Department of
Microelectronics and Information Technology

SWEDISH
INSTITUTE OF
COMPUTER
SCIENCE
SICS

Existence, Identification and Stability of Elephant Flows in IP Traffic

Cecilia Borg

cilla@kth.se

September 5 2002

Bengt Ahlgren, SICS AB, *Supervisor*

Gunnar Karlsson, IMIT KTH, *Examiner*

VETENSKAP
OCH
KONST
KTH

Abstract

Traffic in the Internet today is routed on the shortest path to the destination. This is considered to be the quickest path, but if traffic congestion occurs on the route, packets are dropped and the traffic slows down due to the retransmission of missing packets. If the network resources could be more evenly utilised, some congestions could be avoided and the problem with retransmissions could be reduced. In order to balance the load evenly over a network, the load variation has to be known and predictable.

Other studies of IP traffic have shown that a small number of flows carry the main part of the network traffic; these flows are referred to as elephants. This finding is studied in this report and the stability of these flows is examined. By aggregation with respect to the source and destination networks of the traffic, individual flows are easily identified. This report also discusses how to identify large flows during runtime in order to use their properties when calculating the stability for the future traffic demand. The traffic prediction is based on analysis of logged Internet traffic. The report concludes that the phenomenon with elephant and mice flows can be observed when aggregating traffic artificially by different lengths of their network prefixes. When calculating future stability of flows the network aggregation does not have a major impact.

Preface

This master thesis is done as a part of the IP load optimisation project conducted by SICS AB and Telia Research AB. SICS AB is a non-profit research institute with approximately 100 researchers based in Stockholm, Västerås, Uppsala and Gothenburg.

The writer is conducting her last year of the Master of Science program in Computer Science at KTH, the Royal Institute of Technology in Stockholm.

People that have contributed with knowledge and support during the research and whom the writer would like to thank are: Dr.Bengt Ahlgren, SICS AB; Prof. Gunnar Karlsson, IMIT KTH; Prof. Ingemar Kaj, Dep. of Mathematics Uppsala University; M. Sc. student Johannes Borgström; M. Sc. student Tomas Olsson; and all members of the CNA laboratory at SICS AB.

Table of Contents

1 Introduction

Internet users want high reliable and throughput on their traffic. Traffic problems with Internet today include congestion and ineffective utilisation of network resources. This leads to retransmissions and lower throughput for the users of the network. Backbone operators have so far acted on the problem by applying rules of thumb. In reaction to the increasing bandwidth demand they have doubled their network bandwidth capacity every 12-18 months. Network resources, such as routers and physical links, are expensive and it would be economically desirable to make optimal use of existing resources before upgrading them. If not a proper analysis of the network traffic is made, there could still arise situations with traffic congestion when there exist alternative links that are not fully used.

Routing algorithms within operational networks of today are based on a shortest path algorithm where the link leading to the least costly path is chosen. They are configured with little or no consideration of the current traffic intensity. Load balancing is applied, but is made in a static and manual way, based on rules of thumb. To minimise delay and the risk of congestion a flow optimisation algorithm for internal routing has been devised [Abrahamsson, *et al.* 2000]. An algorithm that evenly balances the traffic over a network has to be based on a correct prediction of the actual traffic flow in order to dynamically avoid unwanted oscillations and traffic congestions in the network. IP traffic behaviour is known to be very complex and volatile. The predicted network traffic is used as input for the flow optimisation algorithm. The bandwidth demand between two boundary routers varies considerably over time.

The purpose of this master thesis is to analyse flow behaviour of Internet traffic with focus on stability over different time scales. The existence, stability and identification of the believed few flows carrying the main part of the traffic volume are examined. These flows are referred to as *elephants*. The properties of these flows are discussed and studied by examining existing Internet traffic logs. How to measure stability is discussed and is modeled as a predictive measure.

Research is done at SICS AB on how to balance more evenly the traffic load over an intradomain network, suffering from congestion and delays. In order to optimise the routing, the stability and predictability of the traffic must be known. The optimisation process would consist of constant measuring of traffic, e.g. through statistical sampling at the boundary routers. The results of the measurments is used as input for a load balance optimsation algorithm. The optimisation algorithm produces how to transfer traffic on the most heavily loaded links to links with more available capacity. When the costs are applied to the network the process starts over again, see Figure 1.

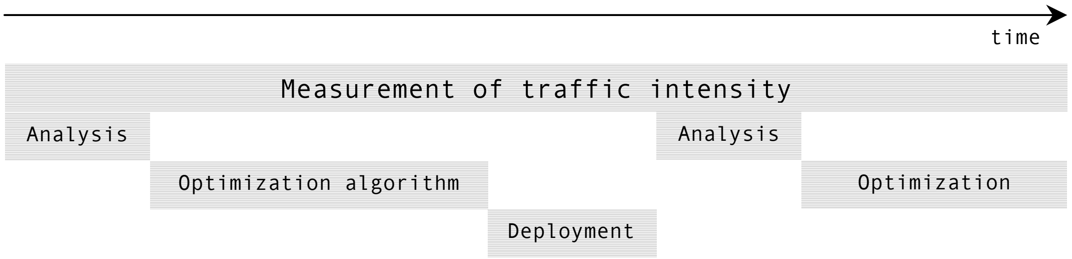


Figure 1. The load optimisation process.

2 The Internet infrastructure

The Internet is composed of inter-connected networks, built on different technologies, to which computers are attached. Computers attached to the Internet are called *hosts* and computers forwarding traffic are called *routers*. Networks are often represented and modeled as graphs. Hosts and routers on the Internet, creating or forwarding data, are referred to as *nodes*. Traffic between two nodes on a network is transmitted over a *link*. A link is a communication path of some medium between two nodes.

In general, each host on the Internet is connected to any other computer. The traffic is passed between routers that are configured to know how to forward the traffic closer towards its destination. Information about the shortest path through Internet is calculated, exchanged and updated using *routing protocols*. A protocol is a set of rules, defining how a certain task is carried out.

Networks are used in different organisations throughout the world and differ in size and technologies. To be able to develop and improve communication between the heterogeneous networks in a structured, efficient and distributed way, the technology of network communication has been divided into different layers of abstraction. The lowest layer deals with low-level technologies like physical interfaces for cables. Each layer gives service to the layer above and should have a well-defined functionality. Responsibilities in one layer can be specified in several different protocols.

In each protocol a Protocol Data Unit format, *PDU,* is specified. A PDU is often constructed with protocol specific information in a header and in an optional tail part and is referred to as a packet, see Figure 2. A PDU from a protocol in a higher level is encapsulated in the data part of the lower layer PDU. When a layer receives a PDU from the layer above, it attaches its own header and possibly a tail with control information and passes it to the next lower layer. If a layer receives a PDU from a layer below, it reads and acts on the information in the header and tail. Before it passes the PDU onto the layer above, it strips off the header and tail. The following two sections describe the two most important network reference models, the OSI reference model and the TCP/IP reference model.

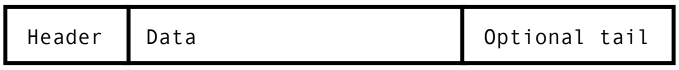


Figure 2. A Protocol Data Unit, PDU, consists of a header and an optional tail part with control information needed by the protocol. The data part could be comprised of a PDU from a protocol in a higher layer.

2.1 The OSI reference model

The International Organisation for Standardisation, *ISO,* proposed the Open System Interconnection Reference Model, *OSI,* in 1983. The model consists of seven layers, see Table 1. There exists a family of protocols related to the OSI model, but they are not widely used today.

Layer	*Responsibility*
Application	User authentication, constraints on data syntax, passwords etc.
Presentation	Converts data into different kinds of presentation to the user.
Session	Deals with the specification of a conversation.
Transport	End-to-end flow control and error control of transmitted data.
Network	Routing and forwarding of data.
Data link	Flow control and error correction on bit level.
Physical	Transmission of bits and the physical link.

Table 1. The OSI reference model.

2.2 The TCP/IP reference model

The extended TCP/IP reference model was designed from the ideas that Vinton Cerf and Robert Kahn presented in 1974 [Cerf, Kahn. 1974]. The responsibilities of the original TCP protocol were separated into the TCP protocol and the IP protocol. The abstracted responsibilities of networking were then formulated in the TCP/IP reference model, see Table 2. Its design goal was to facilitate the connection of heterogeneous networks. A short introduction to the data link, network and transport layer of the reference model follows below.

Layer	*Responsibility*	*Protocols*
Application	Interface towards the user.	Telnet, FTP, SMTP, HTTP, DNS
Transport	Ensures reliable connection over a network.	TCP, UDP
Network	Interface from a host to a host in another network.	IP, ICMP, ARP
Data link	Interface from host to network. Error correction on bit level.	ATM, Ethernet, token ring, FDDI
Physical	Interface for cables and physical equipment used.	Ethernet, token ring, FDDI

Table 2. The extended TCP/IP model consists of five layers. The levels represent different degrees abstraction.

2.2.1 Data link layer

The purpose of the data link layer is to provide somewhat reliable and efficient data communication between two physically connected machines. There is a limitation of the rate at which data can be sent and errors could occur. Some protocols of the data link layer divides the data into frames with a checksum, in order to be able to detect errors and possibly correct them with packet retransmission. Flow control is also considered in some protocols at this layer in order not to send frames faster than the receiver can handle them. All error handling is made with best effort and should not be considered as reliable.

2.2.2 Network layer

The network layer provides mechanisms for delivering packets to the right destination; this process is referred to as *routing*. The most used protocol on the Internet today is the Internet Protocol version 4, *IPv4*. IPv4 has a lot of shortcomings and a new version of the protocol is under deployment (IP version 6, *IPv6*). It will take time before it has replaced IPv4 as the main Internet protocol.

The Internet protocol

The Internet protocol, *IP,* specifies the format for an IP datagram, the packet format used to send IP packets. It consists of a header of at least 20 bytes, see Figure 3, and the data received from the transport layer. The type-of-service field, *TOS*, is a field used to let special traffic get privileges in a router. Routers are however not required to pay attention to the TOS field. IPv4 supports fragmentation of packets if they enter a network whose lower level protocols cannot handle large packets. The fragmented packets are reassembled at the final destination.

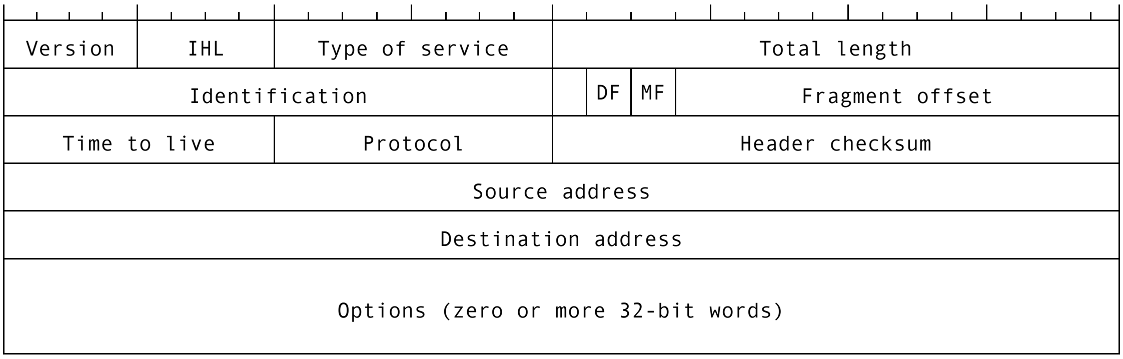


Figure 3. The Internet protocol version 4, IPv4, header format.

The time-to-live value, *TTL,* in the IP header was initially an indication of the maximum actual lifetime of a packet in seconds. It is however difficult to make a precise estimate of the time spent in router queues and links and the time spent there is generally very low. Today, most routers just decrement the TTL value by one, as the packet passes. The value is still a good indication to detect packets in a possible loop. These are discarded so that the network is not filled with unnecessary traffic. If the maximum initial TTL value is set too low, the packet could be discarded before it reaches its destination.

Internet address format

Every computer or network node on the Internet must be uniquely identified by an address. IPv4 uses a 32-bit address, usually notated as divided in groups of bytes separated by dots, decimally represented, e.g. `198.32.64.12`. The address format identifies the first N bits as the network identifier and the rest 32-N as the host identifier. Thus, the length of the network identifier decides the number of possible hosts on the network. An organisation wanting to connect their network to the Internet gets a range of IP addresses that correspond to the size of the network. The IP address format constitutes an address space of 2^{32}, a little less than 4.3 billion possible IP addresses. Since most organisations request more IP addresses than they need, in order to be able to expand their network in the future, there are many unused IP addresses allocated and the Internet is running short of unallocated IP addresses.

Local Internet Service Providers, *ISP'*s, allocate a range of IP addresses to offer customers. The IP addresses are often dynamically distributed in order to be able to reuse an address when a customer no longer needs it.

Another way of sharing, and thus saving addresses is by using network address translation, *NAT*. Part of the IP address space is reserved for private addresses. These cannot be used as identifiers on the Internet, but are used by private networks in order to identify local hosts. On a network where the hosts are addressed by private addresses, a NAT server is connected to the Internet. The local network is identified only by the one IP address assigned to the NAT server. The NAT server then translates the addresses of incoming packets and forwards them to the correct computer on the network. This is not an optimal solution since the NAT server has to know to which computer on the internal network incoming packets are destined for. In TCP sessions the NAT server could easily store the mapping between the internal port and the destination, but if the NAT server times out the connection and throws away the mapping the connection will be broken. Applications communicating via e.g. the transport layer protocol UDP send the source IP address encapsulated in the data stream and the NAT server never gets a chance to translate the address and the returning packets could be lost. The very common File Transfer Protocol, *FTP*, also sends transport layer information in the data stream. Temporary solutions have been proposed to get around these disadvantages.

The Address Resolution Protocol

Every computer on a network has to be identified with an address on the link layer level. The address is called the *medium address,* and consists for Ethernet of 48 bits, usually written with hexadecimal represention in six groups of bytes, divided by colons. When a host has an IP packet destined to a host on the same network, with a destination IP address, the corresponding medium address must be found. The address resolution protocol, *ARP,* is used to find the medium address from an IP address. The host sends out a request to all the hosts on the network containing the destination IP address. It receives an answer from the destination host, containing its medium address. If the packet is destined outside the network, the host has to find the medium address for the preferred router and sends the packet there.

2.2.3 Transport layer

Packets handled by IP in the network layer are sent with the policy of best effort, and they have no guarantees of reaching their destination. Packets are dropped when routers get their buffers full or when the TTL values of the packets have reached zero. The transport layer uses the transmission control protocol, *TCP*, to ensure a reliable connection and the user datagram protocol, *UDP,* for a best effort connectionless communication. TCP takes care of packet reordering and the retransmission of lost packets.

Transmission Control Protocol

The transport control protocol, *TCP*, was designed to provide a reliable communication channel between processes on two hosts. It creates a connection between the process on the destination computer and the client process on the computer. Before it passes the data on to the network layer, it divides the data into discrete entities. The TCP handles buffers and reorders packets arriving out of order before it passes them to the application level. It also makes sure to resend lost packets and to reduce the transmission rate if the receiving computer is slow [Cerf, Kahn. 1974].

The transmitter and receiver are recognised as two *sockets*, where a socket corresponds to one IP address and a 16-bit port number for the communicating application. A TCP connection is defined over one pair of sockets.

TCP also handles congestion control on the network. If a packet is lost during a connection TCP reduces the transmission speed to half and then gradually increases it again. In this way the available bandwidth capacity is shared with the other connections and traffic on the link. The congestion control works fine as long as the link capacity is not exceeded. When the link is fully used, the transmission speed gets slower the more traffic that is being loaded on the link. TCP lacks mechanisms to handle this problem and the problem is solved first when the traffic is reduced. TCP in its original implementation is not well suited for situation where packets are lost for any other reason than congestion, e.g. when bit error occurs often. If for example the connection is wireless with a loss of packets due to poor channel quality, the TCP will still slow down the transmission rate. The TCP header format is shown in Figure 4.

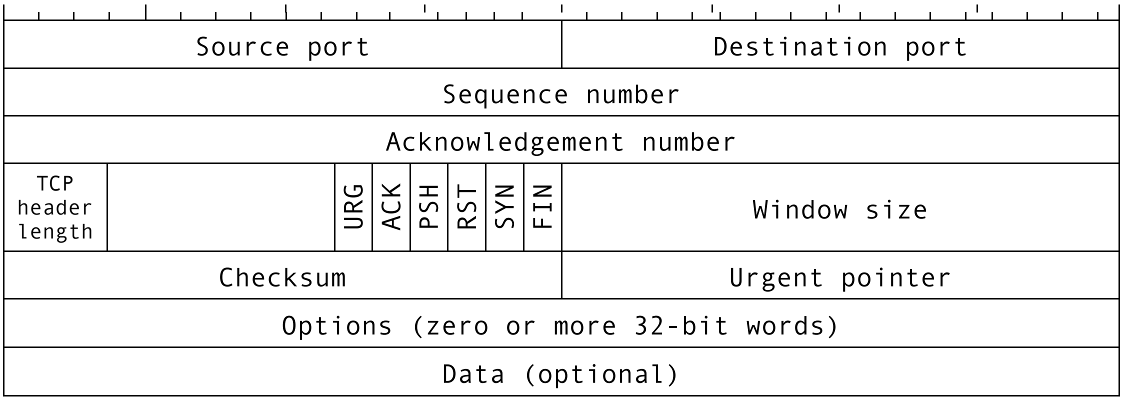
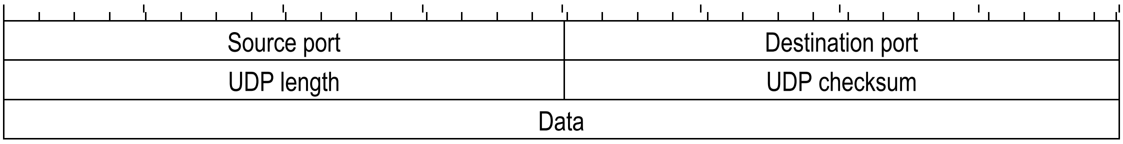


Figure 4. The TCP header format.

User Datagram Protocol

The user datagram protocol, *UDP,* provides the ability to send IP datagrams without establishing a connection. It does not provide error correction or congestion control. It identifies the host port with its 32-bit IP address and a 16-bit port number see Figure 5. Some applications are dependent on receiving the packets in time, e.g. telephony applications or other streaming media. It is better to receive some of the packets in time, rather than wait for retransmission and receive all of the packets after a while. Thus, they use UDP for the data transfer.



Source port	Destination port
UDP length	UDP checksum
Data	

Figure 5. The UDP header format

2.2.4 Communication between two networks

The purpose of the TCP/IP reference model is to be able to let networks built on different technologies to communicate with each other. In Figure 6 the two example networks *N1* and *N2* are pictured with a router *R* connecting them. The two hosts *A* and *B* are connected to *N1* and *N2* and have established a connection between two applications over *TCP*. *N1* uses the fiber distributed data interface technology, *FDDI*, whereas *N2* uses an Ethernet technology.

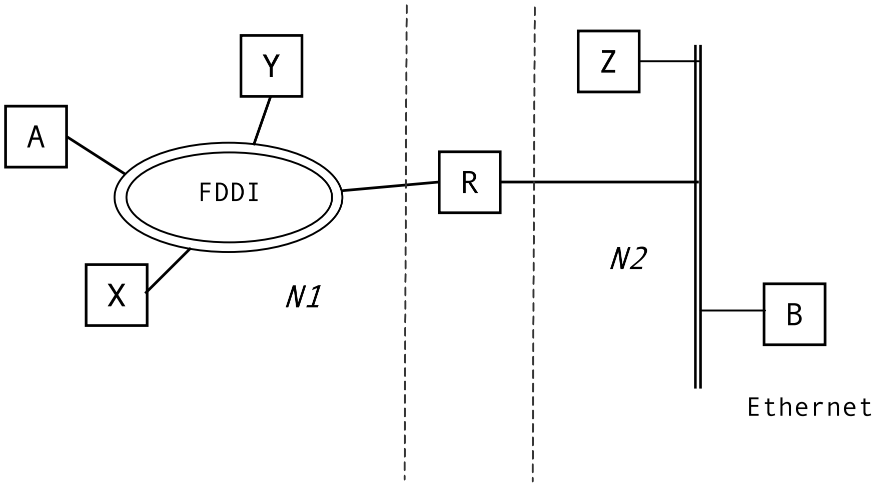


Figure 6. N1 is a FDDI network with the hosts A, X and Y. N2 is an Ethernet network with the hosts B and Z. The two networks are connected through router R.

When host *A* sends data towards host B over an established connection, the data is passed through all the layers in the TCP/IP reference model in Figure 7. The physical layer is omitted in the figure. Host *A* processes the data through all the layers and sends it in a frame on the FDDI network frame packed in a correct FDDI header and tail. Router *R* unpacks the data from the physical network frame up to the network layer and sees that it is destined for *N2*. It properly packs the data in the Ethernet frame format before it sends the frames onto network *N2* addressed to host *B*.

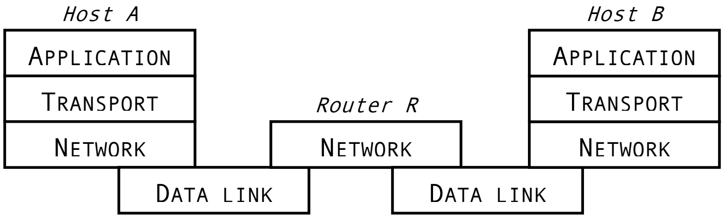


Figure 7. The data gets processed by all layers in the TCP/IP reference model at the hosts A and B, whereas the router R *only needs to unpack the data to the network layer.*

Figure 8 shows how the data is encapsulated on its way through the different layers. The protocols in the different layers add information in headers and tails as the data passes. A router checks the destination information in the network layer. It then constructs the appropriate header and tail for the data link layer, depending on the network technology used on the network where it sends the packet.

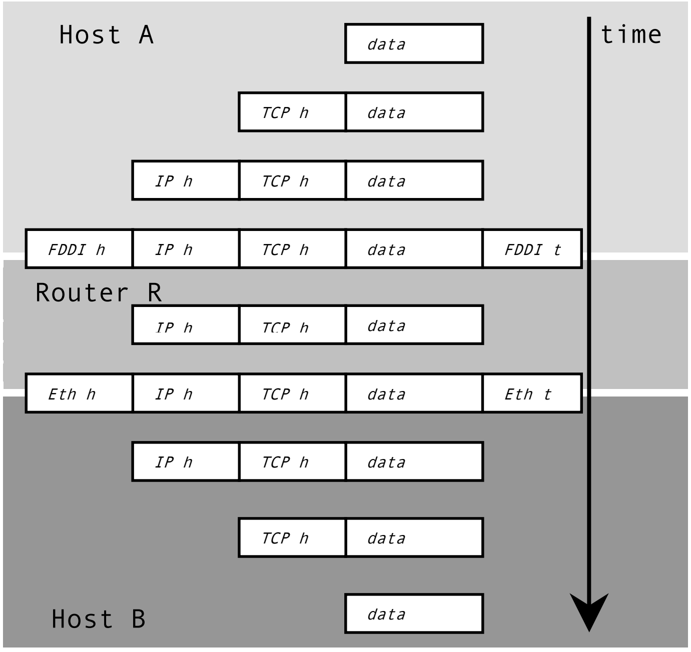


Figure 8. A packet is sent from host A *to host* B *through router* R, h *means header,* t *means tail.*

2.3 Routing algorithms

The main task for a routing algorithm is to calculate how to forward an incoming packet closer to its destination. The Internet routing algorithms are designed to be adaptive to topology changes in the network. Thus there are many alternative routes for each packet.

Networks connected to the Internet must be administered and maintained by some organisation or individual. An administrative set of networks is called an autonomous system, *AS*, and is identified by a 16-bit number. Every AS and its administrators are registered at the Internet Assigned Numbers Authority, *IANA*. Each AS can implement their own routing policies and provide different network services. The number of AS's constituting the Internet is growing and available AS numbers are running short.

2.3.1 Shortest path

In order to choose between different routes, a routing algorithm has to be able to rank the alternatives. To be able to compare different paths, the administrator often assigns each link a cost related to its capacity. The sum of these costs adds up to the total cost for the path, which is used for the calculation of the least costly or "shortest" path to the destination. Routing algorithms based on this additive path metric are called shortest path algorithms.

Problems with shortest path routing

Since the cost is statically configured and does not take the dynamic traffic characteristics into account, the algorithm itself cannot adapt the routes to traffic load. The main part of the traffic could be routed through a subset of all links, while other links are minimally used. Problems with congestion arise if traffic gets routed through a node with insufficient capacity.

Statically configured load sharing

To solve the problem with load sharing over network resources, different solutions have been proposed. These are mainly implemented on a local basis with mechanisms in the internal routing configuration.

To achieve splitting of traffic onto multiple paths in SP algorithms, a mechanism called equal cost multipath*, ECMP,* has been deployed. The mechanism can balance traffic over paths assigned equal costs, thus spreading the traffic over a larger set of nodes. The configuration is however done statically and knowledge of the traffic characteristics is required in order to make an even distribution. It is often hard to manually configure a network to an even distribution and the configuration becomes very sensitive to changes in the traffic pattern. Small changes in the configuration could lead to unexpected changes of the traffic flow through the network and cause even more traffic congestion and delays [RFC 2702].

2.3.2 Intradomain routing

Within an AS, the traffic is routed using an interior gateway protocol, *IGP*. Every router keeps a routing table with information about where to forward incoming packets. The router tables could be statically configure, but it in order to respond quickly to link failure or other changes of the network topology, a dynamically configuring routing protocol should be used in larger networks.

There are different types of intra domain routing protocols, e.g. distance vector routing protocols, distance path routing protocols and link state routing protocols, these are discussed in the following sections. The routing information protocol, *RIP*, is a distance vector protocol, which is easily implemented, but very limited in its performance and should only be used in smaller networks. The open shortest path first protocol, *OSPF*, is an example of a link state protocol, more powerful and complex than RIP and is commonly used in both small and larger networks [Huitema. 2000].

Distance vector routing protocols

In a distance vector routing protocol, every router has only local knowledge about the network. It keeps track of the distance to each router through its neighbors. The routing information is flooded through the network in distance vectors. Each distance vector contains information about the distance for the routers in the network. Every router increases the distances in the vector received, by one. If router C is reached in a distance of two through router A and a distance of one through router B, the router B is preferred to send packets addressed to router C. Traffic is sent on the shortest path calculated by the Bellman-Ford shortest path algorithm.

In order to avoid inconsistent routing information, the protocol specifies a maximum length of the worst path through a network. Every router in a path is added to the total cost of the path. In RIP the length of the worst path is 15 and when paths exceeding this limit are detected the information is discarded and a new network map is created. This is limiting the network size.

If a link failure would occur and the connected routers have not managed to update their routing tables at the same time, inconsistent information might be spread and could cause pairs of routers sending packets between each other. A network in this state only converges when the worst path mechanism detects the inconsistency and resets the routing tables. This is called *the bouncing effect* and *counting to infinity*.

A distance vector routing protocol is simple to implement and use, but on the other hand it has several problems and limitations regarding the network topology and should only be used for smaller networks.

Link state routing protocols

The Internet began as a military research project for Department of Defense in the USA. The first test network was named *Arpanet*. A link state routing protocol was deployed within the Arpanet in order to avoid the problems with distance vector protocols. Instead of exchanging distances to different nodes, every node keeps track of the whole network topology. Updates are flooded through the network only when a change has been made. The traffic is sent on the shortest path to the destination using Dijkstra's shortest path algorithm. Since all the nodes share the same information about the network topology, the routing tables will quickly become stable after a link failure. It takes a little longer to calculate the shortest paths through the network using the Dijkstra algorithm than with the Bellman Ford-algorithm, but in return, the protocol is more stable and converges more quickly after a topology change. The open shortest path first algorithm, *OSPF,* is an example of a link state routing protocol.

Another enhancement with link state protocols is that cost functions other than the amount of hops to a destination could be used, e.g. the link capacity. This opens up possibilities for policy routing and load balancing.

2.3.3 Interdomain routing

The traffic exchanged between AS's is routed with an exterior gateway protocol, *EGP*. The EGP used on the Internet today is the border gateway protocol, *BGP*.

BGP is a path vector protocol, similar to the distance vector protocols and routes the traffic on paths constituted by AS's. Loop prevention is implemented by checking the path for the same AS number appearing twice. Every path has a set of attributes, with information of how preferred the path is from the current AS. Different routing policies are implemented using different preferences for the attributes. This enables for the administrators to prioritise transit for paying customers or letting traffic on a heavily loaded link to be split up on several links.

Two BGP routers exchanging information are called BGP peers. They communicate over a TCP connection. This has the advantage of letting TCP handle retransmission and reordering of packets in the network. One disadvantage is that routing information is treated as ordinary Internet traffic. If congestion occurs and the routers are trying to reconfigure their routing table, the information about the update could be queued and delayed in the same congestion that it was going to solve.

BGP is using incremental updates that are only sent when a change has occurred. This is an advantage, as the network does not get unnecessary loaded with routing information.

Information about how to reach every node on the Internet is fully stored in the routing tables of each BGP router. By saving the current routing table, the Internet configuration could be saved for future research. Each routing table contains enough information to derive paths to every AS available at the time, see Figure 9. This is used when analysing Internet traffic.

```
Network      Next Hop         Metric LocPrf Weight Path

*>4.0.0.0    134.24.127.3     0                    1740 1 i
*            194.68.130.254   2                    5459 5413 1 i
*            158.43.133.48    0      10            1849 702 701 1 i
*            193.0.0.242      0                    3333 286 1 i
*            144.228.240.93   0                    1239 1 i
```

Figure 9. Extract from a BGP table generated by the command "show ip bgp" in a router with the IOS operating system. There are several ways to reach the destination network 4.0.0.0. The path consists of ordered AS numbers. The last AS number shows the origin AS. The i *indicates that the information originates from the internal BGP configuration. The attributes* LocPrf *and* Weight *are attributes used by the BGP prefix election algorithm.*

2.4 Problems and constraints

The Internet is made up of independently administrated subnetworks. Every administrator has individual policies and economical constraints to attend to. This makes it difficult to get an overall picture of the Internet structure and its behavior. Different factors affect how and at what rate the traffic propagates through the network. Some of these factors can be predicted, but others are dependent on the individual configuration of every network. The choice of routing protocol and implementation of the protocol also affects the overall performance.

2.4.1 Propagation delay

Propagation delay between two nodes in a network is dependent on the physical medium between them, but can in most cases be approximated at the speed of light, 300 km/ms. Further delay is added in each router, while deciding where to forward incoming packets.

When comparing delays between nodes in a network, the round trip time, *RTT*, is used. RTT is measured as the time between sending a packet towards a node and until receiving the acknowledgement that the packet has arrived.

2.4.2 Traffic congestion

Traffic congestions could cause delays and packet losses. Traffic congestion occurs if too many packets are routed through the same router at the same time. When router buffers get full, and the router starts to throw away packets. The transport protocol takes this as an indication to reduce the transmission rate. As the packets are buffered within routers, further delay is also added to the traffic.

2.4.3 TCP flow

All the packets belonging to the same TCP flow should be routed along paths with equal propagation delays. If the packets get routed along paths with considerable differences in propagation delay, they could arrive reordered at the destination. If the TCP window size is small, the reordering could cause unnecessary retransmission. For that reason, many routers are configured to forward packets belonging to the same TCP flow on the same path.

2.4.4 Asymmetric routing

An operator of a transit domain might want to route the traffic out from its own network as soon as possible, in order to give the best service to traffic generated from its own paying customers. This policy is called "hot potato routing". If the domain has many adjacent peering domains, it is easy to let traffic exit early. This could cause packets taking a different path back from a destination, so-called *asymmetric routing*, see Figure 10. Asymmetric routing could lead to difficulties in deriving the traversed path for packets afterwards.

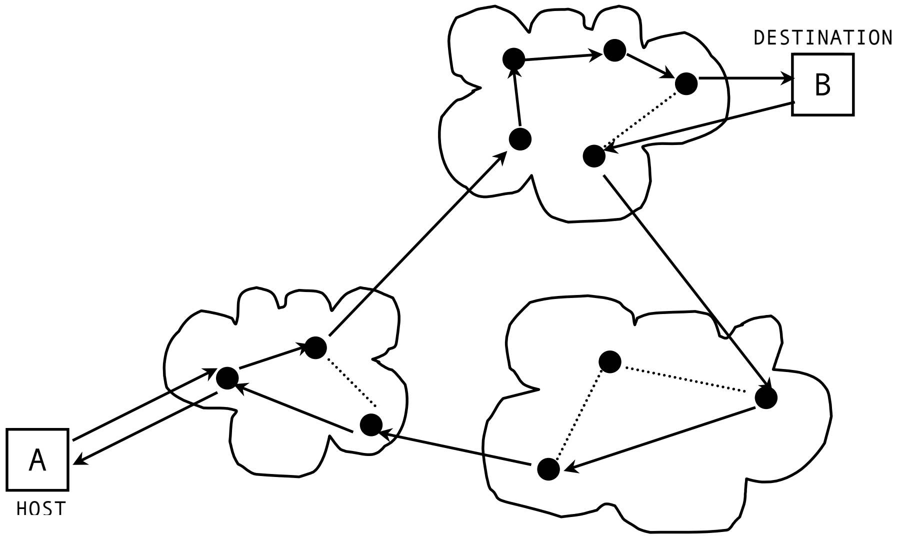


Figure 10. Asymmetric routing. The operators of the subnets want to route the traffic out of their networks as soon as possible. This could cause additional delays and difficulty in deriving the path of a packet.

3 Traffic engineering

Since 1994 when the Web traffic began to grow rapidly, the size and complexity of the Internet has increased considerably [Floyd, Paxson. 2001]. As a consequence of this, it is hard to get a complete overview of the Internet structure and its behaviour. It is important to do research in order to understand the factors affecting traffic and how to optimise utilisation of existing network resources. It is important to develop models of the behaviour in the different kinds of networks composing the Internet. These models are used when developing the future technologies of Internet.

3.1 Traffic engineering working group

The main goal for traffic engineering is to achieve optimal performance of operational networks [RFC 2702]. The Internet Engineering Task Force, *IETF*, has appointed a traffic engineering working group, *TEWG*, with the task to lead the work for efficient and reliable networks and optimisation of network resource utilisation [TEWG]. The underlying network topology is assumed to be relatively static and the goal is to map an existing traffic demand optimally onto it.

TEWG works with the measurement, characterisation, modelling and control of Internet traffic, but does not work with issues concerning the network, e.g. network design. Network engineering could be said to work with long term traffic changes, while traffic engineering works with short term traffic changes [NWG].

Traffic engineering has two objectives in *traffic oriented* and *resource oriented* issues [RFC 2702]. Traffic oriented work concentrates on aspects concerning the quality of service in networks. This includes minimising packet loss and delay while maximising data throughput. Resource oriented work concentrates on optimising the resource utilisation. Bandwidth is a primary resource in a network and it is important to efficiently manage bandwidth resources. Load balancing is resource oriented work and aims at utilising all of the network resources evenly.

One of the most important goals within traffic engineering is to avoid traffic congestion where packets are lost and delays are added. The problem can arise in situations where the network resources are insufficient or inadequate or where the network resources are suboptimally utilised. When the network resources are inadequate or insufficient, adding more resources to the network is the most common and obvious way to solve the problem. Another approach is to apply traffic congestion control techniques in order to fit the traffic to available resources.

Traffic engineering tries to solve the problem with unevenly utilised network resources. One approach to solve the problem has been to apply of load optimisation. This will increase the throughput and decreases the packet loss and delay.

The behaviour of the different routing protocols has a major impact on the performance of the network. There are also other characteristics with Internet traffic that needs to be considered, for example that the packets generated by a TCP flows needs to be kept together in order to avoid reordering and thus receiving a lower performance.

3.2 Load balancing

The main goal with intra-domain load balancing is to make better use of available network resources within an AS in order to minimise the risk of congestion. Hopefully this leads to data transmission with less delay and packet loss. It could however lead to additional propagation delay if the alternative routes are badly chosen. Some applications are very sensitive to delays e.g. voice over IP, *VoIP*; others are more sensitive to packet loss.

Load balancing in this report is considered within an AS or networks controlled by the same operator. Interdomain load balancing is not considered and the problem there is more complex due to poor control and overview of different network configurations.

In SPF algorithms, load balancing cannot be done over links with different assigned costs. When manually configuring load balancing, the traffic demand must be predictable to avoid unanticipated traffic congestions. The administrator responsible for the load-sharing configuration will have to be attentive to changes in the traffic pattern. These changes could come from a change of routing policy in a peering network, a link failure, a change of topology or a sudden change of popularity for an application [Elwalid, *et al.* 2001]. As a result of this instability in traffic flows, the administrator will have to devote a lot of time tuning the configuration to achieve a stable network load balance.

A network is modelled as a graph *G*, with a set of nodes *V* representing the routers and a set of edges *E* to represent the links between the routers, see Figure 11. Each edge is bidirectional. The interior routers are not expected to give any contribution to the traffic flow. Only the flow between boundary routers is taken under consideration.

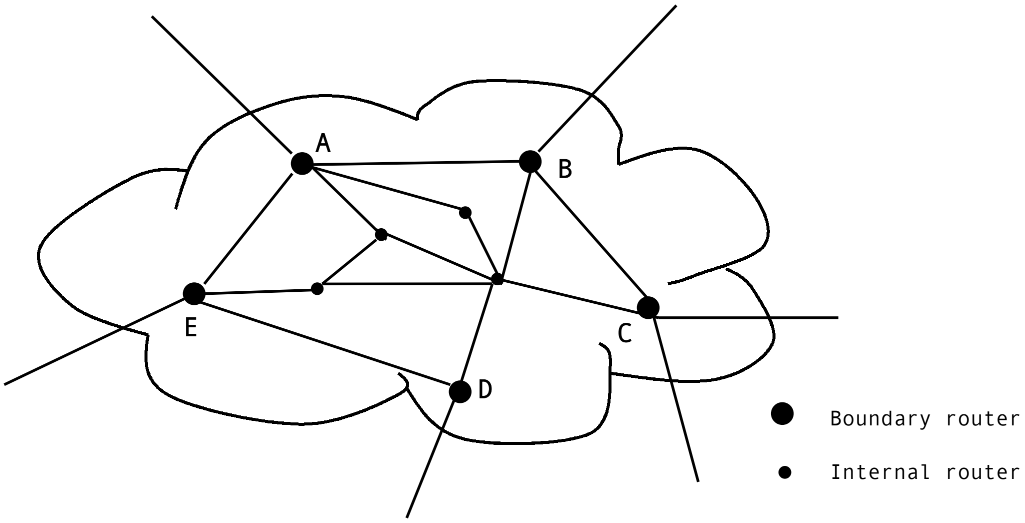


Figure 11. The network is modelled as a bidirectional graph G.

The traffic in a network can be represented in a *traffic matrix*. It consists of the actual bandwidth demand between the boundary routers averaged over some period of time. The predicted future traffic demand is modelled in a *traffic demand matrix* see Figure 12. The row entries correspond to nodes with incoming traffic, called *ingress nodes*. The column entries correspond to outgoing traffic, *egress nodes*. Each matrix entry corresponds to the expected traffic demand, between an ingress node and an egress node. To get a wider understanding of the traffic demand, each entry could be complemented with an estimation of the traffic intensity variation.

	A	B	C	D	E
A	-	5	10	3	2
B	3	-	4	5	2
C	4	2	-	5	6
D	2	1	4	-	3
E	5	6	23	12	-

Figure 12. The demand matrix expresses the expected demand of total bandwidth capacity between every pair of boundary routers in the nearest future.

The optimisation algorithm for load balancing, developed at SICS AB, takes a graph *G* with the demand matrix *D*. Each link is associated with a cost. The algorithm calculates how to reroute traffic from congested parts of the traffic to links with more capacity. In the network, the links must not be loaded too much, in order to be able to handle unexpected bursts in traffic volume. The optimisation algorithm is trying to minimise the total cost of distributing the traffic flows over the network. The algorithm should only reroute traffic on a longer path if a link is heavily loaded.

3.3 Stability measure

To know how long the optimisation is valid, a stability measure must be developed. The stability measure should be applied to the actual traffic demand in the network and determine if the traffic is stable enough to do a valid optimisation over a certain amount of time. If the balanced load is not stable enough, unanticipated congestions could arise in the network.

Stability of IP traffic could be measured in many different ways. Different views of IP traffic stability are described in Section 4. It is important to distinguish features of IP traffic to be able to predict future traffic load.

4 Related work

There are many different research teams studying traffic engineering issues. Many of them think that traffic engineering will play a bigger role in the future than it does now. There are also researchers and networking people who are sceptical against traffic engineering, and who have the belief that network traffic will be more unpredictable in the future and that measuring and optimisation will only slow down traffic in the networks. In the following section some of the work on traffic engineering will be presented. The different research teams try to understand the underlying factors affecting the traffic.

4.1 Concepts and research models

In the search for stable properties of IP traffic flow, the analysis has to be made with respect to different parameters. Every parameter has different impact on the traffic behaviour. Some are known, but their impact is further investigated to see how they interact, e.g. the behaviour of the TCP congestion control, in different situations.

Brownlee and Murray made a classification of the traffic through a network [Brownlee, Murray. 2001]. They refer to *flows* as traffic between the same nodes of a network. *Streams* are defined as the traffic between specific ports on two nodes. *Torrent* is the total amount of traffic on a link. Flows, streams and torrents are aggregates of traffic in both directions between two nodes. They further discuss how to measure traffic on the Internet and how to analyse the data. In their study they have access to more data and information of the network than in this study, therefore their methods could not be adopted or examined here.

Roberts examined the traffic with respect to traffic over TCP and traffic over UDP [Roberts. 2001]. He refers to TCP as elastic traffic, since the congestion control adjusts the traffic flow continuously. Streaming media carried over UDP is referred to as inelastic traffic since UDP lacks the congestion control.

Feldmann and her colleagues presented a methodology and model for traffic demands on IP networks [Feldmann, *et al.* 2001]. They measured traffic demands as the traffic load observed at the ingress nodes and mapped the load against the set of reachable egress nodes. Reachable egress nodes are derived from information in the forwarding tables of the internal routers. If the internal routing configuration is altered, flows could be addressed towards any of the possible egress nodes. If the model only had included one egress point, it would have been dependent on the current routing configuration. Instead, the model is now valid as long as the flow is destined to one of the egress points in its reachable egress set. Feldmann informally divides the traffic based on its origin. Domestic consumers, domestic business users and international traffic are examined separately. The international pattern is recognised as time-shifted business traffic.

You and Chandra examined the traffic from a campus site [You, Chandra. 1999]. They state that it is necessary to identify a level of traffic aggregation that allows a robust traffic characterisation in order to implement services that will give the correct privilege to the right kind of traffic. They look for stable properties in traffic by identifying applications that introduce non-stationary features in the traffic and filter them out from the traffic. Their resulting traffic stream is comprised of 60-70% of the total traffic and can be modeled as a nonlinear threshold autoregressive process.

Bhattacharyya and his colleagues from Sprint Laboratories, California, presented a paper with a study of traffic demands in an IP backbone [Bhattacharyya, *et al.* 2001]. The aim was to evaluate the traffic granularity levels for improving load balancing. They examined if there exist a stable traffic demand between two locations in a backbone. In their model they only described the demand between two points in the network without consideration of the actual routing in between.

4.2 Self-similar properties

IP traffic in general cannot be modelled as a Poisson process as e.g. telephone traffic can. The traffic shows burstiness on many time scales. Paxson and Floyd discuss the possibilities of self-similar properties in their paper from 1995 [Paxson, Floyd. 1995], also Abrahamsson discusses it in his article [Abrahamsson. 1999].

Roberts found self-similar properties with the packet arrival process in his study [Roberts. 2001]. He describes the extreme variation in the size of the observed flows and points at the even more extreme variation caused by the amount of large TCP flows at a millisecond time scale. He therefore suggests describing traffic in terms of larger aggregated flows.

4.3 Elephants and mice

Different researchers [Bhattaracharyya, *et al.* 2001, Feldmann, *et al.* 2001] have independently concluded that a small number of traffic flows carries a large amount of the transferred traffic, see Figure 13. This is an important characteristic of Internet traffic that facilitates optimisation. The few larger flows are referred to as *elephants* and the larger amount of smaller flows are called *mice*. This is a very important characteristic and is useful in deciding how to do load balancing.

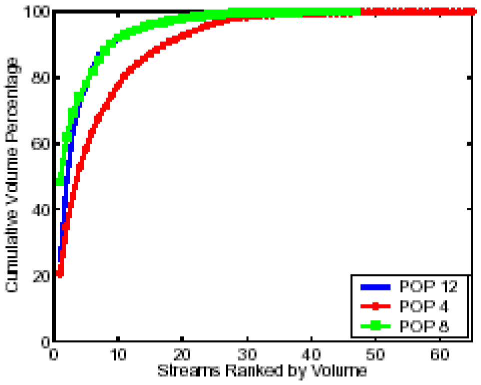


Figure 13. Distribution of traffic aggregated on identical first eight bits in the IP address for a Web-host access link. The streams are sorted by volume and plotted against the cumulative volume distribution. The graph shows the existence of a few streams that carries the main part of the traffic. Figure presented in [Bhattacharyya, et al. 2001]

Anja Feldmann and her colleagues concluded in the experimental analysis that the traffic showed the impact of the elephant and mice phenomenon [Feldmann, *et al.* 2001]. They discuss implications to routing considering only the largest demands. If these large demands should change due to changes in the internal routing configuration or reduced popularity, an unwanted imbalance could come up.

Roberts did also observe a heavy-tailed distribution with elastic traffic in his study of traffic over a backbone link [Roberts. 2001]. He emphasises the difficulty in determining the distribution of the traffic size and suggests implementing traffic control insensitive to the precise size of the transferred document.

4.4 Stability measure

You and Chandra examine stability as the variation in packet intensity in different time windows [You, Chandra. 1999]. They separate packets from different applications and calculate the probability that the traffic could be modeled as a stationary process against different confidence intervals.

Feldmann arranged the traffic streams in descending size order and divided them evenly into numbered quintiles [Feldmann, *et al.* 2001]. Each quintile thus consists of the streams carrying 5% of the total volume. After a time period *h* the demands are measured and ordered again and the proportion of demands that changed quintiles are calculated, see Figure 14. In the first diagram, *h* has the value of 30 min; in the second diagram *h* has the value of 24 hours. The streams in the first quintiles, and thus the largest ones, are the most volatile. The variation increases as the time period *h* extends towards 12 hours and decreases subsequently as *h* approaches 24 hours. In general, streams seem to keep their relative position in size indicated of the fact that jumping between quintiles is generally low and most of the jumps are less than 5 quintiles, 25%. The *x*-axis represents the quintile the streams where placed in at the first point of measurement. The *y*-axis reads to what quintiles the traffic has been found in at the second point of measurement.

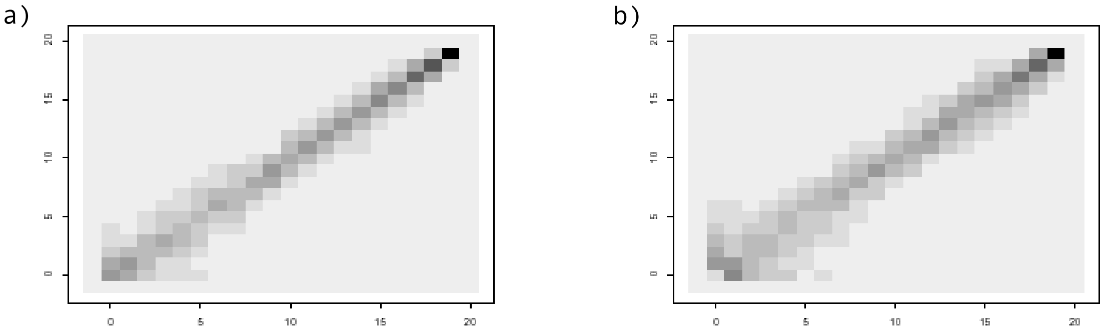


Figure 14. (a) Demands at 1pm and 1:30pm, h=30 minutes. (Nov 3) (b) Demands on Nov 3 and Nov 4, h=24 hours (1pm) Stability of the measured traffic demands across time (two-dimensional histograms). The jumping of streams between different quintiles is considerably low. Figure from article presented at the ACM/SIGCOMM 2000. [Feldmann, et al. 2000]

The largest demands show a substantial variation in size over the time-of-day. They also seem to vary in their time of day pattern [Feldmann, *et al.* 2001].

Bhattacharyya and his colleagues at Sprint labs in California searched for properties of IP traffic [Bhattacharyya, *et al.* 2001]. The analysis of the traffic showed that a small number of the aggregated streams generated a large fraction of the total traffic. In the examined traffic trace they found that approximately ten streams held more than 80% of the total traffic. They denote an aggregated stream with identical first *N* bit of the network prefix as a *pN*-stream. If a *pN* stream is further divided into smaller streams, the phenomenon with a few streams contributing a big proportion of the traffic is again observed. It appeared as if the large aggregates behaved in stable way throughout the day. In the study they compared different types of access links and concluded that traffic from an ISP, a Web host and a peering link behaves considerably different.

Bhattacharyya et al. measured stream stability by ranking them with respect to their carried volume [Bhattacharyya, *et al.* 2001]. The order of rank changes over time is then used as a stability measure. From a plot where the cumulative distribution of rank changes is drawn, it is shown that 70% of the rank changes is less than the order of five for a p8 stream, see Figure 15. They also show that 70% of the top 15 largest streams remain in the top throughout the day. This shows that the largest flows stay large throughout the day and the smaller flows stay small.

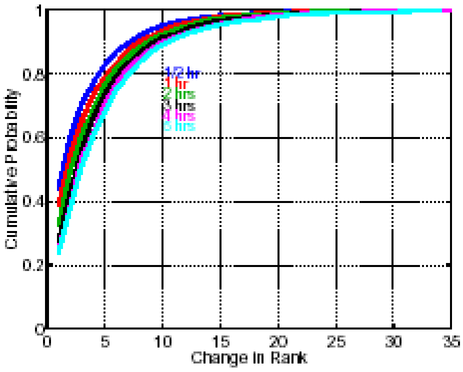


Figure 15. The cumulative distribution of rank changes for p8 traffic stream. The different lines describe time intervals ranging from 30 min to 5 hours. Figure in [Bhattacharyya, et al. 2001]

4.5 Static load balancing

Maintaining a load-balanced network only with the load sharing mechanisms of an IGP has been considered demanding. To facilitate the management, the Multi Protocol Label Switching technique, *MPLS*, has been deployed.

With MPLS, virtual circuits are created within a specified part of a network, referred to as a cloud. Each packet entering the cloud gets one or more circuit labels attached to it, identifying its total path through the network. Every router identifies each path with one of its interfaces. This makes the traffic flow faster through the network, since forwarding is made faster due to the shorter circuit identifier and smaller router tables [RFC 2702]. A disadvantage of MPLS is that a circuit has to be calculated for each source/destination pair. MPLS is still not fully deployed or tested [Fortz, Thorup. 2000].

There exist algorithms achieving load balancing by calculating appropriate weights to the *OSPF* routing algorithm. Optimising the weights setting shows to be a NP-hard problem but Fortz and Thorup have developed a method that takes advantage of a local search heuristic. They are using a simple model of the daily traffic demand. The model does not take into consideration unpredictable bursts in traffic, but treats the difference in morning and evening traffic intensity as two different problems [Fortz, Thorup. 2000].

Roberts found intensity levels in a 5-10 minutes time scale to be predictable from day to day. He modelled the traffic as a random process with a constant intensity, see Figure 16 [Roberts. 2001].

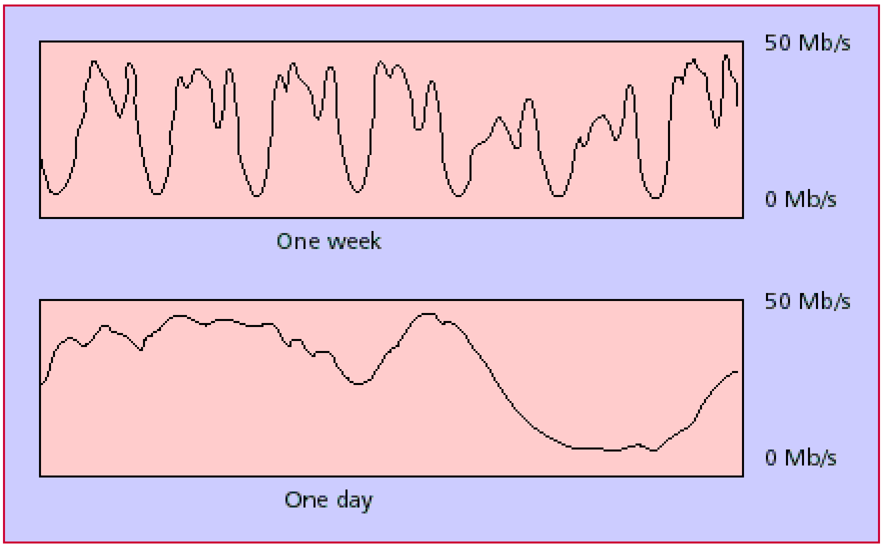


Figure 16. The traffic variation during a day in a backbone link. Figure published in [Roberts. 2001]

5 Problem statement

If some network resources are congested with packet loss and lower throughput as result, the performance could be improved if some of the traffic were routed over parts of the network with less load. The routing mechanism has to have knowledge of the current network traffic load, in order to reroute traffic without creating congestion somewhere else in the network. The estimated future traffic load can be determined from analysis on measurements from the network traffic.

The most important characteristics of the traffic load is its predictiveness and its stability. If the load is too volatile and unpredictive, it can lead to oscillations in the network traffic and cause more instability and cause more traffic congestion.

The purpose of this master thesis is to study the nature of stability with IP traffic on a smaller time scale that can be used for load balancing in a network.

IP traffic is assumed to be comprised of a few larger flows carrying the main part of the traffic called elephants, together with a large number of smaller flows called mice. The existence, identification and stability of the elephants are studied. IP traffic has also shown to possess self-similar properties, thus having sudden bursts in traffic volume.

The analysis of elephants has three objects to determine their

- existence
- identification
- stability

The analysis of elephant existence involves the discussion of how to aggregate traffic into flows. The topological structure of the network should be taken into consideration.

Elephant identification discusses if the same elephants that can be found on a larger time scale also can be seen on a smaller time scale.

Elephant stability involves finding a stability measure that reflects the duration and stability of larger flows. The discussion includes the definition of "stable" properties.

Different stability measures show different characteristics. The stability of IP traffic has been the subject of prior studies, but different requirements apply in the case of dynamic load balancing. There are mainly three different kinds of measures used in the related works:

- Absolute volume variation [Roberts. 2001]
- Change of rank [Bhattacharyya, *et al.* 2001]
- Change of relative size [Feldmann, *et al.* 2001]

Since the traffic intensity is changing with the time of day, an absolute measure of the intensity variation is hard to use. If the stability measure is relative to the intensity at each interval, it is possible to relate and compare values from different intervals.

The identification of elephants is important to sort out all the smaller flows that have occasional peaks in intensity. Smaller flows with sudden bursts could otherwise have too large impact when calculating the actual stability. The identification of these flows is in this study done in a static way that is not considered final and needs to be developed into a more dynamical method.

6 Method

In order to measure stability, the concept itself has to be defined. IP traffic is shown to have bursty properties on many time scales. It is important in this case not to balance the traffic flow based on short bursts. As earlier presented, it is believed that a few flows are carrying the greater part of the traffic volume. Other known properties of IP traffic are useful in the search of ways to determine any stationary behaviour of IP traffic, e.g. the impact of IP traffic carried by TCP.

Traffic could be aggregated in different ways, such as by application, protocol, or network topological endpoints.

In this report the topological locations of the source and destination hosts are used as aggregation parameters. Since the routing configuration is not known, the topological location is artificially approximated by different lengths of the network prefix. Shorter network prefix is used to approximate larger networks. This approximation is not optimal, but it gives a feeling for the impact of traffic from different sizes of networks. The method also abstracts the need of knowing the current internal routing configuration in the network.

The data that is analysed comes from recorded traffic from boundary routers. The first 60 bytes of data from each packet are written to a file for futher analysis. The recorded data shows information about each packet such as when it passed the router, how large it was.

6.1 The packet frames

The first data set consists of traffic traces collected at a router connecting the SICS network to the rest of the SUNET AS via KTH-LAN backbone, see Figure 17. The BGP router table from the same date is downloaded in order to perform a correct mapping of IP addresses against destination network or AS. The internal routing configuration from the time of logging is not known. The traffic log consists of a timestamp, the Ethernet, IP and TCP headers of each packet routed through the boundary router. The source and the destination address together with the length of the IP packet are determined from the IP header. Henrik Abrahamsson collected the traffic at the boundary router. The SICS trace is 24 hours long and was taken on April 14, 1999.

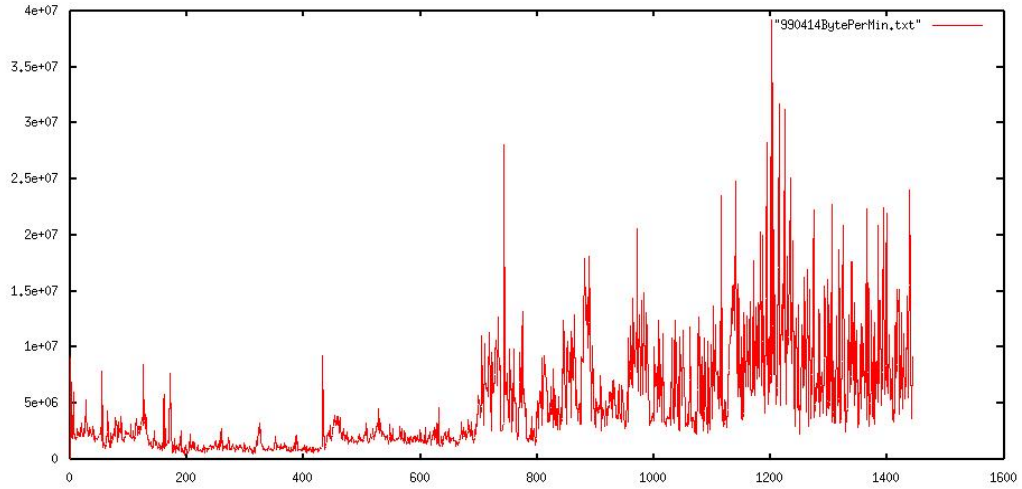


Figure 17. The 24-hour long trace taken from the link connecting the SICS network to the rest of the SUNET AS. The traffic is shown in bytes per minute against every logged minute.

The second trace was taken in Japan on a link over to the USA, see Figure 18. It consists of 67 hours of IP traffic on the 4 Mbps link, logged on May 10-13, 1999. The IP addresses in that trace have been anonymised with the prefix preserving method *tcpdriv* [Xu, *et al* 2001]. Since the trace has been anonymised the mapping between IP address and corresponding AS cannot be done.

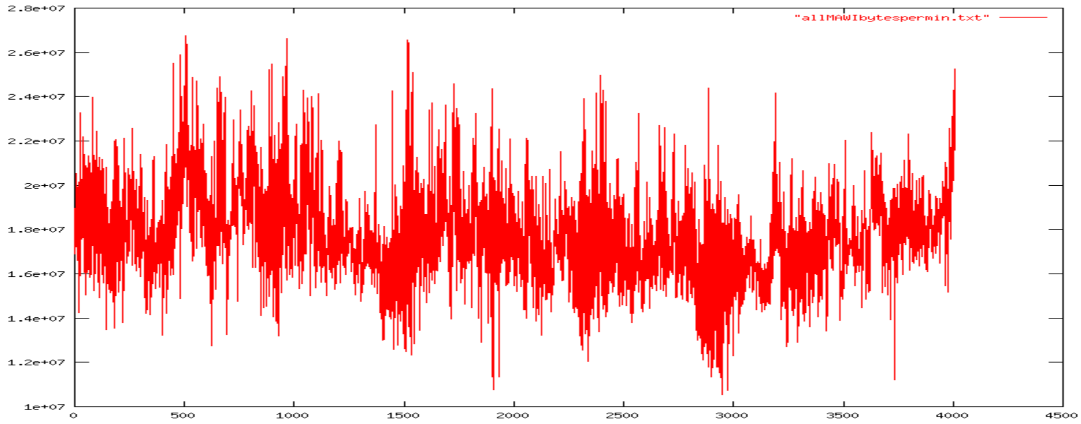


Figure 18. The 67 hour long trace taken on from a link between Japan and the USA. Shown in bytes per min against every logged minute.

6.2 Parameters

6.2.1 The flow concept

Aggregation level

The traffic is analysed with respect to the global network topology. The traffic is divided into *flows* where a flow is every packet to the same network or AS. A network is artificially identified as the addresses sharing the same first bits in the IP address. Packets with addresses sharing the same first *N* bits are referred to as a *pN*-flow; i.e. all packets to 193.x.y.z are collected into one *p*8-flow. *N* takes the values of 8, 16, 24 and 32. The flow ends when there is no traffic between the endpoints in an interval. It is registered as a new flow if the traffic would resume. In the SICS trace the traffic is also aggregated with respect to its destination AS.

Time interval

The traffic is recorded over several hours. The interval is divided into smaller intervals where the traffic is aggregated and analysed. The flows are sorted by volume and elephants are extracted as the largest flows relative to the total traffic volume in the smaller interval. The threshold of when to consider a flow as large depends on the traffic trace. Analysis should first be done to see what ratio there is between elephants and mice. In this study the threshold is set at 80 percent, i.e. the largest flows carrying 80% of the interval traffic is considered as large and their characteristics are further analysed.

In order to analyse the traffic pattern on a larger timescale, the elephant and mice properties of the whole interval are analysed. The interval is treated exactly as one of its smaller interval and the largest flows are extracted and analysed.

6.2.2 Prefix-AS mapping

In order to find the origin AS of a prefix, a mapping is done with the help of the BGP routing tables collected from the Oregon Routeview project site [Oregon Routeview]. A routing table from 1999 has in the order of 100 000 entries with each entry representing an announced prefix and its AS.

To make the look-up process efficient, the routing table was read into the structure of an LC-trie. The LC-trie structure is a trie based on strings with both level and path compression. An uncompressed trie would have had, in this case, an approximate depth of 32 and a width of 2^{31} with variation in density. The information in each node is represented in a 32-bit word that also makes the structure economical in memory. This method of longest prefix match was developed by Stefan Nilsson and Gunnar Karlsson [Nilsson, Karlsson. 1998]. The purpose was to make a fast next hop look-up in routers on the Internet. The structure was modified to produce the corresponding origin AS number to a specific IP address.

6.3 Stability measure

The first traffic trace taken from the SICS access link connecting to the SUNET AS in 1999, is divided into discrete time intervals ranging from 1 second, 60 seconds, 300 seconds, 900 seconds and 3600 seconds. The second trace between Japan and the USA is divided into discrete time intervals with the lengths of 1, 10, 30, 60, 300, 900 seconds. In each smaller interval, the packets are aggregated into flows and their total volume over the time interval is calculated. The largest flows carrying a fraction h of the total traffic volume is further examined. The threshold h has to be configured for every different network and routing configuration; it is assigned the value of 0,8 in this study.

For the optimisation it is interesting to know if the flows that have been considered large for a number of time periods are going to be large in the nearest future. It is interesting to observe the traffic during some time period and from its behaviour earlier be able to derive how stable the traffic will be during the nearest future.

To avoid paying attention to sudden bursts in traffic and smaller flows having a sudden burst in traffic intensity, the main focus is laid on flows that have been considered large for several smaller intervals.

The probability that a flow that has been large in at least x time intervals is large in the next interval is calculated as the quotient between the number of occurrences where a flow could be measured with length $x+1$ and the number of occurrences where it is measured as with the length of x.

Example: Only one flow is found and it is considered to be large for six time intervals, see Figure 19. There are two occasions when the flow could be measured with a length of five. Only at one of these occasions it has also the length of six. This makes the possibility for a flow being of length six if it already has the length of five ½.

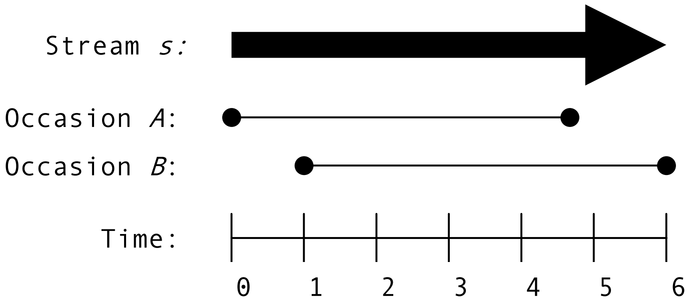


Figure 19. One flow of length six is found. The flow s is measured with length five in both occasion A and B. Occasion A is the only time the flow is measured with the length of six. Thus, the probability is ½=0.5 that the flow s is large in the next interval if it is found large in five consequent intervals.

The probability that a flow that has been large for k time intervals to be large in *k+x* time intervals is calculated in a similar way.

7 Analysis

7.1 Elephant existence

7.1.1 Computing flow volume

The p*N* flows are aggregated on the first *N* bits in the network prefix of the destination address. The volume of each flow is measured as the sum of the volume from each of the first *k* sec interval and all the subsequent intervals where the flow shows non-zero volume. Note that this leads to the exististence of different flows with the same p*N*-identifier, when the flows have stopped and started again. The volume of a flow *s* is shown in Formula 2.

$$\mathrm{Vol}(s) = \{\mathrm{Vol}(s_i) + \mathrm{Vol}(s_{i+1}) + \ldots + \mathrm{Vol}(s_{i+n}) : \mathrm{Vol}(s_{i-1}) = 0, \mathrm{Vol}(s_{i+n+1}) = 0\}$$

*Formula 2. The total volume of a p*N*-flow* s. *The flow is* n *interval long.*

The flow length is used when calculating the flow stability.

7.1.2 Elephants in the whole interval

In Figure 20 a-d the elephant and mice relationship from the Japan trace is indicated for different time intervals. In each diagram, the flows in the whole interval have been ranked by volume in descending order and mapped against the cumulative volume distribution. The x-axis is shown with the percentage of the amount of flows. Since the graph is steep at the left part of the diagram it shows that the largest 10% of the flows stands for an 80%-90% of the total volume, thus indicating the correctness of the elephant and mice assumption. It is also evident that the phenomenon appears independent on the choice of aggregation. The phenomenon is more prominent with longer time interval.

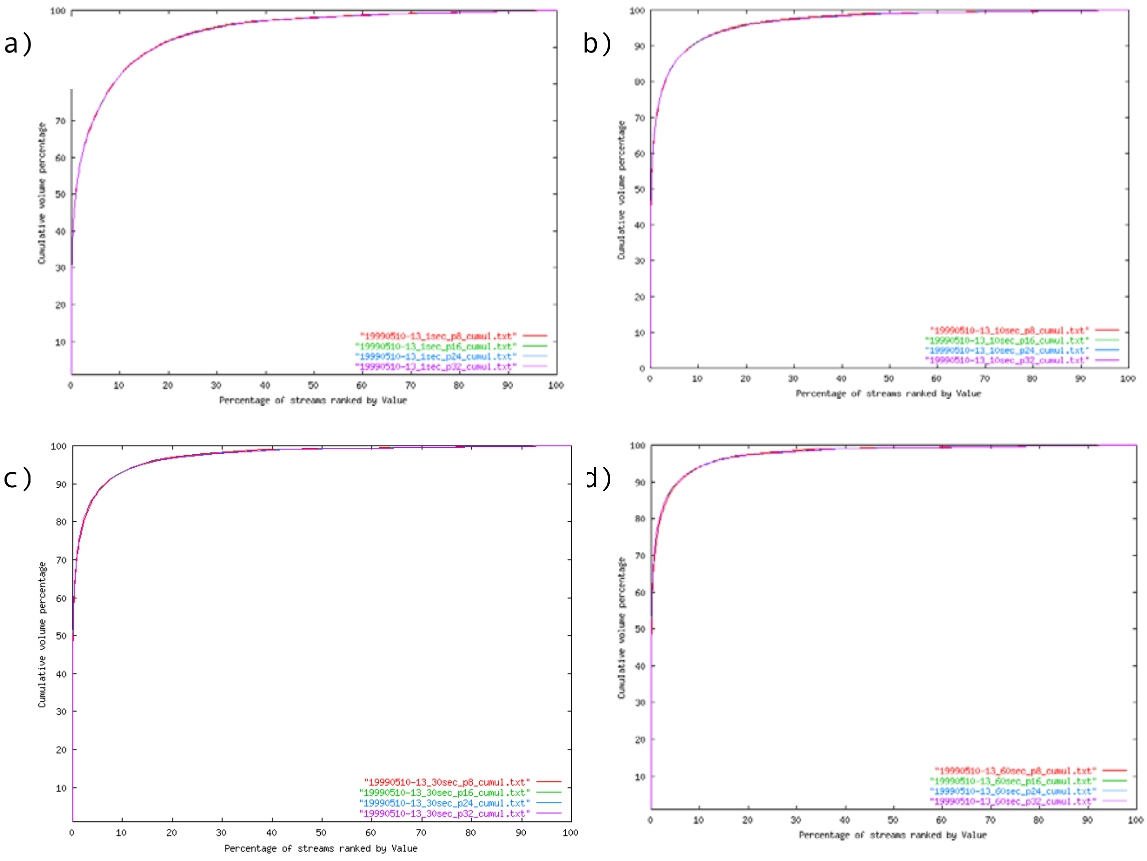
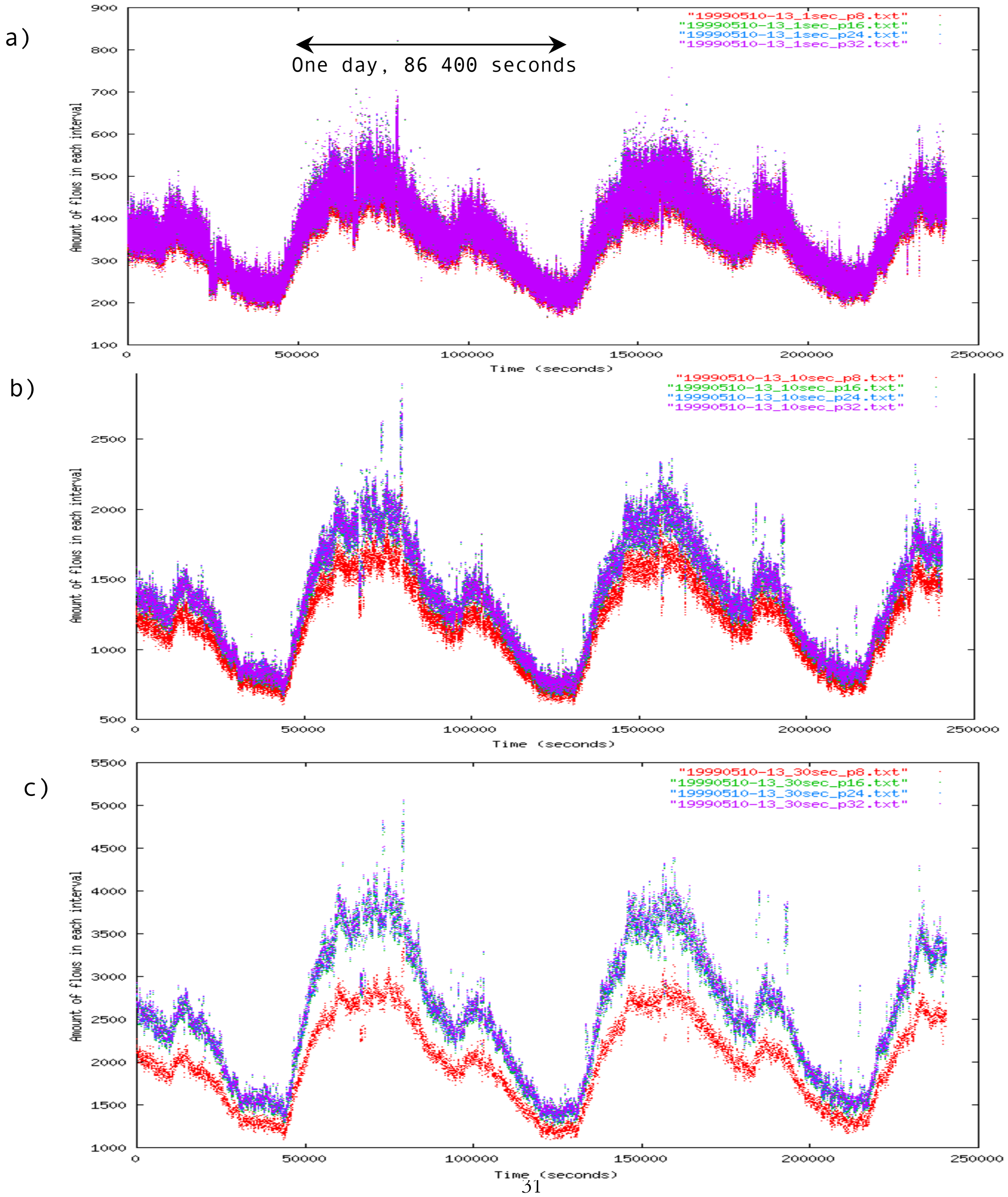


Figure 20. The cumulative volume percentage distribution from the MAWI trace. The flows are ranked in ascending order with respect to their size. (a) Every interval is 1 second long. (b) Every interval is 10 seconds long. (c) Every interval is 30 seconds long. (d) Every interval is 60 seconds long.

7.2 Identification of the elephant flows

In a real-time situation when the traffic is going to be analysed there must be a method of identifying the number of elephants present in order to make use of the stability results. The traffic must be filtered in some way. In Figure 21 a-g the total number of flows in each interval from the MAWI trace is described. A daily recurring pattern is evident here when looking at the total number of flows per interval. The number of flows seems to decrease during nighttime and increase during daytime in a regular pattern. The anonymisation of the traffic could have caused the mapping of some p16, p24 and p32 to the same network prefix, thus explaining the evident division of p16, p24 and p32 flows against the number of p8 flows. There exists fewer P8 flows than the finer aggregation flows in each interval.



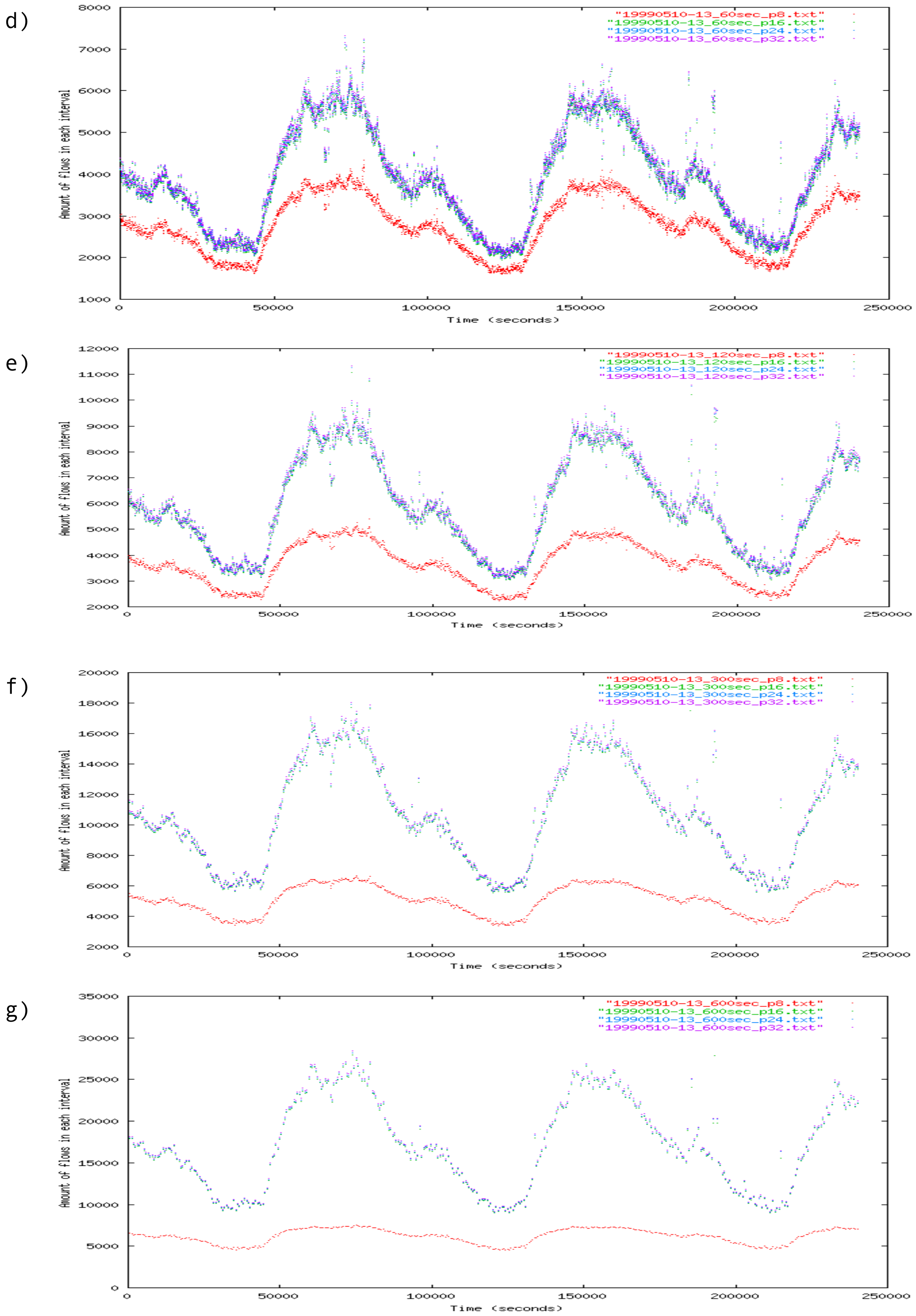
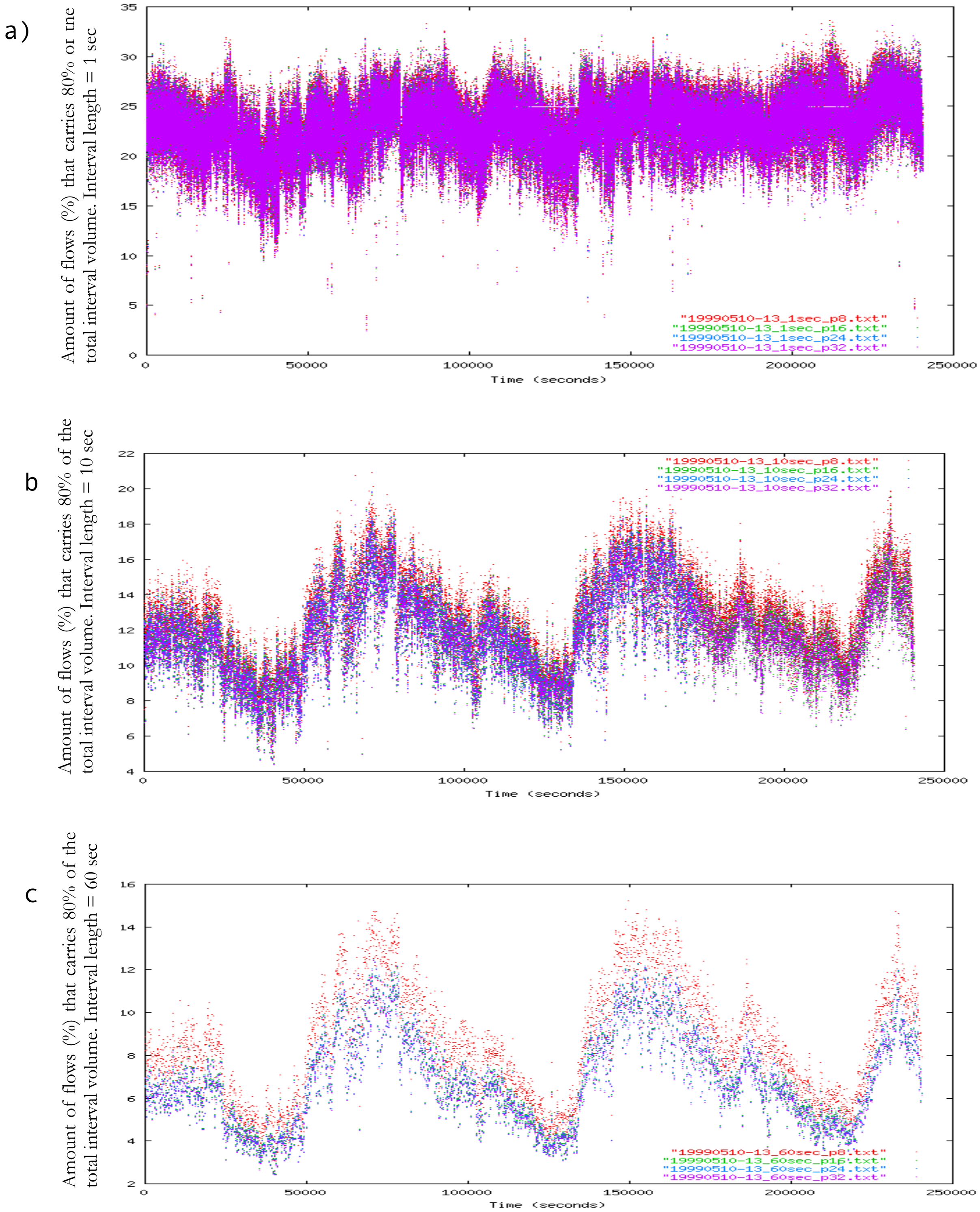


Figure 21. The number of flows in each interval is pictured for different values of interval length. (a) 1 second interval. (b) 10 second interval. (c) 30 second interval. (d) 60 second interval. (e) 120 second interval. (f) 300 second interval. (g) 600 second interval.

The elephant flows discovered in the whole examined interval carried 80 percent of the total traffic, but that need not be the case in each smaller interval. To examine if the same relationship exists in the smaller intervals, the number of flows carrying 80 percent of the total traffic is compared against the total number of flows in the interval. The y-axis shows in Figure 22 how large part of flows in each interval that carries 80% of the interval volume. The results are presented in Figure 22 a-e and it is evident that the elephant and mice phenomenon exists even in the smaller intervals, there is a low amount, ca 6-24% depending on interval length of the total number of flows that are carrying 80% of the total interval volume.



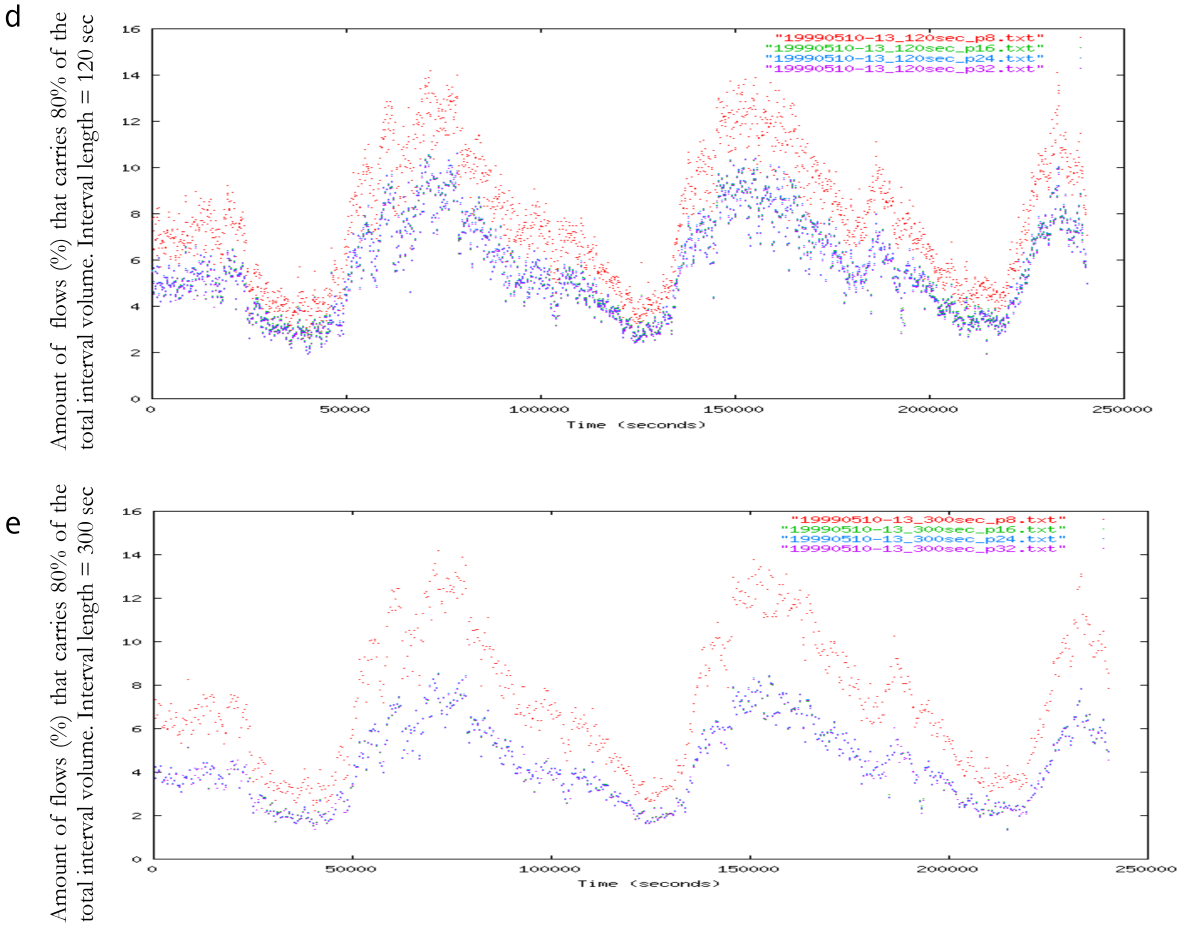


Figure 22 a-e. The amount of flows that 80 percent of the interval traffic volume. The diagrams show the differences between different interval lengths. (a) 1 second interval lengths: (b) 10 seconds, (c) 30 second, (d) 60 second, (e) 120 second.

Stability of elephant flows

The elephants extracted in the smaller intervals are examined for how long they are regarded as being large. If the flow is not regarded as large in one interval, the flow is considered as to have ended. Aggregation seems not to have any major impact on the probability distribution in the MAWI trace as seen in Figure 23.

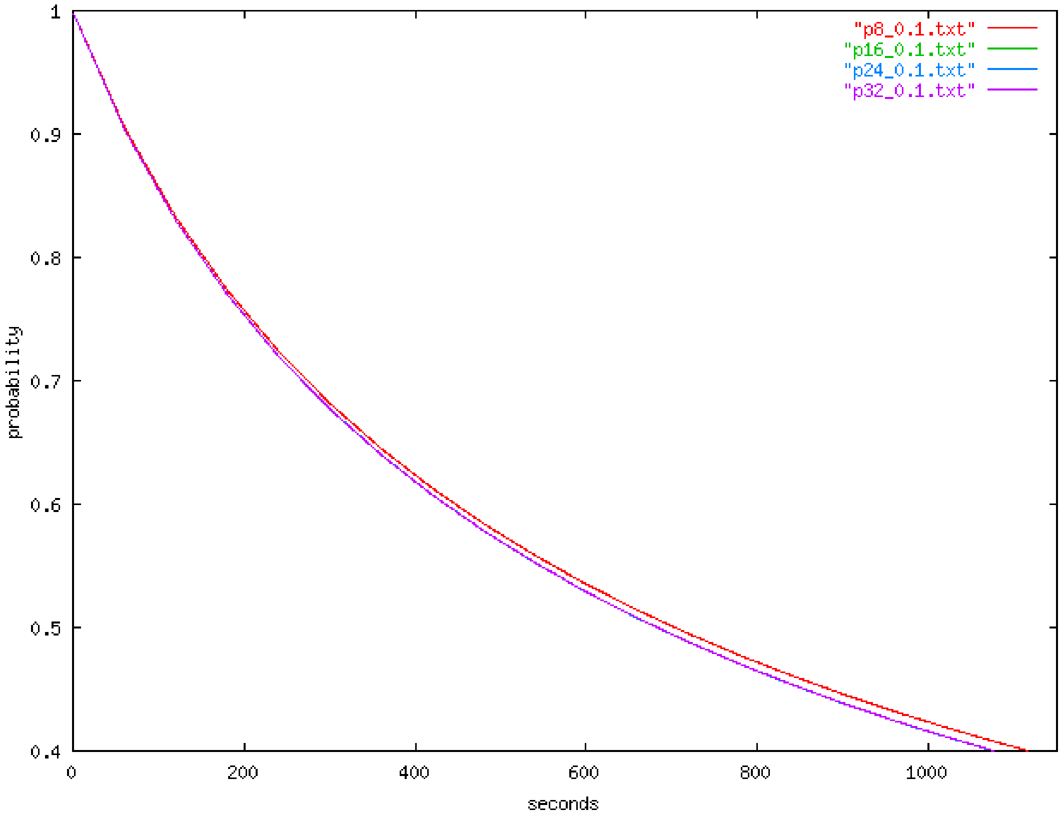


Figure 23. The probability that a flow stays large after being considered to be large for at least 5 minutes for P8-flows.The interval is 60 seconds long.

The diagram and the distribution and mutual relationship between the network aggregation shown in Figure 23 is representantative for the other diagrams showing the other interval lengths. Since no difference in aggregation level can be seen, the different interval length over the same pN-aggregation is shown in Figure 24. Here, a more evident distinction can be made between the impact of interval length on the probability. The probability of a flow staying large *x* seconds after being considered as large for 5 minutes is shown. It seems that the probability is larger when using longer intervals, which is intuitive since there is larger probability for a flow getting interrupted if there is a short interval.

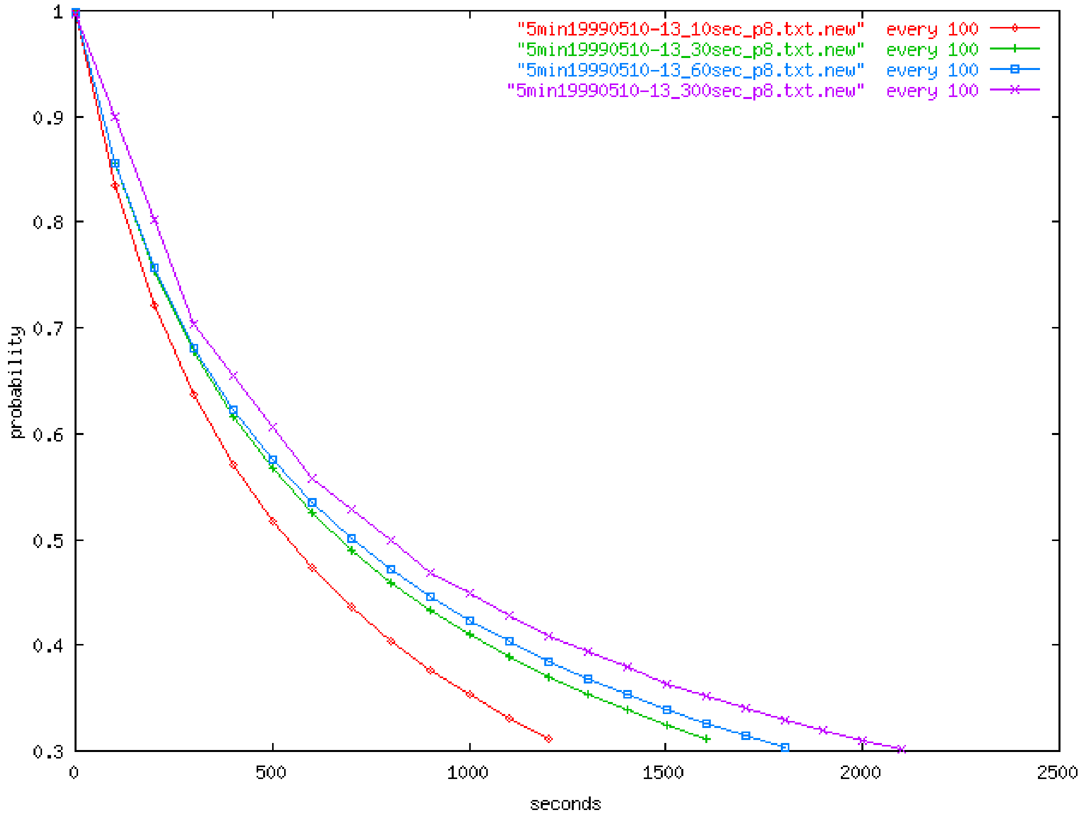


Figure 24. Probability for p8-flows that have considered as being large for 5 minutes staying large for the next x seconds. 1, 10, 30, 60 and 300 seconds interval length.

8 Conclusions

Predicting Internet traffic is a complex and wide field of research and this study has approached the idea of how to find stability in Internet traffic by identifying and analysing the largest flows.

The elephant and mice phenomenon exists independently on the network aggregation level. The same phenomenon appears in the smaller intervals as in the larger ones and this could be used for identification of the larger flows. The larger flows show a matter of stability that could be used when predicting traffic with load balancing. There is a larger probability to find stability when using a longer interval length. The probability that a flow stays large for the next period of time does not seem to be dependent on the network aggregation level. If some distinction between the network aggregation should be made, it seems that more coarse aggregation shows better probability of staying large. The interval length seem to have greater impact on the result. The longer the interval length, the more stable the result. For flows that had been considered large for 5 minutes were the probability 0.5-0.6 that the would stay large for the following 10 minutes, depending on interval length. When looking for probability further into the future, the probability drops fast. Looking at the flows 15 minutes after the first 5 minutes gives a probability of 0.35 to 0.45 that the flows still are considered large and after that the probability drops quickly.

When making a load balance, it is not specific flows that are being moved. The traffic is considered as homogenous and it is only a certain partion of the traffic that gets rerouted. If the link is heavily used, the congestion controls of TCP could have suppressed the actual demand of the TCP flows. When traffic is moved and the capacity is increased, the flows could take up the extra capacity when the occasion is given.

9 Future work

This work has addressed the problem of stability in IP traffic characteristics. Some suggestions and hints have been made in order to continue the work of traffic engineering and stability measurement. Some interesting topics for study are listed below.

- Examine how the total traffic intensity in the interval could be used as a trigger for load balancing.
- How fuzzy logic could be applied in the search for stable large flows, i.e. to measure the probability that a flow will continue to stay large for a certain time.
- With more access to traffic logs and routing configurations, examine the relationship between endpoints and volume of flows.
- Examine how to tune the threshold from which the large flows in the small intervals are going to be separated from the rest. This could be derived from the amount of flows carrying the large part of the traffic in the large interval.
- How the length of each smaller interval influence the extraction of large flows.

10 References

[Abrahamsson. 1999] Abrahamsson, H. Sept. 1999 “*Traffic measurement and analysis*” Technical Report T99:05, SICS. http://www.sics.se/cna/publications/

[Abrahamsson, *et al.* 2000] Abrahamsson, H. Ahlgren, B. Kreuger, P. Oct. 2000.*”IPLDOPT Pre-study Project Report: Related work and Initial Experiment”* Internal report SICS,

[RFC 2702] Awduche, D. Malcolm, J. Agogbua, J. O’Dell, M. McManus, J. Sept 1999. *“Requirements for Traffic Engineering Over MPLS”.* IETF Network Working Group *RFC 2702.* *http://www.ietf.org/rfc/rfc2702.txt?number=2702*

[Bhattacharyya, *et al.* 2001] Bhattacharyya, S. Diot, C. Jetcheva, J. Taft, N. 2001*”Pop-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP”* In the proceedings of the *ACM SIGCOMM Internet Measurement Workshop* *http://www.icir.org/vern/imw2001-papers/58.pdf*

[Brownlee, Murray. 2001] Brownlee, N. Murray, M. 2001*“P Streams, Flows and Torrents: Measuring Stream Distributions in Real Time”,* In the proceedings of *PAM* *http://www.ripe.net/pam2001/Papers/talk_07.ps.gz*

[Cerf, Kahn. 1974] Cerf, V. Kahn, R. May 1974. *“A Protocol for Packet Network Intercommunication”*, *IEEE Transactions on Communications*, vol. 22, pp. 637-648. Summary found on: *http://swig.stanford.edu/pub/summaries/networks/ip.html*

[Claffy. 1994] Claffy, K C. June 1994. *“Internet traffic characterization”* University of California, San Diego. PhD dissertation. *http://www.caida.org/outreach/papers/1994/itc/*

[Elwalid, *et al.* 2001] Elwalid, A. Jin, C. Low, S. Widjaja, I. April 2001. “*MATE: MPLS Adaptive Traffic Engineering*“ In the proceedings of *IEEE Infocom* *http://netlab.caltech.edu/netlab-pub/mate.ps*

[Feldmann, *et al.* 2000] Feldmann, A. Greenberg, A. Lund, C. Reingold, N. Rexford, J. True, F. Aug 2000. *“Deriving Traffic Demands for Operational IP Networks: Methodology and Experience”*, In the proceedings of *ACM SIGCOMM.* *http://www.cs.uni-sb.de/~anja/feldmann/papers.html*

[Floyd, Paxson. 2001] Floyd, S. Paxson, V. Feb 2001. *“Difficulties in Simulating the Internet” IEEE/ACM Transactions on Networking*, vol. **e**9, no. 4, pp. 392-403. *http://www.icir.org/floyd/papers/simulate_2001.pdf*

[Fortz, Thorup. 2000] Fortz, B. Thorup, M. March-April 2000.*“Internet Traffic Engineering by Optimizing OSPF Weights”.* In the proceedings for *INFOCOM.* *http://citeseer.nj.nec.com/fortz00internet.html*

[Huitema. 2000] Huitema, C. 2000.*“Routing in the Internet”.* Prentice Hall ISBN: 0-13-022647-5

[NWG] Network Working Group. July 2001. *“A Framework for Internet Network Engineering”* Internet Draft *http://search.ietf.org/internet-drafts/draft-cheng-network-engineering-framework-01.txt*

[Nilsson, Karlsson. 1998] Nilsson, S. Karlsson, G. June 1999. *”IP-Address Lookup Using LC-Tries”. IEEE Journal on Selected Areas in Communications*, vol 17, no 6, pp 1083-1092.
http://www.comsoc.org/livepubs/sac/public/1999/jun/current.html

[Paxson, Floyd. 1995] Paxson, V. Floyd, S. June 1995. "*Wide-Area Traffic: The Failure of Poisson Modeling*". *IEEE/ACM Transactions on Networking,* vol. 3, no. 3, pp 226-244.
ftp://ftp.ee.lbl.gov/papers/WAN-poisson.ps.Z

[Roberts. 2001] Roberts, J W. Jan 2001.*"Traffic Theory and the Internet" IEEE Communications,* vol. 39, no. 1, pp. 94-99.
http://www.comsoc.org/ci/public/preview/roberts.html

[TEWG] Description of the IETF Internet Traffic Engineering Work Group
http://www.ietf.org/html.charters/tewg-charter.html

[You, Chandra. 1999] You, C. Chandra, K. Oct 1999. *"Time Series Models for Internet Data Traffic"* In the proceedings of the *24th Conference on Local Computer Networks*, IEEE Press, pp 164-171.
http://morse.uml.edu/~kchandra/time_series.pdf

[Xu, *et al.* 2001] Xu, J. Fan, J. Ammar. M, Moon, S B. Aug 2001. *"On the Design and Performance of Prefix-Preserving IP Traffic Trace Anonymization"* Extended Abstract. Presented at the *Internet Measurement Workshop.*
http://www.icir.org/vern/imw-2001/imw2001-papers/69.pdf

[Oregon Routeview] *http://www.pch.net/documents/data/routing-tables/route-views.oregon-ix.net/.*

[MOAT] *Global ISP interconnectivity by AS number – background*
http://moat.nlanr.net/AS/background.html

Websites last accessed March 2002

Appendix A - Glossary

AS	Autonomous System	
BGP	Border Gateway Protocol	*RFC 1771*
ECMP	Equal Cost Multi Path	
EGP	External Gateway Protocol	
IETF	Internet Engineering Task Force	
ICMP	Internet Control Message Protocol	*RFC 792*
IGP	Internal Gateway Protocol	
ISP	Internet Service Provider	
IP	Internet Protocol	
IPv4	Internet Protocol version 4	*RFC 1812*
IPv6	Internet Protocol version 6	*RFC 2460*
ISO	International Organisation for Standardisation	
MPLS	Multiprotocol Label Switching	*RFC 2702*
NAT	Network Address Translation	*RFC 1631*
OSI	Open Systems Interconnection	
OSPF	Open Shortest Path First	*RFC 2178*
PDU	Packet Data Unit	
RIP	Routing Information Protocol	*RFC 1723*
RTT	Round Trip Time	
SUNET	Swedish University Network	
TCP	Transport Control Protocol	*RFC 793*
TEWG	Traffic Engineering Work Group	
TOS	Type Of Service	
TTL	Time To Live	
UDP	User Datagram Protocol	*RFC 768*