Notes on "TCP/IP Illustrated" Version 10 (with changes noted)

Prof. J.H. Davenport

December 17, 2009

Abstract

The book series *TCP/IP Illustrated*, by the late W. Richard Stevens, is a classic of the TCP/IP literature. Volume I is the keystone, describing and illustrating the TCP/IP protocols in action. Nevertheless, it is beginning to be dated in a few places, and it is somewhat Americano-centric. These notes try to update this excellent book. Actual corrections of errors, rather than updates or comments, are given in *italics*. References into the book are generally given in terms of pages ("p.") and lines ("l."). References beginning III are to Volume III.

I am grateful to my friend Mr. I.W.J. Sparry for many comments and corrections, as well as the many explanations he has given me over the years. Many people in BUCS have provided useful explanations, and helped this course evolve as the campus networking, and networking technologies generally, have evolved. Several students have made useful comments over the years I have taught Networking — please continue to do so! In particular Jon Mason pointed me to the updates at [21] and

> Also it's worth noting that the entire book is available online as an eBook for free via the library: `http://proquest.safaribooksonline.com.ezp1.bath.ac.uk/0201633469########`.

Matthew Bewers provided the information in note 41 (page 19).

Preface

p. xviii The statement about SunOS 4.1.x was probably not even true when Stevens was finally published: it is certainly not true now (not least because 4.1.3 had some serious Year 2000 issues). Among UNIX-based implementations, Linux is almost certainly the most popular (by number of hosts) on the Internet. However, a large number of web servers etc. are still based on Suns, running, in general, some version of Solaris. The vast majority of machines on the Net these days[1] are running Windows of some

[1] An article in the *Sunday Times* (19.12.1999) credited Linux with 14% of "new servers", versus 38% for Windows NT.

flavour, now that Windows is shipped with TCP/IP. Stevens' statement that most of the serious research is done with Berkeley-derived systems is still true, though.

Chapter 1

Figs. 1.1/1.2 These figures display the traditional TCP/IP **4-layer** model. In the international standards world it is also common to meet the ISO[2] **7-layer** model. The diagram below shows how the two are related, and gives, analogously to Figure 1.2, an example of each layer in the 7-layer model.

Example	ISO layer	ISO name	TCP/IP name
NFS	7	Application	Application
XDR	6	Presentation	Application
RPC	5	Session	Application
TCP	4	Transport	Transport
IP	3	Network	Network
Ethernet	2	Link	Link
volts/Hz	1	Physical	Link

We will come back to this example in the conclusions, but for the moment let us say that

1. The physical layer describes what passes electrically down the cables, and any physical requirements on plugs/sockets/cables. As we will see, this can differ in different implementations of the Ethernet suite: 10base5 versus 10base2 versus 10baseT etc. The TCP/IP model wraps this in with the next layer. This can matter because the physical layer (wires and hubs) has length limits (500m for 10base5, 195m for 10base2, 100m for 100baseT etc.), whereas the link layer (switches) in theory does not.
2. The link layer specifies the digital interface: which bits in the Ethernet frame mean what. The link layer can be of varying complexity, from a modem link to a large ATM[3] network or a 2.5G/3G mobile network, as we will see in chapter 2.
3. Much the same in the two models.
4. Much the same in the two models. Despite the name, TCP/IP supports two major transport protocols, TCP[4] and UDP[5], as well as many more specialised ones. The existence of two major transport

[2] ISO = International Standards Organisation.

[3] ATM = Asynchronous Transport Mode, a telephony-based system for long-distance large-scale networks.

[4] TCP = Transmission Control protocol: RFC 793, as updated by RFCs 1122, 2581 and 3168.

[5] UDP = User Datagram Protocol: RFC 768, as updated by RFC 1122.

protocols is really a matter of "horses for courses", as we will see in the conclusions.

5. In the various functions that the TCP/IP model lumps under "Application", one concerns the connection of one application and function, typically from client to server. In the example above, this is performed by RPC[6] (see also section 29.2), which connects, say, a "read" request on the client to the procedure to perform this on the server. Since UDP is unreliable, RPC has to build in a re-transmission/time-out system at this level, equivalent to the one that TCP provides at level 4.
6. TCP/IP, and other networking systems, concern themselves with the transmission of bytes (known as 'octets' in RFC terminology), and not with the interpretation of these octets. Since there are several representations of integers ("big-endian" versus "little-endian"; sign-and-magnitude versus two's-complement versus unsigned), floating-point numbers (IEEE, VAX and IBM) etc., conveying information in these formats between heterogeneous hosts requires a neutral standard. XDR[7] (see also section 29.3) is one such. ANS.1/BER[8], as used in SNMP (see pages 386–7), is another one, as is MIME (section 28.4 and III chapter 13), or NVT ASCII (Chapter 25 and elsewhere).. This point is taken up in the conclusion to these notes.
7. NFS[9] (see also chapter 29, especially 29.5) provides for one (client) computer to read/write files, and generally access a UNIX-like model of a filing system, on a remote file server. It uses XDR to transmit 32-bit integers (length of files; modification times etc.), and RPC to indicate which action (read, write, delete etc.) should be performed.

Of course, these diagrams and the associated notes were written from the point of view of a TCP/IP implementor, who would view ATM as one possible level-2 medium[10]. An ATM engineer would view various parts of ATM as providing ISO levels 1–4 (in particular, ATM itself proving layer 3, and the ATM Adaptation Layer providing level 4), and TCP and IP together as providing level 5: the connection between one TCP/IP

[6]RPC = Remote Procedure Call; originally a Sun implementation of a generic concept, but now an Internet RFC (1057).

[7]XDR = eXternal Data Representation; originally a SUN implementation, but now Internet RFC 4506.

[8]ASN.1 = Abstract Syntax Notation 1. BER = Basic Encoding Rules. These are in fact not Internet protocols, but were originally developed in ISO — ISO 8824 (1987) in the case of ASN.1.

[9]NFS = Network File System; again originally a Sun implementation, but now Internet RFC 1094 (for NFS version 2), 1813 (for NFS version 3) and 3010 (RFC 3010 obsoleted the previous RFC 2624) (for NFS version 4). It has now (November 2003) been declared that 3010 obsoletes 1094 and 1813, but is itself obsoleted by RFC 3530.

[10]More details about IP over ATM can be found in chapter 18 of Comer,D.E., *Internetworking with TCP/IP vol. 1: Principles, Protocols and Architecture*. 3rd. ed., Prentice-Hall, 1999.

application session and another. Similarly, a 2.5G/2G mobile network engineer would regard PPP (assuming that's the link layer used by IP) as level 5 layer, and internal parts of the 2.5G/3G network would provide transport, network, link and physical functions.

Fig. 1.3 This figure shows a router as a box which can take in IP packets and forward them appropriately. There can be similar functionality at other levels.

Hubs These are at the Ethernet physical level, and forward packets between two Ethernet segments of the same technology.

Bridges These are often seen with Ethernet, and from the point of view of higher layers, operate at the Link level. They connect two (or more) different Ethernets, and pass Ethernet frames from one net to another as necessary to ensure than any two hosts on the bridged set of Ethernets can communicate as if they were on the same Ethernet. Bridges can be used to connect two Ethernets of the same technology to extend length limitations (e.g. 500m to 1500m for "thick" Ethernet), or to bridge networks of different technology, e.g. a "thick" backbone with various "thin" spurs, or, quite common these days, a 1Gb Ethernet with 10Mb or 100Mb spurs; or 10Gb with 1Gb spurs.

Switches These, sometimes also called bridge/routers, operate with Ethernet at level 2. They will re-broadcast packets from one net to another *if required*, either if programmed, of if they have learnt that the destination is on another segment (so-called "learning bridges"). The hubs required for twisted pair Ethernet fall into the same category.

ATM switches fall into the same category as far as IP is concerned, though not from the ATM point of view.

However, it is worth noting that some devices break this layering, e.g. the CISCO 2950:

> The 2950 is a multilayer switch, it supports layers 2-4 for some services. It can do filtering based on source/destination IP address or port. It also supports QOS based on port number.
>
> It can not do forwarding based on IP address therefore it is not considered a router (layer 3 switch).
>
> The difference is that a layer 3 device normally can also do other functions (NAT) and can do intelligent forwarding base on the IP address. The 2950 can filter at layer 3 and 4, but forwards are based on the MAC addresses only.

Base Stations These are seen with wireless Ethernet, or 2.5G/3G mobile networks. They perform much the same functions as switches, except that the network of base stations has to deal with issues such as handover/takeover as a mobile unit moves from one coverage to another. Typically, they also have to deal with much lossier media than fixed

wiring, so need more sophisticated error detection/correction than cabled Ethernet's traditional "if the CRC doesn't match, throw it away" philosophy.

Routers These, as we have seen, operate at ISO level 3. The higher layers (TCP/UDP and above) do not see them, and from the point of view of level 2, they are just more nodes.

Firewalls These are generally routers (though they may also be switches or even bridges) which may decide not to forward certain IP packets (or Ethernet datagrams if they are level 1/2 objects) because they are in breach of some security policy. Though they essentially operate at one ISO level (normally 3), they may look at level 4 (or higher) information to decide whether the packet should, or should not, be routed. For a good description of firewalls and their rôles, see [6].

Firewalls often have Network Address Transation (NAT) functionality, see Appendix B.

Application gateways These operate at ISO levels 5–7. The classic example today is that of a web cache, which reads the full application-layer request, and either satisfies the request itself, or sends the query on to another machine, collects the response, possibly caches it, and then responds to the original requester. Mail relays (Figure 28.3) are another example.

RFC 3234 provides a taxonomy of these and many other "middleboxes" — a growing phenomenon on the Internet. RFCs 3303 and 3304 address the architecture of middleboxes. One kind in particular are Network Address Translators — see RFC 2663 and Appendix B. RFC 3234 says that the growth of this phenomenon is a matter of concern for several reasons.

- New middleboxes challenge old protocols. Protocols designed without consideration of middleboxes may fail, predictably or unpredictably, in the presence of middleboxes.
- Middleboxes introduce new failure modes; rerouting of IP packets around crashed routers is no longer the only case to consider. The fate of sessions involving crashed middleboxes must also be considered.
- Configuration is no longer limited to the two ends of a session; middleboxes may also require configuration and management: the area addressed by RFCs 3303 and 3304.
- Diagnosis of failures and misconfigurations is more complex.

p. 6 It is important to note that while layering, as described in figure 1.4, is critical to the description of protocols and protocol families, it is *not* necessary for implementation, and indeed may be harmful to a high-performance implementation. The reason for this can be seen in figure 1.7, describing the additions to a piece of data as it passes down the protocol stack. If the implementation is strictly layered, then the user data has to be copied three

times in the process: a truly efficient implementation can generally get by with one copy (in UNIX terms, this should also be the copy from user to kernel space). See [7] or the seminal RFC 817 for an explanation. In a special-purpose router (e.g. Cisco, 3Com), it is normal to arrange that most packet data are never copied, at least for straight-forward cases.

Another example of the violation of layering for performance, in this case overhead minimisation, is given in the discussion on header compression (see the notes to page 31). Firewalls also tend to violate layering, as do some routers (see these notes or Stevens p. 244). Network Address Translators (Appendix B and RFC 2663) also have to violate layering.

Fig 1.5/6 *Class E addresses are defined by the high-order 4 bits all being one. Therefore, in Figure 1.5 the 5th bit from the left, which contains 0 in the class E address, should be removed. The number of bits reserved for the future should be 28, not 27. In Figure 1.6 the ending address of the range for class E addresses should be 255.255.255.255, not 247.255.255.255.* [21]

Fig. 1.6 While this shows the ranges of numbers available, there is more that could be said.

Class	Networks	Hosts/Network	Total Hosts
A	$128 - 2 = 126$	$2^{24} - 2 = 16777214$	2113928964
B	$64 \times 2^8 - 2$=16382	$2^{16} - 2 = 65534$	1073577988
C	$32 \times 2^{16} - 2$=2097150	$2^8 - 2 = 254$	532676100

(The reason for the -2 is that networks and hosts of all 0s or all 1s (in binary) are special — see p. 45.) Thus, although more than half the host numbers are on Class A networks, over 99% of networks are Class C networks. This point is discussed further under CIDR (p. 140).

p. 8 Since a router is merely a specialised host, it also follows that the router in figure 1.3 will have two IP addresses: one for the Ethernet and one for the token ring.

p. 8 Things have moved on in the IP network number allocation business since Stevens wrote this book. IP networks in Europe are allocated by RIPE[11] and in the Asian–Pacific region by AP-NIC from blocks originally sub-allocated to them by the InterNIC. Allocations in the Americas are made by ARIN. See `www.{arin,ripe,apnic}.net`. This allocation method has the additional advantage that networks in a whole range will have a single trans-atlantic route, in general — see the discussion later on CIDR (p. 140). The allocation process is described in RFC 2050. See also `http://www.iana.org/assignments/ipv4-address-space`.

p. 8 Some IP addresses are reserved for private internets (see RFC 1918):

[11] RIPE = Réseaux IP Européens = European IP Networks, a consortium of the major national networks in Europe.

- 10.0.0.0 — 10.255.255.255 (a Class A address)[12];
- 172.16.0.0 — 172.31.255.255 (16 Class B addresses)[13];
- 192.168.0.0 — 192.168.255.255 (256 Class C addresses).

From the point of view of CIDR (see page 140), these can be regarded as[14] 10/8 (network 10, with 8 bits of network ID and 24 bits of subnet/host information), 172.16/12 and 192.168/16.

The use of private internets and their connection to the public Internet via various interfaces has changed the interpretation of IP addresses — see RFC 2101 for an analysis.

p. 12 One half-way house which is used by some Web servers is to create some fixed (but probably configurable) number of threads, and then place new incoming requests on the queue of one thread. This avoids the cost of creating a new process/thread for each request, and limits the maximum load on the system's resources. This is particularly relevant when, as in the case of the Web, the requests are fairly short, but too long to block for.

Conversely, the `inetd` solution, common in UNIX, is to have one concurrent server handling many types of requests, forking not a copy of itself, but the appropriate program, e.g. a mail listener or an FTP server. This cuts down on the number of processes and the occupancy of memory by unused servers.

p. 14 Since the publication of this book, RFC 2119 has been published, which clarifies the meaning of words such as MUST in RFCs.

pp. 14–15 The latest version of the *Assigned Numbers* RFC is RFC 5000 (May 2008), but in fact it is now necessary to go to the various files mentioned in it to get the latest status, notably `http://www.rfc-editor.org/rfcxx00.html`, which is updated daily. For example, 5 new IP options and 4 new ICMP types have been added since RFC1700, as mentioned by Stevens, was published. RFC 3232 describes this database.

RFC 3233 provides an up-to-date definition of the IETF. See also RFCs 2026, 2028, 3777, 3978, 3979, 3932, 4748. *I am not happy with Stevens' "The IETF develops the specifications . . .". This is certainly not true now, and I doubt it ever was. "ratifies" might be a better word.*

p. 15 The *Internet Official Protocol Standards* RFC is now (September 2007) RFC 3700 (July 2004).

[12] This is the Class A address allocated to the original ARPAnet.

[13] In use at Bath for IP addresses on the internal "Classic IP" network, and for ResNet and library docking points. For understanding how the latter accesses the wider Internet, see Appendix B.

[14] Notation is somewhat confused here: both 10/8 and 10/24 have been in use. 10/8 seems to be more common now, and is used in RFC1918. The "official" notation (RFC1518 and the various registries) is <address,mask>, as in `<10.0.0.0,255.0.0.0>`.

p. 15 The revision of RFC 1009 appeared as RFC 1716, but was in turn obsoleted by RFC 1812 (itself updated by RFC 2644).

Fig. 1.9 Unlike the UNIX convention, where time is the number of seconds since 1 January 1970, the `time` protocol is indeed the number of seconds since 1 January 1900. The difference is that in UNIX, `time` returns a (signed) integer, whereas the `time` protocol returns an unsigned integer. So the `time` protocol wraps round in $\frac{2^{32}-1}{60*60*24*365} = 136$ years after 1900, i.e. 2036[15].

p. 16 Various terms are common to describe flavours of networks: Stevens correctly distinguishes any old internet from **The Internet**. Other common terms are given below.

intranet There is no precise definition, but generally it consists of a variety of TCP/IP based services (Web, mailing lists, news groups etc.) running on an internet (generally connected to the Internet via a firewall and/or application gateways) belonging to some organisation, but these services are not visible outside the organisation. Very common in large companies. Such an intranet may well use the "private" IP addresses mentioned under page 8 above.

extranet Confusingly, this word seems to have two different, almost contradictory, meanings.

extranet (1) In opposition to intranet, to mean those Web pages etc. that the organisation *does* want to be visible outside. Often used as in "webmaster" to customer: "Do you want this information just on the intranet, or on the extranet as well"?

extranet (2) Like an intranet, except that the network no longer belongs to a single organisation, but rather to several co-operating institutions. The large car companies, in particular, often have these, which can incorporate the dealers at one end, and suppliers (often going several deep in the supply chain) at the other. Again, the key is that the information is private to the organisations belonging (but greater concern needs to be paid to internal issues of privacy etc.).

LAN =Local Area Network. Originally meaning just a single network (e.g. an Ethernet), but now meaning a collection of inter-connected Ethernets etc. spread across a relatively small area, and under the control of one organisation. A typical example would be the Bath campus LAN: well over 140 Ethernets of various kinds (10Mbps, 100Mbps; co-axial (very little now), UTP and optical fibre) connected by bridges, routers and a backbone[16].

MAN =Metropolitan Area Network. The original hyperbole was that there would be "wired cities", with an all-pervasive network, which

[15] Still true in IPv6, according to RFC 4330.

[16] Initially ATM, then moved to 1Gb Ethernet in 2003, and 10Gb Ethernet in summer 2006.

was a utility like electricity or water. This has not happened (except in a few cases), but in the UK the term is heavily used within academia, as universities are being pushed into regional consortia. Bath used to be in the BWEMAN[17] and is now in SWERN[18]. The term "Metropolis" is somewhat stretched: Glasgow and Aberdeen universities were both in the "Scottish MAN". These networks are (relatively) geographically compact, and are normally under the control of a small consortium.

WAN =Wide Area Network. The UK's national academic network JANET is one example, and large company or government networks are others. The US military's MILNET covers about as wide an area as possible. The management of these, at least up to IP level 3, is often sub-contracted to a specialist company, e.g. SuperJANET 5 (the current incarnation of JANET) is contracted out to Verizon, and the "Fat Pipes"[19] are contracted out to Sprint (as of January 2007).

VPN =Virtual Private Network. As the name implies, not a real network at all, but the use of existing networks on convey a virtual private network. The University of Bath provides one: see `http://www.bath.ac.uk/bucs/ad/vpn/`. This will use some form of IP-in-IP[20], i.e. using IP as a link layer under IP, to get the messages from one end to the other of the virtual network across the underlying physical network. This may or may not also provide security.

p. 19 Estimating the size of the Internet is even more difficult now than it was for Stevens. The number of allocated networks is unmeasurable without knowing the InterNIC's allocation policy (see notes for page 8). Estimating the number of machines is also harder, with the spread of dial-up and broadband services, free ISPs, and the fact that many such machines may be registered with several such ISPs. However, they are unlikely to have dedicated IP numbers: RFC 2050 strongly discourages this, and recommends DHCP or equivalent technology. An estimate published in the Financial Times was that, at the end of 1998, there were 90 million machines with Internet access — more recent estimates are very coarse, though the British Office of National Statistics estimated that, in 1996, "57% of British households are connected to the Internet", whatever that

[17]Bristol and West of England MAN. The main partners were the Universities of Bath, Bristol and West of England, and HP Laboratories in Bristol, but it also served Bath Spa University College, Cheltenham and Gloucester College, the Higher Education Funding Council for England, and over a dozen further education colleges, as well as 150 schools via Bristol City Council.

[18]BWEMAN merged with the network that served Exeter, Plymouth etc. to form the South WEst Regional Network (SWERN).

[19]The "fat pipe" was the name for the one megabit-per-second (Mbps) link connecting JANET to the US' NSFNET in the early 1990s. Demand has mushroomed since then, and at the start of 1999 it was a 155Mbps link, and then (March 2001) went to four such links, with a plan to move to a 2.5Gb link. It is now (January 2007) a 10Gb link with backups.

[20]This is the right level at which to do this. TCP-over-TCP is in practice a disaster, with two sets of time-outs and error recovery fighting each other.

might mean. The Financial Times (19.2.2007) quotes "2500 million people are connected to the Internet". Internet traffic is also growing, doubling every year[21].

The Internet has also been growing in diameter, i.e. the number of routers between two typical points. This has an effect on the "Time-To-Live" field — see the discussion on page 36. However, in 1997–99, the diameter actually decreased, as lengthy IP–IP paths in networks such as JANET or other backbones were replaced by WANs, often by ATM-based WANs, which only count as one IP hop, irrespective of the number of ATM (Level 2 as far as IP is concerned) switches that are traversed. For example, UUNET, one of the geographically largest ISPs, is ATM-based, and always "one hop" as seen at level 3.

The growth in the Internet, the variety and number of machines (particularly routers) running it, and the widely-distributed nature of its management, all mean that evolution is slow. The Appendix to these notes gives an example of how this affected the University of Bath. The discussion on page 50 about a new generation of IP represents probably the biggest incompatible transition that the Internet will need to make, and the magnitude of that transition[22] is worrying many people.

Figure 1.10 Since Stevens wrote this, a version known as NewReno has emerged, and [16, Table 6] reports that this accounts for 76% of the web servers they could classify.

Chapter 2

p. 21 The original DEC/Intel/Xerox Ethernet (at 10Mb/sec: there was an experimental version at 3Mb/sec) is now known as "thick" Ethernet, and was carried on a bulky co-axial cable, with quite severe restrictions on bend radius etc., and with precise specifications for tapping a station into the cable. This had a maximum length of 500 metres, and a delicate means of connecting machines to it — the so-called "vampire tap". This is now known as **10base5** Ethernet, the "10" standing for the 10Mbps transmission rate of frames and the "5" for the maximum length. A newer version, known as "thin" Ethernet or "cheapernet", was carried on much thinner and more flexible co-axial cable, with relatively simple BNC connectors. There was a smaller maximum length (185 metres) but this proved much more suitable for cabling offices etc., particularly in existing buildings. This is known as **10base2**[23] Ethernet. More recently, Ethernet-format

[21] At least until 2002, and there is no reason to suspect a decrease. See [17]. This paper also debunks the "doubling every 100 days" (= 10 times every year) myth.

[22] Estimated at $\$25 \cdot 10^9$ — `http://triangle.bizjournals.com/triangle/stories/2006/01/30/daily41.html`.

[23] The length restriction was to be 200m, and the abbreviation stuck. Anyway, who would say **10base1.85**?

signals can be carried on category 5 UTP[24], when it is known as **10baseT** (limited to 100m to the hub) or on optical fibre **10baseF**.

Higher-speed Ethernet variants are possible on these last two media, and 100Mbps Ethernets (**100baseT** over twisted pair, limited to 100m, or **100baseFX** over optical fibres, limited to up to 20km, depending on the precise optical nature) are available, and 1000Mbps products[25] such as **1000baseT**, limited to 100m[26], or **1000baseSX** (500m) and **1000baseLX** (2km or even 120km — see `http://www.premisesnetworks.com/content/productshowcase/product.asp?docid=0c674b9f-edeb-11d2-94bd-00a0c9b3bdf2`) over fibre. On 13 June 2002, IEEE ratified 802.3ae, 10 Gigabit Ethernet, which allows 300m over multimode fibres and 40km over single-mode fibres. Progress beyond 10Gb seems stalled — see `http://www.channelregister.co.uk/2007/06/18/iee_hssg_talk_about_100gb_e/`.

Ethernet is traditionally thought of as a 'local' protocol, but even attitudes to this are changing.[27]

> Ethernet was initially defined as a LAN technology to interconnect the computer within a small organization. Over the years, Ethernet has become such a popular technology that it became the default OSI Layer 2 mechanism for any data transport. [24, IEEE 802.1ah]

Hence there are new technologies such as IEEE 0802.1ad and 802.1ah (due to become a standard in November 2008) for supporting "Ethernet over Ethernet" Virtual Local Area Networks (VLANs).

There have been numerous demonstrations recently of our dependence on (particularly submarine) optical fibres: see `http://news.bbc.co.uk/1/hi/technology/7794868.stm` (and `http://www.orange.com/en_EN/press/press_releases/cp081219en.html` for the impact onvoice traffic, but the IP impact is probably similar) for the most recent (December 2008).

p. 22 Ethernet "hardware addresses" are assigned by the manufacturer of the Ethernet chip or card, from a range that the manufacturer is allocated by the IEEE[28] — see RFC 1700 for some such allocations, and `http://www.iana.org/assignments/ethernet-numbers` for the most recent list. Hence the address can tell one something about the nature of the machine.

[24] UTP = Unshielded Twisted Pair, i.e. telephone cable. CAT 5 UTP is the version commonly installed in buildings today. It has been estimated (`http://www.grouper.ieee.org/groups/802/3/tutorial/march98/mick_170398.pdf`) that 70% of all installed UTP is CAT 5, and the footage of CAT 5 installed is growing at 30% per annum.

[25] For more details, see Ferrero,A., *The Eternal Ethernet*. 2nd. ed., Addison-Wesley, 1999.

[26] Or possibly 60m. And this seems to require the, as yet unpublished, CAT 6 version of UTP — however, much CAT 5 seems to comply in practice. IEEE has now (2 June 1999) ratified IEEE 802.3ab, a standard for 1000baseT, which is Gigabit Ethernet over *four* pairs of CAT 5 wiring, up to 100m. It also allows auto-negotiation between 100Mbps and 1Gbps. See `http://www.gigabit-ethernet.org/news/releases/062999.html`.

[27] `http://www.bcs.org/server.php?show=ConWebDoc.19150`.

[28] Institute of Electrical and Electronics Engineers: `www.ieee.org`.

Table 1: (Some!) Ethernet formats

name	speed MHz	connection type	distance (maximum)	standard (if any)
10base5	10	thick coax	500m	802.3(8)
Thick Ethernet		(yellow)		
10base2	10	thin coax	195m	802.3(10)
Thin Ethernet		(black)		
10baseT	10	CAT-5 UTP	100m	802.3(14)
10baseF	10	Fibre	?	802.3(15)
(also known as FX, FL)				
100baseT	100	CAT-5 UTP	100m	802.3(24)
(strictly speaking, 100baseTX, as there are obsolete variants)				
100baseFX	100	2×MM fibre (lasers)	2km	802.3(24)
100baseSX	100	MM fibre (LEDs)	300m	
100baseBX	100	SM fibre	10km	802.3
1000baseT	1000	CAT-6(5e) UTP	60m	802.3(40)
1000baseSX	1000	MM fibre	550m	802.3
1000baseLX	1000	MM fibre	550m	802.3
or		SM fibre	2km (10km)	802.3
1000baseLH	1000	SM fibre	100km	vendors
10GbaseT	10000	twisted pair	100m	802.3an
(not yet in production)				
10GbaseSR	10000	MM fibre	$\leq$ 300m	802.3ae
(length depends on fibre type)				
10GbaseLR	10000	SM fibre	10km	802.3ae
10GbaseER	10000	SM fibre	40km	802.3ae
40GbaseSR4	40000	(new) MM fibre	100m	(802.3ba)
40GbaseLR4	40000	SM fibre	10km	(802.3ba)
100GbaseSR10	100000	(new) MM fibre	100m	(802.3ba)
100GbaseLR4	100000	SM fibre	10km	(802.3ba)

(The '4' in 100GbaseLR4 is *not* a typo: see slide 8 of `http://www.ieee802.org/3/ba/public/may08/ganga_02_0508.pdf`.)
(UTP = Unshielded Twisted Pair; MM= multi-mode, SM=single-mode)
802.3ba is not yet (July 2009) ratified, but JANet has run trials (JANET News, June 2009, p. 3).

These will stay with the card or chip for life. So, if a machine is transferred from one University to another, it will keep its Ethernet address, but acquire a new IP address from the range of the new owners. One can think of the Ethernet address as being like the chassis serial number on a car, which is the same even if the car transfers countries (or, in France, departments) and has to be re-registered.

It is normal to say that the Ethernet (or MAC) address is not visible outside the Ethernet it is on. However, JHD was recently in Canada, and logged on from

```
cpe0016b4cac146-cm000e5c6d2bb0.cpe.net.cable.rogers.com
```

where the `cpe` component was in fact the Ethernet address of his laptop.

p. 22 Note that the RFC 894 encapsulation has no "length" field. The length of the Ethernet frame is deducible from the frame (else the hardware would not know which four bytes were the trailer), but of course this will include any padding to the minimum length of 60 bytes (+ trailer). Since IP has its own length field (see page 36), this is not a problem with RFC 894 encapsulation, but the designers of IEEE 802.2/802.3[29] wanted to be able to carry data that was not self-describing, so needed a length field.

p. 22 The 1500 bytes maximum payload was a necessary restriction on 10base5 and 10base2, in order to prevent one site hogging the shared medium for an excessive time. With the move to non-shared media (baseT, baseF), this is not such a problem. However, the frame size has been kept[30] at 1500 to allow bridging at the Ethernet level between different media/speeds: it is common for a 10baseT hub to have a 100baseT outlet to the main network, for example. Some 1000base manufacturers allow for larger frames: Alteon allows "jumbo frames" of 9000 bytes, for example. However, since Ethernet has no provision for fragmentation[31], these cannot be bridged at the Ethernet level to other media/speeds, or even other vendors that don't support this option. Interoperability is a great enemy of change.

p. 22 (This note is only for those of a mathematical inclination.) The sort of CRC used for Ethernets is defined as follows. Choose a Boolean polynomial f (i.e. a polynomial whose coefficients are integers modulo 2, i.e. 0 and 1) of degree 32 such that:

1. the polynomial is irreducible[32], i.e. has no proper factors;
2. the polynomial is primitive, i.e. the powers of x, from x^1 to $x^{2^{32}-1}$ modulo f are all different.

[29] IEEE 802.3 has been re-baptised as ISO 8802-3 (1990).

[30] IEEE 802.11, the standard for wireless Ethernets, says that the maximum length can be at most 2304 bytes (section 6.2.1.1.2).

[31] See Chapter 11 for IP fragmentation.

[32] As a polynomial over the field with two elements, not just irreducible over the integers.

Such a polynomial is $x^{32} + x^7 + x^6 + x^5 + x^4 + x^2 + 1$. Regard the whole message as a polynomial in x, whose coefficients are the bits in the message, and the CRC is the remainder when this is divided by f.

In practice, this is easy to compute in hardware: build a 32-bit linear feedback shift register corresponding to f. In the case above, this would feed out of the top of the register back (via exclusive-or) into bits 7, 6, 5, 4, 2 and 0. Then feeding the entire message through this register leaves the remainder in the shift register. The use of a shift register with feedback explains the word "cyclic" in "Cyclic Redundancy Check".

PPP uses a CRC of the same nature as Ethernet, except that it is, by default, based on a polynomial of degree 16. The polynomial in question is $x^{16} + x^{12} + x^5 + 1$ — see RFC 1662 for details and a fast software implementation for 16 or 32 bits. It is possible for PPP links to negotiate the use of a 32-bit CRC.

This sort of CRC is efficient, in the sense that it will detect all one-bit errors, nearly all single bursts of errors and most more complex errors. It is relatively expensive to compute in software, but easy to compute in hardware. See the discussion on page 36 for a comparison with the IP checksum algorithm.

Section 2.3 This is now completely obsolete.

Section 2.4 SLIP has now gone completely in favour of PPP, and the statements on page 25 line 6 and at the bottom of page 27 are now false. RFC 1812 requires that routers support PPP on all point-to-point links. Nearly all PPP implementations support header compression (the same algorithm as CSLIP), and RFC 1812 mandates it for links up to 19200 baud. Header compression is described in RFCs 2507–9. Proposed improvements are in RFCs 3095, 3096 and 3544.

p. 26 RFC 1548 has been obsoleted by RFC 1661, as updated by RFC 2153. Note that PPP is generally deployed these days over ADSL[33], where speeds are higher, and the considerations in section 2.10 less relevant (though not totally irrelevant).

p.31, l. 8 *It is not the MTU that we are reducing to 256 bytes, but the data length.* If the data length is to be 256, we have to quote an MTU of 296, since TCP, being ignorant of the compression kaing place in the lower layers, will subtract 40 bytes of fixed TCP and IP headers (see page 237) to compute its Maximum Segment Size. However, the calculations assume that the headers are compressed to five bytes, so use 261 as a packet length. *Similarly, the figure of 261ms quoted is based on a SLIP packet*

[33] ADSL stands for "Asymmetric digital subscriber line", and is the most common form of home broadband, at least in the U.K. — in some other countries, such as Canada, broadband over cable television predominates.

of 256 bytes (which would occur if we quoted an MTU of 291, since 291-40+5=256). The correct figure is 272ms, which halves to 136ms. The general conclusion is unaltered.

Header compression, while very valuable (note that a compressed 256+ header byte packet has 1.9% header overload, whereas an uncompressed 256+header byte packet has 13.5% overhead), is a complete, albeit localised, violation of the layering principle, since the CSLIP/PPP implementation has to look up the protocol stack to the IP and TCP layers to perform the compression. We should note that only TCP, which is connection-oriented, benefits from *this*[34] compression: NFS over UDP, which sends many packets to and from the file server, does not, since although there's a logical connection[35], there isn't one at the UDP level, since UDP is not connection-oriented.

p. 31 At today's more common 33,600 baud, the 272ms reduces to 78ms, which is a very acceptable delay. It would be reasonable to consider increasing the maximum data length from 256 bytes to 512 (the MTU from 296 to 552), which would increase the delay to 154ms, but make the overhead 0.9%. This is not a great gain: the greatest gain comes from reducing the number of packets required to carry a message, reducing CPU load and overhead elsewhere in the system: on a non-compressed link following the PPP link, the TCP and IP header overhead would drop from 14% to 7%.

In fact, the greatest gain would come in reliability due to the decrease in fragmentation (see the notes to page 151 and [14]). Consider an NFS `write` command, of 8192 bytes (the usual UNIX block size) plus, say, 60 bytes of UDP and NFS header information from a remote machine. With an MTU of 296, we can fit 276 bytes of UDP information into a fragment, so this takes $(8192 + 60)/276 = 30$ fragments, whereas with an MTU of 552, this takes $(8192+60)/532 = 16$ fragments. If we assume a 1% loss rate on the wider Internet, then an IP packet split into 30 fragments has a 26% chance of being lost (i.e. one fragment is lost, but this loses the whole IP packet), whereas with 16 fragments there is only a 15% chance of it being lost. If we assume a 5% fragment loss rate, then a 30-fragment packet has a loss probability of 79%, requiring on average 4.66 transmissions for success, whereas a 16-fragment packet has a 56% loss rate, requiring 2.27 transmissions on average. At a 10% fragment loss rate (not impossible in practice) the average number of transmissions is 23.59 for 30 fragments and 5.40 for 16.

In practice, we would probably increase the MTU to 576, since this is the default value for Internet MTUs, and may well prevent fragmentation of incoming TCP packets at the PPP interface. This would make the delay

[34] It would be possible to design a compression technique for NFS of course — whether it would be worth it is a different matter.

[35] Manifested by the shared understanding of file handles — see chapter 29

161ms, or 162ms if we allow for the fact that we're probably running PPP rather than SLIP, so should allow for 3 bytes of PPP framing.

Over a broadband connection, it would seem obvious to make the MTU 1500, but apparently it often ends up as 1492 (the RFC1042 MTU for 802.2 Ethernet).

Chapter 3

p. 34 A protocol value of 108 indicates an IP datagram whose payload has been compressed — see RFC 3173.

pp. 34–35 RFC 2780 confirms that the TOS field described here has been officially replaced by a 6-bit differentiated services codepoint (DSCP) field, described in RFC 2474[36], and then a two-bit field which is currently unused[37]. The DSCP field is intended to be used to select the "packet handling behaviour" (PHB) for a particular differentiated services domain. RFC 2474 also contradicts Stevens in the current use of the precedence field, which Stevens describes as "ignored today":

> In short, IP Precedence is widely deployed and widely used, if not in exactly the manner intended in [RFC791].

TOS fields of the form `xxx000xx` (i.e. with DSCP field `xxx000`) are to receive a PHB which is compatible with uses of the precedence field, in particular "common usage of IP Precedence values '110' and '111' for routing traffic." It is worth noting that the mechanism in RFC 2474 was first intended for IPv6 (see notes to p. 50), and has been retro-fitted to IPv4.

p. 36 In 1993, the US academic backbone of the Internet changed, with the phasing out of the old NSFNET. One consequence of this was that it now generally took 3–5 more hops to traverse the new backbone than it did the old. At the same time, there suddenly started to be many complaints about partial or total lack of connectivity between some sites/machines and others. This was finally tracked down to the fact that DEC's VMS, at the time a very popular machine, especially for academic computing services, was shipped with a hard-wired TTL of 32, and the perceived diameter of the Internet (i.e. the number of hops a packet had to travel) now exceeded this. RFC 1700 states that the default TTL should be 64.

pp. 36–37 The IP header checksum is easily calculated in software, since it can be done in 16 or 32-bit chunks, However, it does not prevent against re-arrangement of the (16-bit) words, or insertion/deletion of a word of

[36] The terminology of RFC 2474 has been updated by RFC 3260.

[37] Though RFC 2481, now obsoleted by RFC 3168, describes an experimental use for them to indicate congestion. Section 19 of RFC 2474 contains a useful history of the TOS field.

all 0's or all 1's. Contrast this with the Ethernet CRC described on page 22. However, the IP header checksum is valuable: [22] observes that "we saw a surprising number of IP datagrams with bad headers — it turns out that some LAN chipsets periodically erase 16 bits of the IP header (almost certainly due to a timing error on an internal 16-bit bus). These errors almost all get screened out by the first hop router." .

p. 44 l. 1 *It is not just a case of "how many" bits, but which.* There is no reason in RFC 950 why the sub-net mask need be next to the net mask (the University of Bath's network would work as well if the subnet mask were `ffff00ff`, with the Mathematics network being `138.38.x.96`, rather than `138.38.96.x`, and with the corresponding swaps in the numbers). However, the advent of CIDR (see p. 140) has meant that subnet masks tend to be next to the net masks, and RFC 1812 says that this SHOULD be the case.

While all sorts of tricks can be played, using contiguous bits next to the network's (A/B/C) implicit mask is strongly recommended — see RFC 1219.

p. 44 bottom With the advent of CIDR (p. 140), the distinction between "a host on a different subnet" and "a host on a different net" becomes essentially obsolete. Currently no computation of Class A/B/C should be done — we should check whether (source address & netmask) is equal to (destination address & netmask). If they are equal, the destination is on our subnet, otherwise it's not.

Figure 3.9 With the advent of CIDR (p. 140), the "net-directed" and "all-subnets" broadcasts are essentially obsolete. The only broadcasts are the limited one — "all on my cable" (strictly speaking ISO level 2 network), and the subnet-directed one — "all on my net/subnet". However, as Barry Margolin commented on 22 March 2001

> many routers have subnet-directed broadcasts disabled, because they're used more for SMURF attacks than any legitimate purposes.

RFC 2644, which is "Best Current Practice" amends RFC 1812 (Router Requirements) as follows.

> A router MAY originate Network Directed Broadcast packets. A router MAY have a configuration option to allow it to receive directed broadcast packets, however this option MUST be disabled by default, and thus the router MUST NOT receive Network Directed Broadcast packets unless specifically configured by the end user.

The reservation of -1 for "broadcast" means that the longest sensible netmask is 30 bits: `fffffffc`. In the special context of a point-to-point

link, where broadcasting is not supported, RFC 3021 says that a 31-bit netmask, `fffffffe` is permissible, with the two ends having Host numbers of 0 and 1.

p. 49 The state of allocation of IP numbers in 1999 was described on the Internet recently as "Class B addresses are like hen's teeth". RFC 2050 emphasises that addresses are no longer allocated at the A/B/C borders. RFC 3194 attempts to quantify "allocation efficiency" for addresses. The formula is $\frac{\log(allocated)}{\log(possible)}$, and several instances (e.g. the French move from 8-digit telephone numbers to 9-digits ones) of renumbering have taken place when this reaches 0.87.

p. 50 In fact, the replacement for the current IP (IP version 4, or IPv4), will be known (for historical reasons) as IPv6, and is not really any of the proposals here. It has 128-bit addresses: enough for $5 \cdot 10^{28}$ IP interfaces for every human being alive today. It also has much more variable length headers, whereas in IPv4 we are running out of IP header space (see source routing in the notes to section 8.5) and TCP header length (see the notes to p. 312). The reader should see RFC 2460[38] (which replaces 1883) and the references therein for the details, or [20] for a readable overview. RFC 2893 discusses some issues in the transition from IPv4 to IPv6. `http://www.6net.org` describes a European project for migration to IPv6, of which UKERNA[39], who manage JANET, are a partner. JANET's own efforts are described at `http://www.ja.net/development/ipv6`. RFC 2464 describes a method by which an Ethernet address can be converted into an IPv6 address: however this is not intended for general use.

Vint Cerf, chairman of ICANN and usually described as "an Internet pioneer" said on 2nd November 2007 that "The rate of consumption of available remaining IPv4 numbers appears to be on track to run out in 2010/11."[40] The article went on to say

> Although IPv6 was standardised 10 years ago it has not been rolled out at speed.
>
> While modern computers, servers, routers and other online devices are able to use IPv6, internet service providers have yet to implement the system.
>
> "The reason they haven't — which is quite understandable — is that customers haven't asked for it yet," said Mr Cerf, adding, "my job, whether with my Icann hat on or not, is to persuade them to ask for it. If you don't ask for it, then when you most want it you won't have it."

[38] As updated by RFC 5095, which deprecated the IPv6 version of Source Routing (Stevens section 8.5.

[39] Now (7 June 2007) called Janet (UK).

[40] `http://news.bbc.co.uk/1/hi/technology/7068140.stm`

Separately, JHD has learned (autumn 2007) that one major car manufacturer is planning to make all its cars IP-enabled, so as to be able to communicate with the factory. This will have to be IPv6 because "even `10.0.0.0` isn't large enough."

There is an interesting description of IPv6 take-up at `http://arstechnica.com/news.ars/post/20081113-google-more-macs-mean-higher-ipv6-usage-in-us.html`: France and the U.S. are among the top five countries (admittedly only 0.65% and 0.45% respectively) countries when it comes to percentage of traffic that is IPv6. The reasons are, apparently, that one French ISP "provides home routers that can easily provide IPv6 connectivity", whereas in the U.S. it is Apple's greater market share that accounts for the difference.

On 14 October 2009, it was announced[41] that ENISA would become "the first EU agency to begin offering services over [IPv6]" as part of the EU's IPv6 strategy[42]. This latter, at least, places a lot of the blame on ISPs: "There is evidence that less than half of ISPs offer some kind of IPv6 service". Of course, there are a few large ISPs, who tend to have such an offer, and many small ones who do not, so the comparison may not be fair. JHD would be more inclined to place the blame on content and service providers (also criticised by the EU).

There are various methods for interoperability between IPv6 and IPv4: a good description is at `http://arstechnica.com/articles/paedia/IPv6.ars/4`. Recent (October 2008) progress is described in `http://arstechnica.com/news.ars/post/20081006-ietf-working-on-making-ipv6-and-ipv4-talk-to-ea` `html`.

Chapter 4

p. 57 l. 4 *There is an unfortunate typographical error here: the ARP requests asks for the hardware (Ethernet) address corresponding to the stated protocol (IP) address.*

Chapter 5

JHD wrote in 2001: "RARP is still very much in use, and may even grow as more household appliances become IP devices". This now seems moot, as DHCP now seems to provide the functionality required, and it is probably fair to describe RARP as obsolescent, even if not yet obsolete.

For larger configurations, such as that which the University of Bath runs in its library, with the servers located in the Computing Services building at the

[41] `http://news.zdnet.co.uk/internet/0,1000000097,39807838,00.htm`.

[42] `http://ec.europa.eu/information_society/policy/ipv6/docs/european_day/comm-ipv6-final_en.pdf`

far end of the IP network, it has been replaced by BOOTP (see Chapter 16), or DHCP (RFCs 2131 and 3442).

Chapter 6

p. 71 Information request/reply have been superseded by BOOTP (Chapter 16) or DHCP as means of configuring discless machines. RFC 1700 (and the updating files) lists several other possible ICMP types, but these are not in widespread use.

p. 79 The statement in figure 6.10, that the error message should contain the first eight bytes of the IP data, has been obsoleted by RFC 1812. This says "The IP datagram SHOULD contain as much of the original datagram as possible without the length of the ICMP datagram exceeding 576 bytes" (576 is the minimum MTU that the (IPv4[43]) Internet should support without fragmentation). The reason given for this is "the use of IP-in-IP tunneling and other technologies". Note that RFC 1812 only applies to routers, not hosts, but many ICMP errors (probably most that would use the extra information) will be generated by routers rather than hosts.

RFC 4884 proposed a significant change to the format of ICMP packets to add more information after this "as much as possible". So far, this is only a "proposed standard", though it is very helpful when combined with STUN (page 76).

p. 82 RFC 1812 says that routers SHOULD NOT originate "Source Quench" errors (of course, they may forward them if originated at a host). The justification is that experiments show that generating "Source Quench" (which is ignored by UDP anyway, at least under Berkeley UNIX) actually consumes bandwidth and router resources, so is counter-productive.

p. 82 ICMP error 12 code 1 (required option missing) is used in the US DoD part of the Internet to indicate that a required security option is missing.

Chapter 7

p. 85 The statement in small print is now far more important than it was when the book was written. More and more sites are installing various kinds of firewalls, and ping to hosts is less and less useful. If one knows (or can discover) the route/firewall, ping to intermediate hosts can still be useful.

Chapter 8

p. 101, l. 10 "(42+58/960)" should be "(42+58)/960". [21]

[43] RFC 2460 states that the IPv6 minimum MTU is 1280 bytes. See also `http://www.psc.edu/~mathis/MTU/`.

Section 8.5 Source routing is less and less useful, for two reasons. The first is that the maximum number of slots available in the IP header, nine, is getting less with respect to the diameter of the Internet. The second is that, since source routing can be used to force packets to go via a router that may be more trusted than the normal route, it can be used as a basis of various attacks. Therefore many routers these days are configured to block such packets — Barry Margolin[44] writes: "My guess is that at least 25% of the Internet is inaccessible to source-routed packets".

However, Vernon Schryver writes[45]

> Note: the supposed security problems of source routing have been grossly exaggerated by ignorant trade rag espurts needing something to write about. They've done more harm than good.
>
> The few applications that still use the IP source address for authentication and authorization should use the setsockopt() to turn off any source route that arrived with the SYN. Applications that use real authentication and authorization don't care.
>
> The evils of IP source routes are similar to the evils of raw IP sockets in Windows XP that are going to lead to the end of the Internet realsoonnow. Both can be misused, but both are quite valuable (e.g. 'traceroute -g') and sane defenses against their misuses don't involve outlawing them.

Figure 8.7 The way to read this[46] is that the logical route is "things before the #" (we've already been there), #, real destination field, "things after the #". The IP address immediately after the # is "where to go next", and therefore belongs in the genuine IP destination field, and is put there, instead of taking up space in the options field. Hence we effectively 'gain' one entry.

Chapter 9

p. 117 Of course, an 'unreachable' error is only sent if errors are permissible: see p. 70.

p. 118 The reason that "host unreachable" is now generated instead of "network unreachable" is that it might only be a sub-net that was unreachable, but a host on a different network has no way of knowing the sub-net mask that defines the sub-net. Hence, if "'network unreachable" were generated, the recipient might erroneously conclude that the whole Class A/B/C network was unreachable, whereas in fact other sub-nets might still be reachable. This problem arises because sub-netting is a later addition to the IP suite. The problem gets worse with CIDR (p. 140), since

[44] Message `Jj%V9.36$f83.1107@paloalto-snr1.gtei.net` to comp.protocols.tcp-ip.
[45] Message `bO9i1m$dop$1@calcite.rhyolite.com` to comp.protocols.tcp-ip.
[46] I am grateful to the student who raised this question

there is now no way to determine remotely whether two hosts are on the same network.

p. 119 There are now several more "top-level routing domains": for example, the London router on SuperJANET 5 has to decide whether a packet is destined for:

1. elsewhere in JANET, and if so which other JANET node it should be forwarded to;
2. elsewhere in Britain (e.g. Freeserve), in which case it should be sent to LINX[47] — currently [12, p. 5] over a 40Gb link.;
3. elsewhere in Europe (including the Middle East and parts of Africa), in which case it is sent to GÉANT2[48], which may itself forward it to the Amsterdam Internet Exchange[49];
4. the rest of the world, in which case it is sent over the "fat pipes" to North America.

These routing decisions require knowing where every network is, or can be reached. More accurately, we need to know how every block of networks can be reached: for example an InterNIC or AP-NIC (see the second note of page 8) block of networks can all be aggregated into a single "super-net" and sent as in 4 above, without knowing where the end point is.

p. 123 Note the various checks that 4.4 BSD performs (and other systems should perform). As noted here, a malicious host could generate spurious redirects, this disrupting traffic or directing it via a subverted node. However, the second check 2 is somewhat misleading. One can check that the "indirect is from the current router", but an IP-level check is not very useful, since a host can insert a packet with a false source address[50]. A Level-2 check on the address will not work for PPP (where there are no Level-2 addresses) or in the presence of proxy ARPing (page 60 and the Appendix). Hence this check is not as strong as it looks.

p. 123 Note that sub-netting is an addition to IP after ICMP (in particular redirects) was defined, and hence there is no provision for sending sub-net masks with a redirect. This explains the notes at the end of section 9.5 about having to send host redirects rather than network redirects.

[47] LINX = London INternet eXchange, routing packets between the Internet networks in the U.K., apparently averaging 134Gbs. `http://www.linx.net`

[48] EBONE (mentioned in the text) → TEN155 → DANTE → GÉANT → GÉANT2, where the core is connected by dark fibre (i.e. GÉANT2 lease the fibre "dark", and then illuminate it themselves at whatever frequencies they choose) running multiple 10Gbps links: see `http://www.geant2.net`. Notice also the curious fact in GÉANT2 Bulgaria is connected to Hungary, and not to its neighbour Romania.

[49] AMS-IX: predicted (`http://www.bcs.org/server.php?show=conWebDoc.9665`) to carry 1EB in 2007

[50] This cannot be detected in general, since most packets arrive with a level 2 address of some router, which is different from the Level 3 address of the true origin.

p. 125 Router discovery is not as new as it was in 1994, and more hosts and routers now support it. However, we should note that every router on a sub-net has to support it before it becomes truly useful, so its use is not as wide-spread as could be hoped for.

It should be noted that there is no security on router discovery, and this weakness has been exploited. See `http://www.L0pht.com/advisories/rdp.txt` for details. Firewalls should certainly block these packets.

Since DHCP (see Chaper 16 and notes thereon) tells the client who the routers are, it could be argued that, for most practical purposes, router discovery is obsolete.

Chapter 10

Some additional net references for routing can be found at `http://www.itprc.com/routing.htm` and `http://www.ietf.org/html.charters/idr-charter.html`. There is an excellent graphical tool for showing routes (from Warsaw!) at `http://visualroute.ipartners.pl:81`, and a different one at `http://visualroute.visualware.co.uk`.

Many texts describe BGP as a "link state" protocol because it knows about the state of links. This is *incorrect*, and the description by Stevens (p. 139, l. −8) is correct. A good summary of the differences is in `www.cl.cam.ac.uk/~tgg22/talks/SRCCS.2005.griffin.ppt` at slide 4. One can also admire this brief summary of BGP [23].

> BGP is a path vector protocol. Each BGP advertisement usually includes the sequence of ASes for the path, along with other attributes such as the next-hop IP address. Before accepting an advertisement, the receiving router checks for the presence of its own AS number in the AS path to discard routes causing loops. By representing the path at the AS level, BGP hides the details of the topology and routing information inside each network.

The term "path vector protocol" seems the most appropriate.

Routing spits into intra- and inter-AS routing (AS = 'autonomous system' — see page 139 and these notes on it). A good description of the difference is the following.

> Note that the primary difference between intra-AS and inter-AS routing is that intra-AS routing is usually optimized in accordance with the required technical demands, while inter-AS usually reflects political and business relationships between the networks and companies involved. [10, p. 101].

p. 127 [This text was written in about 2001, and was true at the time] The widespread use of ATM as the link-layer protocol in WANs has meant, oddly enough, that IP routing *within* a WAN is less important than it

used to be. Consider, for example, the SuperJanet IV core network (see `http://www.superjanet4.net/rollout/outline.pdf` for a diagram). The underlying link-layer protocol between the eight Core PoP routers is ATM, so that, as seen from IP's point of view, all eight are directly connected on the same network, and no IP-layer routing is necessary between them. If one of the fibres goes down, so that two sites are no longer directly connected at the fibre level, then it is up to ATM's routing to redirect the ATM cells, so that the IP layer still sees direct connectivity between all the sites.

[2008 text] SuperJanet 5 uses optical ethernet links between the core routers, and between the core routers and regional networks (see the diagrams on slides 40–41 of `http://www.webarchive.ja.net/services/events/networkshop/Networkshop34/presentations/plenary04/RollyTriceV2.pdf`), so IP routing (again) takes place between them. Similar configurations now seem to operate on other major networks, so IP routing within a WAN is back at the level Stevens describes it.

p. 128 OSPF is now definitely more popular than RIP for new networks, for the reasons outlined later. RFC 1812 (updating the small print statement on the page) says that if a router supports any IGP, then it MUST support OSPF[51]. However, it it worth noting (RFC 4822) "There are a number of large-scale RIP deployments today[52] that successfully use manual configuration of RIPv2 Security Associations". Icarus Sparry writes as follows.

> Given the amount of code needed to implement OSPF, I am very willing to believe that there are still devices which only implement RIP. For Marconi we did implement both, but RIP I did in a day, whilst it took a team of 3 several weeks to port the implementation of OSPF we got from FORE.
>
> You might look at `http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/netsh_routing.mspx?mfr=true` and see that OSPF is not available for 64bit windows!

BGP has also replaced EGP in most situations, and again is mandated by RFC 1812 if any Exterior Gateway Protocol (EGP per se, RIP etc.) is supported. Having said which, BGP is a very powerful, hence complicated, protocol ([10] is a 500-page book on BGP). At 5/11/2007 the University of

[51] But it does not necessarily have to participate in it? This is a somewhat curious requirement.

[52] One of the authors of RFC 4822 writes [1]

> Mostly the largish RIPv2 sites that I know about are businesses with a handful of locations (maybe 1 to 5), usually with some sort of IPsec VPN connecting the sites together. ... Usually their internal network topology is quite stable (so convergence times just aren't an issue).

Bath used static routing to connect its (private) AS to SWERN, while the universities of Bristol and the West of England, which have private interconnects as well as to SWERN, used OSPF between them and SWERN. SWERN has since installed more redundancy, and is now a fully-OSPF network. SWERN uses BGP between itself and JANET and WREN, and indeed JANET mandates BGP for all its MAN links.

p. 132 Although the RIP RFC (1058[53]) was written three years after the definition of sub-netting (RFC 950), the RIP RFC was a piece of retrospective documentation, and the RIP code was generally written before sub-netting.

p. 132 The second problem mentioned here is slow convergence. This is often due to a phenomenon known as "counting to infinity" which we illustrate in the context of the diagram on page 131. The initial stable state is as follows (where the entries show "first-hop" routes and metrics).

R1→N2	R1→N3	R2→N1	R2→N2	R2→N3
(direct) 1	(via R2) 2	(via R1) 2	(direct) 1	(direct) 1

R2 is sending RIP messages to R1 indicating that N3 is one hop from it, and therefore R1 deduces that N3 is two hops from it. It therefore broadcasts this fact, but R2 ignores this, since that route would make N3 three hops from it, whereas it is only one.

Now suppose that the interface from R2 to N3 fails. The situation is then as follows.

R1→N2	R1→N3	R2→N1	R2→N2	R2→N3
(direct) 1	(via R2) 2	(via R1) 2	(direct) 1	(unknown) 16

R2 now does not know how to reach N3 directly, so the next broadcast from R1 causes it to accept that message (after all, R1 might have another route to N3, say via a router connecting N1 and N3). This makes the situation the following.

R1→N2	R1→N3	R2→N1	R2→N2	R2→N3
(direct) 1	(via R2) 2	(via R1) 2	(direct) 1	(via R1) 3

R2's next broadcast will advertise a distance of three to N3, which will cause R1 to believe that it is now four from it. This makes the situation the following.

R1→N2	R1→N3	R2→N1	R2→N2	R2→N3
(direct) 1	(via R2) 4	(via R1) 2	(direct) 1	(via R1) 3

The next broadcast from R1 will make R2 now believe that it is five hops from N3. R1 will later believe that it is six hops, and so on.

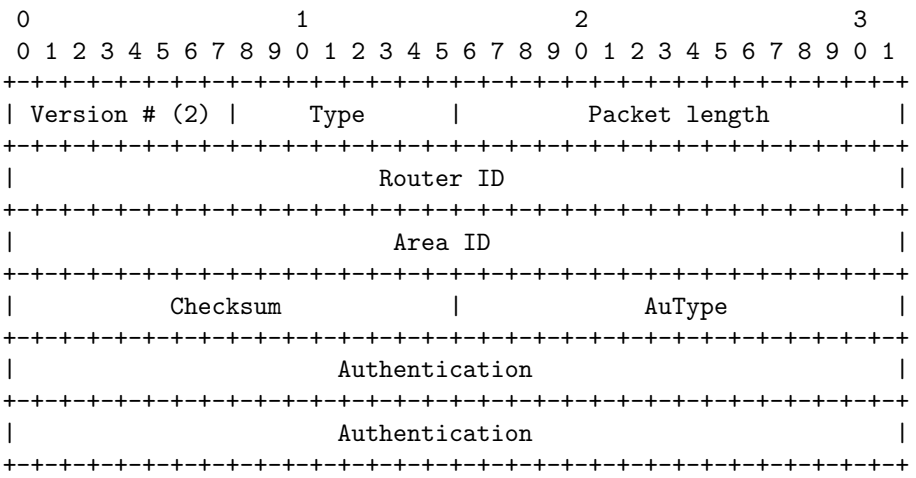
The problem is that R2 does not know that the route R1 is advertising is in fact through R2 (since it could be via some other router). Since "infinity"=16 in RIP, this means that the process takes, on average, four

[53] Updated by RFCs 1388, 1723, 2453 and 4822.

minutes to converge (eight RIP updates from R1 to R2), assuming that no packet is lost. Since packets are bouncing around between R1 and R2, N2 is likely to be overloaded, so packet loss is indeed possible, slowing down the convergence.

p. 137 The current version of OSPF is described in RFC 2328.

p. 138 All OSPF packets begin with a standard header, as defined below.



The various types are as follows.

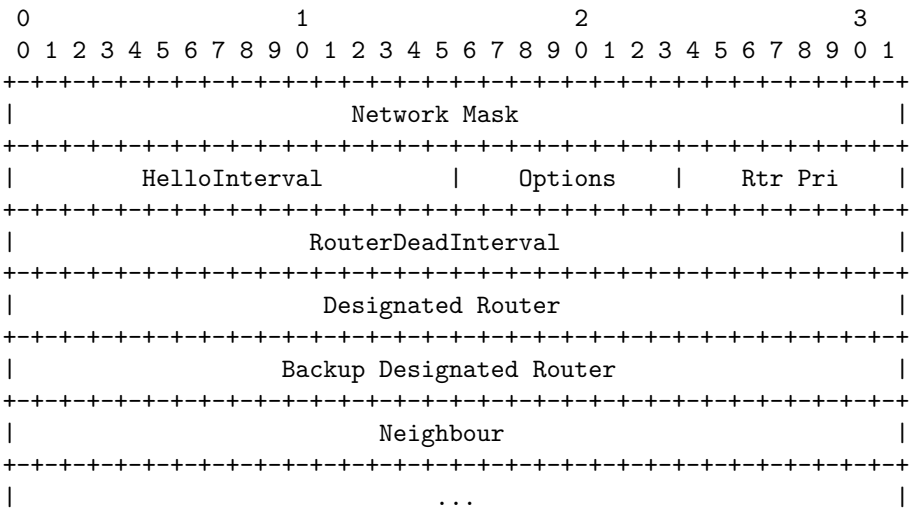
1. Hello — these packets are sent to immediate neighbour routers, to "establish and maintain neighbor relationships"[54].
2. Database Description.
3. Link State Request.
4. Link State Update.
5. Link State Acknowledgment.

Unusually, the checksum, though computed the usual IP way, does *not* include the 64 bits of authentication: this is because authentication is done after checksumming. Various kinds of authentication[55] are possible with OSPF, and the AuType field says which is being used for this packet.

The 'Hello' packet has the following format (after the standard OSPF header).

[54] RFC 2328, p. 193.

[55] The currently defined ones are 'null', 'plain-text password' (which protects against a machine inadvertently joining an OSPF set-up), and 'cryptographic authentication', which uses a password and the MD5 message digest to verify the authenticity of the packet. With cryptographic authentication, the checksum field is not used, since MD5 provides a far more powerful way of detecting corruption.



For the format of the other packets, see RFC2328.

p. 139 BGP version 4 was described in RFC 1771, with a good rationale in the companion RFC 1772. The current definition (Jan 2006) and accompanying material are RFC 4271–4277. The introduction of CIDR has meant that the definition of an AS has changed[56]: RFC 1771 says the following.

> The classic definition of an Autonomous System is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASes. Since this classic definition was developed, it has become common for a single AS to use several interior gateway protocols and sometimes several metrics within an AS. The use of the term Autonomous System here stresses the fact that, even when multiple IGPs and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what destinations are reachable through it.

A further complication is that there are many autonomous systems that ought not to be visible much beyond their boundary. Consider the University of Bath, which is an AS hung off SWERN. This is its only point of access, so it is a stub AS. From the point of view of everyone outside SWERN, it might as well be part of SWERN's AS, since the only route

[56] See RFC 1930 for an explanation. This RFC is not without wishful thinking however: at the start it says the following.

> IDRP (The OSI Inter-Domain Routing Protocol, which the Internet is expected to adopt when BGP becomes obsolete).

There is no sign of this happening.

to it is via SWERN. RFC 3065 introduces the concept of an 'AS confederation', defined as "A collection of autonomous systems advertised as a single AS number to BGP speakers that are not members of the confederation". In this sense, there is an AS confederation including the SWERN core, Bath, Bristol, UWE etc., which can be regarded as a single AS by everyone outside it.

RFC 3065 has now been replaced by RFC 5065. An improvement here is that previously all BGP routers within an AS had to be fully interconnected. Hence n BGP routers required $n(n-1)/2$ BGP-over-TCP connections. To avoid this, a large AS would be split, but this has its own problems, for the it whole Internet.

> Unfortunately, subdividing an autonomous system may increase the complexity of routing policy based on `AS_PATH` information for all members of the Internet. Additionally, this division increases the maintenance overhead of coordinating external peering when the internal topology of this collection of autonomous systems is modified. [RFC 5065, p. 3]

Indeed the whole of SuperJanet IV and all connected MANs could be regarded from outside as a (multi-homed: there are several external links - see `http://www.ja.net/about/topology/workingwiththeworld.pdf`) AS confederation. In fact[57], the University of Bath uses a 'private' AS number, not visible beyond SWERN. SWERN equally uses a 'private' AS number, not visible beyond JANET. This is true for many of the MANs hung off JANET, though some, that have commercial connections, have public AS numbers.

p. 139 The Bath campus is a typical example of a stub AS. It does support more than one network: as well as `138.38`, there are networks for UKOLN, ingenta and EduServ. There are also private networks for the Computer Science Department, wireless, docking points and for ResNet (`172.28`).

AS are identified by a 16-bit number. This has proved inadequate as the Internet has grown (recall that there are 2^{14} possible Class B networks, and many sites only have one, or a few, Class C ones). There has been a partial solution by allocating 65412–65535 as a range of 'private' AS numbers (like 'private' IP addresses), and Bath, for example, has a private AS number, since it is agglomerated into SWERN for external consumption. This is only a partial solution, and it is proposed to move to 4-btye AS numbers (RFC 4893), which will take place compulsorily (for new allocations) from January 2010.

> I am following the uptake of 32-bit AS numbers with great interest. If the industry fails to clear this relatively easy hurdle,

[57] Personal Communication Neil Francis (BUCS) 5.11.2007.

> then the prospects for us making the much larger jump to IPv6 in a timely manner do not look good. [13]

Anyone actually running BGP should look at RFC 5123 (and its errata), since it is possible to receive incorrect, or unauthorised, routes via BGP, and BGP routers "should not believe everything they hear". It should be noted that this *does* happen in practice: in February 2008 Pakistan Telecom cut much of the world off from Youtube. See the Youtube[58] video at `http://www.youtube.com/watch?v=IzLPKuAOe50`, or the technical description at `http://arstechnica.com/news.ars/post/20080225-insecure-routing-redirects-youtube-to-pakistan.html`.

p.140, l. -10 *The number 65,536 mentioned here should be 131,072 — computed as* $2^{32}/(2^7 \times 2^8)$*, where the* 2^7 *comes from the fact that the networks share the top 7 bits (as Stevens says) and* 2^8 *from the fact that we are counting class C networks, rather than addresses.*

p. 141 It was reported at the end of 1999 that some routers were close to having 100,000 entries in their route table. This, while large (and requiring better algorithms than sequential search in the routing mechanism!), is still less than the 2,000,000+ that classic IP without CIDR would imply. There is a graph of routing table growth up to 2000 in [10, p. 68].

10.9 RFC 1466 was replaced by RFC 2050, but this change basically updated the rules on how registries worked. Many more address blocks, often 'reclaimed' class A, have since been allocated to regional registrues to sub-allocate.

Chapter 11

p. 148 The author says "When an IP datagram is fragmented, it is not reassembled until it reaches its final destination." These days, some firewalls will insist on reassembling the packet before deciding whether or not to forward it. See also Appendix B, which explains why.

pp. 148–151 Fragmentation can pose a security problem. There was a bug in Windows NT which would crash it if incompatible (i.e. overlapping) fragments were received. This was used to attack NT machines in the Pentagon when Gates was addressing Congress. Generally, a firewall has little option but to pass a fragment (other than the first, i.e. the one with fragmentation offset zero), since there is no protocol-related information in later fragments. If the first fragment has been dropped, then the subsequent fragments should time out, but the firewall may wish to block the resulting ICMP error, on the grounds that it conveys information that should not be revealed.

[58] Recursion: see recursion.

RFC 815 describes IP fragment re-assembly algorithms. Since the fragments have to be stored in the memory of the IP layer until the packet is complete, there are denial-of-service attacks that flood the target with fragments until the memory is exhausted.

p. 153 The author says "Although most systems do not support the path MTU discovery feature ...". These days, most TCPs do support it. RFC 1191, which recommends it, is a "draft standard"[59].

p. 159 NFS systems that display the ARP timeout bug listed here often have the feature that, after a pause, e.g. lunch, one is greeted on resuming use by a sequence of

```
NFS server XXX not responding: still trying
NFS server XXX OK
```

errors, as the ARP cache is repopulated.

Chapter 12

p. 169, l.3 *Broadcasting and multicasting can apply to protocols other than UDP (though not to TCP). Examples are OSPF (multicast); ICMP (broadcast for router discovery) and IGMP (p. 179).*

pp. 171–2 The "net-directed" and "all-subnets-directed" broadcast are essentially obsolete since CIDR (p. 140).

p. 176 One could ask what the rôle of `224.0.0.1` is — surely it duplicates the "limited broadcast" address `255.255.255.255`? In theory that is true, but in practice `224.0.0.1` means, not Stevens' "all systems on this subnet", but rather "all multicast-capable systems on this subnet". Many systems, e.g. printers, are sold without multicasting, since there is no need for it, and, as we have seen, it complicates the Ethernet interface, device driver, and the IP layer.

Chapter 13

p. 179 RFC 1112, describing IGMP, was updated by RFC 2236 (IGMP v2), itself obsoleted by RFC 3376 (IGMP v3). IPv6 uses Multicast Listener Discovery (MLD) in a similar way. MLD version 1 (RFC 2710) implements the functionality of IGMP version 2; MLD version 2 implements the functionality of IGMP version 3, and RFC 5186 discusses the interaction between these and multicast routing. These enhancements are intended to be upwards-compatible with the original IGMP, which is still the official standard. There are more modifications in RFC 4604, to do with "source-specific" multicasting.

[59] See also RFCs 1453 and 2923.

Multicast routing, which Stevens does not really discuss, is covered in RFC 5110.

It seems fair to say that multicast has *not* been the success that it should have been. While the University of Bath's campus TV is apparently multicast-capable, and the BBC runs a multicast experiment, most commercial providers do not use multicast. This seems to be connected with the fact that many ISPs don't support it: "there's no demand for it". The UK's "Access Grid"[60], widely used among UK Universities, uses multicasting over JANET, and JANET and GEANT (its european equivalent) have well-supported multicasting infrastructure.

Chapter 14

p. 188 RFCs 1034 and 1035 have been updated many times, with the most important general update being RFC 2181. RFCs 4033, 4034 and 4035 contain security extensions to the DNS, with a potentially important extension in RFC 4470 — there's a disucssion of the security extension at the end of these notes to chapter 14. RFC 4343 clarifies the meaning of the term "The DNS is case-insensitive", and introduces an escape mechanism.

Stevens wrote "The most commonly used implementation of the DNS, both resolver and name server, is called BIND". This is still true on Unix systems. Windows systems tend to structure the resolver differently. While most domains will run BIND, performance limitations have become apparent at the very high end[61] of the market. "BIND tends to choke on domains with more than 10^7 entries". At this end of the market, the simple 'zone transfer' mechanism that Stevens described ceases to scale, and the full might of replicated database technology needs to be used.

p. 189 Stevens says that `arpa` is used for address-to-name mappings, but it has recently (RFC 3172) been re-categorised as the "infrastructural domain" for the Internet, which means that it could be used for more. `ip6.arpa` and `e164.arpa` are in use, as well as the common `in-addr.arpa` that Stevens describes.

Just as IPv4 uses 'A' records to store 32-bit IPv4 addresses, so IPv6 uses 'AAAA' records (RFC 2874) to store 128-bit IPv6 addresses. Note that there is no concept of DNSv4 and DNSv6: one DNS tree can store information about both (and indeed much information, such as 'MX' and 'NS' would be the same, except thay they must return AAAA records as well as A records). RFC 2782 introduced 'SRV' records, for service delivery, so that `_www._tcp.bath.ac.uk` would find a web server for the University of Bath. So far, they haven't really caught on, but watch this space.

[60] `http://www.ja.net/services/video/agsc/AGSCHome/whatisaccessgrid.html`.
[61] `http://icannwiki.org/Paul_Kane` quotes $12 \cdot 10^9$ queries/day.

The domain `ip6.arpa` is used for IPv6 PTR queries in much the same way as `in-addr.arpa` is for IPv4. However, instead of being split by octets encoded in decimal (`34.32.138.38.in-addr.arpa`), the addresses are split by nibbles (4-bit chunks) encoded in hexadecimal, as in

`b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA`

(example from RFC 3596).

Telephone numbers can also be stored: the International Telecommunications Union (ITU) has standard E-164 (and the domain `e164.arpa`), so that JHD's office telephone number (international format: +44–1225–386181) would correspond to `1.8.1.6.8.3.5.2.2.1.4.4.e164.arpa`. Hence the university of Bath could ask for delegation of `8.3.5.2.2.1.4.4.e164.arpa`, just as it manages telephone numbers of the form 01225–38xxxx.

p. 189 The country codes are listed in ISO 3166 (one of the more frequently changing ISO standards at the moment!). An on-line version, listing 246[62] country codes as of November 2008, can be found at `http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html`. The European Union has acquired the country domain `.eu`, as well as `.eu.int`. There is a difference, though: `.eu.int` is for the organisations of the Union itself, whereas `.eu` is for any entity in the Union. Similarly[63], `.asia` has been opened up for use (or exploitation[64], if you prefer!). There are other geographical, or cultural, domain, such as `.cat` for "the Catalan linguistic and cultural community"[65]. For a definitive description (assuming you read Catalan!) see `http://www.domini.cat`, which claims nearly 30,000 registrations.

Some two-letter domains are open to what might be described as punning. Tuvalu[66] has long profited from the sale of `.tv` domains — see the Wikipedia article. More recently[67] Montenegro has been profiting from the sale of `.me` domains.

It can be observed that the abbreviations are very "anglo-saxon": for example Finland is `.fi`, whereas `.su` (Suomi) would be more in keeping with the country's own name. Similarly `.gr` for Greece ($\epsilon\lambda\lambda\alpha\sigma$). However, no rule is invariable, and we note `.es` for Spain (Espana). Even more colonial: Morocco is `.ma`, presumably from the French "Maroc".

[62] Up from 239 in December 1999, and 244 in November 2006. One of the most recent was the Åland islands (`.ax`). There have also been changes in French possessions, with `.bl` for Saint-Barthélemy and `.mf` for Saint-Martin.

[63] The Times, Monday 8 October 2007, p. 49.

[64] "In the first two days that `.eu` became available, EUrid, the registry behind the scheme, received 227 applications for `sex.eu`". (*loc. cit.*)

[65] `http://en.wikipedia.org/wiki/.cat`

[66] Formerly the Ellice islands, the world's fourth-smallest country with a population of about 11,800.

[67] `http://technology.timesonline.co.uk/tol/news/tech_and_web/article4385223.ece`, 24 July 2008

p. 189 Several sites outside the U.S.A. use `.com`, `.edu` (largely in Canada, but the University of Bath has registered `bath.edu` for example) and `.net`: two examples are the ISP `.freeuk.com` and JANET itself `.ja.net`[68]. It used to be rare for organisations wholly outside North America to use `.org`, but there are some[69] — the most curious is the British Council offices in Portugal, which live at `www.britishcouncilpt.org`.

In 2001, two new "generic" top-level domains were authorised: `.info`[70] and `.biz`. For more information about these and other new top-level domains (e.g. `.name`, `.museum`, `.coop`[71], `.aero`), see `http://domains.dan.info/structure/new-tlds.html`. There is an interesting question as to how "authoritative" these domain names (or indeed the original ones) are.

> Mr Galvin explained that for a ".org" domain name, a group must in theory be a non-commercial organisation, but in practice "people will go and get a '.org' address if their choice of a '.com' or '.co.uk' isn't available".[72]

Equally, one could doubt whether there is a commercial company behind `www.cynicalbastards.com`.

On the other hand, JHD recalls that, to get the `bath.edu` name, he had to provide a facsimile of the University's Royal Charter, and then engage in some explanation of what this was.

In 2008, ICANN proposed total opening-up of the Top Level Domains[73], but this has yet to take effect.

It may be wondered how popular the 'newer' domains are. JHD analysed his outgoing mail from 2008 (up to 10.11.2008), and found the distribution of correspondents *by e-mail address* shown in table 2. The `.net` ones ranged from an individual (`chrismitchell.net`) to serious network providers. We note no 'new' domains here, and indeed the only time JHD consciously recalls going to a web-site from a 'new' domain was a `.cat` one.

Some other countries (e.g. Austria — `.at`) have a system like the U.K., with `.ac.at` being universities. Others (e.g. Germany — `.de`) do not, and

[68] Not `.janet.net`, which seems to have been taken by a "cybersquatter".

[69] `cec.org` is not the Commission of the European Community (this lives in `eu.int`), `commonwealth.org` is not the British Commonwealth, `eea.org` is the Edmonton Executives Association, not the European Economic Area, and `www.nato.org` is curious. The humorous might care to note that `3.1415926.org` has joined the Internet, and devotees of Cockney rhyming slang might be amused by `www.tasteandsmile.co.uk`. Devotees of long names should study `http://www.llanfairpwllgwyngyllgogerychwyrndrobwllllantysiliogogogoch.com/`

[70] For example `www.health-informatics.info`.

[71] But the MIT bookstore, known as the "Coop", though it is managed by Barnes & Noble, is at `mit.bkstore.com`.

[72] `http://www.bcs.org/server.php?show=conWebDoc.15949`.

[73] See `http://www.bcs.org/server.php?show=ConWebDoc.20927`.

Table 2: JHD's correspondents (by e-mail address)

Domain	addresses	domains
bath.ac.uk	477	1
cs.bath.ac.uk	58	1
Other bath.ac.uk	72	6
Bath total	607	8
Other .ac.uk	89	38
nag.co.uk	9	1
Other .co.uk	17	14
Various .org.uk	10	5
Other .uk	2	2
.edu	6	5
gmail.com	20	2
hotmail.com	13	1
Other .com	43	35
Total .com	76	38
.org	20	14
.net	5	5
.gov	1	1
.int	1	1
.ca	11	6
.de	7	6
.eu	4	2
Other two-letter	37	29
(15 different top-levels		

reliance has to be placed on the format of the next name, e.g. `uni-paderborn.de` means the University of Paderborn. There are also "reverse subdomains", as in `.uk.com`.

p. 190 RFC 2812 describes "best current practice" in the DNS. In particular, it recommends that at least one secondary server be on a different international backbone from the primary server: easier said than done: the University used to have one in the U.S. for many years, but the company providing it (for free under a mutual backscratching arrangement, as is usual) changed hands again, and JHD's contact moved on.

p. 191 Most clients in fact use the combination of the IP address of the server and the identification field to match responses to requests (a 16-bit field by itself is sufficient for an individual resolver, but not for a server that is asking many other servers for answers). It is therefore important (see RFC 2181; section 4) that a multi-homed server uses the same IP address for the reply as the IP address in the request. Similarly, the same port number must be used in the reply as in the request.

pp. 191–2 Stevens does not describe the "additional" records. It is perfectly legitimate, though unhelpful, for a DNS server never to return additional records, and for a client to ignore them if provided. Additional records are used to supply extra information that was not asked for in the query, but which the DNS server thinks the querier might wish to know. Typically these are 'A' records corresponding to names in the answer (or authority) records. The reasons for additional records are two-fold:

- from the point of view of the server, they reduce the load, since it is much cheaper to tack on 'A' records than to reply to a second query;
- from the point of view of the client, the round-trip time is reduced, since only one query need be made.

An analogy would be when some-one comes into my office (1West 2.2) and asks where is 1West 2.3. The answer is "straight through the wall", but I normally also give directions on how to get there.

pp. 193–4; Figs 14.5, 14.8 These are somewhat misleading, since they imply that the query type and query class are properly aligned on 16-bit boundaries, and that (in 14.8) the time-to-live is aligned on a 32-bit boundary. In fact, no padding of any of the domain names (which are just sequences of bytes) is done, so these can start anywhere. This may necessitate some fancy 'C' code to unpick the data structures corresponding to these packet formats, byte-by-byte.

p. 194 TTLs of 0 seem to cause problems with some name servers, and 1 is probably the best value to use for "as little caching as possible". Although Stevens says that 2 days is a common TTL for RRs, it is quite common these days to see 7, or even 14, days.

p. 195 There is an interesting interaction between web browsers and the resolver. Why can't I just write `http://www` and get `www.bath.ac.uk`? Conversely, why does `www.bath.ac.uk` *without* a trailing dot work? The answer is that the browsers, before calling the resolver, automatically put a dot on, since the Web is defined purely in terms of FQDNs (the usual acronym for fully-qualified domain names).

More modern versions of BIND support an additioanl directive (in fact many more), so that `/etc/resolv.conf` on my machines looks like the following (where `water.cs.bath.ac.uk` is `138.38.108.4`).

```
domain cs.bath.ac.uk
search cs.bath.ac.uk bath.ac.uk ac.uk
nameserver 138.38.108.4
nameserver 138.38.108.3
```

The `search` directive takes over one rôle from the `domain` directive Stevens discusses. It means that non-FQDNs should be looked up in `cs.bath.ac.uk`, then `bath.ac.uk`, then `ac.uk`. Hence an attempt to `ping obol` on our machine `melete` produces the following `tcpdump` output (dates and times deleted, and `bath.ac.uk` shortened to `bath`).

```
IP melete.cs.bath.48363 > water.cs.bath.domain:  8464+ A? obol.cs.bath. (36)
IP water.cs.bath.domain > melete.cs.bath.48363:  8464 NXDomain* 0/1/0 (89)
IP melete.cs.bath.48363 > water.cs.bath.domain:  63374+ A? obol.bath. (33)
IP water.cs.bath.domain > melete.cs.bath.48363:  63374 1/3/3 A obol.bath (174)
IP melete.cs.bath > obol.bath: icmp 64: echo request seq 1
IP obol.bath > melete.cs.bath: icmp 64: echo reply seq 1
```

pp. 196–7 In fact, authority need not be delegated on network IDs: since if RIPE (for example) has two Class A sized blocks of Class C addresses which it manages (`194.x.y.z` and `195.x.y.z`), it would make sense for `194.in-addr.arpa` and `195.in-addr.arpa` to be delegated to RIPE by the NIC. It might then sub-delegate Class 'B' sized blocks to individual countries, and so on.

However, delegation *does* have to take place on byte boundaries. So someone who obtains a "super-C" of 16 class C addresses under CIDR, say `195.1.16.x` to `195.1.31.x`, still needs 16 delegations, e.g. `16.1.195.in-addr.arpa` from the owner of the network `195.1`, i.e. the DNS subtree `1.195.in-addr.arpa`. In practice, this is not much work once the delegation has been obtained, but is easily over-looked. Conversely, someone with a "sub-C" allocation has to have delegations for each IP address.

Put bluntly, `in-addr.arpa` and sub-netting/CIDR do not go smoothly together. See RFC 2050 for a description of how the process is managed. RFC 2317 provides a technique for circumventing the problem that the owners of non-octet boundary subnets cannot manage their `in-addr.arpa`

space: essentially we create a `CNAME` record for each machine (although Stevens does not show this, there is no reason why we can't have four levels below `in-addr.arpa`), pointing to a new object in this domain (one per subnet, and describing how the subnet is laid out, e.g. `0/26.x.y.z.in-addr.arpa` for a subnet holding addresses 0 . . . 63 of `z.y.x.0`), which has an NS record pointing to servers of the owners of the relevant subnet. This requires no modification to the DNS lookup mechanism[74]. Since a `CNAME` cannot point to a `CNAME`, though, we cannot apply the trick recursively, i.e. a owner of one of these subnets cannot sub-allocate using this trick.

p. 197 If we analyse the performance of compression on the response in figure 14.3 (p. 203; details shown in p.204), we see that the compressed format would have 213 bytes[75], whereas the uncompressed version (still legal — a server does not need to compress) would be 332 bytes[76].

p. 201 There are many more kinds of DNS records than are discussed here: 47 in the latest on-line update to the Assigned Numbers RFC. RFC 2535[77]. RFC 2535 has now (January 2006) been replaced by RFCs 4033, 4034 and 4035. RFC 2535 introduced KEY records for storing public key encryption parameters, SIG records for signing the data held in a DNS zone, and NXT records to authenticate the *non*-existence of a record in a domain.

p. 201 CNAME records are very heavily used for WWW servers: it has become instinctive to type `http://www.`anything. Hence a company may start out with one machine on the Internet (`startup.co.uk`), with a CNAME of `www.startup.co.uk`, and grow that to a dedicated machine (possibly called `www.startup.co.uk`, but preferably called `www1.startup.co.uk`, with a back-up machine called `www2.startup.co.uk`), and then to a series of web servers, using "round-robin" in BIND to share the load between them.

p. 203 The discussion in the first paragraph is largely correct for Unix achines, which tend to run a name server on them. For windows machines, which tend not to run name servers, and where the resolver is shared between different applications, the resolver tends to have a cache.

p. 207 Stevens states that "caching can reduce the number of packets exchanged [in Figure 14.16]". In fact, if there were no caching, the exchange in 6/7 would be repeated before 10/11. The diagram also assumes that

[74] In theory. In practice some DNS resolver implementations are unhappy with the character `/`, even though it is legal in the DNS (RFC 2181), and one should use a different character, e.g. `-`.

[75] 12 for the header, 23, 20, 34, 24 and 20 for the authority RRs, and 16 each for the additional RRs, since the names here have all occurred previously.

[76] 12 for the header, 29, 32, 40, 40 and 29 for the authority RRs, and 25, 28, 36, 36, and 25 for the additional RRs.

[77] Note that the RSA/MD5 scheme in RFC 2537 (Algorithm code from RFC 2535 =1) has been obsoleted by the RSA/SHA-1 scheme in RFC 3110 (Algorithm code from RFC 2535 =5).

the root servers know the client's/server's name server address directly, which is unlikely in practice, since the root servers generally do not practise recursion. Caching will certainly help here, though, as name-severs for `uk.` are likely to cache most of `ac.uk.`, and research-active universities elsewhere in the world are likely to know a server for `ac.uk.` from their cache, rather than having to go via a root server and `uk.`. Equally, some root servers know about `org.` directly, and some serve `.arpa`, or even `.inaddr.arpa` directly.

root servers On 28.5.2002, the root name servers seemed to be distributed as follows:

`a.root-servers.net` seems to be at Network Solutions Inc. (who run the root zone) in the States.

`b.root-servers.net` is at isi.edu, which runs IANA and RFC-editor.

`c.root-servers.net` is in PSI.net (? near Chicago).

`d.root-servers.net` appears to be at the University of Maryland.

`e.root-servers.net` appears to be in Qwest.

`f.root-servers.net` appears to be at ISC.org - Internet Software Consortium(producers of BIND, the main DNS for UNIX).

`g.root-servers.net` also appears to be in Qwest, or parts of MILNET beyond Qwest.

`h.root-servers.net` appears to be in MILNET.

`i.root-servers.net` is in Sweden (probably originally connected with Craig Partridge, now operated by Netnod.se).

`j.root-servers.net` seems to be at Network Solutions in the States.

`k.root-servers.net` seems to be at LINX (London INternet eXchange).

`l.root-servers.net` seems to be at ICANN.

`m.root-servers.net` seems to be in Japan.

More recently, this has been harder to work out. `d` is still at the University of Maryland, `i` is somewhere in Europe, but probably Netherlands rather that Sweden, `j` is now run by Netnod.se, `k` still seems to be at LINX, and `m` still seems to be in Japan.

general A good comment on DNS performance from Barry Margolin (`barmar@bbnplanet.com`): "We're primary or secondary for over 30,000 domains. Our servers are Sun Ultra 1's with 450MB RAM. The named process is using about half of this." Later: "150-200 queries/sec on our three authoritative servers, if I'm interpreting the statistics correctly. And our regional caching servers handle 200-300 queries/sec."

DNSSEC Stevens correctly says "the DNS is an essential part of any host". The DNS was designed in the days when the Internet was an essentially friendly collaboration, and the protocol is *not* secure against mistakes (see the Pakistan/Youtube incident on page 29 or deliberate attacks: essdentially a 16-bit `identification` field, and a limited range of port numbers, make it vulnerable to "poisoning", i.e. feeding it bad data. This makes the DNS Security extension described in RFCs 4033–5 important. It should be noted that (RFC 4033):

> The DNS security extensions provide origin authentication and integrity protection for DNS data, as well as a means of public key distribution. These extensions do not provide confidentiality.

As of July 2008, the vulnerability was described in `http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1447`. See also `http://tools.ietf.org/html/draft-ietf-dnsext-forgery-resilience-05`.

split horizon It is quite common for sites such as the University of Bath to run distinct views of the DNS internally and externally. This can be seen in the following session (some lines removed for brevity).

```
midge $ /usr/sbin/nslookup
> set debug
> midge.bath.ac.uk
Server:         138.38.32.45
    QUESTIONS:
        midge.bath.ac.uk, type = A, class = IN
    ANSWERS:
    ->  midge.bath.ac.uk
        internet address = 138.38.32.34
    AUTHORITY RECORDS:
    ->  bath.ac.uk
        nameserver = adns1.bath.ac.uk.
    ->  bath.ac.uk
        nameserver = adns0.bath.ac.uk.
> server dns0.cl.cam.ac.uk
> midge.bath.ac.uk
Server:         dns0.cl.cam.ac.uk
Address:        128.232.0.19#53

------------
    QUESTIONS:
        midge.bath.ac.uk, type = A, class = IN
    ANSWERS:
    ->  midge.bath.ac.uk
        internet address = 138.38.32.34
```

```
    AUTHORITY RECORDS:
    ->  bath.ac.uk
        nameserver = tamarin.bath.ac.uk.
    ->  bath.ac.uk
        nameserver = dns0.cl.cam.ac.uk.
    ->  bath.ac.uk
        nameserver = ns-oss.salford.ac.uk.
    ADDITIONAL RECORDS:
    ->  dns0.cl.cam.ac.uk
        internet address = 128.232.0.19
    ->  dns0.cl.cam.ac.uk
        has AAAA address 2001:630:200:4570::d:a0
```

Here we see that querying an *internal* server gives us internal name servers as the authority, whereas querying an *external*server gives us external name servers as the authority.

We also note that Cambridge quotes an AAAA record (as well as the A record) for its name server, which emphasises the fact that there is only one DNS, for both IPv4 and IPv6.

Split horizon DNS also enables one to 'hide' internal-only machines, as shown here. `urbmulti` is visble inside Bath.

```
midge $ /usr/sbin/!!
/usr/sbin/nslookup
> set debug
> urbmulti.bath.ac.uk.
Server:         138.38.32.45
Address:        138.38.32.45#53

------------
    QUESTIONS:
        urbmulti.bath.ac.uk, type = A, class = IN
    ANSWERS:
    ->  urbmulti.bath.ac.uk
        internet address = 239.192.0.252
    AUTHORITY RECORDS:
    ->  bath.ac.uk
        nameserver = adns1.bath.ac.uk.
    ->  bath.ac.uk
        nameserver = adns0.bath.ac.uk.
```

But from outside it is not visible, and indeed the address is a Class D (multicast) one.

```
> server dns0.cl.cam.ac.uk
```

```
Default server: dns0.cl.cam.ac.uk
Address: 128.232.0.19#53
> urbmulti.bath.ac.uk.
Server:         dns0.cl.cam.ac.uk
Address:        128.232.0.19#53

------------
    QUESTIONS:
        urbmulti.bath.ac.uk, type = A, class = IN
    ANSWERS:
    AUTHORITY RECORDS:
    ->  bath.ac.uk
        origin = tamarin.bath.ac.uk
        mail addr = lanmaster.bath.ac.uk
        serial = 2008091749
        refresh = 7200
        retry = 3600
        expire = 604800
        minimum = 86400
    ADDITIONAL RECORDS:
------------
** server can't find urbmulti.bath.ac.uk: NXDOMAIN
```

`pr-pharm2.bath.ac.uk` (`138.38.128.15`) is similarly hidden, but has a Class 'B' (public) address. Attempts to traceroute to it from outside Bath are blocked at the firewall.

```
 1 ge-0-3-5.glas-sbr1.ja.net (146.97.35.225) [AS 65533] 4 msec 4 msec 4 msec
 2 so-1-1-0.warr-sbr1.ja.net (146.97.33.113) [AS 65533] 12 msec 8 msec 12 msec
 3 so-0-2-0.read-sbr1.ja.net (146.97.33.109) [AS 65533] 12 msec 12 msec 12 msec
 4 SWERN-B2.site.ja.net (146.97.42.186) [AS 65533] 20 msec 16 msec 20 msec
 5 bath-fren-ph.swern.net.uk (194.83.94.65) [AS 64529] 20 msec 20 msec 20 msec
 6  *  *  *
```

Chapter 15

p. 209 TFTP is used at the University of Bath to install boot images in the machines in the Library and the computing laboratories. RFCs 2347–2349 update RFC 1350 by adding extra options.

p. 211 The "sorcerer's apprentice" happens as follows (assuming that the client is reading a file from the server).

- A packet (say 10) is delayed after leaving the TFTP process on the server (e.g. because of an Ethernet collision).

- The client times out and sends another ACK of 9, to prompt a re-send of 10.
- The client gets the original 10, and acknowledges it.
- The server sends 10 again, in response to the duplicate ACK of 9.
- The server sends 11, in response to the acknowledgement above.
- The client acknowledges the second copy of 10.
- The server sends 11 again, in response to the acknowledgement above.
- The client sends two acknowledgements for the two copies of 11.
- The server sends two copies of 12 in response to these acknowledgements.

The process continues doing double the required work until the file is received. Worse, the chance of collision is increased, so it is more likely that another packet will be delayed, and so on. See RFC 1123, paragraph 4.2.3.1.

p. 212 The reason that it is important not to tie up the well-known port is that a TFTP server normally serves a family of machines, e.g. in a laboratory. These may well all want service at once, e.g. after a power failure or a change of class.

Chapter 16

Connecting machines to the Internet, other than manually) has basically gone through three phases.

1. RARP (chapter 5) which will give an IP address from the corresponding Ethernet address. Of course, these dasy one wants more, such as a subnet mask etc., which leads to a proliferation of ICMP types (subnet request, router discovery etc.).
2. BOOTP (this chapter), which attempts to place all this, and more, in one protocol.
3. DHCP, which updates BOOTP (using the same packet format), which deals with the case where the machine is self-sufficient, e.g. a laptop, *apart* from its network connection, which may vary from time to time.

   > DHCP is designed to supply DHCP clients with the configuration parameters defined in the Host Requirements RFCs. After obtaining parameters via DHCP, a DHCP client should be able to exchange packets with any other host in the Internet. [RFC 2131]

p. 221, l. 10 *"this value (255)" should be "0".* [21]

p. 222 The definitive RFCs for DHCP[78] are now RFCs 2131/2. An additional

[78] DHCP = Dynamic Host Configuration Protocol.

option is proposed in RFCs 3004 and 3011. RFC 3046 adds an option whereby DHCP relay agents can add information about themselves to a DHCP reply. An authentication option is proposed in RFC 3118. RFC 4833 adds options to specify the time zone, and Daylight Saving Time options, e.g.

```
EST5EDT4,M3.2.0/02:00,M11.1.0/02:00
```

"a timezone that is normally five hours behind UTC, and four hours behind UTC during DST, which runs from the second Sunday in March at 02:00 local time through the first Sunday in November at 02:00 local time. Normally the timezone is abbreviated "EST" but during DST it is abbreviated "EDT"."

Chapter 17

p. 224 It should be noted that the acknowledgement described in the third bullet point is an acknowledgement that the receiving TCP has the segment of data, not that the application has it. When we look at protocols such as SMTP which are built on top of TCP, we will see that there are application-level "acknowledgements" as well to indicate that the application has *successfully* received and processed the data (which may be many segments in the case of a long mail message). The TCP acknowledgement must not be used as a substitute for this. This mistake is illustrated in the following posting[79].

> Hi, everyone I'm having a question about possible duplicate message delivery in TCP. Here's the scene:
>
> Time 1: 3-way handshake has been successfully finished. Client sends a packet with data to server.
>
> Time 2: Server receives the packet, sends the data to application level.
>
> Time 3: Before server sends out ACK, it crashes and reboots.
>
> Time 4: Client times out and resends the packet with data.
>
> Time 5: Server starts up, receives the resent packet and responds with RST.
>
> Time 6: Client tears down connection.
>
> Attention: the data has been accepted by the server (such as picking out 1000 USD from your account) and delivered to the application level but client doesn't know it. If the client tries again (say, picking out another 1000 USD from your account), what a mess?

p. 224 As described here, TCP provides protection against reordering of packets in the network. This was thought to be a very rare occurrence, but

[79] `48babe1f.0208121808.37721440@posting.google.com` to `comp.protocols.tcp-ip`.

currently [3] it is not. However Vernon Schryver (vjs@rhyolite.com) writes on 2 July 2001

> TCP does not work very well with out of order segments. Only a little re-ordering can cause a 20% retransmission rate as well as a large reduction of the congestion window. In practice, routes are fairly stable during the life time of TCP connections or burst of activity on long lived TCP connections. Brand name routers that support load sharing (using multiple next-hops to a single destination) go to lengths to cause any single stream to use a single route.

Barry Margolin (barmar@genuity.net) writes on 25 September 2001

> Although TCP is able to handle packets that arrive out of order, its performance is much better if this doesn't happen. Sending packets along different paths is likely to result in out of order delivery, so modern routers take steps to avoid this. The oldest mechanism they used for this is destination-based route caching: the first time a router needs to send to a particular destination address it selects one of the possible paths and then remembers this for all future packets to that address. Modern, high-performance routers do flow-based route caching; they peek into the TCP layer's header[80] to get the port numbers, and cache a specific path based on the IP addresses and ports.

See also RFC 4653.

Figure 17.2 Two extra flags have been added to TCP by RFC 3168. These are 'ECE' —- Explicit Congestion Notification (ECN) Echo, and CWR — Congestion Window Reduced. In the header, ECE is the bit before URG, and CWR the one before that. See that RFC for the gory details, or [4, section 10.6.3] for a readable summary. ECN also uses two bits in the IP header, part of the TOS field (bit 6 for ECN-capable indication (ECT) and bit 7 for Congestion Experienced (CE)). Routers read 'ECT', and set 'CE' when experiencing congestion. TCPs set 'ECE' when they receive packets with 'CE', and when they receive 'ECE', act as if they were doing the "fast recovery" process, i.e. halve the congestion window. They set 'CWR' to indicate that they have done this. Note the elaborate interplay between the congested router (which does not know the traffic is TCP — layering), the end that is told about congestion, and the end that can do something about it.

[80] Note the violation of the "layering" principle. However, as the information thus obtained is only being used to choose between legal options, it is not significant.

Chapter 18

p. 236 Though MSS is not "negotiated", the sender may send less than the MSS advertised by the recipient, e.g. due to Path MTU discovery (p. 340). Therefore algorithms like delayed acknowledgements (p. 277) must count segments, not bytes in units of MSS.

Equally, though Stevens implies that MSS=MTU−40, the increasing use of TCP options (especially Timestamp, see section 24.5) means that this is not always the case[81]. RFC 2525 (section 2.18) points out that failure to allow for this, especially in the presence of Path MTU discovery (section 24.2) can result in a complete failure to transmit. I have seen Windows 2000 systems advertise an MSS of $1380 = 1500 - 60 - 60$, and refuse to send larger segments than this, even though the other end advertises 1460.

p. 241 Note that this diagram incorporates the corrections from RFC 1122 to the original diagram from RFC 793.

Not all transitions in this diagram are supported by all applications. Barry Margolin (`barmar@genuity.com`) writes as follows.

> Both of these issues (`LISTEN` to `SYN_SENT` transition and simultaneous open) are probably more applicable to peer-peer protocols than client-server protocols. An example might be BGP (note: it doesn't make use of the above features, but it could have been designed to), where the neighboring routers establish TCP connections with each other in order to exchange routing information; if it were specified to use the same port as both the source and destination (which would not be unreasonable, since there's no need for multiple connections between the same two peers, except for the `TIME_WAIT` delay between reuse of the same ports) then a simultaneous open would occur if both peers tried to connect to the other at the same time. And a `LISTEN` to `SYN_SENT` could happen if one router first tries listening for an incoming connection, and when a timeout occurs it then tries to connect it.

p. 243 RFC 793's specification of the MSL as 2 minutes is more honoured in the breach than in the observance. [18] shows that over half of webservers have MSL as 15 seconds, over a third have it as zero, and only about 10% have it as 2 minutes.

p. 246 The MSL quiet time is enforced by no operating system that the readers of `comp.protocols.tcp-ip` know. For a client, this is not too serious, but servers reboot more quickly these days, and there is a potential risk here. One reader points out:

[81] In theory IP options can also invalidate this assumption, but RFC 2525 states that "Arguably, especially since the wide deployments of firewalls, IP options appear only rarely in normal operations."

> Boot time is not the only means by which a crashed IP address may be restarted — for instance, an IP address in a modem pool may be re-assigned within *seconds* of its last use (or even shorter — ISDN, for instance). Add to this any number of 'embedded' devices that are quickly restartable, and there are plenty of systems that might be thought to be subject to such a delay on sending new packets.

And another reader says the following:

> People have effectively wiped out the MSL delay with dynamic IP address assignment, where the addresses are assigned to different clients whenever they are called for — at periods when the pool is busy, of course, that would mean turnover of addresses within a few seconds.
>
> I'd imagine (but I haven't run an ISP, so I don't know) that this has been done for quite some time — with firewalls being implemented at more clients, of course, it's anyone's guess as to how long we'll go before we have to re-implement the MSL delay on assigning IP addresses to new dialups, or explain to our users that they'll be responding with RSTs to random machines for their first two or three minutes of activity :-)

p. 253 RFC 2018, a proposed standard, revives the proposal for Selective Acknowledgements, originally floated in RFC 1072. This adds one SYN-only option (to state that the option should be enabled), and one option that can come with any ACK segment. We describe this in our notes for page 312. RFC 2883 makes a further extension to this in the area of duplicate segments. As of August 2001, Sun's Solaris 7 was one of the few commercial operating systems to support selective acknowledgements, though more are now doing so. By default the support is "passive", i.e. it will be returned in the second SYN if the active open SYN from the other end had it enabled. By patching the kernel[82], it can be made "active", i.e. Solaris will enable it on active opens. See also RFC 3517 on selective acknowledgements in particular the observation "SACK information is advisory and therefore SACKed data MUST NOT be removed from TCP's retransmission buffer until the data is cumulatively acknowledged [RFC2018]". [8] reports that 13%of TCP traces collected would have benefited from SACKs. SACK is recommended (SHOULD) for operation over mobile networks in RFC 3481. [16, Table 7] reports that 68% of web servers, and 88% of clients, were "SACK-capable", though this conceals a variety of behaviours.

p. 254 An option found on many Unix servers is to economise on listeners by having a generic listener, often called `inetd`, which listens on a variety

[82]The command is `ndd -set /dev/tcp tcp_sack_permitted 2`. See `http://www.psc.edu/networking/perf_tune.html`

of ports (TCP and UDP) and forks a specific program to handle any incoming call. The set of ports to listen to and the associated programs is generally listed in a file, often `inetd.conf`. For machines that are rarely servers, e.g. desktop machines, this can economise on other system resources, even though not on the number of `LISTEN` entries in the tables reported by `netstat`.

p. 256 The possibility described here of restricting the local address, has one particular application. As described in III, p. 180 (option 1) some implementations of IP let one bind multiple IP addresses to one link-layer (e.g. Ethernet) address. Each IP address can then correspond to a different `www.X.com` (or whatever) address, and different HTTP servers can be run, each restricting its local address to the address corresponding to the `www.X.com` that it is serving.

p. 260, l. -2 *"active open" should read "active close".* [21]

Chapter 19

p. 265 RFC 2581[83] says that acknowledgements MUST be generated within 500ms. Apparently, Linux systems turn off delayed ACKs during the sender's "slow start" process (as guessed). It also appears (May 2002) that `midge` does not delay the first ACK for data (as opposed to the SYN) on any connection. This is a recommendation in RFC 2757.

p. 267 Not only do tinygrams consume bandwidth, they also consume CPU time in routers. Since today's high-performance routers consume CPU time largely independently of the size of the packet, this may be more serious.

A distressing tendency has been observed of both browsers and web servers to turn off Nagle: this has been observed to cause extra packet sends without improving performance.

The first two sentences of the paragraph begiing "This algorithm" should read "The algorithm says that when a TCP connection has outstanding data that has not yet been acknowledged, small segments cannot be sent until the outstanding data is acknowledged." [21]

Chapter 20

p. 277 The effects also depend on the relative synchronisation of the 200ms (and 500ms) timers on the two machines.

20.6 Stevens says "while this is OK when the two hosts are on the same LAN" and indeed none of his LAN-based examples show slow start, until be gets

[83] Repeated in 5681.

to Figure 20.8 across the wider Internet. Stevens' own machines probably had the code (see appendix C) that suppressed slow start on a LAN, but this has never been standard and, for the reasons in appendix C, probably should not be followed. Note that it is also a violation of layering, in that TCP would be changing its behaviour depending on the state of the connection as seen at the IP layer.

p. 285, ll. 14–15 *"2048" should be "3072" and "4094" should be "6144".* [21]

p. 285 RFC 2581 recommends that slow start should also be used if no data has been sent for more than one RTO, otherwise re-activating an idle connection can let it flood the network. There have been various changes to the initial value of *cwin*: see the discussion for p. 310 and equations (1/2) there. See also RFC 5681.

p. 285 These algorithms look relatively simple, but in fact RFC 2525 (section 2.3) states that *cwnd* is occasionally not set: when the second SYN of the three-way hand-shake does not specify an MSS (hence the MSS defaults to 536), some implementations do not set *cwnd* to the (implied) MSS.

p. 289 The product for an ATM-155 link across the Atlantic, again assuming 60ms RTT, would be

$$\underbrace{\frac{155\text{M}}{8}}_{\text{bytes/sec}} \times \underbrace{\frac{48}{53}}_{\%\text{ useful}} \times \underbrace{\frac{6}{100}}_{60\text{ms}} \approx 1\text{MB}.$$

This assumes "single-user" use: unlikely across the fat pipe during the day. However, a high-bandwidth load would require these sorts of buffer sizes to be effective.

There are also problems with wireless networks: RFC 2757 notes that "A 3rd Generation wireless service offering 2 Mbps with 200-millisecond latency requires a 50 KB buffer."

p. 289 There is a good animation of the effect of the window size to be found at `http://cable-dsl.home.att.net/rwinanim.htm`. The whole site is worth looking at.

This formula assumes no errors: if there is a packet loss rate of p, then the "Reno" TCP implementation is also limited to a maximum sending rate of

$$\frac{1}{RTT * \sqrt{2p/3} + RTO * (3\sqrt{3p/8}) * p * (1 + 32p^2)}$$

segments per second ([19] and RFC 3155).

To summarise chapter 20, there are two effects at work:

- **window**, i.e. flow control imposed by the receiver;

- **congestion window**, i.e. flow control imposed by the sender.

Suppose we have received an ACK acknowledging up to byte a, with a window size of w. Then flow control imposed by the receiver allows us to send up to byte $a + w$. Suppose also the last *cwin* update was to c, with effect from byte s. Then the bytes we can send are those up to byte $\min(a + w, c + s)$.

Chapter 21

p. 300 The calculation of RTO is re-described in RFC 2988. This states that RTO should always be at least one second (older implementations with a 500ms timer essentially did this anyway, but it now a requirement). This is particularly important over mobile networks (RFC 3481).

We should note that the variability of RTT is itself quite variable. [11] says that 60% of the TCP connections they measured had low variability[84], and in absolute terms 50% varied by less than 200–300ms.

p. 301 RFC 2988 makes it a requirement to use Karn's algorithms for deciding which samples to take. The only exception is when TCP timestamps (see section 24.5) are used, when there is no ambiguity.

Sections 21.6–8 These became RFC 2001, which was later re-written into RFC 2581, which can be viewed as an upgrade of RFC 1122 (Host Requirements). 2581 has been replaced by 5681, but without much change of substance.

p. 310 Congestion avoidance assumes that all packet losses are caused by congestion. If this is not the case, then, as pointed out in RFC 3155,

> TCP connections experiencing high error rates on their paths interact badly with Slow Start and with Congestion Avoidance, because high error rates make the interpretation of losses ambiguous - the sender cannot know whether detected losses are due to congestion or to data corruption. TCP makes the "safe" choice and assumes that the losses are due to congestion.
>
> - Whenever sending TCPs receive three out-of-order acknowledgement, they assume the network is mildly congested and invoke fast retransmit/fast recovery.
> - Whenever TCP's retransmission timer expires, the sender assumes that the network is congested and invokes slow start.
> - Less-reliable link layers often use small link MTUs. This slows the rate of increase in the sender's window size during slow start, because the sender's window is increased in

[84] Defined as $\frac{95 \text{ percentile}}{5 \text{ percentile}} < 2$.

> units of segments. Small link MTUs alone don't improve reliability. Path MTU discovery [RFC1191] must also be used to prevent fragmentation. Path MTU discovery allows the most rapid opening of the sender's window size during slow start, but a number of round trips may still be required to open the window completely.

RFC 3649 goes further, and points out that "for a Standard TCP connection with 1500-byte packets and a 100 ms round-trip time, achieving a steady-state throughput of 10 Gbps would require an average congestion window of 83,333 segments, and a packet drop rate of at most one congestion event every 5,000,000,000 packets (or equivalently, at most one congestion event every $1\frac{2}{3}$ hours)" — clearly unrealistic. Furthermore [25] it would take $1\frac{1}{2}$ error-free hours to build up to maximum rate.

p. 310 RFC 2414 suggests experimental initial values of *cwnd* between two and four segments (but never more than $4380 = 3 * 1460$ bytes). This is not yet standard, but RFC 2581 permits two segments, rather than the one in the book/RFC 2001. [18] shows that, as of May 2001, over 90% of Web servers sampled used 2MSS as the initial value. More recently (October 2002) RFC 3390 has the more complicated rule

$$\min(4 * MSS, \max(2 * MSS, 4380\text{bytes})). \quad (1)$$

$4380 = 3(1500 - 2 * 20)$ and therefore three segments of maximum size on 1500-byte MTU links. RFC 3481, now a "best current practice" document, recommends (SHOULD) this for TCP over 2.5G/3G wireless connections. There is a curiosity here, though: how does TCP know that it is transmitting over such a connection. Not only is this a breach of layering[85], it is impractical to discover whether the "other end" is on such a network. But this is precisely the common scenario — user on the end of, say, a BlackBerry wishes to download a fairly small web page, and hence the web *server* should use a larger congestion window. The only logical conclusion, which JHD had not seen elsewhere until RFC 5681, is that this SHOULD applies to all connections.

[16, Figure 2] shows that 96% of web servers seem to set the initial *cwnd* to one or two segments, and they comment "TCP initial windows of three or four segments are seeing very slow deployment in web servers". One possible reason for this is that the "big win" comes in moving from one to two segments, since at this point we don't get hit by a delayed ACK, since the second segment will force an ACK. Since this delay is perceptible to humans using a web browser (2.10 and RFC 1144), we want to avoid it where possible.

[85] It could be argued that Slow Start is fundamentally a breach of layering, since in an ideal world, TCP would not care whether the other end was on the same network or not.

RFC 5681 has the following upper bound on the initial value of *cwin*:

$$\text{initial } cwin \leq \begin{cases} 2S & S > 2190 \\ 3S & 1095 < S \leq 2190 \\ 4S & S \leq 1095 \end{cases} \tag{2}$$

where S is, in practice, the sender's true MSS, i.e. after taking account of the MSS sent from the other end, or the 536-byte default if none was sent..

TCP Vegas (see [5]) makes a slight change by only deciding that congestion has occurred (point 3) if the segment referred to was transmitted *after* the last time congestion was detected. Otherwise it is possible for one overestimate of the sending rate to generate multiple decreases.

p. 311 At the top of the page, it is stated that *cwnd* be updated by $1/cwnd$ segments (i.e. $segsize/cwnd$ bytes) every time an ACK is received. RFC 2581 clarifies this to say that the updating happens for every **non-duplicate** ACK received. This formula may appear mysterious, but the aim is to increase *cwnd* by one segment per round-trip time, and we estimate that there are $cwnd/segsize$ packets in transit.

p. 312 One problem that has caused confusion in the past is the **definition** of a "duplicate ACK". Stevens does not explicitly define it, but RFC 3517 says:

> we define a "duplicate acknowledgment" as a segment that arrives with no data and an acknowledgment (ACK) number that is equal to the current value of HighACK [the greatest ACK value received] ...

Note the "with no data" — since the ACK bit is normally set, packets with data may well duplicate the previous ACK value. There is an even more complicated definition in RFC 5681, requiring

(a) the receiver of the ACK has outstanding data,

(b) the incoming acknowledgment carries no data,

(c) the SYN and FIN bits are both off,

(d) the acknowledgment number is equal to the greatest acknowledgment received on the given connection and

(e) the advertised window in the incoming acknowledgment equals the advertised window in the last incoming acknowledgment.

p. 312 One theoretical problem with the algorithm here is that it assumes that at least the lost packet, plus three to generate duplicate ACKs, can be in transit simultaneously. Because of "slow start", this will often not be the case at the start of a session, or for a small session such as HTTP often generates. A further practical problem with the standard BSD (including

BSD Reno) algorithm is that its definition of when to do a standard re-transmit is based on the RTO estimate calculated from the 500ms timer. [5] notes that it took 1100 ms to spot a time-out, on average, whereas the figure calculated from a perfect clock would be less than 300ms (but note that RFC 2988 mandates a minimum of 1 second for the RTO). They therefore propose a variant, nick-named Vegas, which does the following.

- Uses the system clock (generally microsecond resolution these days) rather than the 500ms timer for its own RTT and RTO calculations.
- Regards a *single* duplicate ACK arriving after its RTO calculations as an indication to re-transmit.
- If the first or second ACK after a re-transmission is *not* duplicate, Vegas checks for any unacknowledged segments which are overdue by its RTO calculations, and re-transmits them.

The "fast recovery" algorithm mentioned in the book, and the variant mentioned above, is fine if only one packet is ever lost at a time (meaning during an RTT). However, we note that a full RTT (plus the arrival times of the packets to generate the duplicate ACKs) is required to get an ACK back acknowledging data beyond the lost segment. If there is more than one lost segment per RTT, the process has to start again immediately (the receiver knows that there is another gap, but can't signal it until all the data up to the gap has been received, since an ACK acknowledges all data up to that point). In practice, what happens is that the original re-transmit algorithm kicks in, and we re-transmit a lot of unnecessary data. RFC 2582 (now replaced by 3782) proposes what to do in these circumstances (see steps 1A, 5 and 6 in its procedures). RFC 2582 implementations are, according to [18], the commonest kind of webserver, with 40%, or more if one includes Windows implementations with a bug in the handling of small pages.

"Selective acknowledgements", originally proposed in RFC 1072, and revised to a proposed standard in RFC 2018[86], solves this problem by adding a new TCP option, which says that "in addition to all the bytes up to the ACK value, I also acknowledge that I have the following blocks of bytes". Once this has been received, the sender know precisely which blocks to re-transmit. Since this is a TCP option, it has to fit in the space allowed by the TCP header length, whose maximum value is 15 (= 60 bytes). This leaves room in principle for selective acknowledgement of four blocks. However, if we are using this option, it is because we have a large fat network, so we probably also have the timestamp option (pp. 349–351) enabled, which cuts down the available space to leave room for three blocks. A block may be more than one segment, of course, and it is likely on a large fat network that losses are bursty: a router has a momen-

[86] As updated by RFC 2883.

tary overload and drops several packets in quick succession[87]. According to [18], 40% of webservers claim to support selective acknowledgements, but in fact only 40% of these actually make use of the information. RFC 2757 recommends selective acknowledgements in the wireless setting.

RFC 2883 extends the proposal for selective acknowledgements in an upwards-compatible way to allow the receiver to specify that duplicate segments have been received. This may help performance in the presence of re-ordering (which otherwise is hard to distinguish from duplication). This extension works by transmitting block numbers *before* the main TCP ACK field, whereas SACK was originally used to report holes in received data transmits block numbers *after* the main TCP ACK field.

p. 312 RFC 2581 slightly revises step 1. Rather than $cwnd/2$, it mandates half the number of unacknowledged bytes (which may be different from *cwnd*, especially if two fast-retransmits happen in succession).

p. 312 A further problem with "fast retransmit" is that there have to be three duplicate ACKs, which means that three more segments have to be there to cause these duplicate ACKs. This means that the data must exist, and the congestion window must be large enough to permit them being sent. In the case of a web server, with lots of small transactions, this may well not be the case, and indeed [2] observes that 56% of the retransmissions were caused by a time-out, and only 44% by "fast retransmit". RFC 3042 proposed some modifications to deal with this problem, notably by responding to the first two duplicate ACKs by sending new data even if this breaches the cwin rules. This would have converted 25% of the RTO-based retransmissions into "fast retransmit" retransmissions.

Emil Naepflein (`Emil.Naepflein@philosys.de`) also observes the following.

> In the early days the windows were often so small that only about 4 packets were in transit most of the time. If you go much higher [than three duplicate ACKs before invoking fast retransmit] with the fast-retransmit trigger you will probably never get any benefit from it. I very often noticed that even 3 duplicate ACKs was too high in case of transmitting packets over a lossy link with high delay and large MTUs.
>
> The fast-retransmit can only achieve good performance if there are a lot of packets in transit so that you get enough duplicate ACKs before you are running out of the congestion window. Otherwise you will get some stop-and-go behaviour with bad performance.

[87] It appears that some router designers do not work on the "equal misery" principle, but would rather drop several packets from one connection than one from several.

RFC 4653 discusses whether 3 is the right number, especially in the presence of packet re-ordering. It suggests increasing it to a congestion window's worth of data, but only in the presence of SACK.

p. 314 RFC 2525 shows that the extra term (256/8 in this case) is bad, and RFC 2581 forbids it.

p. 314, l. 2 "first" should be "third". [21]

p. 316 Storing per-route metrics is very valuable for some higher-level protocols such as FTP (where multiple files can be fetched, each with a different TCP connection — see pp. 419–439) and HTTP, especially to a Web cache. We note the definition of "enough" as 16 windows: many implementations actually use 16 RTT[88] as the measure. The TCP Timestamp option (section 24.5) allows faster collection of RTTs, and is therefore advantageous, particularly in the web case, where most files may well be smaller than 16 windows.

Chapter 24

p. 340 There is a table on support of Path MTU discovery, timestamps, window scale etc. in various operating systems at `http://www.psc.edu/networking/perf_tune.html`. [16, Table 5] gives fairly depressing results on the deployment of Path MTU, with only 41% of web servers deploying it correctly.

p. 344 Since ATM does not do "store-and-forward" at the TCP/IP 2 level, its use decreases the latency, especially across large WANs.

p. 344 Stevens does not show "gigabit over satellite", but this is now achievable, and has a bandwidth-delay product of 62Mb. Satellite links have many other problems, described in RFC 2488.

Equally, he does not discuss modern mobile networking. RFC 3481 points out that the bandwidth-delay product in W-CDMA (used in Europe for 2.5G and 3G mobile networking) has a bandwidth-delay product of 50KB, even for a 384Kb operating over short ranges. This is because what TCP sees as the physical layer, which is actually a complicated mechanism with its own framing, time-outs, retransmission etc., is not amenable to the simple calculations shown here.

Large bandwidth-delay products means that the range of acceptable sequence numbers for a reset (RST flag set) packet is such that denial-of-service attacks become feasible — see RFC 4953.

p. 347 Note that there is a difference between sending (in the initial SYN) a shift count of 0 (a scale factor of $2^0 = 1$) and not sending a window

[88] Exercise: why does 16 windows imply 16 RTT?

scale option at all. The former indicates that the sender will accept and understand window scaling from the other end, though it does not wish to use it. The latter says that the sender does not understand window scaling, and so the other end MUST NOT send such an option. RFC 1323 says "Thus a TCP that is prepared to scale windows should send the option, even if its scale factor is 1" (i.e. what is sent is 0, since $2^0 = 1$). [16] report that 26% of web clients they tested *did* support window scaling, even though 97% of these had a scale factor of 1.

p. 349 Note, however, the timestamps, because they are TCP options, prevent TCP/IP header compression (p. 25 and notes thereto) and therefore are unhelpful on slow links. Slow links do not normally need the accuracy of RTT estimation that timestamps provide, and do not need PAWS protection (next section). Fortunately, slow links tend to occur in situations like dial-up from home, where most connections (certainly HTTP connections) are initiated from the home end, and therefore if this end does not send timestamps, they will not be used.

p. 354 It is astonishing how "Ethernet can only support 100Kbytes/sec" remains in the folklore, even though it is well-exploded by now. Equally, the "fat pipes" across the Atlantic are normally full, so high-speed networking is a reality.

p. 356 As Stevens says, the upper limit is determined by the size of the TCP window, hence the importance of the window scale factor, and the allocation of suitably-sized buffers. This latter point is often ignored[89]. We note (8 December 2008) that the BUCS Solaris servers have a maximum `cwin` of 1MB, which with an RTT of 200ms (reasonable for the West Coast of U.S.A.), means a 40Mb/sec effectie bandwidth limit. This is probably reasonable for general-purpose machines.

Chapter 25

p. 366 *The text after figure 25.7 should read: "...a table containing entries, each containing two simple variables".*

p. 367 *Similarly, Figure 25.9 describes the two simple variables in* `udpEntry`*, rather than* `udpTable`*.*

p. 368 The word "lexicographic" has a subtle meaning in this context, viz. "lexicographic as a sequence of bytes". Hence the byte containing 67 is lexicographically less than the byte containing 161, even though, as ASCII strings, "161" would come before "67".

[89] See Wu & Chien: *GTP: Group Transport Protocol for Lambda Grids* in Proc. 2004 IEEE Symp. Cluster Comp., pp. 228–238. How figure 1, which clearly shows a maximum window of 64K, was published, or how the authors could write without referring to RFC1323, escapes me.

p. 387 SNMPv2 is not on the IETF standards track[90]. SNMPv3 is described in RFCs 2570–5 and is a "Draft Standard". The most significant change in SNMPv3 is described in RFC 2571 as "Address the need for secure SET support, which is considered the most important deficiency in SNMPv1 and SNMPv2c".

Chapter 26

This chapter describes Telnet (the orginal ARPAnet/Internet connection mechanism) and rlogin (the Berkeley Unix mechanism). These days another common connection mechanism is SSH (RFC 4251–5). SSH is substantially more complex than either, since it allows for a number of security options, and also permits operations such as the forwarding of X11 data within a single connection. In terms of the 7-layer model, SSH itself is at the session layer (layer 5), with SSH terminal sessions at layer 7. These use the same pseudo-terminal mechanism as that shown in figure 26.1. SSH also uses the "terminal type" mechanism of Unix/rlogin, but doesn't bother with the speed — largely irrelevant these days.

p. 392 The recommendation against using `.rhosts` files is certainly as valid now as then. Especially in Universities (and Bath is no exception), a Kerberos-based system is used instead.

p. 395 The comparison between client→server signalling and server→client signalling is not quite as clear (in my opinion) as Stevens makes out. In particular, Telnet uses (essentially)[91] in-band in both directions, but has a true escape mechanism, whereas rlogin does not have a way of sending the escape character `0xff` as data.

p. 398 Logically (especially in the context of large fat networks), step 8 need only follow step 2, but in a LAN context (where rlogin is normally used) it will follow step 7.

pp. 401–3 telnet is fundamentally trying to solve a N^2 problem: if there are N different kinds of terminal, there are N^2 possible mappings. Mapping to/from NVT makes it a $2N$ problem, but at the cost of crippling some functionality/response, so a compromise has to be sought. This explains the graph of code size in Figure 26.1

Furthermore, the world is less and less a purely ASCII (especialy 7-bit ASCII) place. RFC 5198 (March 2008), the first genuine update to Telnet (RFC 854, May 1983) for 25 years, specifies Net-Unicode, essentially a UTF-8 (RFC 3629) encoding of Unicode (ISO 10646). This also makes its way ito e-mail via RFC 5335.

[90] According to RFC 3291. The latest edition of the RFC index (May 2008) describes RFC 1441 etc. as "Historic", but that is a fairly recent change.

[91] The data are all in-band, marked with the escape character: however, the urgent pointer is sometimes set to help the other end see these data as soon as possible — see segments 3 and 4 in Figure 26.17.

Figure 26.13 Segments 7 and 8 have, due to the delayed ACK rule, a 20% chance of being combined. So an option can be acknowledged at the same time as it is being replied to, but the TCP client cannot see this.

p. 413 Of course, if the Nagle algorithm is disabled, then every character typed is sent in a separate segment.

Chapter 27

p. 419 FTP is one of the oldest applications for the Internet. It has gained a new lease of life via URLs of the form `ftp://...`, where the browser creates an automatic anonymous FTP, and retrieves the file requested. Indeed, the user may not even see this URL, since it may be embedded in a JavaScript run by a "get updates now" or equivalent button. Various extension to FTP are described in RFCs 2228, 2640, 2773 and 3659. FTP is almost unchanged in the transition to IPv6: Barry Margolin `<barmar@genuity.net>` writes:

> The general operation of the FTP protocol is unchanged. The only additions are the new commands that are used when creating the data channel, since the commands in the v4 protocol contain v4 addresses, and they now need to be able to contain either v4 or v6 addresses. The EPRT and EPSV commands use the familiar dotted-quad format for v4 addresses and colon format for v6 addresses. There's an example right there in the RFC:
>
> ```
> The following are sample EPRT commands:
>
>      EPRT |1|132.235.1.2|6275|
>
>      EPRT |2|1080::8:800:200C:417A|5282|
> ```

Note that "colon format" is also often used for MAC (Ethernet) addresses.

I don't quite agree with Stevens' sentence "FTP handles all the differences ...". FTP *does* use NVT ASCII for the *control* connection: the "different approach" referred to by Stevens refers to the *data* connection.

p. 422 Option (c) at the top of the page, "Page structure", is essentially obsolete, as TOPS-20, the only operating system really to support it, has, despite its elegance, bitten the dust. Hence, in the quote from RFC 1123 at the bottom of the page "exists" should become "exists (or existed)".

p. 422 The second option (c), "compressed mode" is essentially obsolete, as software such as gzip do a far better compression of text files. There is also now an RFC (3173) describing IP-level compression.

p. 423 Not only does SMTP use the same conventions for error codes, so does HTTP. Thus the dreaded "401" is to be interpreted in the light of this convention.

p. 423 Note that, since the PORT command contains IP addresses and port numbers, it must be interpreted by Network Address Translators (see Appendix B).

p. 437 The Unix command `compress`, resulting in files ending in `.Z`, has largely been replaced by `gzip`, or its Windows analogue `winzip`, resulting in files ending in `.gz`, as specified in RFC 1952.

Chapter 28

passim RFCs 821 and 822 have been superseded by 2821 and 2822. Despite their numbers, these RFCs actually date as RFCs from April 2001. In particular, RFC 2821 makes support of EHLO (the extended form of HELO) compulsory for MTAs. However, many MTAs, such as `midge`'s, do not support it. In turn, they have been replaced by 5321 and 5322, but the changes seem to be minor. RFCs 5335 and 5336 extend 2821 and 2822 to include UTF8 in e-mail (in particular 5336 defines an extension UTF8SMTP to RFC 2821), but this seems not to be taken up in RFC 5321/2. Its status is currently (22/11/2009) unclear to JHD. See also RFC 5504.

p. 441 The statistics at the start of this chapter are very dated. For an opposing point of view, see the UK↔US (fat pipe) statistics[92] at `http://bill.ja.net`[93]. However, we also noticed that different universities have very different patterns of usage[94]. The average size of message has also changed drastically — a brief look at my e-mail box shows an average size of 20K bytes.

p. 441 This figure looks somewhat dated today. It still applies to the user interacting directly with a user agent such as PINE[95], but tools such as Webmail, Simeon or Microsoft Outlook employ a more sophisticated model (see the note labelled "Chapter 28" below). Nevertheless, it is important to understand this model first.

[92] For 4 May 2001, 3250GB were received from North America, of which mail was 0.99%, FTP 4.98% and WWW at least 56.36%. For the whole of March 2001, 75Tbytes were received, with mail being 0.99% and FTP 5.54%.

[93] But this service has also been discontinued due, partly, to the cost of measurement at routers.

[94] For example, in March 2001, the University of Bath had 1.89% mail of its 335Gb of traffic from the U.S. (and places relayed via the U.S., e.g. Australia). Conversely, Basingstoke College of Technology had only 0.46% mail, and their traffic was 92.34% Web.

[95] PINE = Pine Is Not Elm, one of the self-referential jokes so common in the Unix world. It has largely replaced ELM as a Unix-based mailer. PINE actually now also comes with an IMAP client, and this is how it is generally used at the University of Bath.

p. 442 Sendmail is indeed the MTA shipped with most Unix systems. However, it is notoriously buggy (the original "Internet worm" exploited a hole in sendmail) and has a weak security model, so many sites, especially universities, have replaced it with a more modern MTA, such as MMDF, qmail or exim. Notice, from Figure 28.1, that changing the MTA might change the way the UA interacts with the MTA (for example, sendmail places incoming mail for `jhd` in `/usr/spool/mail/jhd`, whereas most of the others place it in `~jhd/.mail`), but not the way that MTAs interact with each other, since that is specified by the SMTP protocol.

Figure 28.2 Note that the sender waits for an acknowledgement of each command before sending the next. This is normal in SMTP, though earlier standards do not comment on this. RFC 2920 specifies an SMTP extension that allows a server to declare that it will accept multiple commands (often called "pipelining"), and specifies what the client may then do. This can reduce the number of small packets exchanged, and reduce the latency in mail delivery, particularly when there are a large number of destination addresses. It also avoids the problem mentioned in Exercise 28.4.

p. 445 The "minimal commands" now have the extension EHLO as well as HELO, and RFC 2821 added EXPN to the required list. TURN, SEND, SAML and SOML are now deprecated: TURN has serious security problems. RFC 3461 introduced the extension DSN for Delivery Service Notification. RFC 5336 adds UTF8SMTP, to allow the UTF-8 encoding specified in RFC 5335.

p. 445 The headers are now defined in RFC 2822, with several changes. In particular time zone names (such as MST = "Mountain Standard Time" on the example on p. 443) are deprecated, and should be replaced by numeric values, e.g. −0700 in the case of MST.

p. 446 *Figure 28.3 is in fact a revised version of Figure 28.1.*

Figure 28.3 The University of Bath's configuration is rather like either half of this diagram. The main difference is that there are two relay MTAs: primary `pat.bath.ac.uk` and secondary `mercury.bath.ac.uk`, with the choice determined by MX records (see page 450). If one looks up internally, one also sees `bucs.bath.ac.uk`, which is an alias for all the Unix servers, but this is not advertised externally (an example of so-called "split horizon DNS" — see page 39).

p. 448 RFC 1123's requirement that a mail sender should not give up for at least four days imposes a requirement on MTAs that they should not be down for more than four days. Institutions that close over, say, the Christmas–New Year period[96] will cause mail sent to them to be returned to the sender as undeliverable.

[96] As some otherwise respectable U.K. universities did over 1999–2000.

p. 449 The interaction between an MTA and the DNS is now specified in RFC 2821. In particular, CNAME records should only be looked for if there are no MX records, so step 2 is wrong.

p. 452 While recipient MTAs generally perform a reverse (PTR) lookup on the incoming IP address, and sometimes print a humorous message, it is rare for them to log the failure to match the result of the reverse lookup. The failure might be a consequence of a mis-configured DNS, but it might also indicate an attempt at mail forgery. Indeed RFC 2821 seems to forbid rejecting the mail in these circumstances:

> An SMTP server MAY verify that the domain name parameter in the EHLO command actually corresponds to the IP address of the client. However, the server MUST NOT refuse to accept a message for this reason if the verification fails: the information about verification failure is for logging and tracing only.

However, there is a "get-out-of-jail-free"[97] card in section 7.7:

> It is a well-established principle that an SMTP server may refuse to accept mail for any operational or technical reason that makes sense to the site providing the server.

Section 28.4 Although MIME was originally designed for the SMTP community, it has been taken up, in great volume these days, by the Web community: see III/p. 168, as a way of conveying various types of data over HTTP. The fundamental issues are the same.

p. 452 RFC 1425 has been obsoleted by RFC 1651, itself obsoleted by RFC 1869, and now by RFC 2821.

p. 453 Similarly, RFC 1427 (the SIZE option) has been obsoleted by RFC 1653, itself obsoleted by RFC 1870, and now by RFC 2821. It should be noted that the SIZE option changes the format of the `Mail From` command, and hence should not be used unless the recipient signals that it is acceptable.

p. 454 RFC 1522 has been obsoleted by RFCs 2045–9, themselves updated by RFCs 2184, 2231, 2646, 3023 and 3798.

p. 454 ISO standard 8859-1, commonly known as ISO-Latin-1, is an encoding for the letters found in (western) European latin-based alphabets, such as the "é" in the examples.

p. 454 One also encounters (especially in these days of Russian spam) Windows-1251 as an encoding. It is described [24, Windows-1251] as

[97] I am grateful to Mr. Davis for the terminology.

> Windows-1251 is an obsolete 8-bit character encoding, designed to cover languages that use the Cyrillic alphabet such as Russian, Bulgarian and other languages. It is the most widely used for encoding the Serbian, Macedonian and Bulgarian languages.
>
> In modern applications Unicode is a preferred character set. Windows-1251 and KOI8-R (or its Ukrainian variant KOI8-U) are much more commonly used than ISO 8859-5, which never really caught on. In the future, both may eventually give way to Unicode.

'Obsolete' is, alas, a piece of wishful thinking on the part of the Wikipedist. The Microsoft reference is at `http://www.microsoft.com/globaldev/reference/sbcs/1251.mspx`. An example is

```
From: =?Windows-1251?B?wO3g8u7r6OkgyuDv8ODt7uI=?= <antares@holyrood.ed.ac.uk>
```

which a system understand Windows-1251 would render as

> From: Анатолий Капранов <antares@holyrood.ed.ac.uk>

p. 456 MIME is the mechanism by which attachments are transferred, and hence has become ubiquitous in the Internet. This emphasises Stevens' point, that the MTAs *require* no modification. RFC 1521 has been obsoleted by the family RFC 2045–9, themselves updated by RFCs 2184, 2231, 2646 and 3023 and 3798.

p. 457 Another very common `application` these days is `msword`; there are also `x-excel`, `pdf`, `rtf`, `mac-binhex40` etc.

p. 457 The access-types allowed are `ftp`, `anon-ftp`, `tftp` (rarely used), `local-file` and `mail-server`.

Chapter 28 We said under page 441 that the diagram looked somewhat dated today. With the growth of mobile computing and "e-mail at home", the view of that diagram, that a user only ever interacted with one user agent and a local message store, is fast becoming obsolete. The view that many users adopt today can be seen as

at least as far as receiving is concerned. There are two very different kinds of candidates for ***.

POP =Post Office Protocol, see RFC 1939 (as updated by RFCs 1957, 2449). This is based on the rural American view of a post office: one goes down there, posts some mail previously written, picks up the new mail, and goes back home, to read the mail and write some

more. Here, "goes down" means logging in to the Mailbox (POP) server, sending stored mail, and retrieving all new messages. This is intended for use when dial-up is not cheap, since human reading and writing can be done off-line, but can be very expensive if one receives large attachments, since they are downloaded as well.

IMAP =Interactive Mail Access Protocol, see RFC 2060 (obsoleted by RFC 3501 and numerous updates) and 2683. This, though the later of the two protocols[98], is closer to the traditional model of electronic mail as described by Stevens. Here we assume that the user's software interacts with the mailbox, not because they are on the same machine, but via a well-defined protocol — IMAP.

IMAP views the "mailbox" as a sequence of (possibly hierarchically-nested) folders, each containing messages and one of which is the "inbox", and provides facilities to list the contents of a folder, to retrieve header information about a message (without retrieving the whole message: important if there's a 5Mb attachment!), to retrieve the whole message, to delete/copy messages and so on (24 commands in all). To send messages, the user agent becomes an MTA and communicates the message via SMTP (generally to a relay MTA, as in the top half of Figure 28.3).

IMAP user agents tend to assume that TCP/IP connections to the IMAP server (port 143) are always open, and that SMTP (port 25) connections to the relay MTA (which could, but need not, be the same machine) are always possible.

While IMAP is becoming more popular, the choice between IMAP and POP depends mostly on the user needs, driven themselves by the cost of communication. As regards authentication mechanisms, the two are converging (see RFC 2195).

Chapter 29

p. 461 The chapter largely describes NFS version 2 (RFC 1094). Version 3 is described in RFC 1813 and version 4 in RFC 3010[99].

p. 462, second (3) Technically speaking, it is not the RPC mechanism (RFC 1057) which handles the data translation, but XDR (RFC 1014) or an equivalent. However, all RPC packages actually implement both.

p. 469 Although NFS file handles are meant to be opaque, some implementations had clients which did look at the contents of the file handle. This

[98] The first POP RFC was 918 (1984); whereas the first IMAP RFC was 1064 (1988).
[99] RFC 3010 obsoleted the previous RFC 2624, and is itself obsoleted by RFC 3530.

caused serious compatibility problems when the clients and servers were from different vendors, with different layouts of the file handle[100].

p. 469 However, in WebNFS[101], there is a concept of a "public file handle", described as follows.

> It is an NFS filehandle with a reserved value and special semantics that allow an initial filehandle to be obtained. A WebNFS client can use the public filehandle as an initial filehandle rather than using the MOUNT protocol.

It is an all-zero file handle: more precisely 32 bytes of zero in NFS v2 and a handle with a length of zero in NFS v3.

Incidentally, we should note that the last line of this page should read *increases with version 3 to a maximum of 64 bytes* — NFS v3 file handles have a length field.

p. 469 The generation number exists to deal with the following problem.

Client: Opens file `fred` for reading, gets back a file handle with i-node 9999.

Server: Removes file `fred`, and the i-node is free for re-use.

Server/Client: Creates file `joe`, to which the server allocates i-node 9999.

Client: Reads from the file-handle obtained above, thinking it is reading from `fred`, but in fact reading from `joe` (the NFS read protocol only talks in terms of file handles, not names).

With generation numbers, the fourth step would fail, since there would be a mismatch of generation numbers in the file handle, and the client would print the error message "stale NFS file handle".

In practice, at step 2 the server tries to check (using the locking mechanism) whether `fred` is in use on the client, and, if so, renames it as `.nfs9999` in order to keep the contents around, and preserve the i-node.

NFS is not perfect, however. If two clients[102] C_1 and C_2 are accessing the same files on a server S, say in the order C_1, C_2, C_1, with each access being of the form "read the file, then write it", and the different clients not overlapping (if they did overlap, then the lock manager should intervene),

[100] Why have different layouts? Since the file handle is opaque, i.e. not meant to be read on the client, there is no reason to pass its contents through XDR (other than as a byte array). Hence the same C code, running on big-endian and little-endian machines, would actually generate different byte orders in the file handles.

[101] Documented in RFCs 2054 and 2055.

[102] This scenario is taken from Ric Werne's post to `comp.protocols.nfs` — message `9RTc4.2686$zU5.28853@wbnws01.ne.mediaone.net` dated 5th. January 2000. He states "Most Unixes have 'close-to-open' consistency where an open will check the attributes on the server and invalidate the cache if the file is newer or a different size".

then what may happen is that C_1's second access may not see C_2's changes, since the old C_1 data are still in C_1's cache. Even running `sync` on C_1 (assuming C_1 is a Unix machine) will not help, since this will merely double-check that the data have been sent to S, but will not invalidate C_1's cache. Fundamentally, most file systems pre-date NFS in their design, even though they have been patched since.

p. 473 Although NFS traditionally used UDP, there are great advantages in using TCP when there is any possibility of packet loss: the fixed (and large) time-outs in RPC over UDP are no match for the sophisticated RTT calculations in TCP. "For example, a 1% packet loss rate on a 10MHz Ethernet with classic default NFS/UDP/IP timeouts causes a 90% loss of throughput using 8 KByte buffer sizes and no more read-ahead than done by classic UFS style file systems."[103]

NFS, and network file systems in general, form a major area of research and development. It should be noted that performance of such systems is often counter-intuitive: for example one study [9] shows that 43% of NFS operations are actually `READDIR` and 36.5% are `GETATTR`, with only 16.9% being data transfers. However, the data transfers accounted for 27% of the load on the file server, whereas the `GETATTR`, despite being twice as numerous, accounted for only 20%.

NFS version 2 required all write commands to be *synchronous*, described as follows in RFC 1094 (JHD's emphasis).

> All of the procedures in the NFS protocol are assumed to be synchronous. When a procedure returns to the client, the client can assume that the operation has completed and any data associated with the request is now on stable storage. For example, a client WRITE request may cause the server to update data blocks, filesystem information blocks (such as indirect blocks), and file attribute information (size and modify times). When the WRITE returns to the client, it can assume that the write is safe, even in case of a server crash, and it can discard the data written. This is a *very important part of the statelessness* of the server. If the server waited to flush data from remote requests, the client would have to save those requests so that it could resend them in case of a server crash.

NFS version 3 allowed asynchronous operations, and introduced a `COMMIT` operation to force this commitment to stable storage. There is a good description at `http://nfs.sourceforge.net`[104]. Many people still believe that asynchronous writes are morally dubious.

---

[103] Vernon Schryver (`vjs@rhyolite.com`) in news article `a0c0g6$fg6$1@calcite.rhyolite.com`.

[104] However `http://nfs.sourceforge.net/nfs-howto/ar01s05.html#sync_versus_async`, at section 5.9, says "The default export behavior for both NFS Version 2 and Version 3 protocols, used by **exportfs** in `nfs-utils` versions prior to `nfs-utils-1.0.1` is 'asynchronous'.". This describes the implementation in Linux, not the actual RFCs.

Chapter 30

p. 483 RFC 1288 details the many security loop-holes with finger.

p. 485 WAIS, Gopher and Veronica are essentially obsoleted by WWW.

p. 486 These days, of course, one accesses WWW via any browser.

p. 486 Notice the difference between the X server and the window manager. The X server manages the display (and mouse), redrawing (bits of) the display when the mouse moves, noticing that the mouse has changed window (and probably telling the window manager), redrawing the window as the application (e.g. a full-screen editor) requests, etc. The window manager controls the interaction with the windows, which is on top of which, and so on. It does this by sending commands to the X server, which is the only program that can actually write to the display.

p. 490 A Low Bandwidth X proxy is indeed distributed with X11R6.3.

Answers

p. 513, 10.6 These RIP messages are all discarded, since no-one is listening at that port. However, an ICMP "port unreachable" is not generated, since the source address was broadcast (rule 5, p. 71).

p. 523, 28.3 Of course, the fact that the MTA makes several writes, one for each command, does not guarantee that each write is transmitted in a separate TCP segment, since, with Nagle on, several application writes can be combined into one segment. Hence the "brain-dead" implementations described in the answer are definitely wrong.

Bibliography

The reference Almquist (1993) appeared as RFC 1716 (November 1994), but has been obsoleted by RFC 1812 (1995), itself updated by RFC 2644. This is a good summary of IP addressing rules and the requirements for IP routing.

III Chapter 13

p. 161 The definitive version of HTTP at the moment is 1.1[105]. See also RFC 2145 for the interpretation of the HTTP version number. The growth in header fields has been such that there is now a register to maintain this ever-growing list: see RFC 4229.

[105] Defined in RFC 2616, which obsoletes RFC 2068 (which also described HTTP 1.1), and 2817, which deals with the interoperability of `http` and `https` streams, i.e. mixed secure/insecure pages.

p. 161 The growth in HTTP continues: see the notes to page 441. Indeed (26.4.2007), 60% of all traffic to the University of Bath was YouTube.

p. 163 The note at the top is very prophetic: look at the size of a modern version of Netscape, say.

p. 165 There are more requests in HTTP/1.1: the new ones are POST, DELETE, OPTIONS, CONNECT and TRACE. [15] contains a useful analysis of the changes and their status.

p. 165 The HEAD request is less useful than it used to be, since it is now possible to do a variety of conditionals GET requests.

p. 166 There are many more header fields in basic HTTP/1.1: Accept, Accept-Charset, Accept-Encoding, Accept-Language (meaning human language), Accept-Ranges, Age, Cache-Control, Connection, Content-Language (also human), Content-Location, Content-MD5, Content-Range, ETag (short for "entity tag"), Expect, Host, If-Match, If-None-Match, If-Range, If-Unmodified-Since, Max-Forwards, Proxy-Authenticate, Proxy-Authorization, Range, Retry-After, TE (for "transfer encodings acceptable"), Trailer, Transfer-Encoding, Upgrade, Vary, Via and Warning. In fact, they have been growing at such a rate that the RFC mechanism cannot keep up, and RFC 4227 describes an IANA register for new extensions, with an *initial* list of 116 such header fields.

p. 167 Many new status codes have been devised: 100, 101, 203, 205, 206, 300 (one which says that there are multiple representations of the URL, and invites the user/browser to choose), 303, 305, 307, 405–417, 504, 505.

p. 167 In the example, note how the connection shifted from NVT ASCII (up to and including the blank line sent by the server) to the binary in the GIF. TCP itself, of course, does not care, since it merely transmits bytes. The client is told about this shift via the `Content-type` header. Note that, once we have done this `0xff` is no longer the NVT ASCII IAC escape character, so no interpretation of bytes is necessary or possible by the server or HTTP part of the browser, and the bytes get passed to the GIF renderer, which *does* interpret them.

p. 170 The four-connection behaviour of Netscape is deprecated by RFC 2616, which says (p. 31) that a single-user client SHOULD NOT maintain more than two connections to any server. Nevertheless, it is still very common in most brands of browser — selfishness rules over good behaviour!

p. 176 The HTTP/1.1 RFC (2616) describes persistent connections and says that implementations SHOULD implement them (p. 30 of the RFC). However, [15] shows that many sites do not implement these, often by explicitly turning them off in their implementation of a server that can support them. There are also problems with persistent connections, in that they stay idle for a while, having transmitted enough data that *cwin*

is large, and then suddenly spring back into life with a large burst of data: see section 4.1 of RFC5681.

general There is a survey[106] of "the average web page". However, this refers to 'page' as seen by the user, not necessarily 'page' as transmitted in a single HTTP GET. Such figures would be interesting. The final caveat is worth noting.

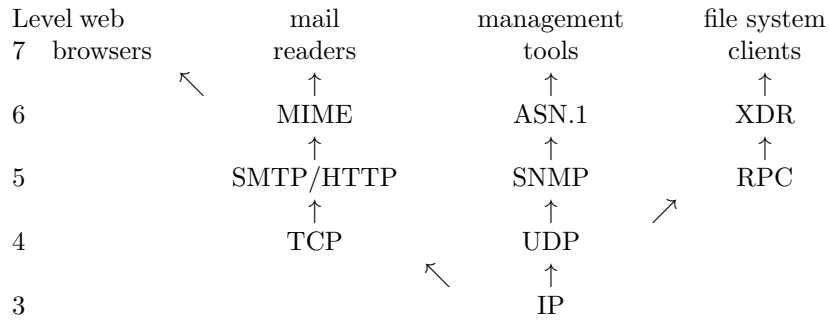
> However as web pages become more complex, each individual component adds size and HTTP requests to the total page payload.

Conclusions

1. IP dates back to RFC 791 (1981), and the last major revision to IP as it affected the IP layer on hosts (essentially the kernel as regards Unix machines) was sub-netting (RFC 950, 1985), and the new requirements were consolidated in RFC 1122 (1989). For some (but generally not stub) routers, the introduction of CIDR (RFC 1518, 1993) was another major change. RFC 1123 (1989) was the last major re-definition of the requirements for existing applications. Routers have to change faster than hosts, and the last list of requirements for routers was in RFC 1812, though updated by RFC 2644, which changed the default handling for directed broadcasts: RFC 1812 said that the default MUST be to permit te=hem, but, as 2644 observed "their use on the Internet backbone appears to be comprised entirely of malicious attacks on other networks.", and the default is now not to forward them.

   In computing terms, this is relatively stable, since the changes were engineered to be upwards-compatible, as far as possible. The change from IPv4 to IPv6 will be far more fundamental.

2. We have now seen much of the TCP/IP world, so we can review the layering structure we described in the notes to Figures 1.2/1.3.



[106] `http://www.optimizationweek.com/reviews/average-web-page`.

We note that there are now several different level-6 encodings. XDR was defined to solve the comparatively simple problem of transport of integers etc. across a binary protocol (RPC). MIME was invented to transfer structured (e.g. multi-part messages) binary data across a fundamentally 7-bit ASCII connection (SMTP). ASN.1 deals with more complex trees than either.

3. Layering is a very important conceptual model. However, we should note three important points.

   Implementation It is *not* necessary to implement each layer separately. Doing so tends to lead to unnecessary traversals of the data. A good implementation of TCp/IP/Ethernet should copy the data once from user to kernel space, computing the TCP checksum in the process, and then from kernel space onto the Ethernet, computing the Ethernet frame check sequence in the process.

   Interfacing We *may* need interfaces between the layers — ARP is one obvious example between the link and network layers.

   Violations These may be necessary for performance or security reasons, but should always be carefully reasoned, and preferably be "fail-safe". We have seen several so far.

   - Header compression in SLIP and PPP (p. 31). Here PPP is definitely better, since it negotaites compression, whereas a compressed SLIP talking to an uncompressed SLIP just generates incomprehension.
   - Slow start in TCP, which is mandated (RFC 1122) for non-local connections. JHD prefers to see this as slow start being mandated, with[107] an optional optimisation *not* to do it on local connections. This is also related to the choice of initial `cwin`, which SHOULD follow equation (1) for 2.5G/3G wireless connections (i.e. connections with such a link in them), and therefore logically has to happen all the time, since not only would it be a violation of layering, it is in practice impossible to detect the presence of such links at a TCP level.
   - Network Address Translators (Appendix B), though they logically work at the network layer, need to look at TCP/UDP port numbers to define their mapping, and may need to poke at other layers, e.g. for FTP.

4. We can see that different higher-level protocols (RFC model Layer 4) handle data formats in different ways.

   Telnet NVT ASCII throughout.

[107] At the time Stevens wrote. These days RFC 5681 mandates it for all connections, for reasons given in Appendix C.

FTP NVT ASCII on the control connection, whatever is spcified on the data connection.

SMTP NVT ASCII with MIME-encoded data in other forms, unless `8BITMIME` is in use.

HTTP NVT ASCII in the "control phase" (up to the blank line — see notes on p. 169), and MIME-based thereafter.

5. This also lets us contrast UDP with TCP. Most "long-distance" applications use TCP, with its sophisticated (some would say complicated) mechanisms for adjusting to the available bandwidth, and recovering from dropped packets. "Local" applications such as NFS, other RPC-based systems such as Sun's NIS[108], TFTP etc. use UDP. The DNS, which normally transfers a small amount of data but which can require to transfer much more, and NFS are now capable of using either, depending on the situation.

   There was an attempt to define a protocol[109] with the basic features of UDP but with reliability, however it never went beyond "experimental" status.

References

[1] Atkinson,R.J., Private Communication. E-mail `5B44E53A-4D39-4C31-96EA-D29698347932@extremenetworks.com`, 15 March 2007.

[2] Balakrishnan,H., Padmanabhan,V., Seshan,S., Stemm,M. & Katz,R. TCP Behavior of a Busy Web Server: Analysis and Improvements. Technical Report UCB/CSD-97-966, August 1997. `http://nms.lcs.mit.edu/~hari/papers/csd-97-966.ps`.

[3] Bennett,J.C.R., Partridge,C. & Shectman,N., Packet Reordering is not Pathological Network Behavior. *IEEE/ACM Trans. Networking* **7** (1999) pp. 789–798.

[4] Bradford,R.J. *The Art of Computer Networking.* Pearson, 2007. ISBN-13: 9780321306760.

[5] Brakmo,L. & Peterson,L. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE J. Selected Areas in Communication*, **13** (1995) pp. 1465–1480. `http://www.cs.arizona.edu/xkernel/www/people/brakmo-papers/abstracts.html`.

[108] NIS = Network Information Services. Sun used to call it "Yellow Pages" until a trademark law-suit was threatened by British Telecom. It provides shared user names/passwords etc. across a local area network, but has several security weaknesses.

[109] RDP = Reliable Datagram Protocol, described in RFC 1151.

[6] Cheswick.W.R. & Bellovin,S.M., *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley, 1994. `research.att.com/dist/internet_security/firewall.book/*`

[7] Crowcroft,J., Wakeman,J., Wang,Z. & Sirovica,D., Is Layering Harmful? *IEEE Network* **6** (1992) 1 pp. 20–24. The seven missing figures from this appear in issue 2 of the same volume.

[8] Fall,K. & Floyd,S., Simulation-based comparisons of Tahoe, Reno and SACK TCP. *Computer Communications Review*, July 1996. `ftp://ftp.ee.lbl.gov/papers/sacks.ps.Z`.

[9] Gibson,G.A., Nagle,D.F., Amiri,K., Chang,F.W., Feinberg,E., Gobioff,H., Lee,C., Ozceri,B., Riedel,E. & Rochberg,D., A Case for Network-Attached Secure Disks. CMU Technical Report CMU-CS-96-142, 26 September 1996. `http://www.pdl.cmu.edu/PDL-FTP/NASD/TR96-142.pdf`.

[10] Halabi,S. (with D. McPherson), Internet Routing Architectures, second edition. CISCO Press, 2000 [In UoB Library: 518.731 HAL].

[11] Jaiswal,S., Iannaccone,G., Diot,C., Kurose,J. and Towsley,D., Inferring TCP Connection Characteristics Through Passive Measurements. Proc. IEEE InfoComm 2004 vol. 3, pp. 1582–1592. Digital Object Identifier `10.1109/INFCOM.2004.1354571`.

[12] *Janet News* **1**(2007).

[13] Karrenberg,D., Twin hurdles that could trip internet upgrade plans. `http://resources.zdnet.co.uk/articles/comment/0,1000002985,39625225,00.htm`, 10 March 2009.

[14] Kent,C.A. & Mogul,J.C., Fragmentation Considered Harmful. *Computer Communications Review* **17** (1987) pp. 390–401.

[15] Krishnamurthy,B. & Arlitt,M., PRO-COW: Protocol Compliance on the Web. HP Labs Technical Report 1999-99, August 1999. `http://www.hpl.hp.com/techreports/1999/HPL-1999-99.html` `http://www.usenix.org/events/usits01/krishnamurthy.html`

[16] Medina,A., Allman,M. & Floyd,S., Measuring the Evolution of Transport Protocols in the Internet. *SIGCOMM Comput. Commun. Rev* 35(2005) 2 pp. 37-52.

[17] Odlyzko,A.M., Internet traffic growth: Sources and implications. Optical Transmission Systems and Equipment for WDM Networking II, B. B. Dingel, W. Weiershausen, A. K. Dutta, and K.-I. Sato, eds., Proc. SPIE, vol. 5247, 2003, pp. 1–15. `http://www.dtc.umn.edu/~odlyzko/doc/itcom.internet.growth.pdf`.

[18] Padhye,J. & Floyd,S., On Inferring TCP Behavior. To appear in Proc. SIGCOMM '01. http://www.aciri.org/tbit/.

[19] Padhye, J., Firoiu, V., Towsley, D. & J.Kurose, "TCP Throughput: A simple model and its empirical validation". SIGCOMM Symposium on Communications Architectures and Protocols, August 1998.

[20] Stallings,W., The New and Improved Internet Protocol. BYTE **21**(1996) 9 pp. 55–56.

[21] The literary executors of W.R. Stevens, http://www.kohala.com/start/tcpipiv1.html.

[22] Stone,J. & Partridge,C., When the CRC and TCP Checksum Disagree. Proc. ACM SIGCOMM 2000. http://www.acm.org/sigs/sigcomm/sigcomm2000/conf/paper/sigcomm2000-9-1.ps.gz.

[23] Wang,F., Mao,Z.M., Wang,J., Gao,L. & Bush,R., A measurement study on the impact of routing events on end-to-end internet path performance. Proc. SIGCOMM '06, ACM Press, New York, 2006, pp. 375–386. DOI http://doi.acm.org/10.1145/1159913.1159956.

[24] The Wikipedia Foundation, *Wikipedia, the free encyclopedia.* www.wikipedia.org.

[25] Xu,L., Harfoush,K. & Rhee,I., Binary Increase Congenstion Control for Fast Long Distance Networks. http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/index_files/Page444.htm.And Proc INFOCOM 2004.

A A Backwards Compatibility Problem

This describes an example which actually occurred at the University, showing how evolution in the IP standards can cause a substantial backwards-compatibility problem. It should be read after Chapter 4 of Stevens, as an illustration of the material so far. See also the notes to pages 118 and 123, which illustrate several infelicities due to the fact that sub-netting was added after other ICMP features were defined.

When the University of Bath installed its first Ethernet (in mathematical Sciences in 1986), the University had, thanks to a far-sighted member of the Computing Service, already obtained a Class B address (138.38.x.y) — such an allocation would be impossible today. It was already clear that a certain allocation policy would be needed, and RFC 950, describing sub-netting, had just been published, even though it was not widely supported, and Mathematical Sciences was allocated 138.38.96.x (and up to 138.38.103.x). So the three Sun machines in Mathematical Sciences formed a nice little private network,

unconnected to anything else[110]. By 1988, the Sun operating system supported sub-netting.

Mathematical Sciences also had some High-Level Hardware Orion machines, and they soon supported Ethernet and TCP/IP, but not sub-netting, and they were added to this Ethernet. As long as this network remained separate, there was no problem, since the fact that the Orions thought that they were machines `96.10` etc. on Class B network `138.38` and the Suns thought that they were machines `3` etc on sub-net `138.38.96` did not matter, since the same IP addresses were used, and no routing was taking place.

Once the Sun `138.38.96.1` became multi-homed, with a second Ethernet connection named `138.38.32.254` on the campus backbone (then Ethernet), there was a problem with the Orions. The Sun `138.38.96.3`, with a sub-net mask of `ffffff00`, knew that `138.38.32.1` was on a different subnet to it, so it routed to it via its route default, i.e. `138.38.96.1`, but the Orion `138.38.96.10`, with no sub-netting, i.e. a mask of `ffff0000`, thought that the two were on the same net, so would not route, but ARP for `138.38.32.1`.

This problem was solved by "proxy ARPing" (p. 60), in that `138.38.96.3` was instructed to reply to ARP requests for `138.38.32.1` (and other machines) by giving the Ethernet address of `138.38.96.1`. `138.38.96.10` was then satisfied, and sent the packet for `138.38.32.1` out on the Mathematical Sciences Ethernet, with the Ethernet address of `138.38.96.1`, but believing that this was the Ethernet address of the destination. On arrival at `138.38.96.1`, there was no way to tell this packet from a correctly routed packet (e.g. from `138.38.96.3`), and it was routed on to `138.38.32.1`. A reply packet was routed by `138.38.32.1`, which understood sub-netting, to its non-default router `138.38.32.254` for the `138.38.96.x` subnet, and passed on to `138.38.96.10`.

There were some drawbacks to this scheme. The ARP table on `138.38.96.3` had to be added to each time the system booted, so a list of `arp -s pub` (see p. 63) commands was added to the start-up script. Every time the Orions wanted to communicate with a new machine not on the sub-net, this list had to be extended. Also, `138.38.96.3` became an additional point of failure, since if it was down, the Orions could not communicate with any machine on the rest of `138.38.x.y` (outside `138.38.96.y`) unless the entry was already in the ARP cache[111]. Thus a failure of `138.38.96.3` was manifest as a slowly-growing series of complaints of the form "my Orion won't talk to ...". An attempt was made to solve this by using `138.38.96.1` as the proxy server, but this failed, as it also answered ARP requests for `138.38.32.1` on the `138.38.32.y` sub-net. `138.38.96.2` was also used as a proxy server, with another manually-maintained list of ARP commands in its start-up. Most of the time this worked, but occasionally people would forget to add machines to both lists. This was not

[110]Initially. Soon an application gateway was installed to convert e-mail from SMTP-based TCP/IP mail to "Yellow Book" X.25-based e-mail (and *vice versa*) for communication with JANET, which at that point was running X.25 rather than TCP/IP.

[111]In theory, and as mandated by RFC 1122, such an entry should time out after 20 minutes, but, as pointed out on page 60, such timeouts tend to be, and on the Orions were, restarted each time the entry was used.

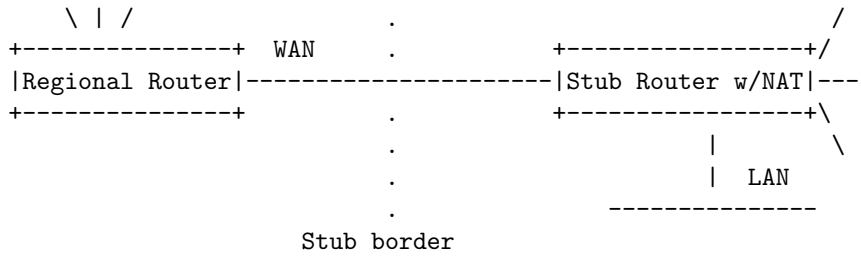
a problem as long as both machines were up, but if one failed, a few connectivity problems would occur, as an Orion tried to contact a machine for which only the failed machine was answering.

RFC 1122 (October 1989) made sub-netting support compulsory, but in fact High-Level Hardware never upgraded their system, and this hack continued to run until the Orion kernels were patched to give them the correct submasks in 1993, and then they were turned off in 1995.

B Network Address Translators

This appendix deals with network addres translation *within* IPv4. Translation between IPv4 and IPv6 is a separate, and complex, subject. See RFC 4966 for some of the complications.

These have become wide-spread in the Internet since Stevens wrote his book. They generally translate between "private" Internet addresses (see RFC 1918 and the note to p. 8), as used by most Internet Serive Providers, including the University's ResNet, and "public" (i.e. non-RFC 1918) addresses, though they don't have to. This form of Network Address Translation is described in RFC 3022. The following diagram is taken from RFC 2663, a dictionary of Network Address Translation (NAT) terminology.



The LAN has an RFC 1918 address, which we will call 192.168.1.0. The stub router has an address in this range for its LAN connection (clearly) but a public address on the WAN, say 1.2.3.4. When a host on the LAN (say 192.168.1.1) wishes to browse `www.bath.ac.uk` (138.38.32.5), its initial TCP SYN packet (see chapter 17) will have some source port number (say 1234) and destination port 80. This packet cannot be sent out on the WAN, since 192.168.1.1 is a private address.

In the simple case, the NAT router changes the IP source address to 1.2.3.4 (updating both the IP header checksum (p. 36) and the TCP[112] checksum, the latter since the IP address is in the pseudo-header — see section 11.3). The NAT router also remembers that the TCP quad (see p. 255)

138.38.32.5 80 1.2.3.4 1234

[112]It should be noted that the TCP checksum need not be recomputed (which might involve re-assembling fragments): it can merely be adujusted, since it is a linear function of the data — see section 4.2 of RFC 3022. The same applies to UDP checksums, except that `0` has to be treated as a special case.

corresponds to the TCP pseudo-connection

138.38.32.5 80 192.168.1.1 1234.

Packets that arrive at 1.2.3.4 (TCP port 1234) are forwarded to 192.168.1.1 port 1234, with the destination address and checksums altered.

If TCP port 1234 on 1.2.3.4 is already in use, then either the connection fails (Basic NAT — highly undesirable) or the NAT process[113] has to assign another port number (say 2468), and it then has to remember that

138.38.32.5 80 1.2.3.4 2468

corresponds to the TCP pseudo-connection

138.38.32.5 80 192.168.1.1 1234.

Packets that arrive at 1.2.3.4 (TCP port 2468) are forwarded to 192.168.1.1 port 1234, with the destination address, port number and checksums altered.

A simplistic NAT as described above will work with many TCP and UDP protocols, but not all. A notorious problem is FTP (chapter 27) since the FTP control connection sends messages containing the IP addresses and port numbers for the data channel (such protocols are known as "bundled session applications" in RFC 3027). Hence if the NAT router is to support FTP, it most be "FTP-aware" — typically via an Application-Layer Gateway (ALG) that understands the FTP protocol. Contrary to the usual principle that fragmentation and TCP packaging are end-to-end procedures, the NAT will have to assemble fragments, and even TCP segments, to see the complete PORT command: several translators do (did?) not do this, with bizarre "sometimes does not work" effects. A fuller description, and a list of other protocols with NAT problems, is in RFC 3027.

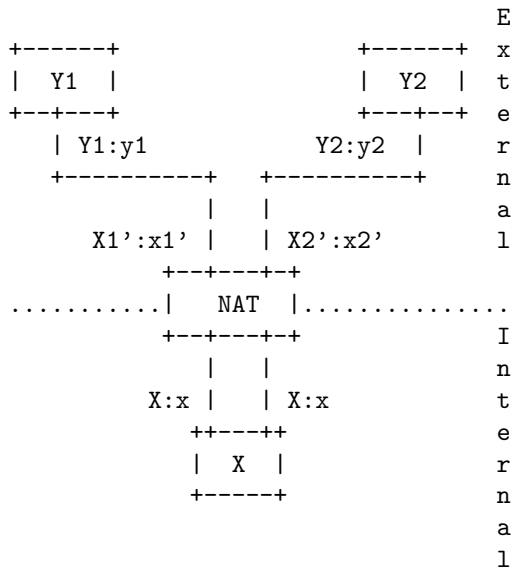
X windows (Stevens Chapter 30) is also a complication with NAT because, although based on a "client–server model", the 'server' (see note after Stevens' Figure 30.1) is actually the user's only machine. Hence applications runnng outside the NAT firewall will not be able to connect back to the X server running on the user's computer, which has done a *passive* open of port 6000+n. SSH solves this problem by creating a virtual X-windoes server at `localhost:17` (for example), and then tunneling X traffic back over the SSH connection.

ICMP is also complicated. To quote RFC 2663:

> All ICMP error messages (with the exception of [the] Redirect message type) will need to be modified, when passed through NAT. The ICMP error message types needing NAT modification would include Destination- Unreachable, Source-Quench, Time-Exceeded and Parameter-Problem. NAT should not attempt to modify a Redirect message type.

[113] Known as NAPT: Network Address/Port Translator.

Figure 1: Port Mapping and NAT



> Changes to ICMP error message will include changes to the original IP packet (or portions thereof) embedded in the payload of the ICMP error message. In order for NAT to be completely transparent to end hosts, the IP address of the IP header embedded in the payload of the ICMP packet must be modified, the checksum field of the same IP header must correspondingly be modified, and the accompanying transport header. The ICMP header checksum must also be modified to reflect changes made to the IP and transport headers in the payload. Furthermore, the normal IP header must also be modified.

RFC 3235 gives guidelines on how to design applications that are NAT-friendly. Conversely, RFC 4787 gives requirements on NATs that make them UDP-friendly. The following diagram (Figure 1) comes from RFC 4787. This RFC specifies that the NAT must perform "Endpoint-Independent Mapping", which it defines as follows.

> The NAT reuses the port mapping for subsequent packets sent from the same internal IP address and port (X:x) to any external IP address and port. Specifically, X1':x1' equals X2':x2' for all values of Y2:y2.

NATs work reasonably well for client/sever architectures, but less well for peer-to-peer networks, where the 'session' may be originated outside the NAT. RFC 5128 discussed this issue, and says the following.

> Currently deployed NAT devices are designed primarily around the client/server paradigm, in which relatively anonymous client machines inside a private network initiate connections to public servers with stable IP addresses and DNS names. NAT devices encountered en route provide dynamic address assignment for the client machines. The illusion of anonymity (private IP addresses) and inaccessibility of the internal hosts behind a NAT device is not a problem for applications such as Web browsers, which only need to initiate outgoing connections.
>
> ... In the peer-to-peer paradigm, Internet hosts that would normally be considered "clients" not only initiate sessions to peer nodes, but also accept sessions initiated by peer nodes. The initiator and the responder might lie behind different NAT devices with neither endpoint having a permanent IP address or other form of public network presence.

RFC 3489 proposed to make NATs more controllable, by introducing STUN — Simple Traversal of User datagram protocol (UDP) through Network address translators (NATs), but this has been replaced by RFC 5389, which cunningly re-uses the acronym!

> The protocol defined in this specification, Session Traversal Utilities for NAT, provides a tool for dealing with NATs. It provides a means for an endpoint to determine the IP address and port allocated by a NAT that corresponds to its private IP address and port. It also provides a way for an endpoint to keep a NAT binding alive.

C Allman's e-mail

In response to JHD's query, Mark Allman, one of the authors of RFC 5681 (and many others) sent the following message, with permission to discuss it with my students.

```
Date: Wed, 16 Dec 2009 00:03:09 -0500
From: Mark Allman <mallman@icir.org>
To: Professor James Davenport <jhd@cs.bath.ac.uk>
Subject: Re: RFC 5681
```

"A few notes ...

- The notion of not using slow start when two endpoints are on the same local network has never been standardized.
- Stevens' books are BSD-centric — which was more than fine when they came out, but is now pretty outdated. And, BSD did in fact have some code that eliminated slow start when two hosts were on the same local network. My hunch is the current versions of the main BSD systems

(Free, Net and Open) no longer have this notion of local hosts not using slow start. But, I wouldn't swear to it.

- Ultimately, not using slow start locally has been viewed as a bad idea for a couple of reasons.

  First, how do we define "locally"? Are two hosts connected via a hub local? A switch? A campus router? These intermediary devices do have queues and can get busy and so contention can occur.

  Second, even if we can define something as local how do we figure out if our peer meets the definition? How do we know some "local" host isn't really on a ratty wireless link? Or, a dialup? (This latter of course going away ...) Do we make the determination based on IP address? There are no good heuristics for doing so in a global fashion. So, does this basically come down to having a host locality table on each machine?

  Third, even if we can agree on a definition of "local" and we can somehow determine a peer is "local" then what good does it do? Even fat enterprise networks have pretty modest delay*bandwidth. So, the window you need to build is not great. Further, it happens quite fast.

I hope those shed some light on the subject."

With respect to the last point "it happens quite fast", JHD observed, and Allman agreed: "Especially now that most hosts are starting cwin off at more than one segment, so that the 'delayed ACK' doesn't get in the way."

D A note on terminology

See table 3. So 0 or more bytes = 1 segment = 1 packet = 1 or more datagrams.

Table 3: Terminology for messages

Layer	Data is in ?
Application	Whatever the application wants
⇕	Byte stream protocol
TCP	Internal buffers
⇕	segments
IP	packets ⇔ Fragments
⇕	If possible ↙
Link Layer	datagrams

E Changes

This appendix details the significant[114] changes made since the author started keeping a change log. Only the ones since 2.10.2009 are printed here, though,

[114] Excluding changes to wording, or updates to RFC numbers unless the text also changes.

to avoid confusion.

E.1 2/10/2007 –

2/10/2009 Call this version 10, and archive off the old change log. Fix the URL for the e-book.

13/10/2009 Update p. 26 to point out that ADSL normally carries PPP.

19/10/2009 Added (thanks to Matthew Bewers) the information in note 41 (page 19), and information on the transition to 32-bit AS numbers.

22/11/2009 Upgraded start of SMTP Chapter to point to the UTF8 work.

30/11/2009 Upgraded start of Chapter 26 to mention where SSH fits in.

12/12/2009 Updated chapter 13 to say more about the take=up of multicast.

14/12/2009 Updated HTTP section.

16/12/2009 Added more on slow start (including Appendix C following e-mail exchange with Allman.

17/12/2009 Added brief description of X and NAT to Appendix B.