

THE TCP/IP PROTOCOL SUITE
Tjaart Blignaut

Index

1. Introduction	page 2
2. The internet Protocol	page 3
2.1 The IP Packet	page
4	
2.2 Internet Control Message Protocol (ICMP)	page 7
2.3 Internet Group Management Protocol (IGMP)	page 8
3. Transmission Control Protocol (TCP)	page 10
4. Establishing a connection with TCP	page 12
5. Conclusion	page 13
6. Bibliography	page
14	

Introduction

In the past people were forced to rely on messengers and mail to deliver information across the globe. Today the World Wide Web has practically taken over the function of every traditional communication. Electronic mail and web browsing has become a primary and very important source of communication in this day and age.

What controls the underlying data transfer of the internet?

TCP/IP

The web is built on a large internetwork more commonly known as the internet. Most information is transferred via the main protocols of the internet. This collective of protocols is called the TCP/IP suite. The TCP/IP protocol suite is so named because of its two main protocols:

TCP (Transmission Control Protocol)

IP (Internet Protocol)

We will call this suite of protocols TCP/IP for the sake of abbreviation.

The internet consists of the following groups of networks:

Backbones: Large networks that are meant to connect other networks, some of them include NSFNET in the US and EBONE in Europe.

Regional networks that connect Colleges and Universities for example.

TCP/IP has long ruled the internet world and probably will for quite some time in the future. Every internet application including FTP, HTTP, GOPHER

IP

The Internet Protocol

IP is the protocol that hides the basic physical network by creating a virtual network view. It is a best effort protocol which means that it sends packets without following them up or checking if they reach their destination. Packets may be lost or even duplicated. IP relies on higher level protocols to address these problems. IP is a connectionless protocol. This means it sends packets without knowing if the receiver even exists.

Addressing

IP uses a 32-bit number addressing system separated by dots. A typical IP address would look like this: 196.125.3.1

The lowest number for IP is 1.0.0.0 and the highest is 255.255.255.255. IP addresses are used to uniquely identify a host on an IP network.

Each host can have numerous IP's but require one as a minimal. An IP packet contains a source IP address and a destination IP address. This is to ensure that data is sent to the right computer on the network.

Classes

For the sake of order IP addresses have been split up into five groups namely A-E.

An A class address would be masked as network.host.host.host. The network is the part of the IP that represents which network it belongs to. The host is mostly random numbers between 0 and 255 within that network range. A class networks range from 1.0.0.0 to 127.0.0.0.

B class addresses range from 128.1.0.0 to 191.254.0.0 and are used by very large corporations who require many IP addresses. They are masked network.network.host.host

C class addresses are in the form of network.network.network.host . These are typically used by smaller corporations. One good example of a C class network is most small ISP's like M-Web. They buy several of these C class ranges to accommodate their users.

D and E class networks will not be discussed as they are not as important as the first three classes.

Reserved IP addresses

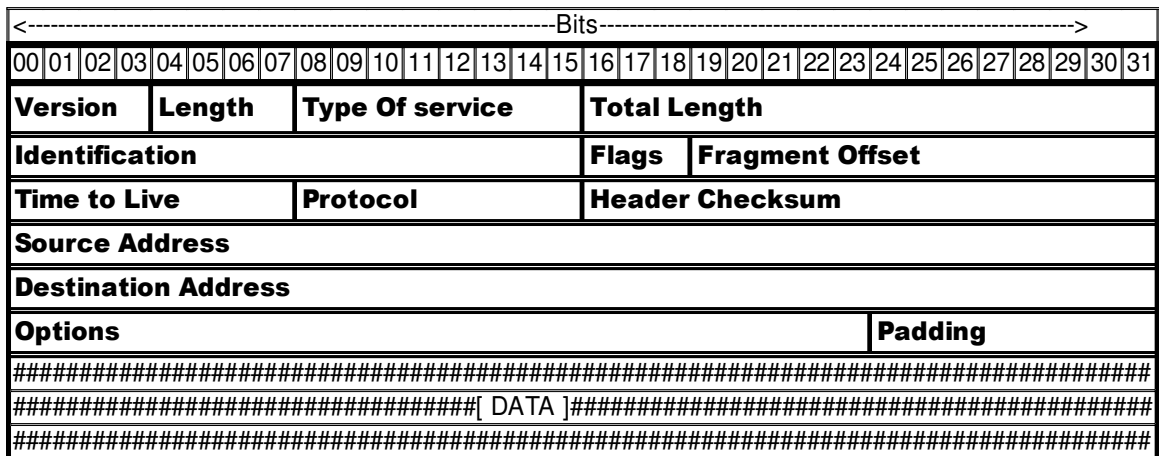
In addition to these Classes there is another there is another range reserved specifically for local area networks. 192.168.0.1 – 192.168.255.255

Another reserved address range is the loopback addresses. These addresses are used for local network interfaces and do not physically access a network. 127.0.0.1 Is the most typical loopback address for a network card.

How IP sends Information

The IP Packet

IP sends its information in small packages more commonly known as packets. A packet does not only contain the data that needs to be sent but also very necessary information of what the data is for, where it originated from, its original checksum, which higher level protocols are involved and some options for the receiver to know how to treat the packet. An IP packet is illustrated in this diagram in a visual and easy to read manner. All IP packets are usually just a bunch of binary 1's and 0's which don't really make a lot of sense to the human brain.



Fields in the IP header

Version

The version field will contain information about the version of IP protocol currently used. For now this number is four but can be soon expected to change because of some problems with this version.

Internal header length (IHL)

This value tells the receiver of the packet how large the header of the IP packet is so that it can easily calculate where relevant data starts. The header size can differ from 20 – 60 bytes because of varying fields and options. The value of the header is expressed as byte divided by 4 to make the size of this field minimal. The maximum value for this field is 15 and the minimum is 5.

TOS - Type of service

This field is specifically for routers to know how to handle an IP-datagram (IP packet). Most commercial applications do not use this field at all or implement it poorly. Here are the codes for the various Types of Service

- | | | | |
|--------------------------|-------------------|-------------------------------|----------------|
| 0 Routing | 1 Priority | 2 Immediate | 3 Flash |
| 4 Flash Override | 5 Critical | 6 Internetwork Control | |
| 7 Network Control | | | |

Total Length

This is the total length of the packet that is to be sent. It includes the size of the header and the other higher level protocols. Computers are supposed to be able to receive packets with a size up to 576 bytes. Anything larger than that must be **fragmented**.

Identification

This number identifies a single packet uniquely so that packets can be reassembled in the correct order when they reach their destination. All packets that are fragmented have the same identification numbers so that that packet will be reassembled correctly.

Flags

This field is used to determine if a packet should be fragmented or not. If the second bit is set to 1 it is DF (don't fragment). If the third bit is set to 1 it means MF (more fragments). Hence the packet has to be fragmented. The first bit is reserved and not used to my knowledge.

Fragment Offset

This field is put there specifically for the destination host to reassemble fragmented packets in the right order.

TTL - Time to Live

Packets are routed to get to their destination. If packets were infinitely routed and never reached their hosts they could travel on the internet practically forever. Therefore each packet has a time to live. When a packet reaches a router one second is taken off the TTL value and it is routed to the next. If the TTL reaches zero before it reaches it's destination it will be discarded by a router.

Protocol

IP does only the basic packet delivery. It requires a higher level protocol to ensure safe delivery and perform more complex tasks. The most popular of these protocols is TCP which will be explained at a later stage.

Some common protocol values are:

Decimal	Protocol	Detailed Description
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol

Header Checksum

This is an online dictionary definition of a checksum

"A computed value which depends on the contents of a block of data and which is Transmitted or stored along with the data in order to detect corruption of the data. The receiving system recomputes the checksum based upon the received data and compares this value with the one sent with the data. If the two values are the same, the receiver has some confidence that the data was received correctly. "

The checksum field is used to calculate if the packet has been received successfully and completely. Predetermined value is in the field. When it reaches the destination host the checksum is calculated from the packet and if the checksums match the packet is kept. Otherwise the packet will be discarded.

Source IP address

This is the address of the sender of the packet. This field is very necessary if a reply is to be sent back.

Destination IP Address

This is the IP of the destination host. It will be used by routers to get the packet to the right place.

Options

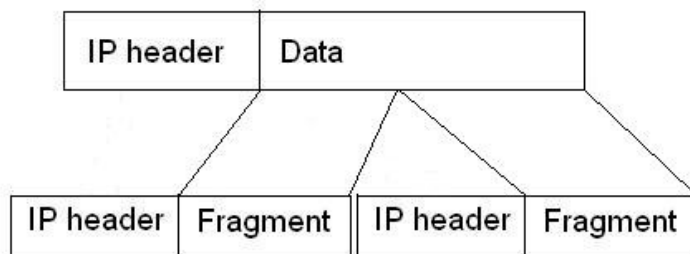
Options are an extra field for adding more functionality to IP. It does not have to be used and can vary in length.

Fragmentation

Each IP network has a limit on packet size. This size is called the MTU (maximum transmission unit). To accommodate these size limitations on networks a router must be able to break packets into smaller chunks called fragments and reassemble them into larger chunks if it's MTU is larger than the fragments it receives.

Every fragment of a fragmented datagram has its own IP header. The identification field of the IP header is used to identify from which datagram the originating packet comes.

The fragment offset field in the IP header is used to determine where each fragment fits on to another to form the original packet.

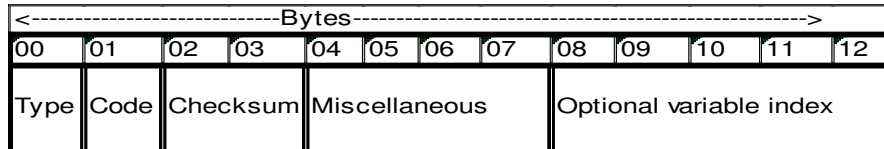


Internet Control Message Protocol (ICMP)

ICMP is a lower level protocol like IP but it performs the function of error messages and notification. ICMP packets are embedded in IP packets and form a very important part of TCP/IP. It is mainly used for notification but also to check if a host exists, which is known as a ping.

ICMP can be used to send information that it is not intended to by means of Tunneling. This is when other non ICMP data is sent through an ICMP packet. Since many firewalls block all incoming ICMP packets this method is not widely used.

The ICMP Packet



ICMP Type

The ICMP type gives us an idea of what the ICMP message is about. The codes are:

- | | |
|-----------------------------------|-------------------------------------|
| 0 - Echo reply | 3 - Destination unreachable |
| 4 - Source Quench | 5 - Redirect |
| 8 - Echo | 9 - Router advertisement |
| 10 - Router solicitation | 11 - Time exceeded |
| 12 - Parameter problem | 13 - Time-Stamp request |
| 14 - Time-Stamp reply | 15 - Information request (obsolete) |
| 16 - Information reply (obsolete) | 17 - Address mask request |
| 18 - Address mask reply | 30 - Traceroute |
| 31 - Datagram conversion error | 32 - Mobile host redirect |
| 33 - IPv6 Where-Are-You | 34 - IPv6 I-Am-Here |
| 35 - Mobile registration request | 36 - Mobile registration reply |
| 37 - Domain name request | 38 - Domain name reply |
| 39 - SKIP | 40 - Photuris |

ICMP Code

The meaning of a code value depends on its type. It gives us an elaboration of the code in the type.

Checksum

This is the same checksum method used for the IP datagram. It also performs the exact same function.

Miscellaneous

This is a variable field that depends on the type of ICMP message. Some

Optional Variable Appendix

This is a field that also depends on the type of information to give more information. For instance to give an address in a domain name reply.

Internet Group Management Protocol (IGMP)

Multicasting and groups

In a large network where one packet needs to be sent to multiple stations imagine how hard it would be to separately send a message to each of its destination, or imagine sending one message to all of the stations on that network. Traffic load is a very important factor in networking and IGMP is responsible for making sure that only the right hosts receive their packets. It uses a user system that can easily be changed and therefore it is very flexible and dynamic. It was invented for Ipv4 networks that support *multicasting.

If a packet needs to be sent to multiple stations there are three ways of doing this. They are:

Unicasting is when information is sent to each station which it is meant for, one by one. This is the most primitive method for sending information to multiple stations.

Broadcasting on the other hand lightens the load on the server. It sends a single packet to a whole network and each receiver must then decide if they want the packet or not.

Unicasting has an adverse effect on the server sending the packets to their destinations. Broadcasting is faster on the sender's side but has a slowing effect on the **whole network**. **Multicast* combines the advantages of both unicasting and broadcasting in a way. Multicast messages are sent to a certain "host group" in a network once. Members of this group will receive the message respectively.

The IGMP packet

The IGMP packet is similar to the ICMP header in many ways. It also uses types and codes to identify different events.

<-----Bytes----->							
00	01	02	03	04	05	06	07
Type	Code	Checksum		Group address			

Type

This field defines what the message is about.

Here are the possible values of an IGMP type:

- 17 IGMP Membership Query
- 18 IGMPv1 Membership Report
- 19 DVMRP
- 20 PIM version 1
- 21 Cisco Trace Messages
- 22 IGMPv2 Membership Report
- 23 IGMPv2 Leave Group
- 24 IGMPv2 Leave Group
- 25 IGMPv2 Leave Group
- 26 IGMPv2 Leave Group
- 27 IGMPv2 Leave Group
- 28 IGMPv2 Leave Group
- 29 IGMPv2 Leave Group
- 30 Multicast Traceroute Response
- 31 Multicast Traceroute
- 32 Multicast Traceroute
- 33 Multicast Traceroute
- 34 IGMPv3 Membership Report

IGMP Code

Some IGMP packets also contain a code that points out more specific information on a type. Only DVMRP and PIM version one currently have codes:

DVMRP Codes

- 1 Probe
- 2 Route Report
- 3 Old Ask Neighbors
- 4 Old Neighbors Reply
- 5 Ask Neighbors
- 6 Neighbors Reply
- 7 Prune
- 8 Graft
- 9 Graft Ack

PIM Version 1 Codes

- 0 Query
- 1 Register
- 2 Register-Stop
- 3 Join/Prune
- 4 RP-Reachable
- 5 Assert
- 6 Graft
- 7 Graft Ack
- 8 Mode

Checksum

The IGMP checksum has exactly the same function as the IP and ICMP checksum and is calculated in almost exactly the same way.

Group Address

Group addresses use class D IP addresses which are specially set out for this function. Addresses range from 224.0.0.0 to 239.255.255.255 where the address 224.0.0.0 is never used and 224.0.0.1 is the "permanent group of all IP hosts". It is basically an identifier of for a group of hosts.

Transmission Control Protocol (TCP)

TCP is a higher level protocol than IP and exists on the transport layer. It ensures safe transfer of data. This means that data that is not lost, altered or corrupted when it reaches its destination. It uses IP to route packets to one host to another, but also makes sure that chunks of data reach their destination successfully. Unlike IP TCP is a connection orientated protocol. This basically means that it exchanges data in a specific order.

1. a connection is established
2. Data is exchanged
3. the connection is closed

Since IP is not reliable for safe data transfer TCP has taken various steps for overcoming the problems with IP data transfer

TCP segments

TCP uses segments to split up data that has to be sent. It is very much the same as fragmentation of IP packets.

The maximum segment sizes (MSS) of networks differ. Therefore in a TCP connection the following happens:

1. Host A connects to host B, and sends his MSS
2. Host B accepts the request and sends his MSS
3. Both compare their MSS to the other's and then choose the smallest one

Because of this TCP packets are called segments because they are not IP packets.

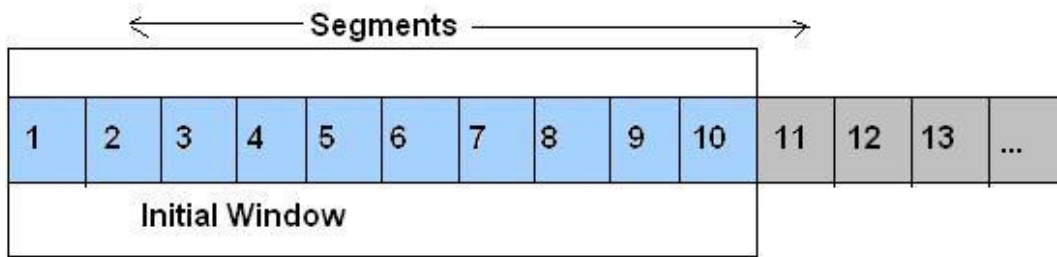
TCP timers and Safe delivery

In order for data to be safely transmitted the sender must know if the receiver ever got the data. Therefore TCP was designed so that the receiver confirms every segment by sending a confirmation. When the sender initially sends the data a retransmission timer is started. If the timer stops before a confirmation is received from the destination host the segment is resent.

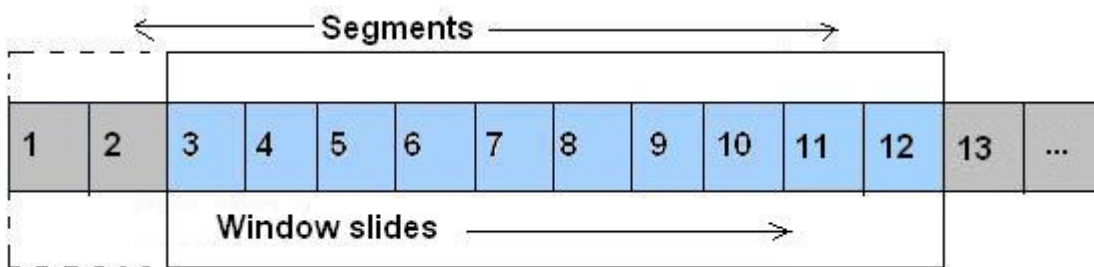
In order to make this timer accurate in a world of differing network speeds there have been several different methods of calculation which are relatively accurate.

Sliding window

TCP doesn't send one segment at a time but rather multiple segments simultaneously; these segments are part of a window.



The figure above shows an initial TCP window that allows TCP to send 10 segments simultaneously. Grey segments have not been sent. Blue segments have been sent and await acknowledgement. When acknowledgement is received for a segment the window slides to the right. Each time acknowledgement is received the window slides. The sliding window is illustrated in the next figure.



Flow control is necessary in this procedure because a server with a very large window could flood a small slow connection with a small window. Therefore we have window advertisements. It basically works like this: the receiver sends a window advertisement with every acknowledgement; this tells the sender how large the window should be. The sender then adjusts his window size according to this advertisement. If a window advertisement is set to 0 the transfer will stop.

Establishing a connection with TCP

TCP is a connection orientated protocol. Which means one hosts connects to the other before transferring data. But how does this really work. With a procedure called the “*Three-way-handshake*”.

Terms in this chapter:

*SYN*chronize – synchronise flag set on a segment

*ACK*knowlegde - acknowledge flag set on a segment

*FIN*ish - finish flag set on a segment

RST – reset flag on a TCP segment

Step One – Request

The host requesting the connection sends a SYN TCP segment (synchronize) with a valid source and destination port number, a SYN flag, no data and the ISN.

The ISN is the Initial Sequence Number; this is a number to identify a connection primarily for security reasons.

The client now registers a dynamic port (49152 to 65535) and connects to a port on the server for instance 80 which is HTTP.

Step 2 – Reply

When the server receives the synchronize request he can choose to accept or refuse the connection. If the server wants the client to connect it will send a segment with a SYN and an ACK flag. If the server rejects the connection it will sent a segment with an RST flag.

Step 3 – connection

The client has sent the request to connect to the server, the server replied with a valid checksum and a sequence number. The client now sends an ACK segment to acknowledge the sequence number of the server. The client also sends his Window advertisement in this ACK segment. If anything goes wrong at this time the server will send an RST segment (reset) and therefore close the connection.

Step four – closing a connection

Closing a connection happens in four steps. This is because both the client and the server might be sending and receiving data from each other at the same time.

If the client wants to close the connection it sends a FIN and an ACK segment containing the client and server sequence numbers. After this the client does not send anymore data and waits for the server to acknowledge his FIN segment. If the timer for the Fin segment runs out the client will send it again.

The server responds by acknowledging the FIN segment it received. It finishes sending its data and then sends a FIN segment to the client. The connection is now closed.

Conclusion

TCP/IP is a complex set of protocols with an ingenious design that have made internetworking across the globe a reality today. Their seamless functionality with practically any application has won them the rightful place of the most popular and widely used protocols on the globe.

Even though IP is a very great protocol the current version has a few shortcomings. All newer operating systems support the new version of IP which will probably be implemented within the next few years. Ipv6 is an even simpler protocol than IP and much better. It also supports more IP addresses. Because of the recent flooding of the class C address range IP's are becoming rare and expensive to buy. Hopefully Ipv6 will solve this problem and make IP the more reliable and secure.

When TCP/IP was first developed security was easily exploited and these methods are still used today to exploit networks across the world. Sending a Fin packet for instance where a Syn packet was expected.

TCP/IP has definitely proven it's scalability with the rising of the internet over the past decade. The internet is growing and will keep on growing. Will these protocols keep up?

I definitely think so. Nothing wasn't accounted for when they were created and nothing was left to chance.

These are definitely the best set of protocols around and will probably be for a very long time.

Bibliography

21/04-2000.

Re-edited 6/21/01.

Published on [Blacksun Research Facility {BSRF}](#) -

Written by WATER

Notable events during TCP/IP

Historyhttp://www.cs.utexas.edu/users/chris/think/Early_Days_Of_TCP/Introduction/

Internetworking - The Basics about IPv4, ICMP and IGMP

Internetworking Part 2 - The Transport Layer

[blacksun.box.sk](#) tutorials

TCP/IP Tutorial and technical overview

IBM redbooks – www.ibm.com/redbooks

