiment more clear by explaining the experimental procedures as well.

3.1.1 Overview of the PlanetLab nodes involved in the Experiment

In the PlanetLab testbed, there are many nodes among them nodes were unreliable so we dropped them out. Besides that, some nodes have firewalls or some other functionalities which prevented us from reaching them. The best nodes that were selected for the experiment are listed in table 3.1. We selected 54 nodes where 23 nodes are from North America, 2 nodes are from Brazil, 20 nodes are from Europe and 9 nodes from Asia and Australia. The table highlights most relevant information about nodes such as geography along with the ISP and Autonomous system number.

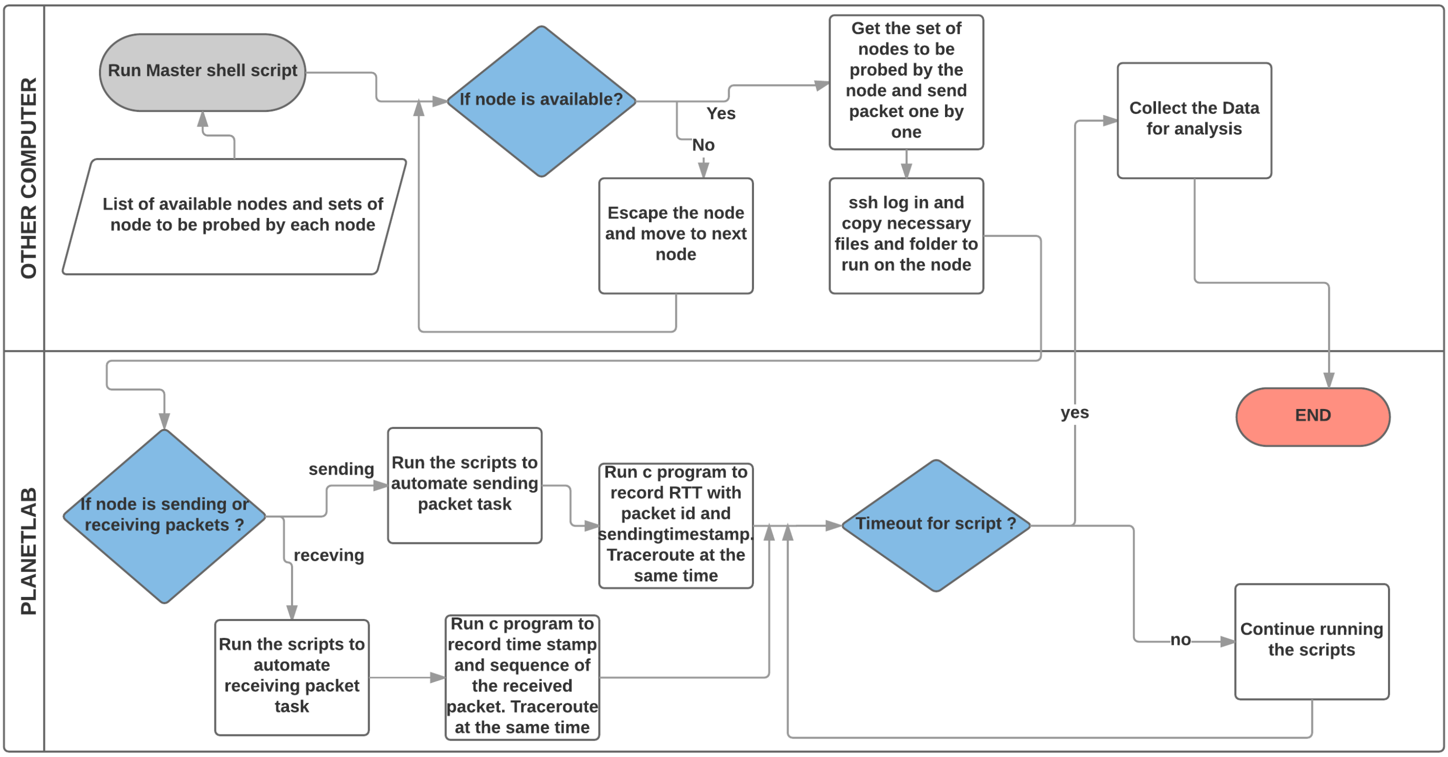


Figure 3.2: The Flow chart for Experiment.

3.1.2 Hardware and System Information

All nodes run Linux. Most of the machines have Fedora (Linux) and some of them also have CentOS. More precisely, CentOS release 6.4 (Final),CentOS release 6.8 (Final),Fedora release 14 (Laughlin),Fedora release 8 (Werewolf) Linux distribution are deployed on PlanetLab nodes. The nodes have different hardware, for example, they have different numbers of processors with varying number of CPU cores and capacity. Most of the processors use hyper threading functionality as well.The number of processors varies from 2 processors to 16 processors. Then the number of CPU cores in each processor varies from 2 CPU cores to 8 CPU cores. The capacity of CPU varies from 2.4GHz to 3.6GHz. Most of the nodes have 4GB of RAM. The disk quota on each node is 9.6GB however in some nodes it varies from several Gigabytes to Terabytes.

3.1.3 Experiments details

Selection of Nodes for experiment

In the PlanetLab website, we can see more than 300 nodes are available. However, the information is not up to date as most of the nodes are dead or unreachable. Therefore, the first step was to find all the nodes

SN	Nodes	ASN/Location	country
1	mars.planetlab.haw-hamburg.de	AS680 DFN Verein zur Foerderung eines Deutschen Forschungsnetzes	Germany
2	merkur.planetlab.haw-hamburg.de	AS680 DFN Verein zur Foerderung eines Deutschen Forschungsnetzes	Germany
3	node2.planetlab.mathcs.emory.edu	AS3512 Emory University	United States
4	pl1.cs.montana.edu	AS13476 Montana State University	United States
5	pl1.eng.monash.edu.au	AS56132 Monash University	Australia
6	pl1.ucs.indiana.edu	AS87 Indiana University	United States
7	pl2.6test.edu.cn	AS23910 China Next Generation Internet CERNET2	China
8	pl2.pku.edu.cn	AS4538 China Education and Research Network Center	China
9	pl2.ucs.indiana.edu	AS87 Indiana University	United States
10	plab1.cs.msu.ru	AS2848 MSU Vorobjovy Gory, Moscow, Russia	Russian Federation
11	planet-lab-node1.netgroup.uniroma2.it	AS137 ASGARR Consortium GARR	Italy
12	planet1.pnl.nitech.ac.jp	AS2907 Research Organization of Information and Systems, National	Japan
13	planet2.pnl.nitech.ac.jp	AS2907 Research Organization of Information and Systems, National	Japan
14	planetlab-2.cse.ohio-state.edu	AS159 The Ohio State University	United States
15	planetlab-5.eecs.cwru.edu	AS32666 Case Western Reserve University	United States
16	planetlab-coffee.ait.ie	AS1213 HEANET	Ireland
17	planetlab-js1.cert.org.cn	AS4134 Chinanet	China
18	planetlab-js2.cert.org.cn	AS4134 Chinanet	China
19	planetlab02.cs.washington.edu	AS73 University of Washington	United States
20	planetlab04.cs.washington.edu	AS73 University of Washington	United States
21	planetlab1.cesnet.cz	AS2852 CESNET2	Czech Republic
22	planetlab1.cs.du.edu	AS14041 University Corporation for Atmospheric Research	United States
23	planetlab1.cs.okstate.edu	AS5078 Oklahoma Network for Education Enrichment and	United States
24	planetlab1.cs.otago.ac.nz	AS38305 The University of Otago	New Zealand
25	planetlab1.dtc.umn.edu	AS57 University of Minnesota	United States
26	planetlab1.ifi.uio.no	AS224 UNINETT UNINETT, The Norwegian University and Research	Norway
27	planetlab1.net.in.tum.de	AS12816 MWN-AS	Germany
28	planetlab1.pop-mg.rnp.br	AS1916 Associacao Rede Nacional de Ensino e Pesquisa	Brazil
29	planetlab1.unr.edu	AS3851 Nevada System of Higher Education	United States
30	planetlab1.virtues.fi	AS47605 FNE-AS	Finland
31	planetlab2.cesnet.cz	AS2852 CESNET2	Czech Republic
32	planetlab2.citadel.edu	AS53257 The Citadel	United States
33	planetlab2.cs.cornell.edu	AS26 Cornell University	United States
34	planetlab2.cs.du.edu	AS14041 University Corporation for Atmospheric Research	United States
35	planetlab2.cs.otago.ac.nz	AS38305 The University of Otago	New Zealand
36	planetlab2.cs.ubc.ca	AS393249 University of British Columbia	Canada
37	planetlab2.cs.uoregon.edu	AS3582 University of Oregon	United States
38	planetlab2.inf.ethz.ch	AS559 SWITCH Peering requests: <peering@switch.ch>	Switzerland
39	planetlab2.pop-mg.rnp.br	AS1916 Associacao Rede Nacional de Ensino e Pesquisa	Brazil
40	planetlab2.rutgers.edu	AS46 Rutgers University	United States
41	planetlab2.tlm.unavarra.es	AS766 REDIRIS RedIRIS Autonomous System	Spain
42	planetlab2.utdallas.edu	AS20162 University of Texas at Dallas	United States
43	planetlab2.utt.fr	AS2200 Reseau National de telecommunications pour la Technologie	France
44	planetlab3.cesnet.cz	AS2852 CESNET2	Czech Republic
45	planetlab3.cs.uoregon.edu	AS3582 University of Oregon	United States
46	planetlab3.eecs.umich.edu	AS36375 University of Michigan	United States
47	planetlab3.inf.ethz.ch	AS559 SWITCH Peering requests: <peering@switch.ch>	Switzerland
48	planetlab3.mini.pw.edu.pl	AS12464 PW-NET	Poland
49	planetlab4.inf.ethz.ch	AS559 SWITCH Peering requests: <peering@switch.ch>	Switzerland
50	planetlab4.mini.pw.edu.pl	AS12464 PW-NET	Poland
51	planetlab5.eecs.umich.edu	AS36375 University of Michigan	United States
52	ple2.cesnet.cz	AS2852 CESNET2	Czech Republic
53	salt.planetlab.cs.umd.edu	AS27 University of Maryland	United States
54	stella.planetlab.ntua.gr	AS3323 NTUA	Greece

Table 3.1: List of PlanetLab nodes with location information

that are accessible. After getting a list of accessible nodes we checked the functionalities and programs that are required for running the experiment are available or not. If the program and service are lacking then we tried to install them manually. We tried to fix minor issues like repository errors, DNS error,etc. Thereafter we begin filtering the nodes by dropping the nodes which can not be maintained for running the experiment. In the process of selecting nodes, we got around 70 available nodes and after dropping the nodes which are unreliable. Thus,we end up with 54 suitable nodes for conducting experiments.

Selection of Node pairs

After getting a list of suitable PlanetLab nodes the next task is to generate the inter-domain links from them for each node. The task carried out by applying two algorithms shown in Algorithm1 and Algorithm2 respectively. Here, we have used a term inter-domain links for node pairs which are from different networks.

Algorithm 1: Select_best_nodes

Input: L = { l_1,l_2,............,l_n} as list of the all nodes
Result: B={b_1,b_2,...........,b_n } as list of Best nodes for each nodes in L

begin
  $B \leftarrow \emptyset$ `// Empty set initialisation`
  $i \leftarrow 1$ `// Loop iterator for selecting current node`
  while $l_i \in L$ **do**
    $b_i \leftarrow$ `get_best_node(`l_i, L`)`
    $B \leftarrow B \cup b_i$
    $i \leftarrow i + 1$
  Procedure `get_best_node(`l_i, L`)`
    $l_{(\text{Hop})i} \leftarrow$ `trace_route_all(`l_i, L`)` `// List of nodes with number of hop count`
    $l_{(\text{City\_AS})i} \leftarrow$ `get_city_and_AS(`$l_{(Hop)i}$, L`)` `// List of nodes with AS,City, hop count`
    $l_{(\text{Sorted\_City\_AS})i} \leftarrow$ `get_unique_AS_and_city(`$l_{(City\_AS)i}$, L`)`
    `// filtered and sorted by unique city and AS count`
    $l_{(\text{Final})i} \leftarrow$ `sort_nodes_by_hop_count(`$l_{(Sorted\_City\_AS)i}$`)`
    `// Final filtered and sorted by hop count list`
    return $l_{(Final)i}$

The detailed steps to follow to find the most suitable nodes to be picked up by a node is explained in Algorithm1. At the very first step, we used the list of suitable PlanetLab nodes as input. We selected all those nodes one by one as a current node and proceed to find out the list of nodes that fulfil the criteria of the experiment.Then the selected current node tracerouted all the nodes of the list L and record nodes with their corresponding hop counts from that node. Afterward, we found out the AS and city of those nodes and recorded in a list along with hop counts. Thereafter we filtered that list such that it contained only the nodes having different ASes and belonged to different cities. Eventually, we sorted the filtered list according to ascending order of the number of hop counts from that nodes and created a final list that is a list of best nodes for the current node. We repeated the process for all the nodes in List L in order to get a list of best nodes for them.
In Algorithm2 we describe all the steps going to be followed for the sake of obtaining the inter-domain links that were used in the experiment. First we gathered all the corresponding lists of best nodes for all nodes in List L. Then we had to determine how many nodes will a node be allowed to probe and up to how many links it will be involved into. We had to set limits for a node so it could form Inter-domain links in more appropriate way. There were two reasons behind this, one was we had only limited number of nodes so that we could get a limited number of combination of the nodes and another big reason was that we were selecting nodes for

Algorithm 2: Select_Inter-domain_links_per_node

Input: L = { l_1,l_2,.............,l_n} as list of the all nodes

Result: D={d_1,d_2,............,d_n } as list of Inter-domain links for each nodes in L

1 **begin**

2 $G_{list} \leftarrow \emptyset$ `// Initially Global list of links all node is empty`

3 $P_n \leftarrow$ `number_of_node_to_probe()` `// number of Inter-domain links per node`

4 $B \leftarrow$ *Select_best_nodes* `// list of the best nodes for each node from Algorithm2 such as B={`$b_1, b_2, \ldots\ldots\ldots\ldots, b_n$ `}`

5 $i \leftarrow 1$ `// Loop iterator for selecting current node`

6 **while** $l_i \in L$ **do**

7 **if** l_i *is First node* **then**

8 **for** $Inter - domain\_links\_count \leqslant P_n$ **do**

9 $d_i \leftarrow$ `add_node_to_list_as_link(`l_i, b_i`)` `// Iink such as` $l_i \rightarrow b_i$ `elements(ascending order of number of hop count)`

10 **else**

11 $d_i \leftarrow$ `get_inter-domain_link(`b_i, G_{list}, l_i `)`

12 $G_{list} \leftarrow$ `add_to_global_list(`d_i`)`

13 $D \leftarrow D \cup d_i$

14 $i \leftarrow i + 1$

15 **Procedure** `get_inter-domain_link(`b_i, G_{list}, l_i`)`

16 $n_p \leftarrow$ `max_links_involved_by_node()` `// maximum number of links can a node be involved in`

17 $T_{list} \leftarrow \emptyset$ `// Temporary list to save links for a node set as empty`

18 **for** $Inter - domain\_links\_count \leqslant P_n$ **do**

19 `check_route_exists_in_global_list(`b_i, G_{list}, l_i`)`

20 **if** *route exists* **then**

21 go to next element in b_i

22 **else**

23 $count \leftarrow$ `node_presence_count(`b_i, G_{list}, l_i`)` `// check how many times node is present in global list`

24 **if** $count \leqslant n_p$ **then**

25 $T_{list} \leftarrow$ `add_node_to_list_as_link(`l_i, b_i`)` `// add node to Temporary list like line 9`

26 **else**

27 go to the next element in b_i

28 **return** $T_{(list)i}$

inter-domain links from the lists where node were sorted according to hop counts. Hence for each node, we tried to find the farthest nodes as far as possible. Therefore, if we did not limit the presence of the node on the process farthest node will be probed several times and closet nodes will be probed rarely. Then we began to find inter-domain links for each node sequentially. If a node currently being processed was a first node of the list then we simply added nodes maintaining limitation from its list of best nodes to the list of its inter-domain links. Subsequently, we added to the global list which will be used to check for the presence of duplicate links for coming up nodes. For other nodes, we need to check in the global list of inter-domain links for the avoidance of duplicate links and at the same time, we would be checking how many the nodes were involved in forming the links as well. Thereafter, if the link satisfied both criteria, we added it to the list of the corresponding node and then to the global list respectively. This process was repeated util we got the list of inter-domain links for all the nodes in the set L.

Development of Programs and Scripts

To measure Latency, we designed a C program that sends packets to the list of the nodes and receives back from them. To measure the Round Trip Time the sending and receiving timestamp were logged in separate files. Similarly, we designed the shell scripts that monitors the load in the system and traceroutes from sender to receiver and vice versa. In addition, a python script was created to calculate the RTT and loss between links. A separate script to collect the data every day via cronjob was prepared. Besides that reformatting and arranging data was done via other scripts as well. In general, we created several scripts for individual purpose and combined all to 2 parts one for running experiments and collecting data and other for rearranging and reformatting the collected data as per requirements

Running experiment

we set up a separate Laptop computer for running the experiments and collecting data. The experiment was run as a cronjob which runs every week. The experiment was arranged to run for 3 weeks.
The experiments began with the main shell scripts which prepared all the requirement for running other programs by installing required services and copying all the relevant file to the respective nodes. Basically, we could divide the experimental task into 3 different tasks as following.
Probing nodes with packets : We ran a C code to probe node by sending packets from one end of the link to the node on the another end. One node acted as a sender and sent the packet to receiving node meanwhile another C code was running on the receiving side to receive packets and send back the same packet to the sender. We created two separate thread for sending the packet and receiving packet so that the sending and receiving task are independent and do not interfere with each other. We had also set

the sending rate such as we sent the packet every 200ms. We logged the timestamp and sequence number of the packet on both the sending and the receiving end of the link.
Tracerouting the nodes: While we were probing nodes by sending the packets, we were also tracerouting the nodes. More precisely, we tracerouted in a two-way fashion from sender to receiver and receiver to sender at the same time. We logged the timestamp when we tracerouted and output of traceroute.
Monitoring resource and the load in the node: During the experiment, we also kept track of resource consumption and load on the nodes. We used top command to check the CPU utilization, memory consumption, I/O waiting and average load on the node. The main purpose of the monitoring the node was to confirm that if there is congestion on the particular link then the load and resource on the node are not culprits on that context.

Data Collection

The data was collected every day from remote PlanetLab nodes to a local laptop by running a script as a cronjob. We collected data every day because of two reasons 1)We could use data every day for analysis and did not need to wait for the experiment to be completed. 2) The disk on the remote server has a limited quota and we can get rid of disk quota exceeded problem.

The detail steps involved for collecting the results from remote server to local computer is mentioned on Algorithm 3. First of the file to be collected is identified thereafter those are located.The located files are copied to new files respectively so that we do not loose the data in between the copying process. Then the files are compressed and sent to the local computer. If the files are successfully transferred, then we just delete them in order to maintain free disk space at the nodes.

Data Rearrangement

After completion of data collection task, we need to arrange the data in more appropriate away for future access. First of all, we uncompress all the data and then we selected the desired file. Afterward, we merged the corresponding files into a single file.Thereafter we saved those files to a new path in such a way we can recognise the files belongs to which links and in which direction of the links. The figure 3.3 below gives a more clear image about this. All the nodes that have been probed are put under probed_nodes folder along with all the log files. The nodes which probed a node are put under probing_nodes folder along with all corresponding log files.

Algorithm 3: Collect_data_Every day

Input: L = { l_1,l_2,............,l_n} as list of the all nodes
Result: R={r_1,r_2,...........,r_n } as zipped data from each nodes in L

begin
  $R \leftarrow \emptyset$ `// Empty set initialisation`
  $File = \{sending\_packet\_log, receivining\_packet\_log, traceroute\_log, top\_log\}$
  `// list of files to collect`
  $i \leftarrow 1$ `// Loop iterator for selecting current node`
  while $l_i \in L$ **do**
    $r_i \leftarrow$ `get_data_from_remotenode(`l_i, *File*`)`
    $R \leftarrow R \cup r_i$
    $i \leftarrow i + 1$
  Procedure `get_data_from_remotenode(`l_i, *File*`)`
    $File_{\text{rotation}} = \{\text{sending\_packet\_log}_{\text{new}}, \text{receivining\_packet\_log}_{\text{new}}, \text{traceroute\_log}_{\text{new}}, \text{top\_log}_{\text{new}}\}$
    `// original files rotated to new log files`
    `locate_required_logfile(` *File*`)`
    $File_{\text{rotation}} \leftarrow$ `rotate_located_file(`*File*`)` `// rotate file to new supplied file names respectively`
    $File_{\text{compressed}} \leftarrow$ `compress_rotated_files(`$File_{rotation}$`)`
    `// compress files after log rotation`
    return $File_{compressed}$
    `delete_compressed_file(`$File_{compressed}$`)`

```
.
└── mars.planetlab.haw-hamburg.de
    ├── probed_nodes
    │   ├── pl2.6test.edu.cn
    │   │   ├── final_receiving_information.txt
    │   │   ├── final_sending_information.txt
    │   │   ├── loss.txt
    │   │   ├── one_way_loss_client.txt
    │   │   ├── one_way_loss_server.txt
    │   │   └── rtt.txt
    │   ├── planetlab1.cs.okstate.edu
    │   │   ├── final_receiving_information.txt
    │   │   ├── final_sending_information.txt
    │   │   ├── loss.txt
    │   │   ├── one_way_loss_client.txt
    │   │   ├── one_way_loss_server.txt
    │   │   └── rtt.txt
    │   ├── planetlab2.cs.ubc.ca
    │   │   ├── final_receiving_information.txt
    │   │   ├── final_sending_information.txt
    │   │   ├── loss.txt
    │   │   ├── one_way_loss_client.txt
    │   │   ├── one_way_loss_server.txt
    │   │   └── rtt.txt
    │   └── planetlab2.inf.ethz.ch
    │       ├── final_receiving_information.txt
    │       ├── final_sending_information.txt
    │       ├── loss.txt
    │       ├── one_way_loss_client.txt
    │       ├── one_way_loss_server.txt
    │       └── rtt.txt
    └── probing_nodes
        ├── pl1.ucs.indiana.edu
        │   └── final_receiving_information.txt
        └── planetlab2.cs.otago.ac.nz
            └── final_receiving_information.txt
```

Figure 3.3: Tree view of File arrangement

Calculation of Metrics

We calculated the latency by computing Round Trip Time (RTT) for each packet. The RTT was computed in microseconds first and then converted to milliseconds. Besides that, we also calculated loss as another metrics. The loss can be 1)Two-way loss 2) one-way loss from sender to receiver 3) one-way loss from receiver to sender.

Part III

Analysis and Results

Chapter 4

Latency Analysis

First of all, we begin with the analysis on the basis of Round Trip Time measured on the various links and find the congested inter-domain links among them. In this part, we will cover the detail explanation on the latency analysis and the results obtained in the respective sections.

4.1 Classification of Datasets

According to the experimental design, we had more than 200 node pairs from the combination of 54 nodes from all the continents. The very first approach was, to narrow down the analysis process by dividing the node pairs to meaningful sets. Most of the nodes were from North America and Europe. Few nodes from Asia and very few nodes from Australia, Oceania, and South America. In order to make suitable sets, we put South American nodes to North American set and made a new set as an America. Similarly, we put nodes from Australia and NewZealand to the set Asia. Thus, ultimately we classified the nodes to three sets as shown in the figure below. After dividing the nodes into sets, we made new sets of a combination of those sets. We grouped the node to node links according to the continent to continent sets.Hence we divided ultimately to 9 on the basis of the continents as shown in the figure below. The detailed information of the sets is included in the appendix section.

Continent-set	Number of nodes
America	25
Europe	20
Asia	9

Continent to continent combination	Number of inter-domain links
America to America	17
America to Europe	57
America to Asia	18
Europe to Asia	23
Europe to Europe	15
Europe to America	42
Asia to America	8
Asia to Europe	18
Asia to Asia	8

Figure 4.1: The continent to continent sets and node pairs involved

4.2 Creating Time series data

First, the whole data was binned by an hour bucket so that we have 24 buckets for respective hours. All the data were put into a relevant bin and computed statistics such as mean, median and percentile in order to reduce the minor observations.In this way, we obtained time series data for all the inter-domain links for plotting latency trend over the each hour of a day.

4.3 Latency Trend over time

After we converted all the data to time series data, we visualized the RTT trend over the time for the all node pairs of the respective sets. We plotted severals graph and on the basis of computed statistics such as mean, median and percentiles. we came to a point that the percentile five statistics had a more consistent pattern. The results presented on sections below are based on percentile five of RTT over the per hour time in GMT. We have divided the latency analysis into the sections according to Continent to Continent combinations obtained from the classification of data sets. For the sets having few nodes, we did not use local time zone. For bigger sets with varieties of nodes having geographical diversity and time zone variance, we also use the local time zone analysis to make analysis convenient and meaningful.

4.3.1 Latency analysis on links from Asia to other continents

we have few nodes which belong to Asia and some links from Japan to other continents were down during the course of the experiment. We do not have enough elements to further classify those sets with respect to the local time zone. We simply visualize the links where the nodes of from Asia, Australia, and NewZealand probed to nodes from another continent.

The aim is to find the hours when the links have peak RTT values and how the RTT varies over 24 hours. The plots below depicts the RTT trend of the links over time from Asia to other continents.

Figure 4.2 pictures the RTT trend of the links from Asia to Europe. The nodes that are probing to another node are mostly from China and very few nodes are from Australia and NewZealand. The nodes probed in Europe are distributed all over Europe.One thing clearly noticed in the graph is that many links have a very high value of the RTT. Different links depicted different RTT trend over 24 hours time. However, we found the hours when the links have high RTT values. In the links planetlab-js1.cert.org.cn -> stella.planetlab.ntua.gr, pl1.eng.monash.edu.au ->stella.planetlab.ntua.gr, and planetlab-js1.cert.org.cn ->planetlab1.ifi.uio.no, we can point that the links experience RTT fluctuation at hour 5 and also between hours 12 and 16.

Figure 4.3 represents the RTT trend of the links from Asia to America. During experiment some links were not accessible because the participating nodes were down as matter of fact we got only 3 links to analyze. Here the nodes from china are probing to nodes in Brazil and the US. In the context of the link between china and the US, we did not observe meaningful variance on RTT trend. In the link from china to Brazil we noticed two RTT spikes at hour 4 and hour 14.

Figure 4.4 depicts the RTT trend of the links from Asia to Asia set. The nodes that are probing to another node are from China and a node is from Australia. The nodes being are probed belongs to China, Australia, and NewZealand. Here also the nodes from Japan to another country were not accessible because nodes in Japan went down during the experiment as a consequence we have fewer links to analyse. From the insight to the graph we can notice that most of the links don’t show the variation in RTT over time. One link from Australia to China has variation on RTT trend over time. The RTT values rises gradually and reaches at peak at hour 14 and after hour 14 RTT values decreases gradually to a lowest value.

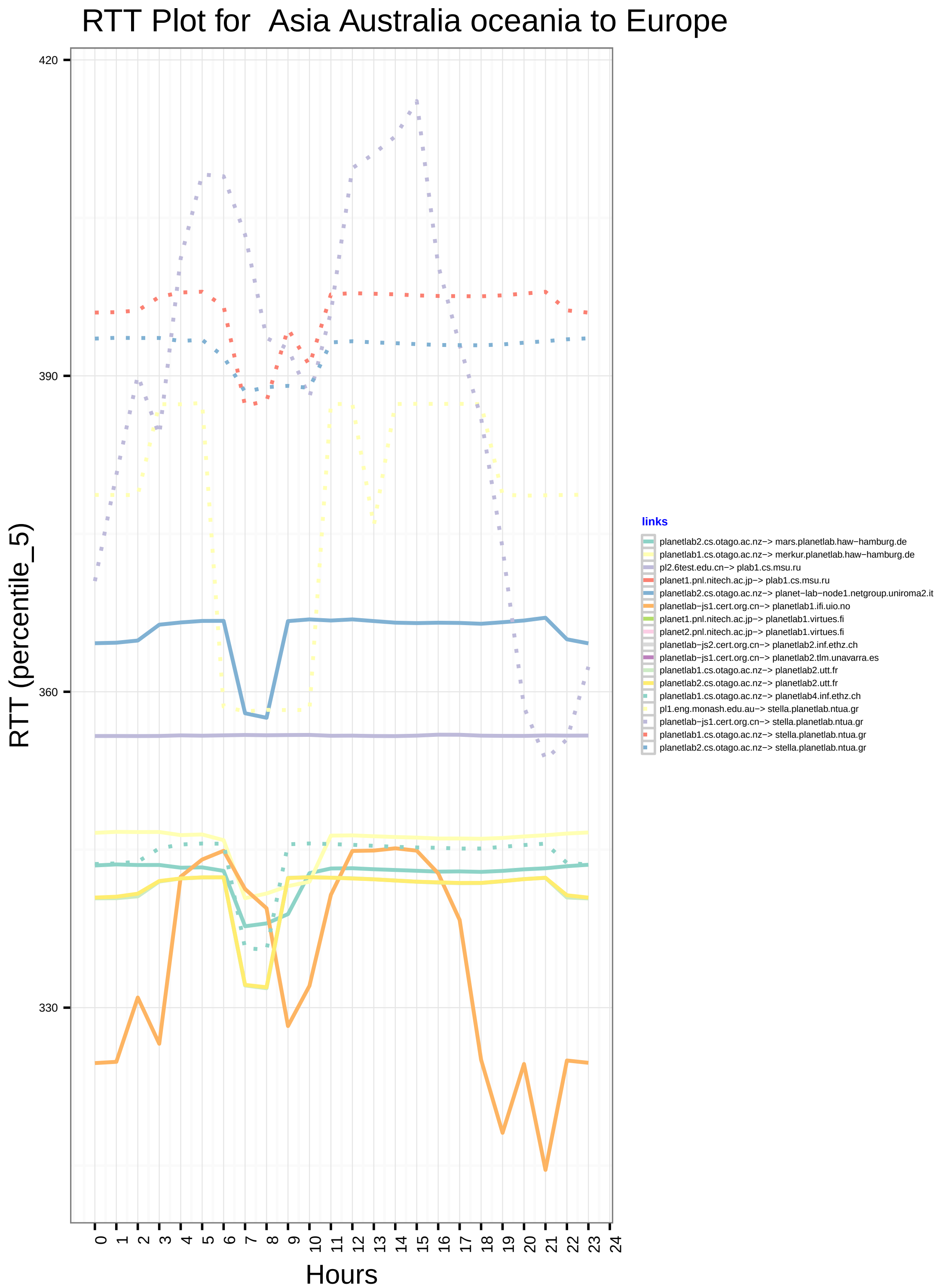


Figure 4.2: Latency trend over hours on the links from Asia to Europe Set

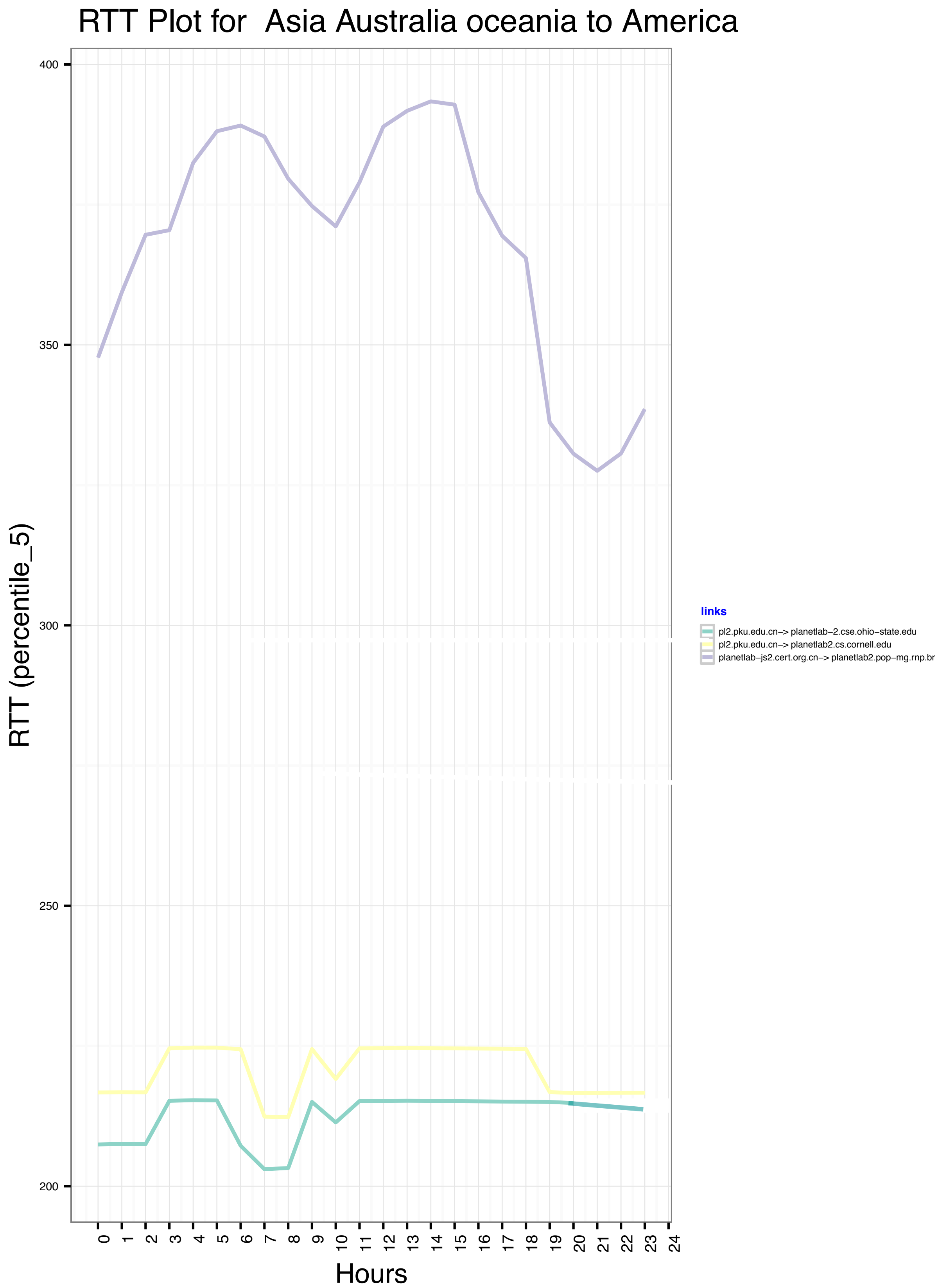


Figure 4.3: Latency trend over hours on the links from Asia to Europe Set

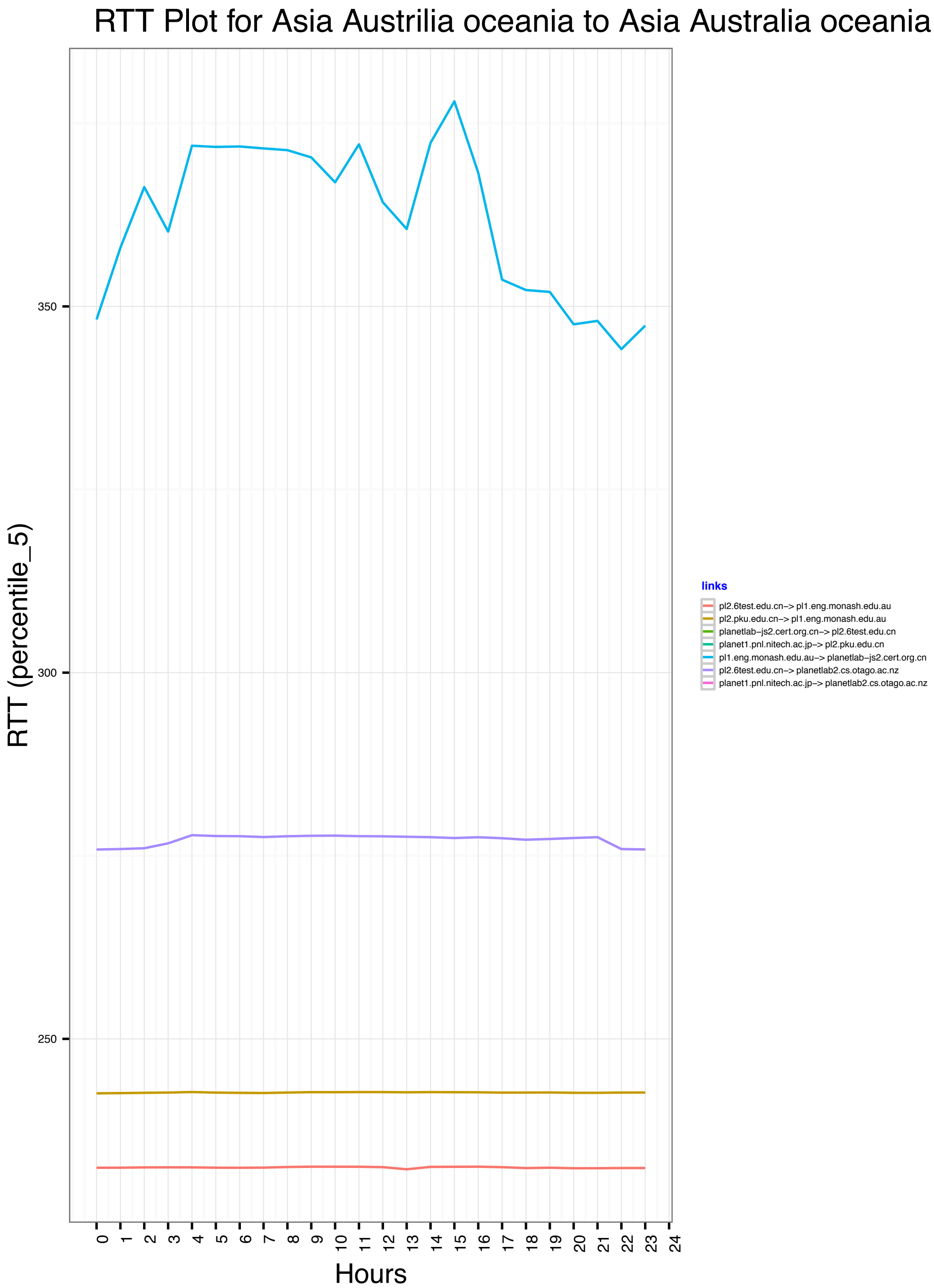


Figure 4.4: Latency trend over hours on the links from Asia to Europe Set

4.3.2 Latency Analysis on Links from America to Europe and vice versa

latency Trend over time within Local time zone

As a matter of the fact, the nodes in the US are distributed in such way that the time difference is varying with high degree and having several local time zones, the local time issue clearly rises while analyzing links where probing nodes are located in the US. Another problem with the continent to continent sets was that the set holding a lot of numbers of links was also difficult to analyze by visualizing the data. For the sake of making the analysis process simpler and more effective, we classified the nodes by local time zone and redesign the sets according to the local time zone of nodes involved. Actually, the time zone classification doesn't suit for sets with few nodes like Asia to America, Asia to Europe, Asia to Asia whereas for the continent to continent sets such as America to Europe and Europe to America which has sufficient number of links and time zone variance,it is an appropriate method of classification. Hence we redesigned the continent to continent sets from America to Europe and Europe to America as shown in figure 4.5 and figure 4.6 respectively.

Time zone to Time zone	Number of links
Eastern Daylight Time (EDT) to Central European Summer Time (CEST)	16
Eastern Daylight Time (EDT) to Eastern European Summer Time (EEST)	6
Mountain Daylight Time (MDT) to Eastern European Summer Time (EEST)	1
Mountain Daylight Time (MDT) to Central European Summer Time (CEST)	9
Pacific Daylight Time (PDT) to Eastern European Summer Time (EEST)	5
Pacific Daylight Time (PDT) to Central European Summer Time (CEST)	9
Central Daylight Time (CDT) to Central European Summer Time (CEST)	2
Central Daylight Time (CDT) to Eastern European Summer Time (EEST)	6

Figure 4.5: Classification of links in America to Europe by local time zone

The detail information about those time zone classified sets are included in appendix section.

Time zone to Time zone	Number of links
Central European Summer Time (CEST) to Eastern Daylight Time (EDT)	11
Central European Summer Time (CEST) to Mountain Daylight Time (MDT)	3
Central European Summer Time (CEST) to Pacific Daylight Time (PDT)	10
Central European Summer Time (CEST) to Central Daylight Time (CDT)	6
Eastern European Summer Time (EEST) to Eastern Daylight Time (EDT)	3
Eastern European Summer Time (EEST) to Pacific Daylight Time (PDT)	2

Figure 4.6: Classification of links in Europe to america by local time zone

After the classification of the links according to local time zone, we started to analyze the latency fluctuation for the links from America to Europe and Europe to America in the respective sections as followings.

Latency analysis America to Europe

In this part, we discuss latency analysis on the links from America to Europe. The main aim is to see the RTT pattern of the links that belongs to same local time zone over per hour time of a day. We will present some graph and pinpoint the hours where RTT values are high and also observe the variation on RTT over time series.

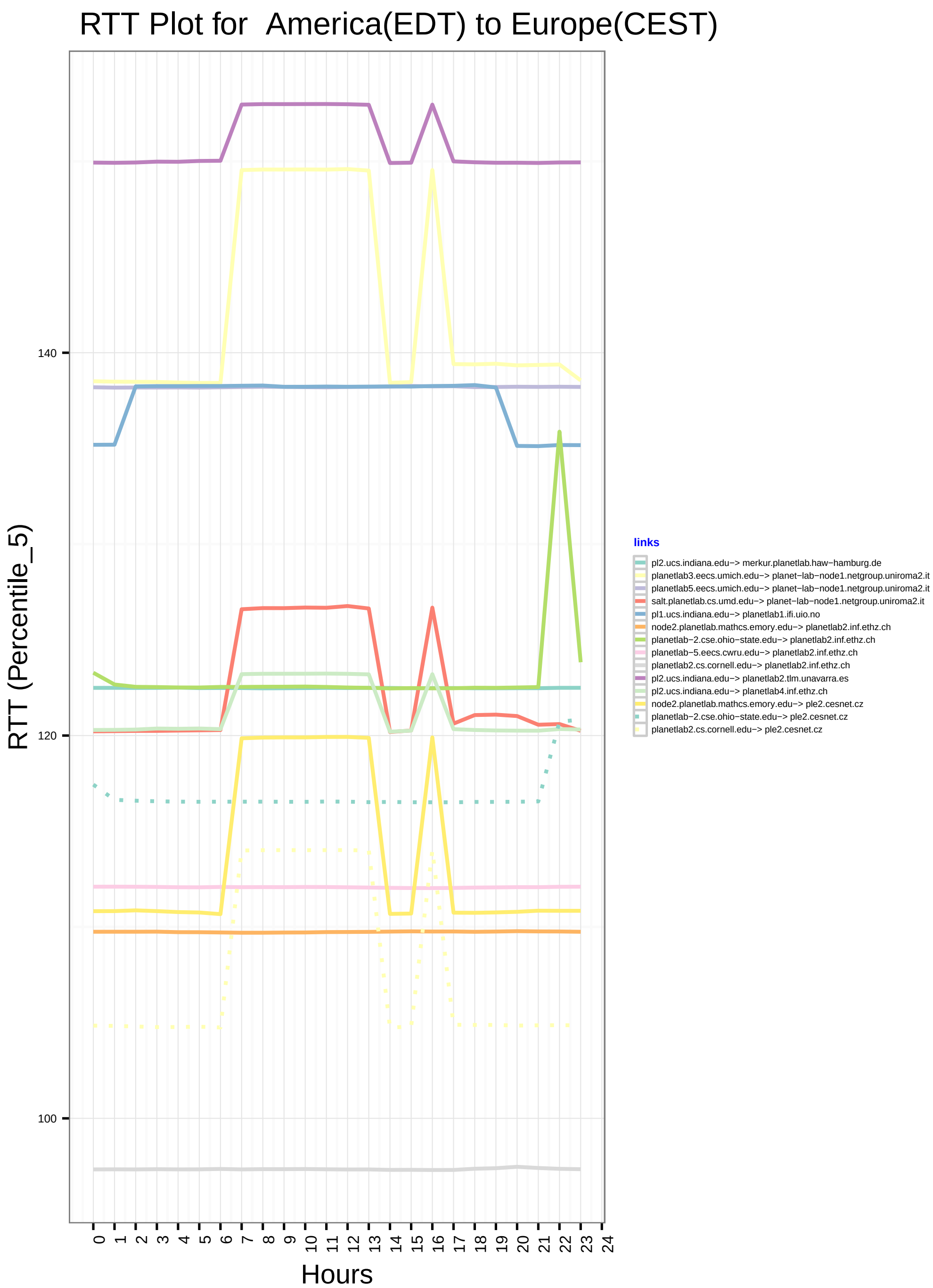


Figure 4.7: RTT Trend of links between Eastern US to Central Europe

Figure 4.7 depicts the RTT trend of the links where the nodes in the US having Eastern Day Time as local time are probing the nodes in Europe having Central European Summer Time as local time. Here we noticed that the RTT value increases after hour 7 and continues until hour12.

We again observe a spike in the graph at hour 16. Most of the links seem to be congested on during these periods. In the link planetlab-2.cse.ohio-state.edu -> planetlab2.inf.ethz.ch, we found that the spike at hour 22.

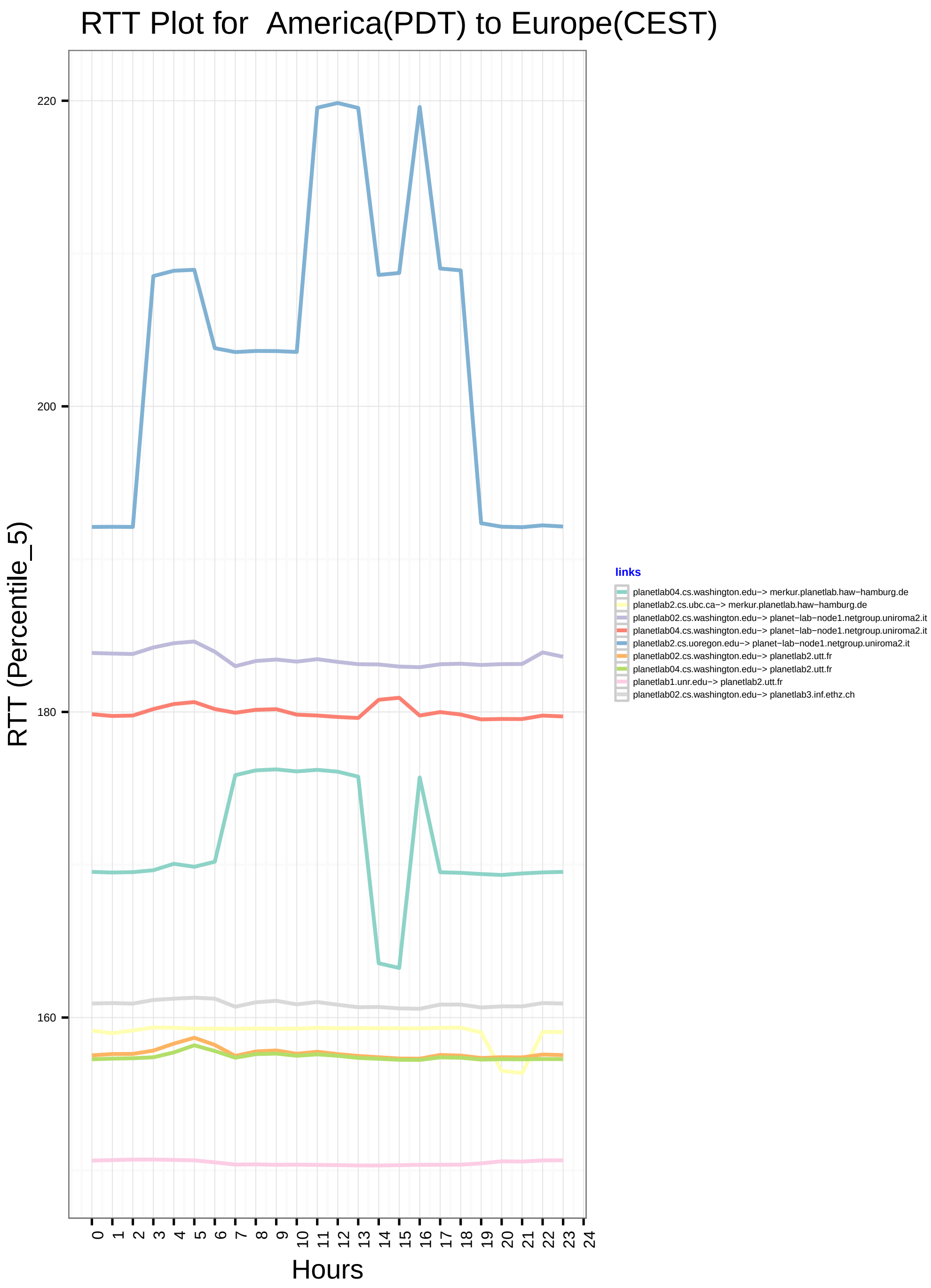


Figure 4.8: RTT Trend of links between Western US to Central Europe

In figure 4.8 we observed most of the node pairs do show fluctuation whereas some node pairs have a fluctuation in RTT. In the link planetlab04.washington.edu >merkur.planetlab.haw-hamburg.de we saw that RTT values rise from hour 7 until hour 12 and again we saw the peak at hour 16. Another link from Washington to Rome also depicted RTT peak value during hours 5, 7, 16 respectively.

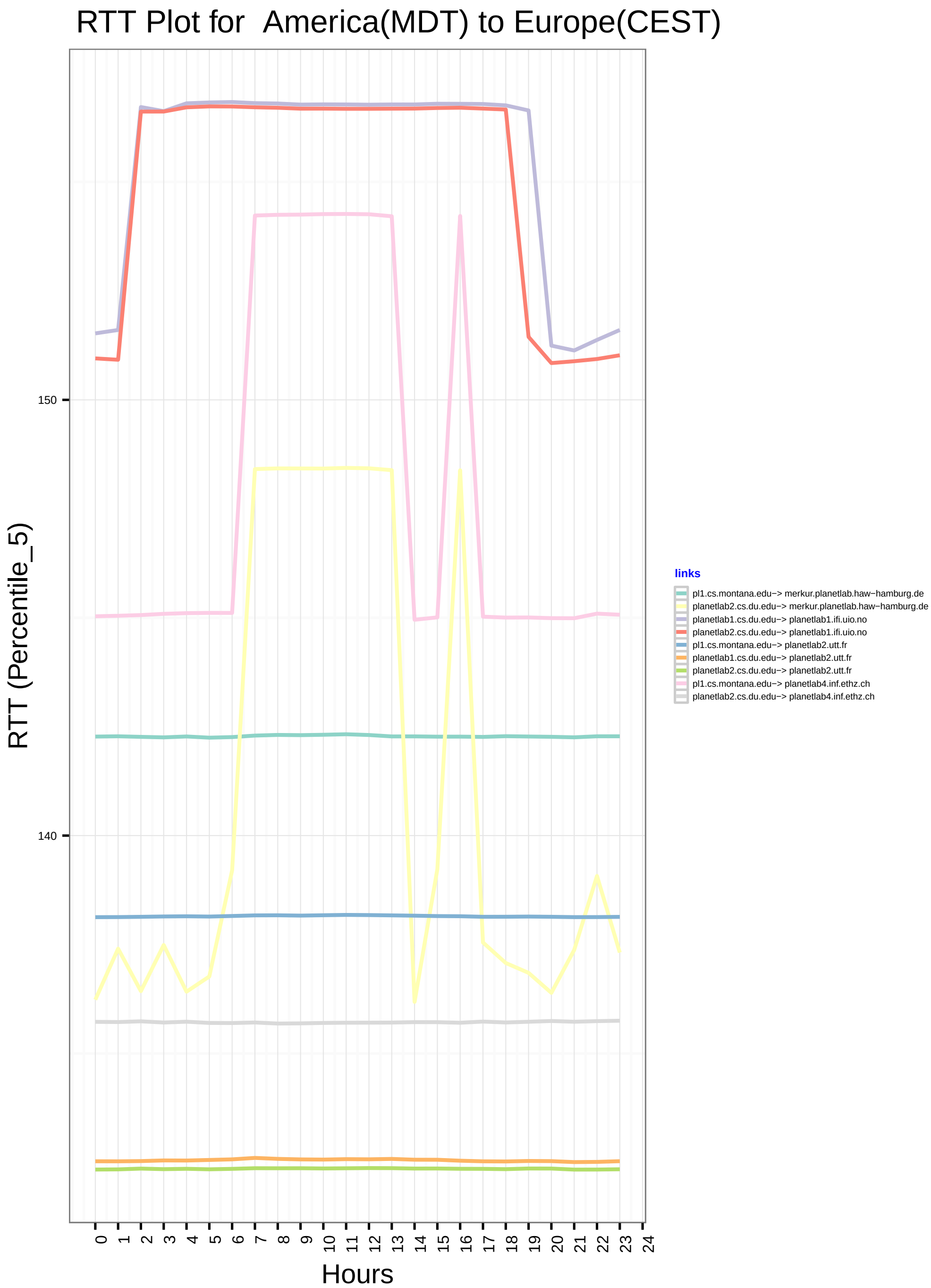


Figure 4.9: RTT Trend of links between Pacific Day Time zone in US to Central Europe

In figure 4.9 we can notice that several nodes have no variation on RTT over per hour time. However some of the links depicted peak RTT value during hours 7 and 12. The another spike appears at hour 16 as well.

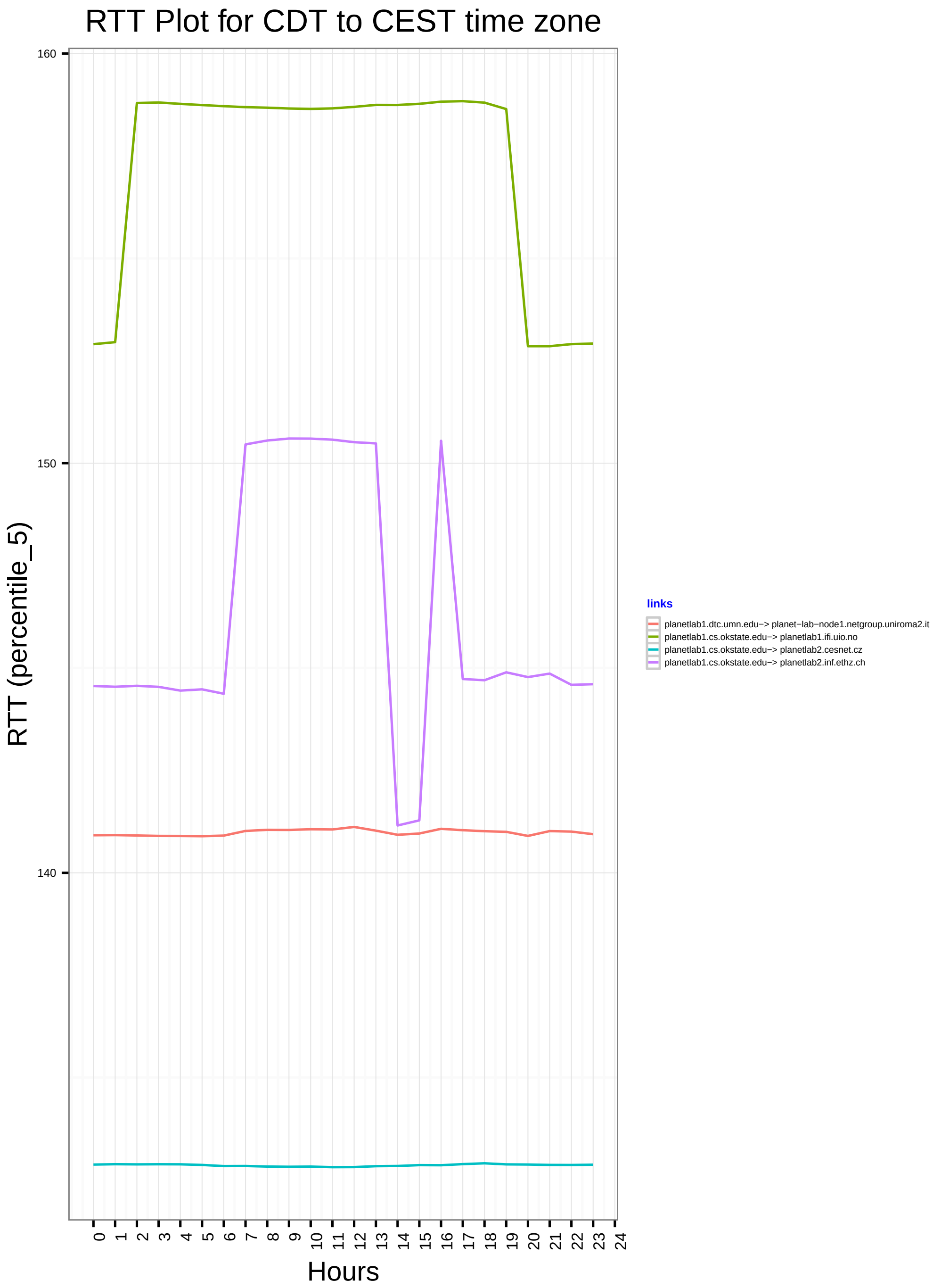


Figure 4.10: RTT Trend of links between Central Day Time zone in US to Central Europe

Figure 4.10 shows the RTT variation over time between the links from Central US to Europe. Here we observed only one node from Oklahoma to Switzerland has RTT variation over per hour time. The fluctuation are observed between period hour 7 to 12 and at hour 16 as well like previous links.

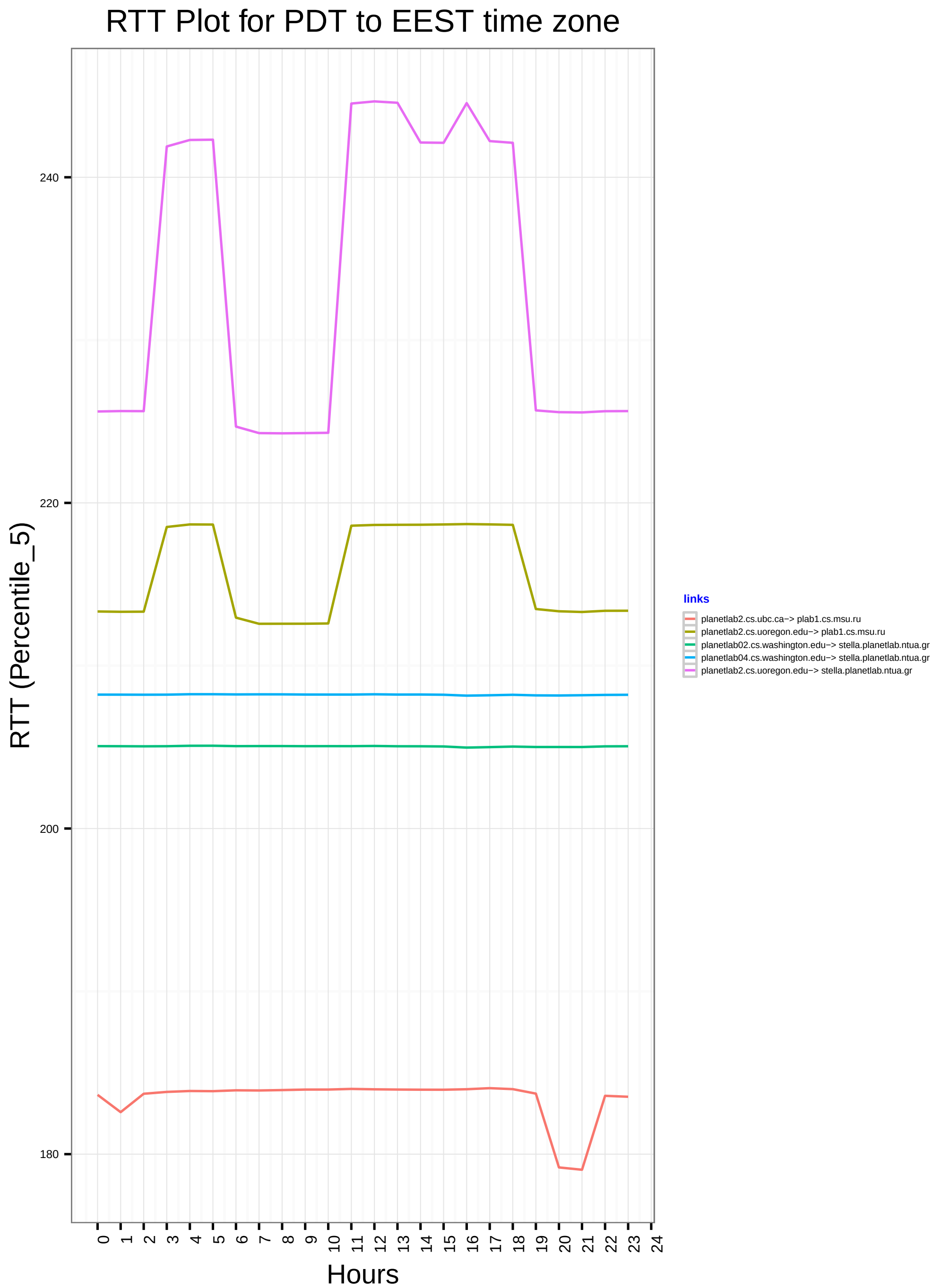


Figure 4.11: RTT Trend of links between Pacific Day Time zone in the US to Eastern Europe

Figure 4.11 represent the RTT pattern over time between links from Pacific Day Time in the US to Eastern Europe. There were few links in Eastern Europe as a matter of the fact we have fewer items for analysis on that zone.

The links in another time zone of the US did not reflect many variations in RTT observations. However, we noticed slight fluctuation on RTT on two links at hour 4 and period between hour 12 and 17.

Latency analysis of links from Europe to America

In this section, we present the result obtained from analyzing the links from Europe to America. We visualize the RTT trend of the links having a common local time zone and pinpoint the hours where the link has high RTT value. We plotted the graph for all the sets based on the local time zone mentioned in the above section. We will explain most relevant result among them as follows. Figure 4.12 shows the RTT trend of the links from Central European Summer Time to Central Day Time in the US. The insight to the figure revealed that most of the links have same RTT trend over the time series while some links remained non-fluctuated. We also noticed that several nodes from Europe probing to only 2 nodes in the US and observed pattern resembles for all with the only difference in the RTT values. The RTT values started to rise at hour 7 and become steady until hour 12. After that RTT value comes to original trend and again all of sudden the RTT value peaks at hour 16.
Figure 4.13 depicts the RTT variation on the links from Central Europe Summer Time to Pacific Day Time in the US. In the figure, we observed that most of the links have a similar pattern. We can see that RTT values don not fluctuate except between hours 20 and 21 where there is a slight drop in RTT. Apart from this we still noticed that in the link between planetlab3.cesnet.cz and planetlab04.cs.washington.edu there is fluctuation in RTT. The RTT values peak at hour 7 and continues until hour 12. After hour 12 the RTT value catches the normal value and all of sudden the RTT value peaks again at hour 16.
We also performed the analysis on the links from Europe to Eastern Day Time and Mountain Day Time. On the process of visualizing the RTT trend over the time, we did not notice much fluctuation. For instance, we presented the scenario in figure 4.14 where we see that there is no variation in RTT over time series.

4.3.3 Latency analysis of links from Europe to Asia

In this section, we present the result of analysis on links from Europe to Asia. First, we visualize all links in from Europe to Asia, Australia and, NewZealand as whole in one. After this, we visualize the RTT fluctuation on the links from Europe with local time zone as Central European Summer Time to China separately. In that context,we visualize the RTT trend while taking local time zone on consideration since China exercises only one time zone as China Standard Time.

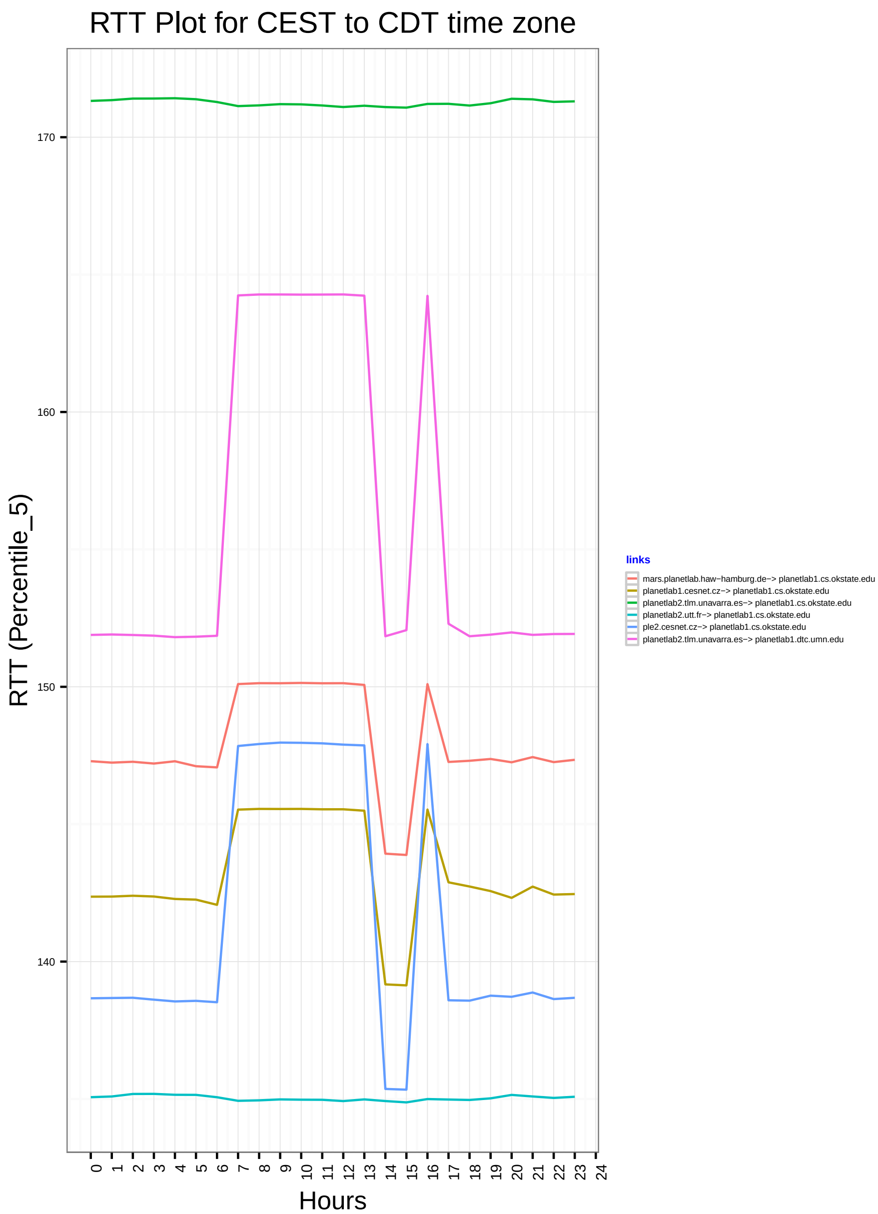


Figure 4.12: RTT Trend of links between Central Europe and Central US

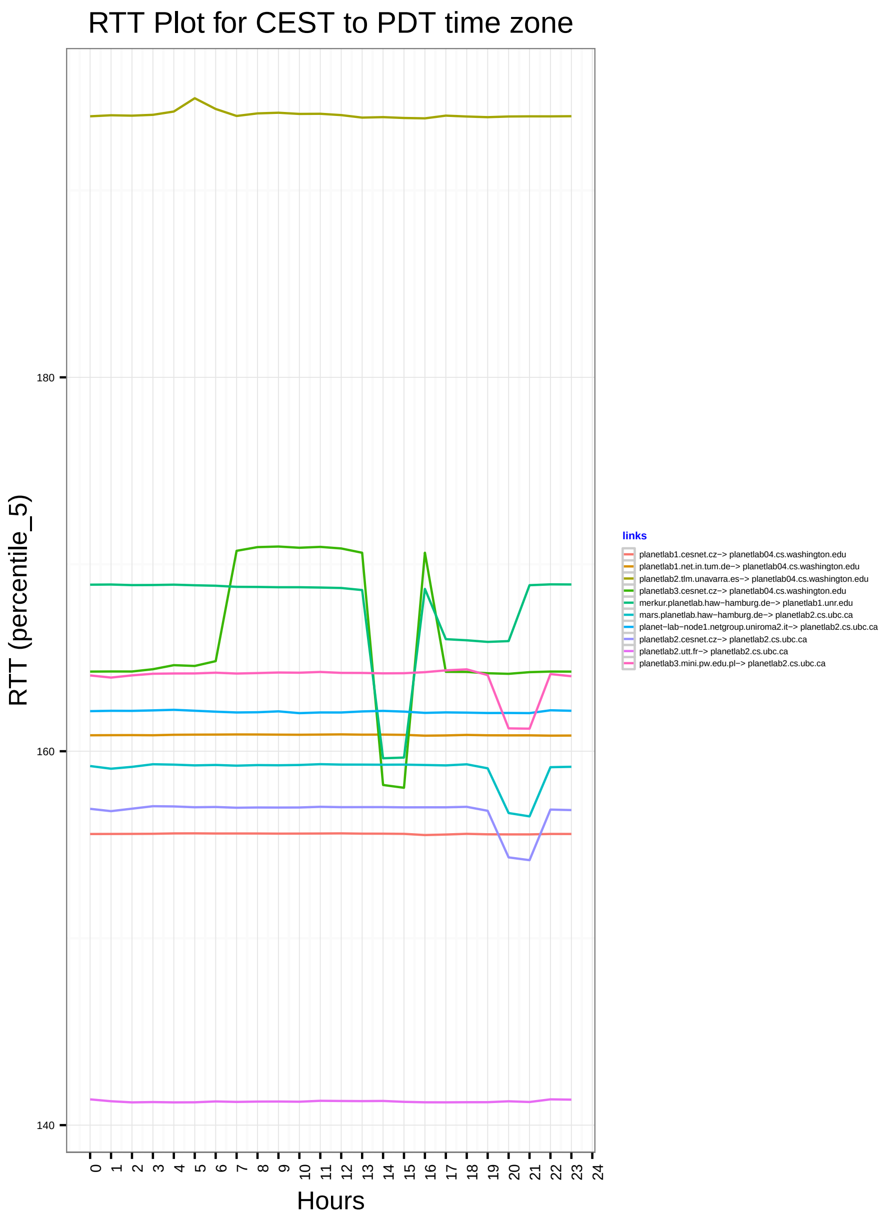


Figure 4.13: RTT Trend of links between Europe to Pacific Day Time USA

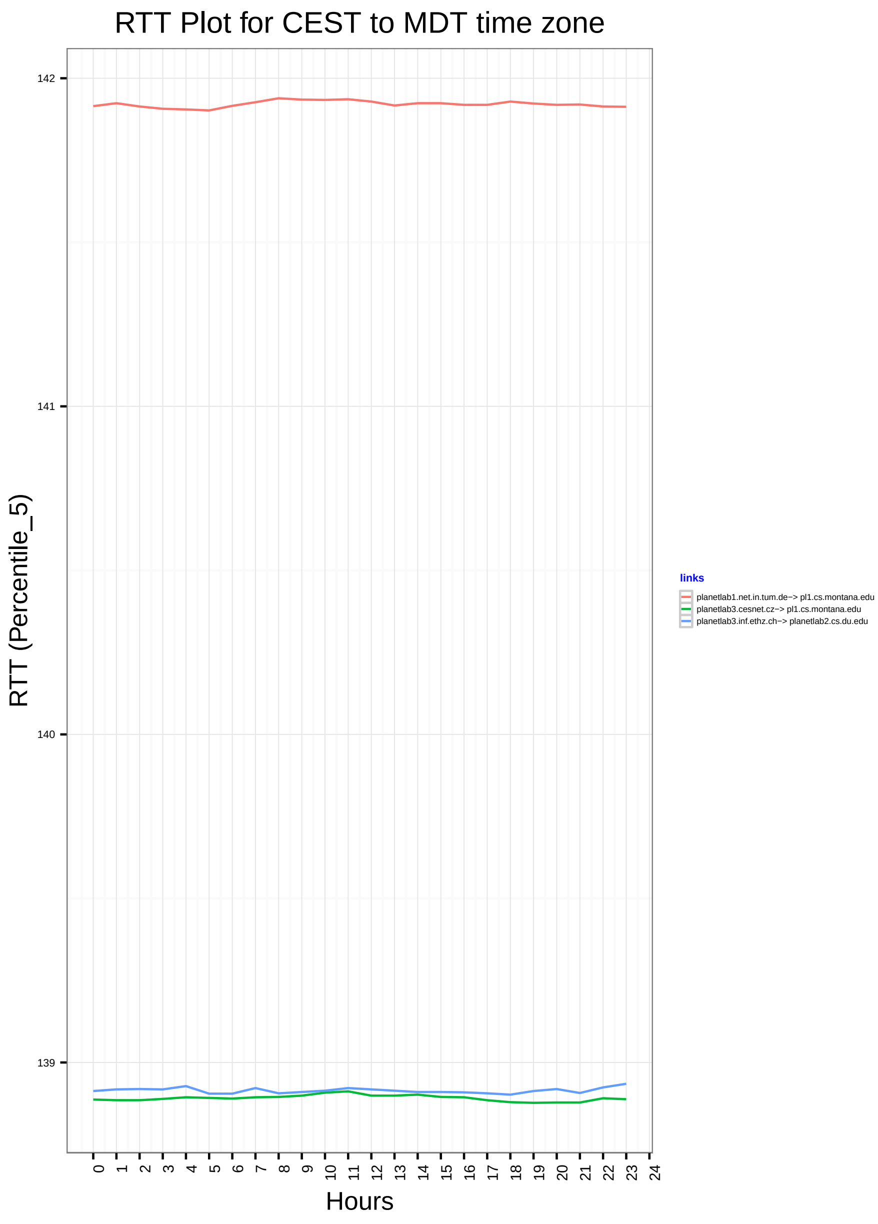


Figure 4.14: RTT Trend of links between Europe to Mountain Day Time US

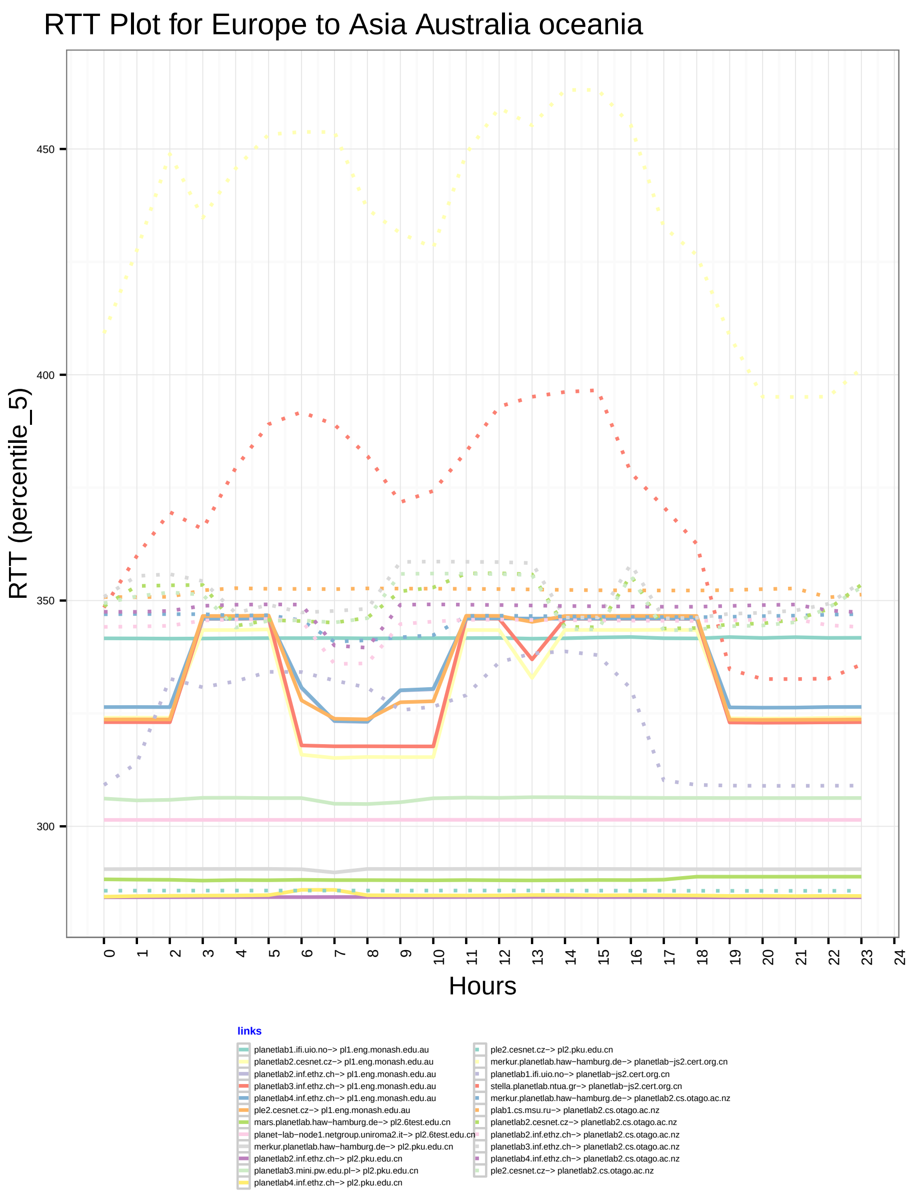


Figure 4.15: RTT Trend of links between Europe and Asia, Australia Oceania

Figure 4.15 pictures the RTT trend over hours of a day for all the links from Europe to Asia , Australia, and NewZealand. In the figure, we can see that different links are showing different RTT patterns over hours of a day. With close observation on the graph, we noticed that a lot of links have very little variation while some links have shown fluctuation between hour 3 and 6 and between hour 12 and 18.

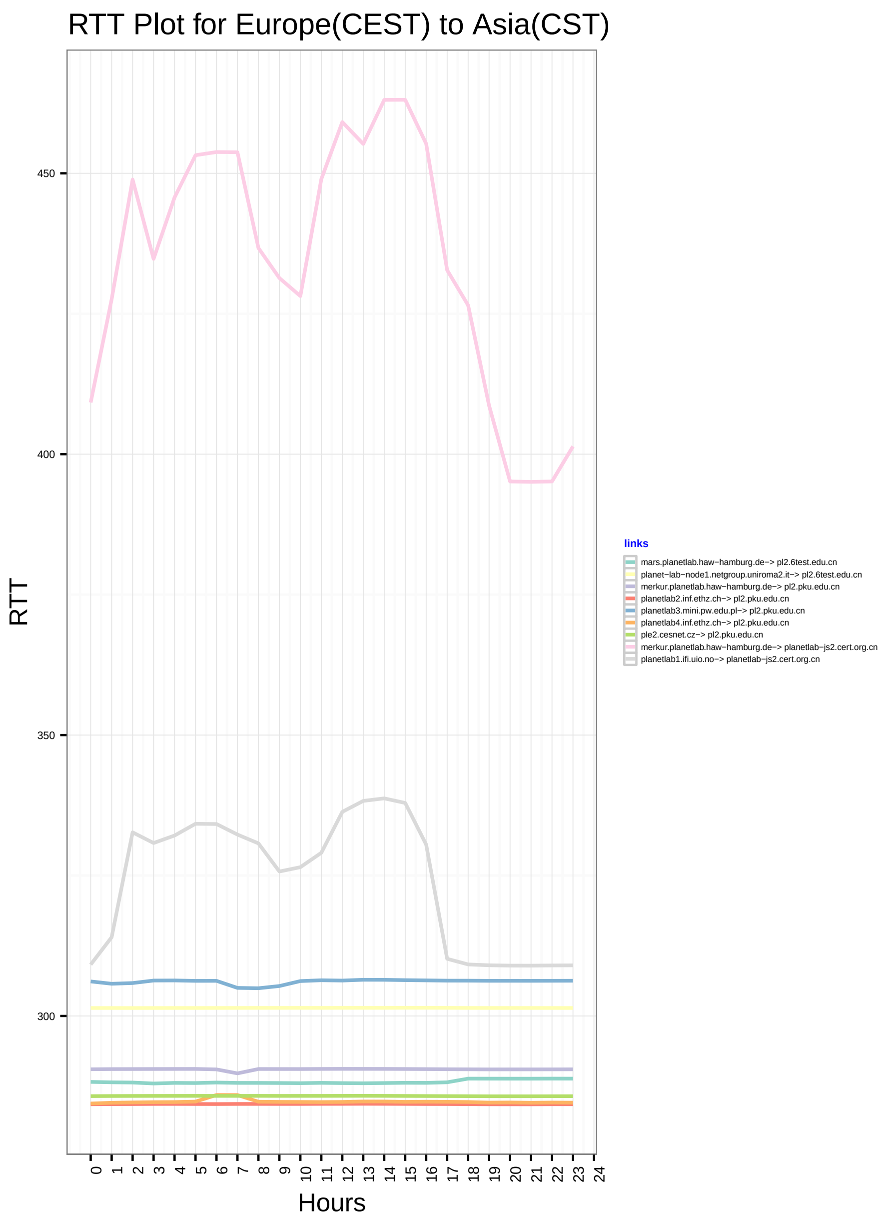


Figure 4.16: RTT Trend of links between Europe and China

The RTT trend over hours of a day from links having Central European Summer Time to China Standard Time is shown in figure 4.16. Seven node pair links in the graph do not have many variations on RTT over time. However, we observe some fluctuation on RTT on two links during hour 5 to 9 and 12 to 16.

4.3.4 Latency analysis of links from America to Asia

In this section present the latency analysis performed on the links from America to Asia having the common local time zone. In figure 4.17 we show the RTT trend over time on the links from Central Day Time in the

US to China Standard Time in Asia. We notice that the RTT peaks at hour 5 and 16 respectively in the link from planetlab1.dtc.umn.edu to planetlab-js2.cert.org.cn.

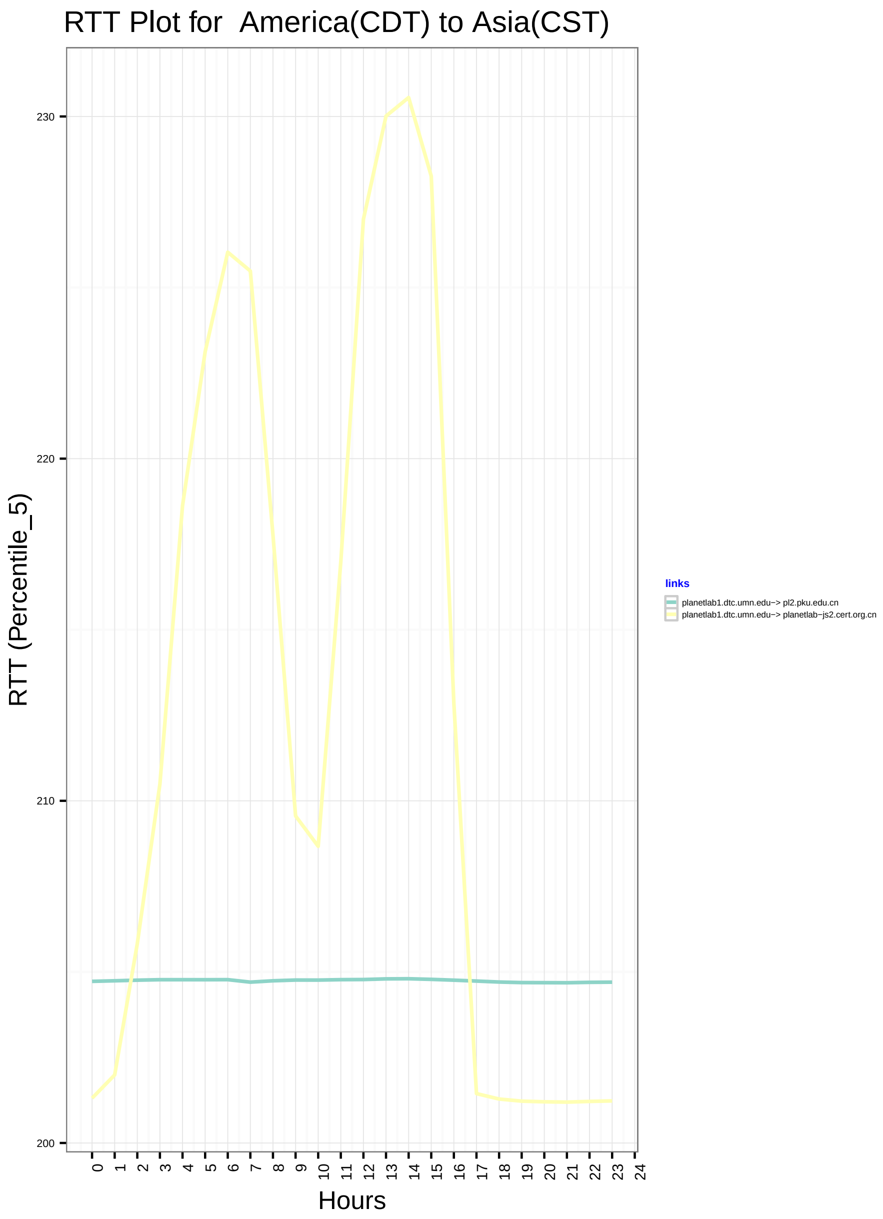


Figure 4.17: RTT Trend of links between Central Day Time zone in US to China

Figure 4.18 depicts the RTT variation over hours of a day on the links from Eastern Day Time in the US to China Standard Time in Asia. We observed that most of the links have same pattern of the RTT variation over time. The RTT fluctuation was noticed during hours 3 to 8 and 11 to 16.

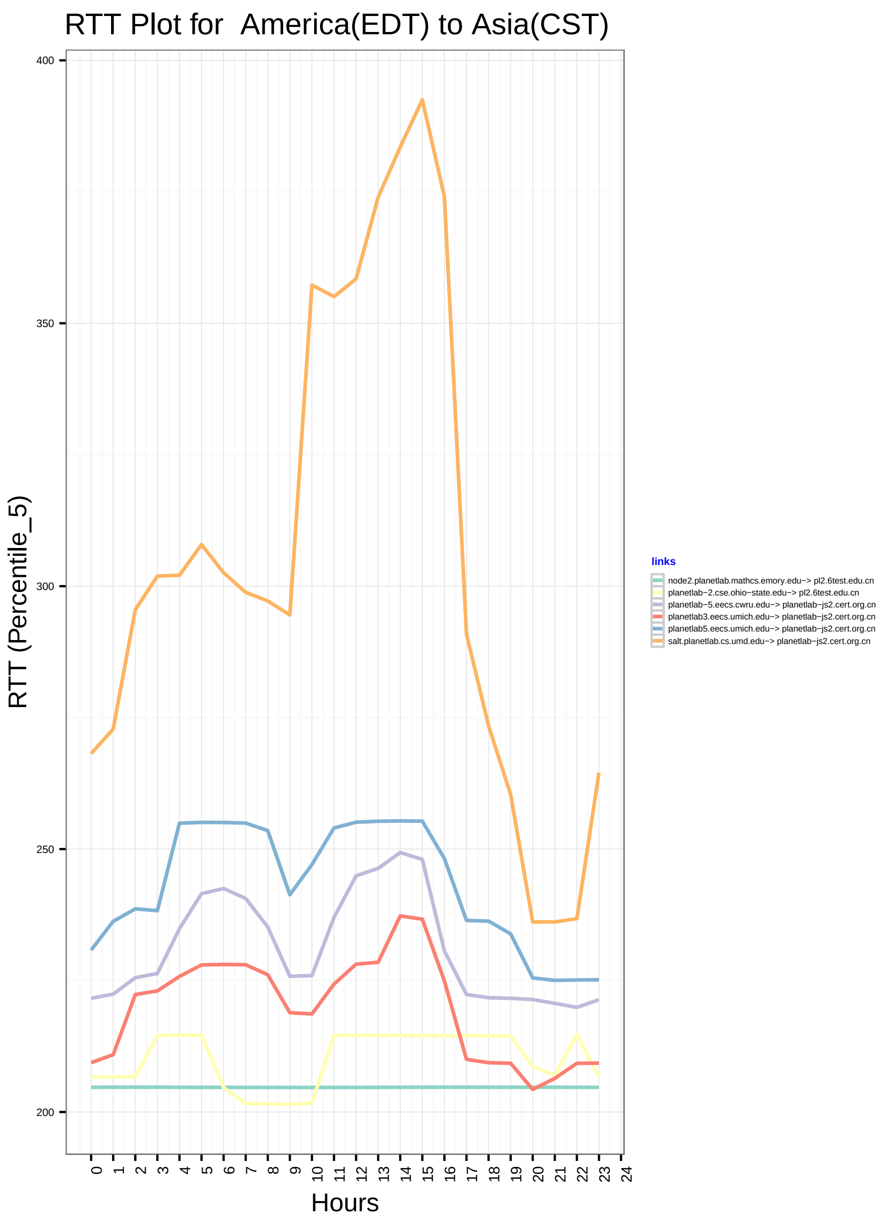


Figure 4.18: RTT Trend of links between Eastern Day Time zone in US to China

4.3.5 Latency analysis of links from America to America and Europe to Europe

We performed latency over set America to America and Europe to Europe following the exact process that we implied for sets explained above. We did not find many variations on the links to pinpoint and infer some meaning from that, therefore, we do not present any visual graph in this section.

4.4 Idetification of congested links

After completion of latency analysis on all links. We pointed the links having high RTT fluctuation over hours of a day. The links showing the RTT peaks over time series is identified as congested links among all the links. The links showing meaningful RTT fluctuation are shown in a table below. We will dig more into these in the next chapter.

SN	links
1	mars.planetlab.haw-hamburg.de to planetlab1.cs.okstate.edu
2	node2.planetlab.mathcs.emory.edu to ple2.cesnet.cz
3	pl1.cs.montana.edu_to_merkur.planetlab.haw-hamburg.de
4	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr
5	pl1.ucs.indiana.edu_to_stella.planetlab.ntua.gr
6	pl2.ucs.indiana.edu_to_merkur.planetlab.haw-hamburg.de
7	pl2.ucs.indiana.edu_to_planetlab2.tlm.unavarra.es
8	pl2.ucs.indiana.edu_to_planetlab4.inf.ethz.ch
9	planetlab-2.cse.ohio-state.edu_to_pl2.6test.edu.cn
10	planetlab-2.cse.ohio-state.edu_to_ple2.cesnet.cz
11	planetlab-js1.cert.org.cn_to_planetlab2.tlm.unavarra.es
12	planetlab-js1.cert.org.cn_to_stella.planetlab.ntua.gr
13	planetlab-js2.cert.org.cn_to_planetlab2.inf.ethz.ch
14	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu
15	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch
16	planetlab1.cs.otago.ac.nz_to_stella.planetlab.ntua.gr
17	planetlab1.dtc.umn.edu_to_planetlab-js2.cert.org.cn
18	planetlab1.virtues.fi_to_planetlab2.tlm.unavarra.es
19	planetlab2.cs.cornell.edu_to_ple2.cesnet.cz
20	planetlab2.cs.du.edu_to_merkur.planetlab.haw-hamburg.de
21	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru
22	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it
23	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr
24	planetlab2.tlm.unavarra.es_to_planetlab1.dtc.umn.edu
25	planetlab2.utt.fr_to_planetlab1.cs.okstate.edu
26	planetlab3.cesnet.cz_to_planetlab04.cs.washington.edu
27	planetlab4.mini.pw.edu.pl_to_planetlab-2.cse.ohio-state.edu
28	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn
29	salt.planetlab.cs.umd.edu_to_planet-lab-node1.netgroup.uniroma2.it
30	salt.planetlab.cs.umd.edu_to_planetlab-js2.cert.org.cn
31	planetlab-2.cse.ohio-state.edu_to_planetlab2.inf.ethz.ch

Figure 4.19: List of links having congestion

Chapter 5

Traceroute Analysis

In the previous chapter, we identified the links which have problems among all links. Based on those results, we move in the direction to identify the links in the path between two ends of those links which are contributing to the congestion. For the sake of identification of the all the links in between source and destination and troubleshooting, them, Traceroute is a handy tool. Hence we move forward to analyzing the traceroute output of respective links. In fact, we have run traceroute from one end of the link to another end every 10 seconds for 3 weeks. . The detail analysis process and result are discussed in the sections below as following.

5.1 Parsing and retrieving data in designated format

While running traceroute from source to destination, we have sent 10 packets from source to destination therefore at each hop we have 10 RTT values ,among them, we picked up minimum value and corresponding IP address. At the same time, we subtract the RTT values of that hop with the previous hop's RTT value in order to calculate the hop delay. This process continues until the end of the file. Eventually, we got the file with information about each hop number along with corresponding RTT, IP address, and delay to reach to that hop from the previous hop.

5.2 Generating Time series data for each hop from source to destination

We have recorded every instance of traceroute along with Unix timestamp so we can easily convert that Unix timestamp to hours . Thereafter, we create a bin with a size of an hour thus we have 24 bins for generating time series data. We put the data into their corresponding bins. More precisely, we binned RTT and hop to hop delay for all the hops. To finalize time series data creation , we compute some statics like mean, median, and percentile to reduce the number of observation. In this way, we obtained one data point per each hour so that we can visualize the value over the time series plot.

In order to perform per hop analysis with time series data, we need to have time series data for each hop that lies in the path from source to destination. Hence we extracted data that belongs to particular hop and make time series data for that hop. By applying this process for every hop we finally got time series data for each hop in the path between source and destination.

5.3 Analysis by correlation

After we obtain time series data of RTT and hop to hop delay for each hop on all the links that were identified as a congested link from latency analysis, the next task is to identify the links in between them which is contributing to the congestion. In this process, we use Pearson correlation method [2] to find a correlation between RTT and hop to hop delay for each hop in the path between source to destination.We apply the method for all the congested links. We selected all the hops showing more than 50 percent correlation as links that are responsible for congestion in overall path. Thereafter we found the position of that links in the network for instance if they belong to last mile or transit. Similarly, we also checked that links are intra-domain links or Inter-domain links. The detail information of correlation analysis is attached in the appendix. The files contain all the information about the links such as the

5.4 Results

In this section, we present the results obtained from the analysis process by visualizing them as graphs. The aim is to find how many links are contributing to congestion in a whole path, locate the position of the links and find out either they are from same network or different networks.

In figure 5.1 we plotted all the links which affected overall delay and locate where do they belong to in the routing path. We found that the links from the Transit network are higher than the last mile. From the figure, we can infer that there is more congestion in transit network than in the last mile for many links which were identified as congestion link from RTT trend analysis.

Figure 5.2 represents the number of Inter-domain and Intra-domain links contributing to the congestion. From the observation, at the figure, we found that there are more Intra-domain links responsible for congestion than the Inter-domain links. Figure 5.3 is simply a graph combining both figure 5.1 and 5.2. Here we how the Intra-domain links and Inter-domain links which are contributing to congestion and high RTT are distributed in the last mile and Transit network. With insight into the figure, we noticed that slightly higher number of the Intra-domain link lies in the last

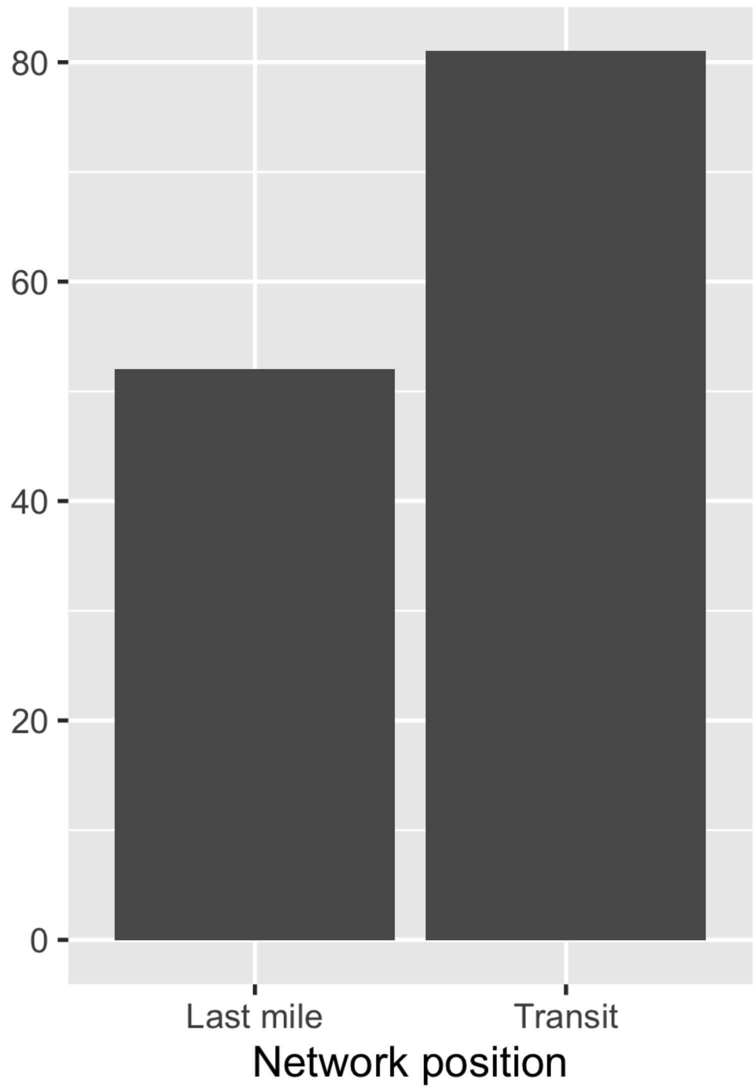


Figure 5.1: Number of the links with network position

mile than Transit network. In figure 5.4, we plotted the frequency of presence of the autonomous system of the participating links. We sorted out all the autonomous numbers that are forming links at each hop. After that, we counted their presence on the links we are analyzing. The aim was to find which Autonomous system is present in high frequency and contributing more to the delay and congestion. We observed that AS20965 is owned by GEANT The GEANT IP Service in Great Britain which has a high frequency of presence. Similarly, AS11537 is owned by ABILENE - Internet2 in the US and AS4134 owned by CHINANET-BACKBONE in Beijing have also a high frequency of the presence as shown in the figure 5.4.

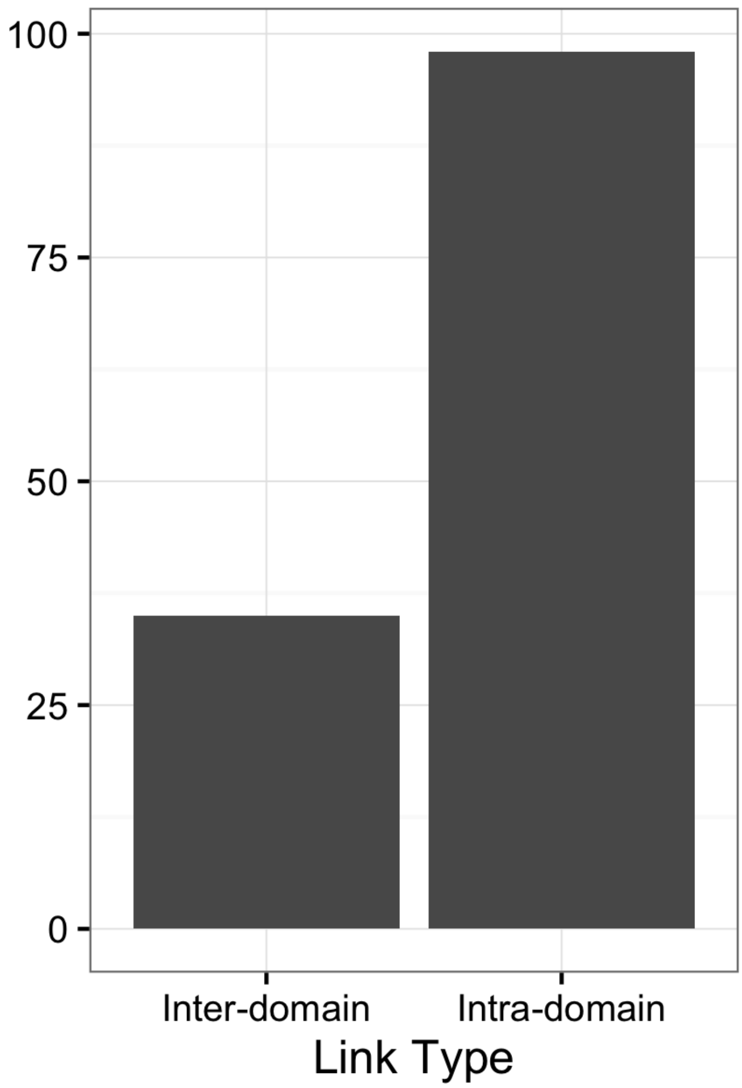


Figure 5.2: Number of the links with Link type

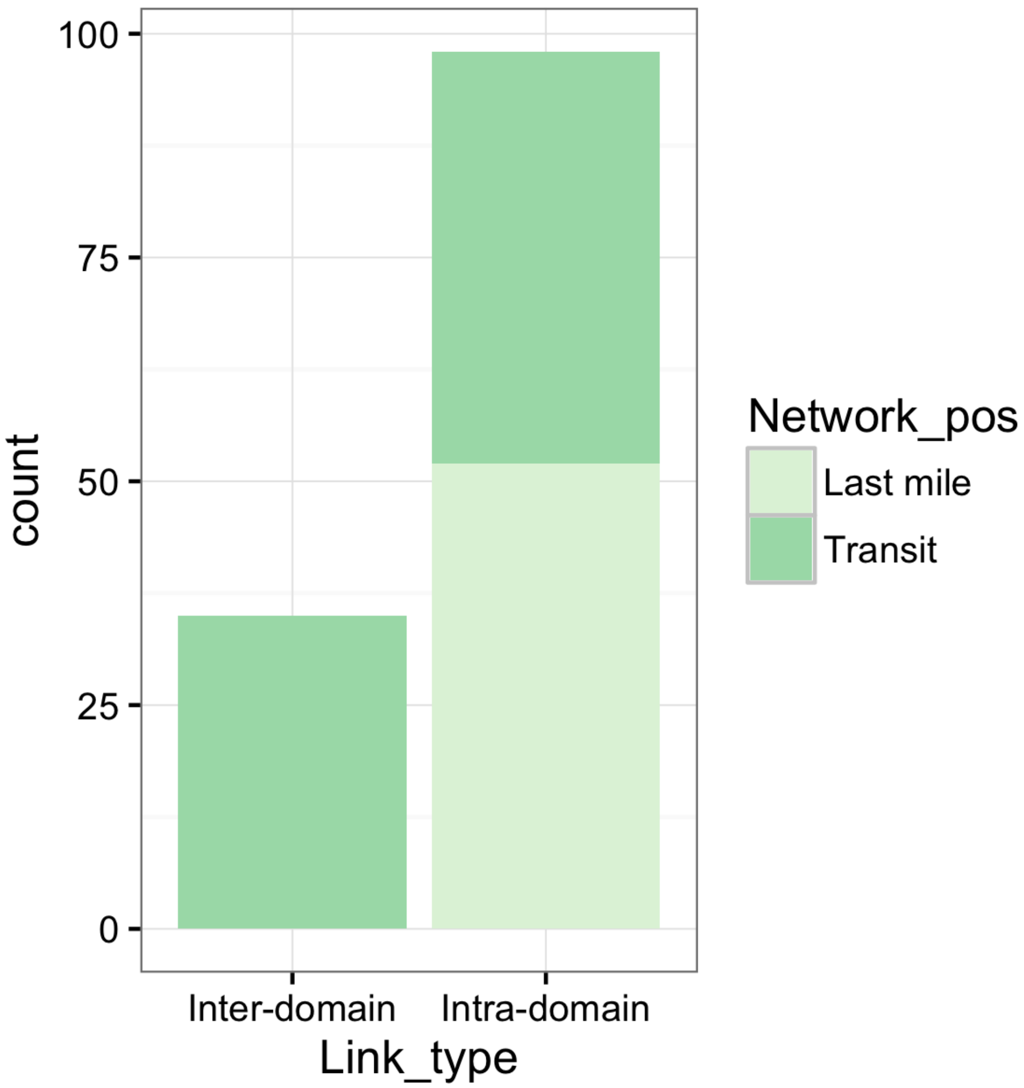


Figure 5.3: Number of the links with network position and link type

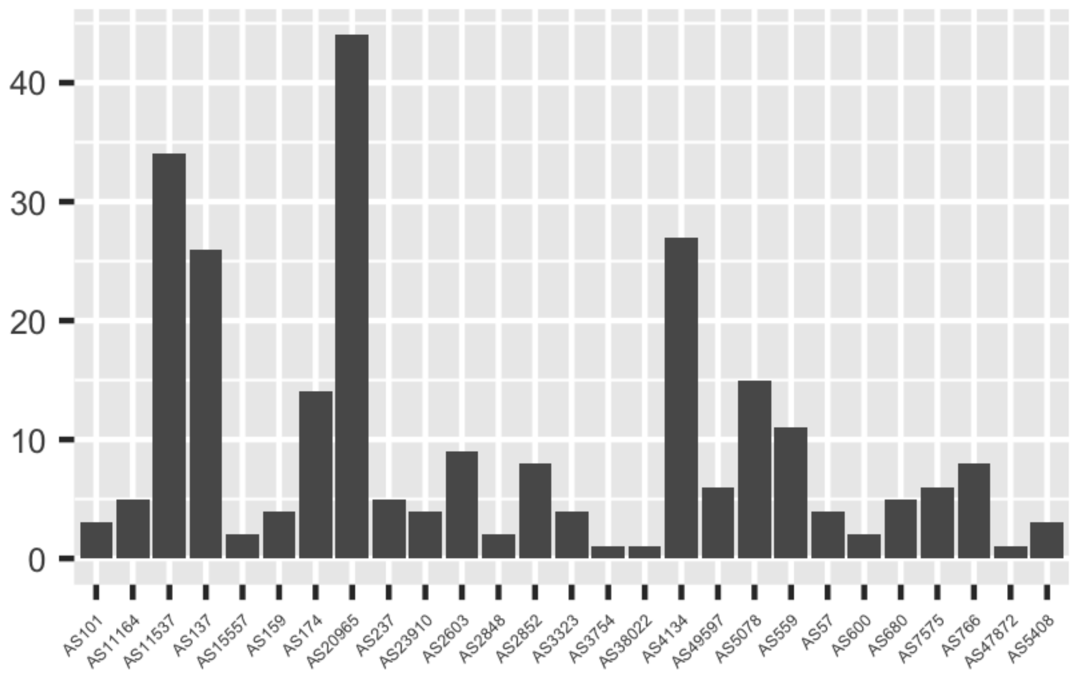


Figure 5.4: Number of the links with network position and link type

Chapter 6

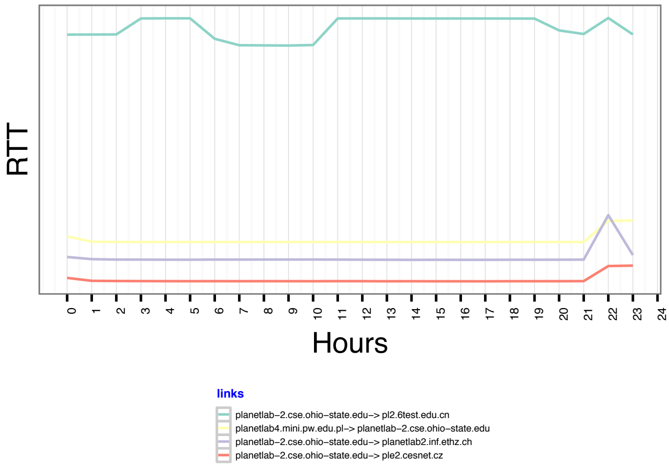
Discussion and Conclusion

6.1 Discussion on results from latency analysis

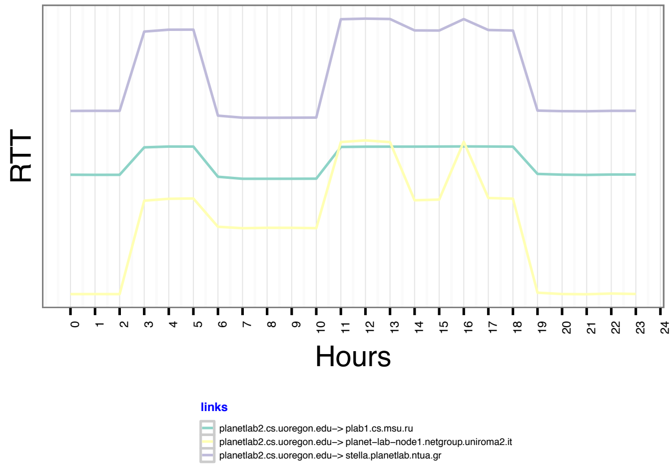
In this section we present some highlights of result from latency analysis. we analyzed around 200 node pair links investigating the latency trendover time. As a result, we found 31 node pair links showing peak RTT values over the time. From a close observation of results from latency analysis, we found 4 patterns by gathering the links with similar patterns together. The latency trend among those node pair links as shown in figure 6.2 below. Figure 6.1 depicts conversion of GMT to respective local time where the shaded area represents the business hours of respective time zones.

GMT	EDT US	PDT US	MDT US	CDT US	CST China	CEST Europe	EEST Europe
0	8 PM	5 PM	6 PM	7 PM	8 AM	2 AM	3 AM
1	9 PM	6 PM	7 PM	8 PM	9 AM	3 AM	4 AM
2	10 PM	7 PM	8 PM	9 PM	10 AM	4 AM	5 AM
3	11 PM	8 PM	9 PM	10 PM	11 AM	5 AM	6 AM
4	12 AM	9 PM	10 PM	11 PM	12 PM	6 AM	7 AM
5	1 AM	10 PM	11 PM	12 AM	1 PM	7 AM	8 AM
6	2 AM	11 PM	12 AM	1 AM	2 PM	8 AM	9 AM
7	3 AM	12 AM	1 AM	2 AM	3 PM	9 AM	10 AM
8	4 AM	1 AM	2 AM	3 AM	4 PM	10 AM	11 AM
9	5 AM	2 AM	3 AM	4 AM	5 PM	11 AM	12 PM
10	6 AM	3 AM	4 AM	5 AM	6 PM	12 PM	1 PM
11	7 AM	4 AM	5 AM	6 AM	7 PM	1 PM	2 PM
12	8 AM	5 AM	6 AM	7 AM	8 PM	2 PM	3 PM
13	9 AM	6 AM	7 AM	8 AM	9 PM	3 PM	4 PM
14	10 AM	7 AM	8 AM	9 AM	10 PM	4 PM	5 PM
15	11 AM	8 AM	9 AM	10 AM	11 PM	5 PM	6 PM
16	12 PM	9 AM	10 AM	11 AM	12 AM	6 PM	7 PM
17	1 PM	10 AM	11 AM	12 PM	1 AM	7 PM	8 PM
18	2 PM	11 AM	12 PM	1 PM	2 AM	8 PM	9 PM
19	3 PM	12 PM	1 PM	2 PM	3 AM	9 PM	10 PM
20	4 PM	1 PM	2 PM	3 PM	4 AM	10 PM	11 PM
21	5 PM	2 PM	3 PM	4 PM	5 AM	11 PM	12 AM
22	6 PM	3 PM	4 PM	5 PM	6 AM	12 AM	1 AM
23	7 PM	4 PM	5 PM	6 PM	7 AM	1 AM	2 AM

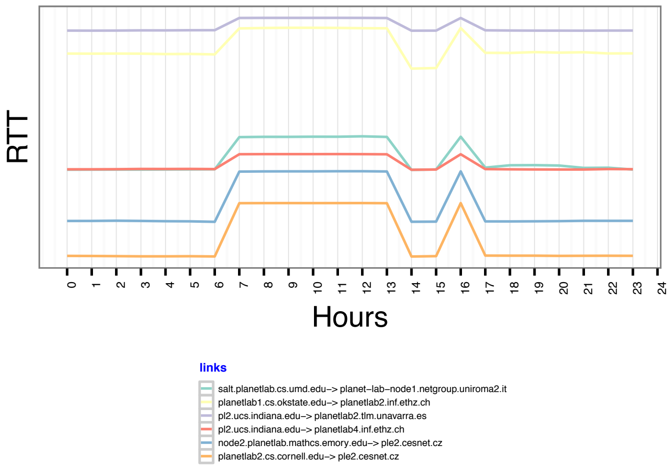
Figure 6.1: GMT to Local time chart



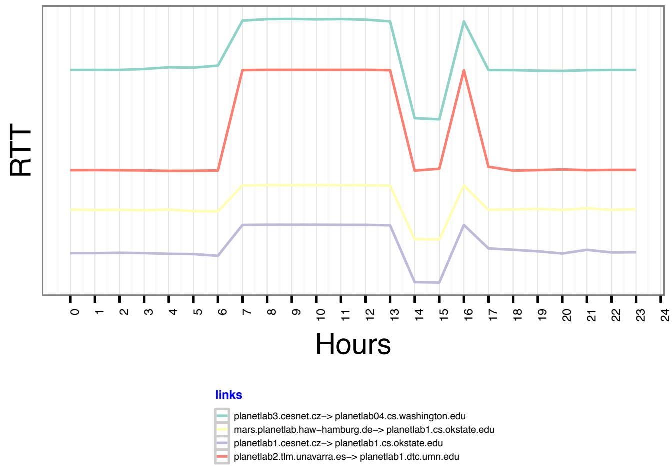
(a) Ohio Europe and China



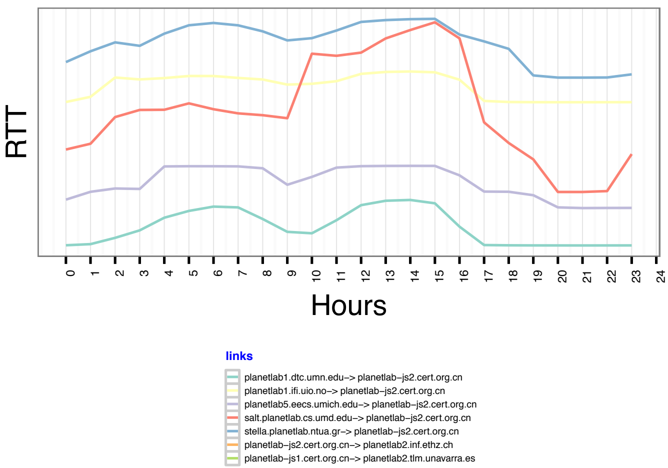
(b) Oregon and Europe



(c) America to Europe



(d) Europe to America



(e) America Europe China

Figure 6.2: Patterns of congested links after gathering the links with similar RTT trend together

In figure 6.2(a), we see common latency pattern of node pairs between Ohio, Europe, and China where all the links have latency peak during hours 22 GMT to 23 GMT. Referring to Eastern Day Time zone (EDT) in Ohio we detected that links are congested during peak hours 6pm to 7pm. links between Ohio and Europe and Europe to Ohio are congested during 11pm to 12am with reference to Central European Summer Time (CEST). Similarly, a link between Ohio and China is congested between 5am to 6am considering the China Standard Time (CST). From the observation of traceroute result for all these links, we found that more congestion is contributed by last mile networks. The total number of congested links in the core and in the edge is shown in bars labeled as pattern1 in figure 6.3. In addition,transit networks Ohio Academic Resources Network (OARnet) in Ohio, Abilene in the US and GEANT in Europe has also contributed for high latency.

Figure 6.2(b) depicts the common pattern of latency trend for links between node in Oregon in the US and nodes in Europe. We noticed latency elevated between hours 3 GMT to 6 GMT and also between hours 11 GMT to 18 GMT. With reference to Pacific Day Time(PDT) in Oregon, the links from Oregon to Europe remain congested for the whole morning and again got congested during peak evening hours 7pm to 9pm when the resource demand is high because streaming services for example, video streaming via NETFLIX. With reference to Eastern European Summer Time (EEST) in Europe, the links are congested in the morning during 6pm to 9pm and also are congested for whole business hours until 8 pm. When examining these links closer using traceroute, we detected that both last mile networks and transit networks contributed for elevated latency. However, we observed more congestion in transit networks. The exact number of links causing congestion in the core and the edge is depicted in the bars labelled as pattern2 in figure6.3. More precisely, ABILENE in the US, GEANT in the UK, NORDUNET in Norway and FIBERNET Corp in Orem are the backbone networks which are causing high latency.

A common latency fluctuation pattern of node pairs between China and Europe and China and the US is shown in figure 6.1(e). We noticed that the RTT gradually increases from hour 1 GMT and reaches to peak at hour 5 GMT. After hour 5 GMT the RTT drops slowly until hour 9 GMT. Thereafter, the RTT increased gradually from hour 10 GMT and reaches to peak at hour 15 GMT . After hour 15 GMT it gradually decreases and levels to the normal RTT value after hour 17 GMT. Regarding local time in China, the links between china and other nodes are congested during late business hours 7pm and 10 pm. In addition, the links are congested after midnight and remain congested until 2am.The links between China and the US are congested during business hours 11am and 2pm and couple of hours after midnight. With respect to European local time, the links are congested in morning from 7 am to 10 am and also during business hours 2pm and 5pm. Correlating with traceroute results as shown in pattern3 bars in figure 6.3, we detected that there are more congested links in the core

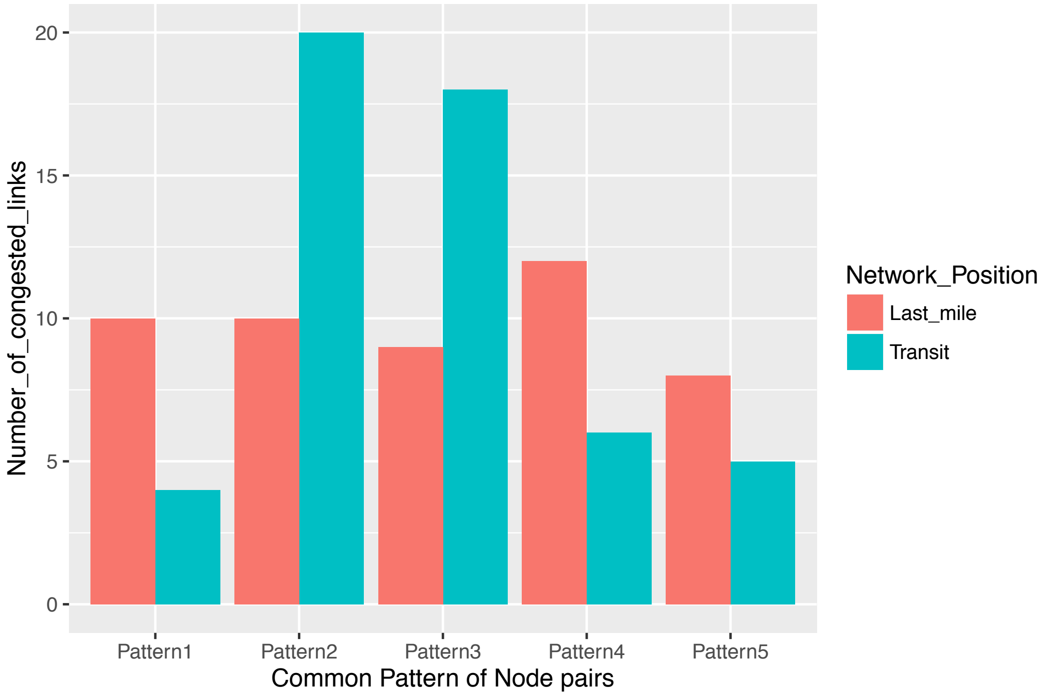


Figure 6.3: Number of congested links along with Network position for RTT patterns shown in figure 6.2

and CHINANET-BACKBONE (backbone network) in china is contributing most for the measured delay. In addition, ENDAV-AS in Bulgaria , Cogent, and INTERNET2 in the US also have a contribution for a high delay.

Another common pattern of latency found among links from America to Europe and Europe to America is depicted in figure 6.1(d) and 6.1(e) respectively. Here,we can see the high latency is observed between 7 GMT and13 GMT. We also found another peak RTT value at hour 16 GMT. Linking the pattern of congestion with the local time zones of nodes in the USA, we observed that the links are congested during business hours 9am to 11am in the US. In addition, the links are congested between 1am and 8 am. Regarding the European local time, we observed that links are congested for long during business hours 9 am and 5pm. We investigated more on these links using traceroute, as shown in bars in pattern4 and pattern5 in figure 3. In figure, we detected that there are more congested links in the edge than in the core. In case of congestion in the core networks, we found that backbone networks namely ABELINE in the US and GEANT in Europe have contributed most for congestion.

Based on patterns mentioned in figure 6.2 and in figure 6.3, we found that links are congested normally at business hours and hours in the evening. In addition, we also observed the congestion during a night and in the morning for some links. Apart from that, we noticed that big backbone networks such as ABELINE, GEANT, and CHINANET-BACKBONE are contributing more to the congestion in the core networks (a detailed analysis on these backbone networks are presented in the next section). However, the edge networks are also causing the congestion.The exact comparison of congestion in the core networks and the edge networks including all congested node pairs is presented in next section of discussion.

From insight into the latency analysis results, we observed that only node pairs links having end nodes from the different continent are congested. This means that links with very long path crossing global continents are congested. In figure 6.2, we can see that all congested node pair links having end nodes from different continents. In figure 6.4 , the latency trend of links having end nodes only in America is depicted. Similarly, figure 6.5 shows the RTT trend of the links where end nodes belong to Europe only. In the both figures, we did not notice significant RTT variation.

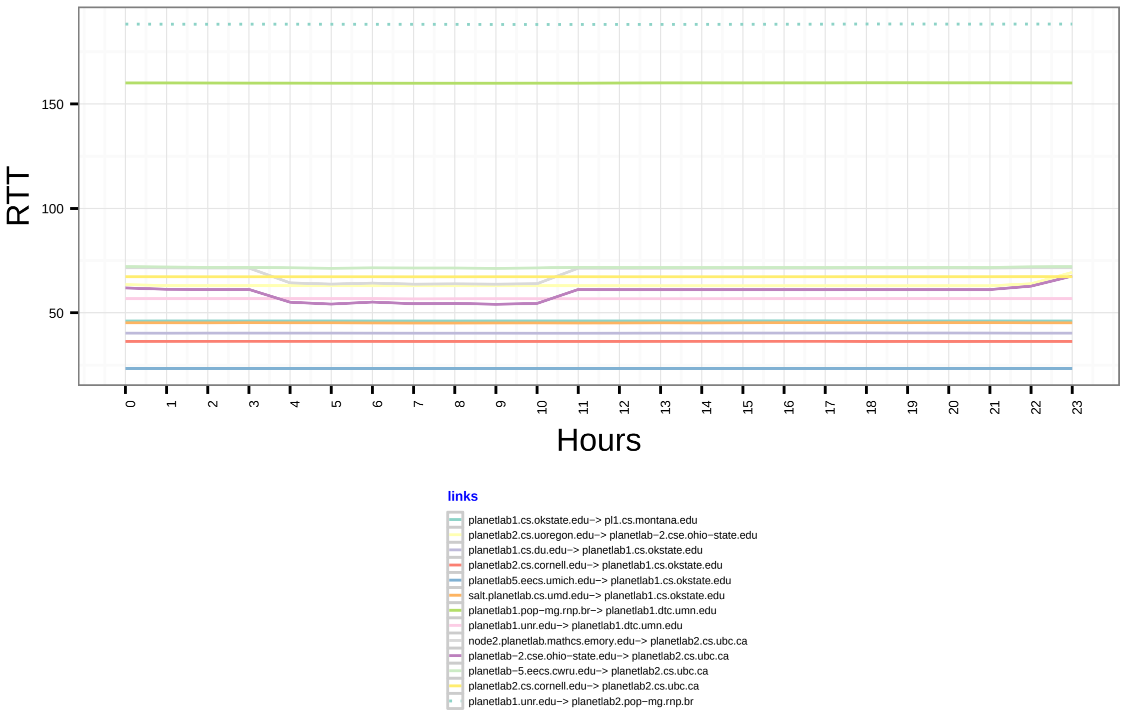


Figure 6.4: RTT trend America to America

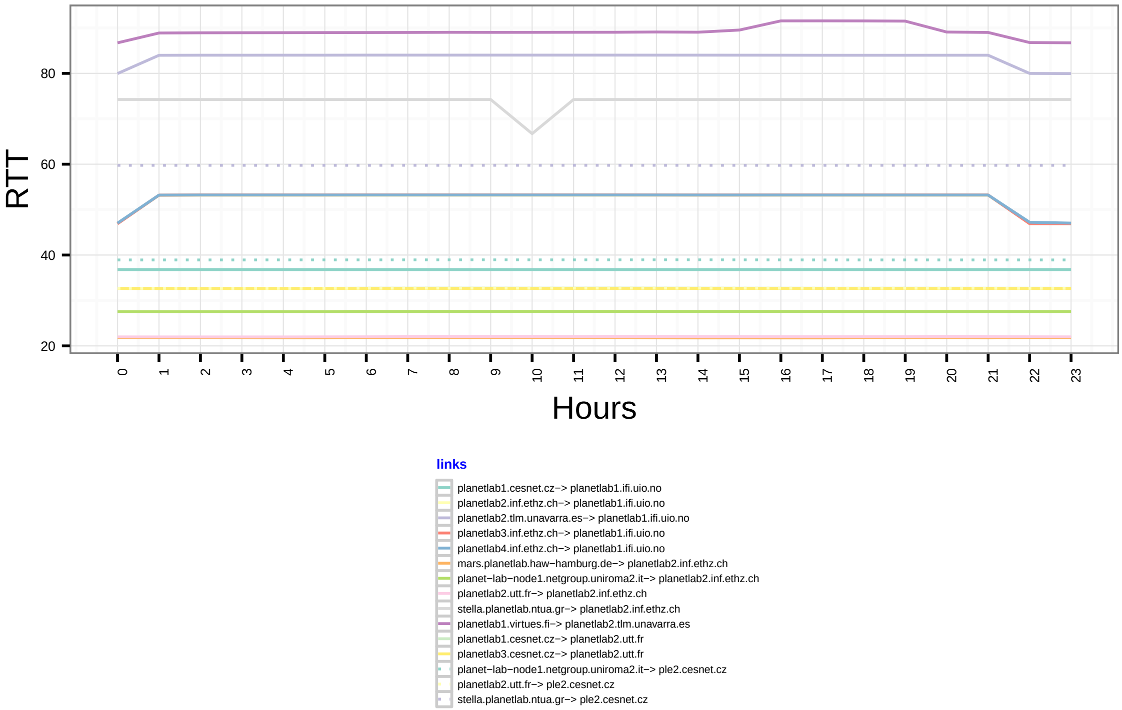


Figure 6.5: RTT trend Europe to Europe

6.2 Discussion on traceroute analysis results

We investigated more on the identified congested node pairs using traceroute in order to find out links causing congestion in a path between end nodes. The term link we are using in this section is a link between consecutive routers. This will be clearer if we look at correlated traceroute result included in the appendix. We identified the links that are causing a high delay in a path by correlating RTT and hop by hop delay. From correlated traceroute results, we found that around 58% of congested links are in transit networks and 42% of congested links lies in last mile networks(networks containing source or destination node). Among all the congested links, around 78% of them were Intra-domain links and around 22% of then were Inter-domain links. Moreover than 50% of total Intra-domain links are present in transit networks.

We found 134 congested links on the path between source to the destination from all congested node pairs links and among them 70 congested links were from just 3 backbone networks. In figure 6.5, we present detailed information about the congested links along with corresponding backbone networks.

AS	AS owner	Total links	Network Position	Total Inter-domain Links	Total Intra-domain Links
AS20965	GEANT The GEANT IP Service, GB	27	Transit	10	17
AS11537	ABILENE - Internet2, US	25	Transit	13	12
AS4134	CHINANET-BACKBONE No.31,Jin-rong Street, CN	18	Transit	9	0
AS4134	CHINANET-BACKBONE No.31,Jin-rong Street, CN	18	Last mile	0	9

Figure 6.6: Number of the links with network position and link type for GEANT, ABILENE and CHINANET-BACKBONE backbone networks

We noticed that packets in all links from America to Europe need transit from AS11537 when leaving America and need transit from AS20965 when entering Europe and vice versa. Whereas AS4134 provides transit for packets entering or leaving china and also provides connection service for regional networks. From the information in figure 6, we can see that almost 50% of the congested links lies within these ASes.

6.3 Limitations

We have conducted experiments in PlanetLab testbed and it has some limitations. All the nodes in PlanetLab are at universities or research centers and they are likely to use educational networks such as ABILENE and GEANT therefore what we see here is limited to PlanetLab only but the Internet is very broad in a scope. Therefore, we fail to measure real access networks as experienced by regular users. PlanetLab nodes are used by a large number of researchers and students sharing limited resources hence it can also be congested. Hence, our finding might have been affected by congestion in the PlanentLab. Apart from this, the measurement is limited to RTT measurement between node pairs of the PlanetLab nodes. We have not performed loss analysis correlating RTT analysis hence the RTT analysis might not be precise. Moreover, we are using Traceroute which infers latency and gives a nice overview of network hops but it is not precise. As traceroute gives latency value on the basis of the only forward path, the latency value is not precise if there is congestion in reverse path. Furthermore, traceroute might not insight properly where is congestion when there are multiple routing policies and asymmetric routes in the network.

6.4 Conclusion

Networking is growing rapidly in terms of size and complexity challenging the performance of the network. In order to cope this, we have invested a lot of money to increase capacity, speed and upgrade technologies. However, the performance of the network is not satisfactory as the underlying problem is congested links which are the bottleneck for the network performance. In addition, we are mostly focused on only improving the performance of the edge networks. However, the performance of the network is degraded in the core networks nowadays [30, 41]. Hence, in this thesis, we devised experiments in the PlanetLab testbed to examine congestion in the edge networks as well as in the core networks. We measured end to end delay of more than 200 node pairs where nodes are distributed all over the world and detected the congested node pairs among them. Using traceroute analysis on the congested node pairs, we found the congested links between node pairs and located them in the network. Hence, we detected around 42% congestion in the edge networks whereas around 58% congestion in the core networks. Moreover, we observed that intra-domain links contributed more for congestion than inter-domain links.

6.5 Future works

Although we calculated packet loss, we did not have time to analyze the loss statistics that were gathered. Therefore, we can correlate the result from traceroute analysis with loss statistics and make the result more precise. We are limited to only a few nodes in the PlanetLab so that we can increase the number of nodes in the future and make the experiment complete and more meaningful. Similarly, we can increase the duration of experiment significantly for instance more than 6 weeks so that we might see more appropriate variation on latency. Another improvement can be made by adding people's home networks for the experiments. In this thesis, we are limited only to educational and research networks in the PlanetLab. If we add home networks we can broaden the scope as commercial ISP networks will be added up for analysis.

Bibliography

[1] National Media Museum Bradford BD1 1NQ. "A Brief History of the Internet." In: (2011).

[2] Alan Agresti and Barbara F Agresti. "Statistical Methods for the." In: *Social Sciences. CA: Dellen Publishers* (1970).

[3] A de A Antonio et al. "Revisitando Metrologia de Redes: Do Passadoas Novas Tendências." In: ().

[4] Jose M. Barcelo, Juan I. Nieto-Hipolito, and Jorge Garcıa-Vidal. "Study of Internet autonomous system interconnectivity from BGP routing tables." In: (2004).

[5] Steven Bauer, David D Clark, and William Lehr. "The evolution of internet congestion." In: TPRC. 2009.

[6] Jean-Chrysotome Bolot. "End-to-end packet delay and loss behavior in the Internet." In: *ACM SIGCOMM Computer Communication Review*. Vol. 23. 4. ACM. 1993, pp. 289–298.

[7] Michael S Borella et al. "Internet packet loss: Measurement and implications for end-to-end QoS." In: *Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing., 1998 Proceedings of the 1998 ICPP Workshops on*. IEEE. 1998, pp. 3–12.

[8] CJ Bovy et al. "Analysis of end-to-end delay measurements in Internet." In: *Proceedings of ACM Conference on Passive and Active Leasurements (PAM), Fort Collins, Colorado, USA*. 2002.

[9] Robert L Carter and Mark E Crovella. "Measuring bottleneck link speed in packet-switched networks." In: *Performance evaluation* 27 (1996), pp. 297–318.

[10] Marshini Chetty et al. "Why is my internet slow?: making network speeds visible." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, pp. 1889–1898.

[11] Chiara Chirichella and Davide Rossi. "To the Moon and back: are Internet bufferbloat delays really that large?" In: *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE. 2013, pp. 417–422.

[12] Brent Chun et al. "Planetlab: an overlay testbed for broad-coverage services." In: *ACM SIGCOMM Computer Communication Review* 33.3 (2003), pp. 3–12.

[13] Cisco. "Border Gateway Protocol." In: *Internetworking Technologies Handbook*. Cisco Press, Sept. 2003. Chap. 41, pp. 663–673.

[14] David D Clark et al. "Measurement and Analysis of Internet Interconnection and Congestion." In: *2014 TPRC Conference Paper*. 2014.

[15] Douglas E. Comer. *The Internet Book: Everything You Need to Know about Computer Networking and How the Internet Works*. 4th ed. Prentice Hall, 2006. ISBN: 0132335530.

[16] Constantine Dovrolis. "The Evolution and Economics of Internet Interconnections." In: ().

[17] Aiguo Fei et al. "Measurements on delay and hop-count of the Internet." In: *IEEE GLOBECOM*. Vol. 98. 1998.

[18] Lixin Gao. "On inferring autonomous system relationships in the Internet." In: *IEEE/ACM Transactions on Networking (ToN)* 9.6 (2001), pp. 733–745.

[19] Daniel Genin and Jolene Splett. "Where in the Internet is congestion?" In: *arXiv preprint arXiv:1307.3696* (2013).

[20] Jim Gettys and Kathleen Nichols. "Bufferbloat: dark buffers in the internet." In: *Communications of the ACM* 55.1 (2012), pp. 57–65.

[21] Ningning Hu et al. "A measurement study of internet bottlenecks." In: *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. Vol. 3. IEEE. 2005, pp. 1689–1700.

[22] Ningning Hu et al. *Locating Internet bottlenecks: Algorithms, measurements, and implications*. Vol. 34. 4. ACM, 2004.

[23] Gianluca Iannaccone, Martin May, and Christophe Diot. "Aggregate traffic performance with active queue management and drop from tail." In: *ACM SIGCOMM Computer Communication Review* 31.3 (2001), pp. 4–13.

[24] Olumuyiwa Ibidunmoye. "Performance Problem Diagnosis in Cloud Infrastructures." In: (2016).

[25] Olumuyiwa Ibidunmoye, Francisco Hernández-Rodriguez, and Erik Elmroth. "Performance anomaly detection and bottleneck identification." In: *ACM Computing Surveys (CSUR)* 48.1 (2015), p. 4.

[26] Cisco Visual Networking Index. "Global Mobile Data Traffic Forecast Update 2014–2019. White Paper c11-520862." In: *Available on http://www. cisco. com/c/en/us/solutions/collateral/service-provider/visual-networking-indexvni/white_paper_c11-520862. html* ().

[27] Cisco Visual Networking Index. *Global Mobile Data Traffic Forecast Update 2015–2020 White Paper*.

[28] Ivan Kaminow and Tingye Li. *Optical Fiber Telecommunications IV-B: Systems and Impairments*. Academic Press, 2002. ISBN: 0-12-395172-0.

[29] Dan Komosny et al. "PlanetLab Europe as Geographically-Distributed Testbed for Software Development and Evaluation." In: *Advances in Electrical and Electronic Engineering* 13.2 (2015), pp. 137–146.

[30] Tom Leighton. "Improving performance on the internet." In: *Communications of the ACM* 52.2 (2009), pp. 44–51.

[31] Chuan Lin et al. "Research on bottleneck-delay in internet based on IP united mapping." In: *Peer-to-Peer Networking and Applications* (2016), pp. 1–13.

[32] Matthew Luckie et al. "Challenges in Inferring Internet Interdomain Congestion." In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM. 2014, pp. 15–22.

[33] Ivan Marsic. *Computer Networks: Performance and Quality of Service*. Ivan Marsic, 2010. ISBN: N/A.

[34] Ingrid Melve et al. "A study on network performance metrics and their composition." In: *Campus-Wide Information Systems* 23.4 (2006), pp. 268–282.

[35] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. "The Akamai network: a platform for high-performance internet applications." In: *ACM SIGOPS Operating Systems Review* 44.3 (2010), pp. 2–19.

[36] Bruno Quoitin et al. "Interdomain traffic engineering with BGP." In: *Communications Magazine, IEEE* 41.5 (2003), pp. 122–128.

[37] Lawrence G Roberts. "Beyond Moore's law: Internet growth trends." In: *Computer* 33.1 (2000), pp. 117–119.

[38] Seungwan Ryu, Christopher Rump, and Chunming Qiao. "Advances in active queue management (AQM) based TCP congestion control." In: *Telecommunication Systems* 25.3-4 (2004), pp. 317–351.

[39] Halabi Sam. *Internet routing architectures*. Pearson Education India, 2008.

[40] Ankit Singla et al. "The internet at the speed of light." In: *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*. ACM. 2014, p. 1.

[41] Ramesh K Sitaraman. "Network performance: Does it really matter to users and by how much?" In: *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*. IEEE. 2013, pp. 1–10.

[42] Internet Society. *Global Internet Report 2014 Internet Society*. Tech. rep. Internet Society, May 2014.

[43] Michael Welzl. *Network congestion control: managing internet traffic*. John Wiley & Sons, 2005.

[44] Mark Winther. *Tier 1 ISP s : What They Are and Why They Are Important*. WHITE PAPER. Global Headquarters: 5 Speen Street Framingham, MA 01701 USA: IDC, May 2006.

[45] Yang Richard Yang and Simon S Lam. "General AIMD congestion control." In: *Network Protocols, 2000. Proceedings. 2000 International Conference on*. IEEE. 2000, pp. 187–198.

Appendices

Timezone of all nodes

Node	Location	Time zone	UTC/GMT with offset
node2.planetlab.mathcs.emory.edu	Georgia USA	Eastern Daylight Time (EDT)	UTC/GMT -4
pl1.cs.montana.edu	Montana USA	Mountain Daylight Time (MDT)	UTC/GMT -6
pl1.ucs.indiana.edu	Bloomington Indiana USA	Eastern Daylight Time (EDT)	UTC/GMT -4
pl2.ucs.indiana.edu	Bloomington Indiana USA	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab-2.cse.ohio-state.edu	Columbus Ohio USA	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab-5.eecs.cwru.edu	Cleveland Ohio USA	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab02.cs.washington.edu	Seattle Washington USA	Pacific Daylight Time (PDT)	UTC/GMT -7
planetlab04.cs.washington.edu	Seattle Washington USA	Pacific Daylight Time (PDT)	UTC/GMT -7
planetlab1.cs.du.edu	Denver Colorado USA	Mountain Daylight Time (MDT)	UTC/GMT -6
planetlab1.cs.okstate.edu	Stillwater Oklahoma USA	Central Daylight Time (CDT)	UTC/GMT -5
planetlab1.dtc.umn.edu	Minneapolis Minnesota USA	Central Daylight Time (CDT)	UTC/GMT -5
planetlab1.pop-mg.rnp.br	Belo Horizonte Minas Gerais Brazil	Brasília Time (BRT)	UTC/GMT -3
planetlab1.unr.edu	Reno Nevada USA	Pacific Daylight Time (PDT)	UTC/GMT -7
planetlab2.citadel.edu	Charleston South Carolina USA	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab2.cs.cornell.edu	New York USA	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab2.cs.du.edu	Denver Colorado USA	Mountain Daylight Time (MDT)	UTC/GMT -6
planetlab2.cs.ubc.ca	Vancouver Canada	Pacific Daylight Time (PDT)	UTC/GMT -7
planetlab2.cs.uoregon.edu	Eugene Oregon USA	Pacific Daylight Time (PDT)	UTC/GMT -7
planetlab2.pop-mg.rnp.br	Belo Horizonte Minas Gerais Brazil	Brasília Time (BRT)	UTC/GMT -3
planetlab2.rutgers.edu	New Jersey United States	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab2.utdallas.edu	Dallas Texas USA	Central Daylight Time (CDT)	UTC/GMT -5
planetlab3.cs.uoregon.edu	Eugene Oregon USA	Pacific Daylight Time (PDT)	UTC/GMT -7
planetlab3.eecs.umich.edu	Michigan USA	Eastern Daylight Time (EDT)	UTC/GMT -4
salt.planetlab.cs.umd.edu	Maryland USA	Eastern Daylight Time (EDT)	UTC/GMT -4
planetlab5.eecs.umich.edu	Michigan USA	Eastern Daylight Time (EDT)	UTC/GMT -4
mars.planetlab.haw-hamburg.de	Hamburg Germany	Central European Summer Time (CEST)	UTC/GMT +2
merkur.planetlab.haw-hamburg.de	Hamburg Germany	Central European Summer Time (CEST)	UTC/GMT +2
plab1.cs.msu.ru	Moscow Russia	Far Eastern European Time (FET)	UTC/GMT +3
planetlab-coffee.ait.ie	Athlon Ireland	Irish Summer Time (IST)	UTC/GMT +1
planetlab1.cesnet.cz	Prague Czech Republic	Central European Summer Time (CEST)	UTC/GMT +2
planetlab1.ifi.uio.no	Oslo Norway	Central European Summer Time (CEST)	UTC/GMT +2
planetlab1.net.in.tum.de	Munich Germany	Central European Summer Time (CEST)	UTC/GMT +2
planetlab1.virtues.fi	Vantaa, Finland	Eastern European Summer Time (EEST)	UTC/GMT +3
planetlab2.cesnet.cz	Prague Czech Republic	Central European Summer Time (CEST)	UTC/GMT +2
planetlab2.inf.ethz.ch	Zurich Switzerland	Central European Summer Time (CEST)	UTC/GMT +2
planetlab2.tlm.unavarra.es	Madrid Spain	Central European Summer Time (CEST)	UTC/GMT +2
planetlab2.utt.fr	Troyes France	Central European Summer Time (CEST)	UTC/GMT +2
planetlab3.cesnet.cz	Prague Czech Republic	Central European Summer Time (CEST)	UTC/GMT +2
planetlab3.inf.ethz.ch	Zurich Switzerland	Central European Summer Time (CEST)	UTC/GMT +2
planetlab3.mini.pw.edu.pl	Warsaw Poland	Central European Summer Time (CEST)	UTC/GMT +2
planetlab4.inf.ethz.ch	Zurich Switzerland	Central European Summer Time (CEST)	UTC/GMT +2
planetlab4.mini.pw.edu.pl	Warsaw Poland	Central European Summer Time (CEST)	UTC/GMT +2
ple2.cesnet.cz	Prague Czech Republic	Central European Summer Time (CEST)	UTC/GMT +2
stella.planetlab.ntua.gr	Athens Greece	Eastern European Summer Time (EEST)	UTC/GMT +3
pl1.eng.monash.edu.au	Victoria Australia	Australian Eastern Standard Time (AEST)	UTC/GMT +10
pl2.6test.edu.cn	Beijing China	China Standard Time (CST)	UTC/GMT +8
pl2.pku.edu.cn	Beijing (Haidian) China	China Standard Time (CST)	UTC/GMT +8
planet1.pnl.nitech.ac.jp	Tokyo Japan	Japan Standard Time (JST)	UTC/GMT +9
planet2.pnl.nitech.ac.jp	Tokyo Japan	Japan Standard Time (JST)	UTC/GMT +9
planetlab-js1.cert.org.cn	Nanjing Jiangsu China	China Standard Time (CST)	UTC/GMT +8
planetlab-js2.cert.org.cn	Nanjing Jiangsu China	China Standard Time (CST)	UTC/GMT +8
planetlab1.cs.otago.ac.nz	Dunedin New Zealand	New Zealand Standard Time (NST)	UTC/GMT +12
planetlab1.cs.otago.ac.nz	Dunedin New Zealand	New Zealand Standard Time (NST)	UTC/GMT +12

All link with time zone informations

Node1	Node2	Continent_to_Continent	Timezone_to_Timezone
node2.planetlab.mathcs.emory.edu	ple2.cesnet.cz	America_to_Europe	EDT_to_CEST
node2.planetlab.mathcs.emory.edu	planetlab2.inf.ethz.ch	America_to_Europe	EDT_to_CEST
pl1.cs.montana.edu	planetlab2.utt.fr	America_to_Europe	MDT_to_CEST
pl1.cs.montana.edu	merkur.planetlab.haw-hamburg.de	America_to_Europe	MDT_to_CEST
pl1.cs.montana.edu	planetlab4.inf.ethz.ch	America_to_Europe	MDT_to_CEST
pl1.ucs.indiana.edu	mars.planetlab.haw-hamburg.de	America_to_Europe	EDT_to_CEST
pl1.ucs.indiana.edu	planetlab-coffee.ait.ie	America_to_Europe	EDT_to_IST
pl1.ucs.indiana.edu	stella.planetlab.ntua.gr	America_to_Europe	EDT_to_EEST
pl1.ucs.indiana.edu	planetlab1.ifi.uio.no	America_to_Europe	EDT_to_CEST
pl2.ucs.indiana.edu	merkur.planetlab.haw-hamburg.de	America_to_Europe	EDT_to_CEST
pl2.ucs.indiana.edu	planetlab4.inf.ethz.ch	America_to_Europe	EDT_to_CEST
pl2.ucs.indiana.edu	planetlab2.tlm.unavarra.es	America_to_Europe	EDT_to_CEST
planetlab-2.cse.ohio-state.edu	ple2.cesnet.cz	America_to_Europe	EDT_to_CEST
planetlab-2.cse.ohio-state.edu	planetlab2.inf.ethz.ch	America_to_Europe	EDT_to_CEST
planetlab-5.eecs.cwru.edu	planetlab2.inf.ethz.ch	America_to_Europe	EDT_to_CEST
planetlab02.cs.washington.edu	planetlab3.inf.ethz.ch	America_to_Europe	PDT_to_CEST
planetlab02.cs.washington.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	PDT_to
planetlab02.cs.washington.edu	stella.planetlab.ntua.gr	America_to_Europe	PDT_to_EEST
planetlab02.cs.washington.edu	planetlab2.utt.fr	America_to_Europe	PDT_to_CEST
planetlab04.cs.washington.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	PDT_to
planetlab04.cs.washington.edu	merkur.planetlab.haw-hamburg.de	America_to_Europe	PDT_to_CES
planetlab04.cs.washington.edu	stella.planetlab.ntua.gr	America_to_Europe	PDT_to_EEST
planetlab04.cs.washington.edu	planetlab2.utt.fr	America_to_Europe	PDT_to_CEST
planetlab1.cs.du.edu	stella.planetlab.ntua.gr	America_to_Europe	MDT_to_EEST

planetlab1.cs.du.edu	planetlab1.ifi.uio.no	America_to_Europe	MDT_to_CEST
planetlab1.cs.du.edu	planetlab2.utt.fr	America_to_Europe	MDT_to_CEST
planetlab1.cs.okstate.edu	planetlab2.cesnet.cz	America_to_Europe	CDT_to_CEST
planetlab1.cs.okstate.edu	planetlab1.ifi.uio.no	America_to_Europe	CDT_to_CEST
planetlab1.cs.okstate.edu	planetlab2.inf.ethz.ch	America_to_Europe	CDT_to_CEST
planetlab1.dtc.umn.edu	plab1.cs.msu.ru	America_to_Europe	CDT_to_EEST
planetlab1.dtc.umn.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	CDT_to_CEST
planetlab1.pop-mg.rnp.br	planetlab3.mini.pw.edu.pl	America_to_Europe	BRT_to_CEST
planetlab1.unr.edu	planetlab2.utt.fr	America_to_Europe	PDT_to_CEST
planetlab2.citadel.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	EDT_to_CEST
planetlab2.citadel.edu	plab1.cs.msu.ru	America_to_Europe	EDT_to_EEST
planetlab2.cs.cornell.edu	ple2.cesnet.cz	America_to_Europe	EDT_to_CEST
planetlab2.cs.cornell.edu	planetlab2.inf.ethz.ch	America_to_Europe	EDT_to_CEST
planetlab2.cs.du.edu	planetlab1.ifi.uio.no	America_to_Europe	MDT_to_CEST
planetlab2.cs.du.edu	planetlab2.utt.fr	America_to_Europe	MDT_to_CEST
planetlab2.cs.du.edu	merkur.planetlab.haw-hamburg.de	America_to_Europe	MDT_to_CEST
planetlab2.cs.du.edu	planetlab4.inf.ethz.ch	America_to_Europe	MDT_to_CEST
planetlab2.cs.ubc.ca	merkur.planetlab.haw-hamburg.de	America_to_Europe	PDT_to_CEST
planetlab2.cs.ubc.ca	plab1.cs.msu.ru	America_to_Europe	PDT_to_EEST
planetlab2.cs.uoregon.edu	plab1.cs.msu.ru	America_to_Europe	PDT_to_EEST
planetlab2.cs.uoregon.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	PDT_to_CES
planetlab2.cs.uoregon.edu	stella.planetlab.ntua.gr	America_to_Europe	PDT_to_EEST
planetlab2.pop-mg.rnp.br	plab1.cs.msu.ru	America_to_Europe	BRT_to_EEST
planetlab2.utdallas.edu	stella.planetlab.ntua.gr	America_to_Europe	CDT_to_EEST
planetlab2.utdallas.edu	planetlab2.tlm.unavarra.es	America_to_Europe	CDT_to_CEST

planetlab2.utdallas.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	CDT_to_CEST
planetlab3.eecs.umich.edu	plab1.cs.msu.ru	America_to_Europe	EDT_to_EEST
planetlab3.eecs.umich.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	EDT_to_CES
planetlab3.eecs.umich.edu	planetlab1.virtues.fi	America_to_Europe	EDT_to_EEST
planetlab5.eecs.umich.edu	plab1.cs.msu.ru	America_to_Europe	EDT_to_EEST
planetlab5.eecs.umich.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	EDT_to_CES
salt.planetlab.cs.umd.edu	plab1.cs.msu.ru	America_to_Europe	EDT_to_EEST
salt.planetlab.cs.umd.edu	planet-labnode1.netgroup.uniroma2.it	America_to_Europe	EDT_to_CEST
node2.planetlab.mathcs.emory.edu	planetlab2.cs.ubc.ca	America_to_America	EDT_to_PDT
pl1.cs.montana.edu	planetlab2.rutgers.edu	America_to_America	MDT_to_EDT
pl2.ucs.indiana.edu	planetlab2.rutgers.edu	America_to_America	EDT_to_EDT
planetlab-2.cse.ohio-state.edu	planetlab2.cs.ubc.ca	America_to_America	EDT_to_PDT
planetlab-5.eecs.cwru.edu	planetlab2.cs.ubc.ca	America_to_America	EDT_to_PDT
planetlab1.cs.du.edu	planetlab1.cs.okstate.edu	America_to_America	MDT_to_CDT
planetlab1.cs.okstate.edu	pl1.cs.montana.edu	America_to_America	CDT_to_MDT
planetlab1.pop-mg.rnp.br	planetlab1.dtc.umn.edu	America_to_America	BRT_to_CDT
planetlab1.unr.edu	planetlab2.pop-mg.rnp.br	America_to_America	PDT_to_BRT
planetlab1.unr.edu	planetlab1.dtc.umn.edu	America_to_America	PDT_to_CDT
planetlab2.cs.cornell.edu	planetlab1.cs.okstate.edu	America_to_America	EDT_to_CDT
planetlab2.cs.cornell.edu	planetlab2.cs.ubc.ca	America_to_America	EDT_to_PDT
planetlab2.cs.ubc.ca	planetlab2.rutgers.edu	America_to_America	PDT_to_EDT
planetlab2.cs.uoregon.edu	planetlab-2.cse.ohio-state.edu	America_to_America	PDT_to_EDT
planetlab2.pop-mg.rnp.br	planetlab-2.cse.ohio-state.edu	America_to_America	BRT_to_EDT
planetlab5.eecs.umich.edu	planetlab1.cs.okstate.edu	America_to_America	EDT_to_CDT
salt.planetlab.cs.umd.edu	planetlab1.cs.okstate.edu	America_to_America	EDT_to_CDT

node2.planetlab.mathcs.emory.edu	pl2.6test.edu.cn	America_to_Asia	EDT_to_CST
planetlab-2.cse.ohio-state.edu	pl2.6test.edu.cn	America_to_Asia	EDT_to_CST
planetlab-5.eecs.cwru.edu	planetlab-js2.cert.org.cn	America_to_Asia	EDT_to_CST
planetlab-5.eecs.cwru.edu	pl1.eng.monash.edu.au	America_to_Asia	EDT_to_AEST
planetlab1.dtc.umn.edu	planetlab-js2.cert.org.cn	America_to_Asia	CDT_to_CST
planetlab1.dtc.umn.edu	pl2.pku.edu.cn	America_to_Asia	CDT_to_CST
planetlab1.pop-mg.rnp.br	pl2.6test.edu.cn	America_to_Asia	BRT_to_CST
planetlab1.pop-mg.rnp.br	planetlab1.cs.otago.ac.nz	America_to_Asia	BRT_to_NST
planetlab1.unr.edu	pl2.6test.edu.cn	America_to_Asia	PDT_to_CST
planetlab2.citadel.edu	pl2.pku.edu.cn	America_to_Asia	EDT_to_CST
planetlab2.citadel.edu	planetlab-js2.cert.org.cn	America_to_Asia	EDT_to_CST
planetlab2.cs.ubc.ca	pl2.pku.edu.cn	America_to_Asia	PDT_to_CST
planetlab2.pop-mg.rnp.br	pl2.pku.edu.cn	America_to_Asia	BRT_to_CST
planetlab2.pop-mg.rnp.br	planetlab-js1.cert.org.cn	America_to_Asia	BRT_to_CST
planetlab2.utdallas.edu	pl1.eng.monash.edu.au	America_to_Asia	CDT_to_AEST
planetlab3.eecs.umich.edu	planetlab-js2.cert.org.cn	America_to_Asia	EDT_to_CST
planetlab5.eecs.umich.edu	planetlab-js2.cert.org.cn	America_to_Asia	EDT_to_CST
salt.planetlab.cs.umd.edu	planetlab-js2.cert.org.cn	America_to_Asia	EDT_to_CST
mars.planetlab.haw-hamburg.de	pl2.6test.edu.cn	Europe_to_Asia	CEST_to_CST
merkur.planetlab.haw-hamburg.de	planetlab-js2.cert.org.cn	Europe_to_Asia	CEST_to_CST
merkur.planetlab.haw-hamburg.de	pl2.pku.edu.cn	Europe_to_Asia	CEST_to_CST
merkur.planetlab.haw-hamburg.de	planetlab2.cs.otago.ac.nz	Europe_to_Asia	CEST_to_NST
plab1.cs.msu.ru	planetlab2.cs.otago.ac.nz	Europe_to_Asia	EEST_to_NST
planet-labnode1.netgroup.uniroma2.it	pl2.6test.edu.cn	Europe_to_Asia	CEST_to_CST
planetlab1.ifi.uio.no	planetlab-js2.cert.org.cn	Europe_to_Asia	CEST_to_CST

planetlab1.ifi.uio.no	pl1.eng.monash.edu.au	Europe_to_Asia	CEST_to_AEST
planetlab2.cesnet.cz	planetlab2.cs.otago.ac.nz	Europe_to_Asia	CEST_to_NST
planetlab2.cesnet.cz	pl1.eng.monash.edu.au	Europe_to_Asia	CEST_to_AEST
planetlab2.inf.ethz.ch	pl2.pku.edu.cn	Europe_to_Asia	CEST_to_CST
planetlab2.inf.ethz.ch	planetlab2.cs.otago.ac.nz	Europe_to_Asia	CEST_to_NST
planetlab2.inf.ethz.ch	pl1.eng.monash.edu.au	Europe_to_Asia	CEST_to_AEST
planetlab3.inf.ethz.ch	planetlab2.cs.otago.ac.nz	Europe_to_Asia	CEST_to_NST
planetlab3.inf.ethz.ch	pl1.eng.monash.edu.au	Europe_to_Asia	CEST_to_AEST
planetlab3.mini.pw.edu.pl	pl2.pku.edu.cn	Europe_to_Asia	CEST_to_CST
planetlab4.inf.ethz.ch	pl2.pku.edu.cn	Europe_to_Asia	CEST_to_CST
planetlab4.inf.ethz.ch	planetlab2.cs.otago.ac.nz	Europe_to_Asia	CEST_to_NST
planetlab4.inf.ethz.ch	pl1.eng.monash.edu.au	Europe_to_Asia	CEST_to_AEST
ple2.cesnet.cz	pl2.pku.edu.cn	Europe_to_Asia	CEST_to_CST
ple2.cesnet.cz	planetlab2.cs.otago.ac.nz	Europe_to_Asia	CEST_to_NST
ple2.cesnet.cz	pl1.eng.monash.edu.au	Europe_to_Asia	CEST_to_AEST
stella.planetlab.ntua.gr	planetlab-js2.cert.org.cn	Europe_to_Asia	EEST_to_CST
mars.planetlab.haw-hamburg.de	planetlab1.cs.okstate.edu	Europe_to_America	CEST_to_CDT
mars.planetlab.haw-hamburg.de	planetlab2.cs.ubc.ca	Europe_to_America	CEST_to_PDT
merkur.planetlab.haw-hamburg.de	planetlab1.unr.edu	Europe_to_America	CEST_to_PDT
plab1.cs.msu.ru	node2.planetlab.mathcs.emory.edu	Europe_to_America	EEST_to_EDT
plab1.cs.msu.ru	planetlab-2.cse.ohio-state.edu	Europe_to_America	EEST_to_EDT
plab1.cs.msu.ru	planetlab1.pop-mg.rnp.br	Europe_to_America	EEST_to_BRT
planet-labnode1.netgroup.uniroma2.it	planetlab2.cs.ubc.ca	Europe_to_America	CEST_to_PDT
planetlab-coffee.ait.ie	planetlab2.cs.cornell.edu	Europe_to_America	IST_to_EDT
planetlab-coffee.ait.ie	planetlab-2.cse.ohio-state.edu	Europe_to_America	IST_to_EDT

planetlab-coffee.ait.ie	planetlab1.cs.okstate.edu	Europe_to_America	IST_to_CDT
planetlab-coffee.ait.ie	planetlab1.unr.edu	Europe_to_America	IST_to_PDT
planetlab1.cesnet.cz	planetlab04.cs.washington.edu	Europe_to_America	CEST_to_PDT
planetlab1.cesnet.cz	planetlab1.cs.okstate.edu	Europe_to_America	CEST_to_CDT
planetlab1.ifi.uio.no	planetlab2.rutgers.edu	Europe_to_America	CEST_to_EDT
planetlab1.ifi.uio.no	planetlab2.citadel.edu	Europe_to_America	CEST_to_EDT
planetlab1.net.in.tum.de	planetlab04.cs.washington.edu	Europe_to_America	CEST_to_PDT
planetlab1.net.in.tum.de	planetlab2.rutgers.edu	Europe_to_America	CEST_to_EDT
planetlab1.net.in.tum.de	planetlab2.pop-mg.rnp.br	Europe_to_America	CEST_to_BRT
planetlab1.net.in.tum.de	pl1.cs.montana.edu	Europe_to_America	CEST_to_MDT
planetlab1.virtues.fi	planetlab2.pop-mg.rnp.br	Europe_to_America	EEST_to_BRT
planetlab1.virtues.fi	planetlab1.unr.edu	Europe_to_America	EEST_to_PDT
planetlab1.virtues.fi	planetlab2.cs.cornell.edu	Europe_to_America	EEST_to_EDT
planetlab2.cesnet.cz	planetlab2.rutgers.edu	Europe_to_America	CEST_to_EDT
planetlab2.cesnet.cz	planetlab2.cs.ubc.ca	Europe_to_America	CEST_to_PDT
planetlab2.tlm.unavarra.es	planetlab1.dtc.umn.edu	Europe_to_America	CEST_to_CDT
planetlab2.tlm.unavarra.es	planetlab04.cs.washington.edu	Europe_to_America	CEST_to_PDT
planetlab2.tlm.unavarra.es	planetlab1.cs.okstate.edu	Europe_to_America	CEST_to_CDT
planetlab2.utt.fr	planetlab1.cs.okstate.edu	Europe_to_America	CEST_to_CDT
planetlab2.utt.fr	planetlab2.cs.ubc.ca	Europe_to_America	CEST_to_PDT
planetlab3.cesnet.cz	planetlab04.cs.washington.edu	Europe_to_America	CEST_to_PDT
planetlab3.cesnet.cz	planetlab2.rutgers.edu	Europe_to_America	CEST_to_EDT
planetlab3.cesnet.cz	pl1.cs.montana.edu	Europe_to_America	CEST_to_MDT
planetlab3.inf.ethz.ch	planetlab2.cs.du.edu	Europe_to_America	CEST_to_MDT
planetlab3.mini.pw.edu.pl	planetlab-2.cse.ohio-state.edu	Europe_to_America	CEST_to_EDT

planetlab3.mini.pw.edu.pl	planetlab2.cs.ubc.ca	Europe_to_America	CEST_to_PDT
planetlab3.mini.pw.edu.pl	planetlab-5.eecs.cwru.edu	Europe_to_America	CEST_to_EDT
planetlab4.mini.pw.edu.pl	planetlab-2.cse.ohio-state.edu	Europe_to_America	CEST_to_EDT
planetlab4.mini.pw.edu.pl	planetlab-5.eecs.cwru.edu	Europe_to_America	CEST_to_EDT
planetlab4.mini.pw.edu.pl	pl2.ucs.indiana.edu	Europe_to_America	CEST_to_EDT
planetlab4.mini.pw.edu.pl	planetlab2.citadel.edu	Europe_to_America	CEST_to_EDT
ple2.cesnet.cz	planetlab1.cs.okstate.edu	Europe_to_America	CEST_to_CDT
stella.planetlab.ntua.gr	planetlab2.cs.ubc.ca	Europe_to_America	EEST_to_PDT
mars.planetlab.haw-hamburg.de	planetlab2.inf.ethz.ch	Europe_to_Europe	CEST_to_CEST
planet-labnode1.netgroup.uniroma2.it	ple2.cesnet.cz	Europe_to_Europe	CEST_to_CEST
planet-labnode1.netgroup.uniroma2.it	planetlab2.inf.ethz.ch	Europe_to_Europe	CEST_to_CEST
planetlab1.cesnet.cz	planetlab1.ifi.uio.no	Europe_to_Europe	CEST_to_CEST
planetlab1.cesnet.cz	planetlab2.utt.fr	Europe_to_Europe	CEST_to_CEST
planetlab1.virtues.fi	planetlab2.tlm.unavarra.es	Europe_to_Europe	EEST_to_CEST
planetlab2.inf.ethz.ch	planetlab1.ifi.uio.no	Europe_to_Europe	CEST_to_CEST
planetlab2.tlm.unavarra.es	planetlab1.ifi.uio.no	Europe_to_Europe	CEST_to_CEST
planetlab2.utt.fr	ple2.cesnet.cz	Europe_to_Europe	CEST_to_CEST
planetlab2.utt.fr	planetlab2.inf.ethz.ch	Europe_to_Europe	CEST_to_CEST
planetlab3.cesnet.cz	planetlab2.utt.fr	Europe_to_Europe	CEST_to_CEST
planetlab3.inf.ethz.ch	planetlab1.ifi.uio.no	Europe_to_Europe	CEST_to_CEST
planetlab4.inf.ethz.ch	planetlab1.ifi.uio.no	Europe_to_Europe	CEST_to_CEST
stella.planetlab.ntua.gr	ple2.cesnet.cz	Europe_to_Europe	EEST_to_CEST
stella.planetlab.ntua.gr	planetlab2.inf.ethz.ch	Europe_to_Europe	EEST_to_CEST
pl1.eng.monash.edu.au	stella.planetlab.ntua.gr	Asia_to_Europe	AEST_to_EEST
pl2.6test.edu.cn	plab1.cs.msu.ru	Asia_to_Europe	CST_to_EEST

planet1.pnl.nitech.ac.jp	planetlab1.virtues.fi	Asia_to_Europe	JST_to_EEST
planet1.pnl.nitech.ac.jp	plab1.cs.msu.ru	Asia_to_Europe	JST_to_EEST
planet2.pnl.nitech.ac.jp	planetlab1.virtues.fi	Asia_to_Europe	JST_to_EEST
planet2.pnl.nitech.ac.jp	planetlab1.net.in.tum.de	Asia_to_Europe	JST_to_CEST
planetlab-js1.cert.org.cn	planetlab2.tlm.unavarra.es	Asia_to_Europe	CST_to_CEST
planetlab-js1.cert.org.cn	stella.planetlab.ntua.gr	Asia_to_Europe	CST_to_EEST
planetlab-js1.cert.org.cn	planetlab1.ifi.uio.no	Asia_to_Europe	CST_to_CEST
planetlab-js2.cert.org.cn	planetlab2.inf.ethz.ch	Asia_to_Europe	CST_to_CEST
planetlab1.cs.otago.ac.nz	stella.planetlab.ntua.gr	Asia_to_Europe	NST_to_EEST
planetlab1.cs.otago.ac.nz	planetlab2.utt.fr	Asia_to_Europe	NST_to_CEST
planetlab1.cs.otago.ac.nz	merkur.planetlab.haw-hamburg.de	Asia_to_Europe	NST_to_CEST
planetlab1.cs.otago.ac.nz	planetlab4.inf.ethz.ch	Asia_to_Europe	NST_to_CEST
planetlab2.cs.otago.ac.nz	mars.planetlab.haw-hamburg.de	Asia_to_Europe	NST_to_CEST
planetlab2.cs.otago.ac.nz	planet-labnode1.netgroup.uniroma2.it	Asia_to_Europe	NST_to_CEST
planetlab2.cs.otago.ac.nz	stella.planetlab.ntua.gr	Asia_to_Europe	NST_to_EEST
planetlab2.cs.otago.ac.nz	planetlab2.utt.fr	Asia_to_Europe	NST_to_CEST
pl1.eng.monash.edu.au	planetlab-js2.cert.org.cn	Asia_to_Asia	AEST_to_CST
pl2.6test.edu.cn	planetlab2.cs.otago.ac.nz	Asia_to_Asia	CST_to_NST
pl2.6test.edu.cn	pl1.eng.monash.edu.au	Asia_to_Asia	CST_to_AEST
pl2.pku.edu.cn	pl1.eng.monash.edu.au	Asia_to_Asia	CST_to_AEST
planet1.pnl.nitech.ac.jp	pl2.pku.edu.cn	Asia_to_Asia	JST_to_CST
planet1.pnl.nitech.ac.jp	planetlab2.cs.otago.ac.nz	Asia_to_Asia	JST_to_NST
planetlab-js1.cert.org.cn	planetlab2.cs.otago.ac.nz	Asia_to_Asia	CST_to_NST
planetlab-js2.cert.org.cn	pl2.6test.edu.cn	Asia_to_Asia	CST_to_CST
pl1.eng.monash.edu.au	planetlab2.rutgers.edu	Asia_to_America	AEST_to_EDT

pl2.6test.edu.cn	planetlab2.rutgers.edu	Asia_to_America	CST_to_EDT
pl2.pku.edu.cn	planetlab2.rutgers.edu	Asia_to_America	CST_to_EDT
pl2.pku.edu.cn	planetlab-2.cse.ohio-state.edu	Asia_to_America	CST_to_EDT
pl2.pku.edu.cn	planetlab2.cs.cornell.edu	Asia_to_America	CST_to_EDT
planet2.pnl.nitech.ac.jp	planetlab-2.cse.ohio-state.edu	Asia_to_America	JST_to_EDT
planet2.pnl.nitech.ac.jp	planetlab2.cs.cornell.edu	Asia_to_America	JST_to_EDT
planetlab-js2.cert.org.cn	planetlab2.pop-mg.rnp.br	Asia_to_America	CST_to_BRT

Correlated Traceroute Result

Router1(R1)	Router2(R2)	ASN-R1	ASN-R2	Link_type	Network_pos	Node pairs	Source-asn	dest-asn	Corr-coff
9	10	AS11537	AS11537	Intra_domain	Transit	mars.planetlab.haw-hamburg.de_to_planetlab1.cs.okstate.edu	AS680	AS5078	0.933400275764225
7	8	AS20965	AS20965	Intra_domain	Transit	mars.planetlab.haw-hamburg.de_to_planetlab1.cs.okstate.edu	AS680	AS5078	0.938308725886617
9	10	AS20965	AS20965	Intra_domain	Transit	node2.planetlab.mathcs.emory.edu_to_ple2.cesnet.cz	AS3512	AS2852	0.900032336965431
7	8	AS20965	AS20965	Intra_domain	Transit	node2.planetlab.mathcs.emory.edu_to_ple2.cesnet.cz	AS3512	AS2852	0.999465455993886
4	5	AS11537	AS11537	Intra_domain	Transit	pl1.cs.montana.edu_to_merkur.planetlab.haw-hamburg.de	AS13476	AS680	0.987540994176509
12	13	AS20965	AS20965	Intra_domain	Transit	pl1.cs.montana.edu_to_merkur.planetlab.haw-hamburg.de	AS13476	AS680	0.999985332322064
4	5	AS7575	AS7575	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.611655298764645
6	7	AS7575	AS7575	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.700638407542856
15	16	AS20965	AS20965	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.755691114341418
13	14	AS20965	AS20965	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.891816814332189
17	18	AS20965	AS20965	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.996801210802361
9	10	AS11537	AS11537	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.999312722686815
7	8	AS7575	AS101	Inter_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.999916937841621
8	9	AS101	AS11537	Inter_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.999990922385921
11	12	AS11537	AS20965	Inter_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	0.999998577806617
25	26	AS174	AS174	Intra_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	1
29	30	AS3323	AS20965	Inter_domain	Transit	pl1.eng.monash.edu.au_to_stella.planetlab.ntua.gr	AS56132	AS3323	1
15	16	AS20965	AS5408	Inter_domain	Transit	pl1.ucs.indiana.edu_to_stella.planetlab.ntua.gr	AS87	AS3323	0.999655928916299
12	13	AS20965	AS20965	Intra_domain	Transit	pl1.ucs.indiana.edu_to_stella.planetlab.ntua.gr	AS87	AS3323	0.999992696566316
18	19	AS174	AS3323	Inter_domain	Transit	pl1.ucs.indiana.edu_to_stella.planetlab.ntua.gr	AS87	AS3323	1
8	9	AS20965	AS20965	Intra_domain	Transit	pl2.ucs.indiana.edu_to_merkur.planetlab.haw-hamburg.de	AS87	AS680	0.56207766528503
19	20	AS766	AS766	Intra_domain	Last_mile	pl2.ucs.indiana.edu_to_planetlab2.tlm.unavarra.es	AS87	AS766	0.703152479348513
16	17	AS766	AS766	Intra_domain	Last_mile	pl2.ucs.indiana.edu_to_planetlab2.tlm.unavarra.es	AS87	AS766	0.908570517890833
15	16	AS766	AS766	Intra_domain	Last_mile	pl2.ucs.indiana.edu_to_planetlab2.tlm.unavarra.es	AS87	AS766	0.999961958071532
14	15	AS559	AS559	Intra_domain	Last_mile	pl2.ucs.indiana.edu_to_planetlab4.inf.ethz.ch	AS87	AS559	0.999998341466699
9	10	AS11537	AS11537	Intra_domain	Transit	planetlab-2.cse.ohio-state.edu_to_pl2.6test.edu.cn	AS159	AS23910	0.514013350799751
24	25	AS23910	AS23910	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_pl2.6test.edu.cn	AS159	AS23910	0.578100504417482
8	9	AS11537	AS11537	Intra_domain	Transit	planetlab-2.cse.ohio-state.edu_to_pl2.6test.edu.cn	AS159	AS23910	0.733967965120264
23	24	AS23910	AS23910	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_pl2.6test.edu.cn	AS159	AS23910	0.883627860028192
10	11	AS20965	AS20965	Intra_domain	Transit	planetlab-2.cse.ohio-state.edu_to_planetlab2.inf.ethz.ch	AS159	AS559	0.518564237158281
19	20	AS559	AS559	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_planetlab2.inf.ethz.ch	AS159	AS559	0.774775476169892
23	24	AS559	AS559	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_planetlab2.inf.ethz.ch	AS159	AS559	0.868983306399728
22	23	AS2852	AS2852	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_ple2.cesnet.cz	AS159	AS2852	0.81203234629168
20	21	AS2852	AS2852	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_ple2.cesnet.cz	AS159	AS2852	0.865766700902611
21	22	AS2852	AS2852	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_ple2.cesnet.cz	AS159	AS2852	0.921661449635271
18	19	AS2852	AS2852	Intra_domain	Last_mile	planetlab-2.cse.ohio-state.edu_to_ple2.cesnet.cz	AS159	AS2852	0.943731531302794
20	21	AS174	AS174	Intra_domain	Transit	planetlab-js1.cert.org.cn_to_planetlab2.tlm.unavarra.es	AS4134	AS766	0.568885314617022
23	24	AS174	AS174	Intra_domain	Transit	planetlab-js1.cert.org.cn_to_planetlab2.tlm.unavarra.es	AS4134	AS766	0.571063041133371
6	7	AS4134	AS4134	Intra_domain	Last_mile	planetlab-js1.cert.org.cn_to_planetlab2.tlm.unavarra.es	AS4134	AS766	0.650264726293847
17	18	AS174	AS174	Intra_domain	Transit	planetlab-js1.cert.org.cn_to_planetlab2.tlm.unavarra.es	AS4134	AS766	0.931174925265074
4	5	AS49597	AS4134	Inter_domain	Transit	planetlab-js1.cert.org.cn_to_planetlab2.tlm.unavarra.es	AS4134	AS766	0.999567151978398
9	10	AS4134	AS174	Inter_domain	Transit	planetlab-js1.cert.org.cn_to_stella.planetlab.ntua.gr	AS4134	AS3323	0.60933856570042
21	22	AS174	AS174	Intra_domain	Transit	planetlab-js1.cert.org.cn_to_stella.planetlab.ntua.gr	AS4134	AS3323	0.63444439623229
6	7	AS4134	AS49597	Inter_domain	Transit	planetlab-js1.cert.org.cn_to_stella.planetlab.ntua.gr	AS4134	AS3323	0.998086770207016
6	7	AS4134	AS4134	Intra_domain	Last_mile	planetlab-js2.cert.org.cn_to_planetlab2.inf.ethz.ch	AS4134	AS559	0.523852120265762
4	5	AS49597	AS4134	Inter_domain	Transit	planetlab-js2.cert.org.cn_to_planetlab2.inf.ethz.ch	AS4134	AS559	0.999228807973143
27	28	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.531559164885293
25	26	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.542551634832966
23	24	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.663717744643837
26	27	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.858005837913624
28	29	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.89706799612332
6	7	AS11537	AS11537	Intra_domain	Transit	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.946001017154462
29	30	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cesnet.cz_to_planetlab1.cs.okstate.edu	AS2852	AS5078	0.99859428076938
11	12	AS20965	AS20965	Intra_domain	Transit	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch	AS5078	AS559	0.560781422516707
6	7	AS5078	AS5078	Intra_domain	Last_mile	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch	AS5078	AS559	0.821946258064024
28	29	AS559	AS559	Intra_domain	Last_mile	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch	AS5078	AS559	0.870646886596061
15	16	AS559	AS559	Intra_domain	Last_mile	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch	AS5078	AS559	0.951805847718884
13	14	AS20965	AS20965	Intra_domain	Transit	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch	AS5078	AS559	0.98478784829175
9	10	AS11537	AS20965	Inter_domain	Transit	planetlab1.cs.okstate.edu_to_planetlab2.inf.ethz.ch	AS5078	AS559	0.999736859197319
5	6	AS38022	AS7575	Inter_domain	Transit	planetlab1.cs.otago.ac.nz_to_stella.planetlab.ntua.gr	AS38305	AS3323	0.799512046271256
11	12	AS101	AS11537	Inter_domain	Transit	planetlab1.cs.otago.ac.nz_to_stella.planetlab.ntua.gr	AS38305	AS3323	0.98003078490474
20	21	AS20965	AS20965	Intra_domain	Transit	planetlab1.cs.otago.ac.nz_to_stella.planetlab.ntua.gr	AS38305	AS3323	0.988680716214557
14	15	AS11537	AS20965	Inter_domain	Transit	planetlab1.cs.otago.ac.nz_to_stella.planetlab.ntua.gr	AS38305	AS3323	0.996509909558949
15	16	AS49597	AS4134	Inter_domain	Transit	planetlab1.dtc.umn.edu_to_planetlab-js2.cert.org.cn	AS57	AS4134	0.659392558106116
9	10	AS11164	AS11164	Intra_domain	Transit	planetlab1.dtc.umn.edu_to_planetlab-js2.cert.org.cn	AS57	AS4134	0.667473762260682
14	15	AS4134	AS49597	Inter_domain	Transit	planetlab1.dtc.umn.edu_to_planetlab-js2.cert.org.cn	AS57	AS4134	0.723285711374078
13	14	AS49597	AS4134	Inter_domain	Transit	planetlab1.dtc.umn.edu_to_planetlab-js2.cert.org.cn	AS57	AS4134	0.856052941074462
21	22	AS766	AS766	Intra_domain	Last_mile	planetlab1.virtues.fi_to_planetlab2.tlm.unavarra.es	AS47605	AS766	0.967063734261476
9	10	AS2603	AS47872	Inter_domain	Transit	planetlab1.virtues.fi_to_planetlab2.tlm.unavarra.es	AS47605	AS766	0.986407282991879
7	8	AS3754	AS11537	Inter_domain	Transit	planetlab2.cs.cornell.edu_to_ple2.cesnet.cz	AS26	AS2852	0.790951427553443
15	16	AS680	AS559	Inter_domain	Transit	planetlab2.cs.du.edu_to_merkur.planetlab.haw-hamburg.de	AS14041	AS680	0.818321881780382
14	15	AS680	AS680	Intra_domain	Last_mile	planetlab2.cs.du.edu_to_merkur.planetlab.haw-hamburg.de	AS14041	AS680	0.999967518145
13	14	AS680	AS680	Intra_domain	Last_mile	planetlab2.cs.du.edu_to_merkur.planetlab.haw-hamburg.de	AS14041	AS680	0.99998809473464
11	12	AS2603	AS2603	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.571263852740072
9	10	AS2603	AS2603	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.748993630974505
17	18	AS2848	AS2848	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.913695559638673
5	6	AS11537	AS11537	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.981417069665777
12	13	AS2603	AS2603	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.99016607228739
2	3	AS3582	none	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.996285476523749
4	5	AS11537	AS11537	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.999582899227326
8	9	AS2603	AS2603	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.999727869088249
3	4	none	AS11537	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_plab1.cs.msu.ru	AS3582	AS2848	0.999995477286086
18	19	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.610937138089236
20	21	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.882075350465838
17	18	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.914342246676446
8	9	AS20965	AS20965	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.989266858406753
6	7	AS11537	AS20965	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.999777602884492
4	5	AS11537	AS11537	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.999867467484379
3	4	none	AS11537	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	0.999996183255074
22	23	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	1
24	25	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	1
26	27	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	1
27	28	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	1
28	29	AS137	AS137	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS3582	AS137	1
17	18	AS3323	AS3323	Intra_domain	Last_mile	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.550974147594531
8	9	AS20965	AS20965	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.879671014577892
15	16	AS20965	AS5408	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.927420365924471
2	3	AS3582	none	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.99660793888375
12	13	AS20965	AS20965	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.998983723557582
6	7	AS11537	AS20965	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.999779423083993
4	5	AS11537	AS11537	Intra_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.999879969372592

3	4	none	AS11537	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	0.999994569229884
25	26	AS20965	AS5408	Inter_domain	Transit	planetlab2.cs.uoregon.edu_to_stella.planetlab.ntua.gr	AS3582	AS3323	1
7	8	AS20965	AS20965	Intra_domain	Transit	planetlab2.tlm.unavarra.es_to_planetlab1.dtc.umn.edu	AS766	AS57	0.55029420277329
16	17	AS57	AS57	Intra_domain	Last_mile	planetlab2.tlm.unavarra.es_to_planetlab1.dtc.umn.edu	AS766	AS57	0.597895850050854
15	16	AS57	AS57	Intra_domain	Last_mile	planetlab2.tlm.unavarra.es_to_planetlab1.dtc.umn.edu	AS766	AS57	0.667320671038322
2	3	AS15557	AS15557	Intra_domain	Transit	planetlab2.utt.fr_to_planetlab1.cs.okstate.edu	AS2200	AS5078	0.694999154941189
5	6	None	AS2200	Inter_domain	Transit	planetlab2.utt.fr_to_planetlab1.cs.okstate.edu	AS2200	AS5078	0.963107173646707
13	14	AS11537	AS5078	Inter_domain	Transit	planetlab2.utt.fr_to_planetlab1.cs.okstate.edu	AS2200	AS5078	0.989338001882767
5	6	AS20965	AS11537	Inter_domain	Transit	planetlab3.cesnet.cz_to_planetlab04.cs.washington.edu	AS2852	AS73	0.708934458497623
8	9	AS11537	AS11537	Intra_domain	Transit	planetlab3.cesnet.cz_to_planetlab04.cs.washington.edu	AS2852	AS73	0.818349497236063
6	7	AS11537	AS11537	Intra_domain	Transit	planetlab3.cesnet.cz_to_planetlab04.cs.washington.edu	AS2852	AS73	0.941821902228253
21	22	AS159	AS159	Intra_domain	Last_mile	planetlab4.mini.pw.edu.pl_to_planetlab-2.cse.ohio-state.edu	AS12464	AS159	0.567679832696613
16	17	AS600	AS600	Intra_domain	Transit	planetlab4.mini.pw.edu.pl_to_planetlab-2.cse.ohio-state.edu	AS12464	AS159	0.600716322238201
22	23	AS159	AS159	Intra_domain	Last_mile	planetlab4.mini.pw.edu.pl_to_planetlab-2.cse.ohio-state.edu	AS12464	AS159	0.970125115398838
13	14	AS4134	AS4134	Intra_domain	Last_mile	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.788803893413037
19	20	AS237	AS237	Intra_domain	Transit	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.796713835108847
6	7	AS237	AS11164	Inter_domain	Transit	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.798573949511338
20	21	AS237	AS4134	Inter_domain	Transit	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.801635836343897
17	18	AS4134	AS4134	Intra_domain	Last_mile	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.907308542654655
21	22	AS4134	AS237	Inter_domain	Transit	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.962924448296706
7	8	AS11164	AS11164	Intra_domain	Transit	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.996691057873774
11	12	AS4134	AS4134	Intra_domain	Last_mile	planetlab5.eecs.umich.edu_to_planetlab-js2.cert.org.cn	AS36375	AS4134	0.99996372797571
20	21	AS137	AS137	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS27	AS137	0.505121678142989
15	16	AS137	AS137	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS27	AS137	0.740900376492062
21	22	AS137	AS137	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS27	AS137	0.97377542604463
27	28	AS137	AS137	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS27	AS137	1
29	30	AS137	AS137	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planet-lab-node1.netgroup.uniroma2.it	AS27	AS137	1
20	21	AS4134	AS4134	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planetlab-js2.cert.org.cn	AS27	AS4134	0.621231608793494
18	19	AS4134	AS4134	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planetlab-js2.cert.org.cn	AS27	AS4134	0.712881919831286
25	26	AS4134	AS4134	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planetlab-js2.cert.org.cn	AS27	AS4134	0.718867892643994
21	22	AS4134	AS4134	Intra_domain	Last_mile	salt.planetlab.cs.umd.edu_to_planetlab-js2.cert.org.cn	AS27	AS4134	0.871838693227803
16	17	AS174	AS174	Intra_domain	Transit	salt.planetlab.cs.umd.edu_to_planetlab-js2.cert.org.cn	AS27	AS4134	0.986795495697199