Prologue: The Internet

(FIXME: How should I handle references in this section?)
A younger second cousin of mine, Tom, once asked me about how life was before the Internet. Let us meet again in a few years, and continue that conversation.

History

TOM: Could you use the Internet when you went to school?

NIELS: No, I got my first account an a machine with a real Internet connection some year after I entered university, October 24, 1992.

TOM: So that was the first time you used email?

NIELS: Not quite, I got a student account at the CS department some year earlier. The computer system for the students was not directly connected to the Internet, presumably for the protection of the Internet, but one could send and receive email, both locally, to other students and teachers, and to the outside world.

Since email was the first big application for the network, it was common with gateways between Internet email, and other systems. I remember having a summer job in the early 1990s at a telecommunications company. We had email, but no direct connectivity to the Internet. It was not possible to connect directly to FTP servers around the world, but some file archives offered an FTP by email service, which we used.

TOM: So how did people exchange files before the Internet?

NIELS: Computer communication was fairly obscure; some people used modems to dial in to bulletin board systems, and hacker groups would exchange data on floppy disks sent via postal mail. It was common that computer magazines and books published program listings, but to try the program, you had to first type it into your computer by hand.

TOM: When did Sweden get connected to the Internet?

NIELS: I think the first network that was connected was the university network SUNET. A 56 kbit/s satellite link was set up between the Nordic university networks, and the John von Neumann Center at Princeton, New Jersey. Due to a delay on the

American side of the link, Sweden missed the outbreak of the first Internet Worm, released by Morris on November 2, 1988[1]. The satellite link was operational a week later[2].

Tom: It's hard imagine how it was before the Internet, I use email and instant messaging constantly to keep in touch with friends.

Niels: The phone system is of course many decades older than the Internet, so that's what people were using to keep in touch. Today, the Internet is used for everything, not just email and file transfer, but phone calls, radio, film distribution, etc. Before this convergence, we had a couple of large but separate communication systems, starting with radio broadcast and the (wired) telephone network, then television (different systems for terrestrial, satellite and cable), and finally mobile telephone networks.

Circuit switching vs packet switching

Tom: How does that work, then? What makes the Internet different from the telephone network?

Niels: The telephone systems were traditionally circuit switched. Originally, when making a phone call, you had an operator manually patch together a physical electrical circuit between the two telephones. Later on, this manual patching was replaced by automatic electromechanical switches, reacting to each digit of the phone number as you dialed, and then by digital systems.

Tom: This seems easy enough for local calls; there's a cord from each telephone to a huge switch board at the telephone switch, and to connect two telephones, you patch the corresponding two cords together. But how could you patch together a remote call?

Niels: For the duration of a call, you need a path reserved through the network, from one switch to the next. Between neighboring switches, your call might use a separate cord, a frequency band in a cable using frequency division multiplex, or certain time slots in digital system with time division. This means that every switch on the path must be aware of your call, and change their state when the call is set up or teared down.

Tom: I've heard that the Internet is "packet switched", what does that really mean?

Niels: Too see the difference between circuit switching and packet switching, say you want to transport fuel from the harbor at Värtahamnen to Arlanda airport. One alternative is to construct a pipeline between these two points. A different way to solve the problem is to load the fuel onto trucks, and let each truck find its way

[1] The worm infected VAX computers and Sun workstations. It is guessed to have infected 6 000 out of the roughly 60 000 computers connected to the Internet at the time.

[2] Kaarina Lehtisalo, *The History of* NORDU*net—Twenty-five years of networking cooperation in the Nordic countries*, `http://www.nordu.net/history/book.html`

from the harbor to the airport independently. Pipelines are sure useful in certain circumstances, but using a general purpose road network is more flexible. You can freely mix vehicles of different sizes and with different cargo, while you can't use the same pipeline system to transport petrol and beer.

Routing

TOM: But unlike trucks, packets have no drivers?

NIELS: There have actually been some research on "active networking", where each packet contains the intelligence needed for it to find its way through the network[3]. But in the Internet, packets are not smart enough to find their way by themselves.

TOM: So when I send a packet over the network, how does the network know where to send it?

NIELS: Packets are identified by its source and destination address. These are the IP-addresses, 32 bits, usually written as, e.g., `130.237.43.158`[4]. Have you ever manually configured the IP address of your computer?

TOM: I think I had to to that sometimes, but that was many years ago, so I don't quite remember. Nowadays it's always automatic. But I remember having to type in some numbers for the address, and also for something called "netmask" and "default gateway", whatever that meant.

NIELS: That's the first step of routing. Your computer needs to know which addresses belong to computers on the same local network as you, and that's what the netmask is; it gives the size of the prefix identifying your local network. On the KTH network, my address is `130.237.43.158`, and the netmask is `255.255.0.0`. That tells my computer that all addresses starting with `130.237.` belong to computers on the same local network. This is the network prefix, in this case, it's 16 bits long.

When I send a packet, the IP stack first uses the network mask to check if the destination address is on the same local network. If it is, the packet can be sent to destination directly, without passing through any intermediate routers.

TOM: So that's what the netmask is for. And the "default gateway"?

NIELS: That's the IP address of a router, which must be located on the local network, that routes packets to and from the outside world. All packets sent to destinations that are not on your local network, are sent to the default gateway, which then passes them on to other routers. It's called "default", because it is possible to set up more complex routing rules, and the default gateway is used when no other rule match.

TOM: So how does the gateway know what to do with my packets?

[3]This goes back at least to the Softnet project in the early 1980s, developed by Jens Zander and Robert Forchheimer at Linköping University.

[4]For IP version 6, the addresses are 128 bits.

Niels: The routing system is more or less automatic. Each router has a couple of in- and outgoing links. It keeps track of which parts of the network are connected, directly or indirectly, to each link. Within an organization, routes can be configured manually, or via protocols such as ospf (Open Shortest Path First), which exchanges routing information with neighboring routers. The ospf protocol finds the shortest, or lowest cost, path between each pair of routers. Between organizations, routing depends not only on network topology, but also on private service agreements between Internet Service Providers (isp), basically, on who is paying whom. This routing information is exchanged between routers, taking policy into account, using bgp (Border Gateway Protocol).

No matter which protocol or method is used to configure the routing, the end result is a *routing table*, a large lists of network prefixes, and for each prefix, the outgoing link and neighbor router for that prefix. The destination address of each incoming packet is looked up in the routing table to find the longest matching prefix, and then the packet is transmitted to the corresponding router.

Routers close to the network edge often have a default gateway; a neighboring router which is more central, and which is used for all packets not matching any other rule. Routers in the core network don't have any default gateway, and this core is sometimes called the "default-free zone".

TCP/IP

Tom: When people talk about tcp/IP, what does that mean? Is it just a more complicated way to say "Internet"?

Niels: Right, it's more or less the same thing. At least ip, which simply stands for the Internet Protocol.

If packet switching is analogous to a general purpose road network, then the ip protocol is the standardized freight container for inter model transport. Before 1970, container transport suffered from several national or company-specific standards that did not inter-operate. Inter-modal freight transport took off after the iso standardization of container sizes around 1970, transforming the world economy in unexpected ways[5]

Tom: So there were other packet switching protocols before ip?

Niels: Sure. I'm not very familiar with any of them, but there were many proprietary networking protocols, e.g., sna from ibm and decnet from Digital Equipment Corporation. The x.25 protocol was standardized by ccitt in 1976, primarily for use in networks run telephone companies. tcp/ip were developed during the 1970s, by the Defense Advanced Research Projects Agency (darpa) in the United States, as communication protocols for interconnecting different networks. iso standards

[5] Marc Levinson, *The Box—How the shipping container made the world smaller and the world economy bigger.*

for networking were developed during the 1980s, accepting x.25, but not TCP/IP, as a part of the Open Systems Interconnection (OSI) standard.

TOM: But TCP/IP is still not an ISO standard? So what happened with that?

NIELS: When NORDUnet were implementing wide area networking in the late 1980s, the only part of ISO networking standards that was in real use was x.25. And x.25 had both technical and economical problems[6]. So the choice was between the OSI standards, which were not mature, various proprietary networking technologies, and TCP/IP[7]. I imagine the considerations were similar at other organizations planning or building networks.

The first version of NORDUnet was a wide area Ethernet that supported all of IP, DECnet and x.25. Some year later, the network was upgraded to a pure IP network, but still with support for DECnet and x.25 on top of IP.

Architecture

TOM: So IP "won" the protocol war. Was that just due to good timing, or what is it that makes IP different?

NIELS: As the name implies, IP was designed for the interconnection of heterogeneous networks. I think it managed to find the right, close to minimal, interfaces for doing that. Transmission of IP packets is straightforward to implement on top of virtually any link technology[8]. And on top of IP, end hosts can implement more sophisticated communication and application protocols. The transmission control protocol (TCP) provides a bidirectional reliable connection between end hosts. Most application protocols you use are specified as working on top of a bidirectional data stream, and are used on top of TCP. Some that don't need a connection, or don't need reliable transmission, can work with IP directly[9]

TOM: Isn't that the obvious way to design a networking protocol?

NIELS: It may seem obvious today. But for earlier networks, the applications and the link technology was more tightly coupled. E.g., in the telephone system, all links and connections were of fixed bandwidth (around 4 kHz for analog transmission and 60 kbit for digital transmission), chosen for supporting a voice service of reasonable quality. And there were few applications besides voice.

[6]Quoting Hans Wallberg, manager of SUNET, "x.25 became terribly expensive when usage grew, since the cost was based on the amount of data transferred. The performance was also too poor. Even if you had a 64 kbit/s connection you never got more than 2400 bit/s due to all overhead."

[7]The so called "protocol wars" are described in some more detail in Kaarina Lehtisalo's book, which is one of my main sources for the history of the early years.

[8]An experimental specification for the transmission of IP packets with carrier pigeons, RFC 1149, was published on April Fool's day 1990. A decade later, this specification was implemented for the first time, in a cooperation between the Linux and BSD User Group in Bergen, and Vesta Brevdueforening.

[9]Actually, these protocols usually don't work with IP directly, but with the UDP protocol, which is a very thin layer on top of the IP packet delivery service.

Tom: But now your comparing computer network to the telephone system that is may decades older. That's cheating. What if you compare IP to DECnet or to the OSI protocols?

Niels: I don't know many details of DECnet works, but one important difference is that IP is an open, non-proprietary standard. You can buy devices that speak IP, network equipment, computers, various gadgets, from a large number of different companies, and they will work together. As for OSI, those protocols were complex committee products, where everybody's favorite feature was included. For example, the OSI model specifies two more layers than in TCP/IP, and the OSI network layer supports both datagram-oriented and connection-oriented services. In TCP/IP, the split between IP and TCP means that the network layer need not be aware of connections. The abstraction of a connection between two hosts is created by the TCP-implementation in the end nodes. This is an example of the end-to-end principle.

Tom: The "end-to-end principle"? What in the world is that?

Niels: The end-to-end principle states that in a communications system, as much as possible of the protocol logic and state should be located in the communication end-points. This is important for the scalability of the system. In practice, one consequence is that a TCP connection can survive a reboot of routers along the path, as well as routing changes.

Tom: Quite different from the old telephone system.

Niels: In the telephone system, the telephone devices are simple, and all the complexity is in the network. In the Internet, the network is stupid, and the end-points have to be quite complex[10].

Congestion control

Tom: When driving, I often get stuck in the traffic, waiting and waiting in some queue. Do packets get stuck too, on the Internet?

Niels: That's congestion: When there is more traffic than the network can handle. In the Internet, each router has some incoming and some outgoing links of limited capacity. For example, if a router has two incoming links where 80 Mbit of packets arrive every second, and all packets are routed to the same outgoing link, with a capacity of only 100 Mbit/s, then that router is overloaded. Routers buffer packets at the outgoing link, so when the router is overloaded, packets are queued up in that buffer. But unlike traffic queues, the buffer size, and hence the queue size, is limited. When the limit is reached, arriving packets are not added to the queue, they are thrown away or "dropped onto the floor".

[10] But not as complex as one might think. Adam Dunkel's TCP stack `uip` supports 8-bit micro controllers, and needs about 5 Kbyte for code and a few hundred bytes of RAM, depending on the application.

Tom: Not quite like road traffic then.

Niels: No, it's as if there were trapdoors in the roads, located some 100 m before each intersection. The trapdoor leads to a bottomless hole, and they open automatically whenever there's a queue all the way from the trapdoor to the intersection.

Another important difference compared to the road network is the traffic pattern. The bulk of the traffic on the Internet is transfer of fairly large files. A typical file will be divided into somewhere from a hundred packets for a moderate size image, to several thousands of packets for larger files. All these packets have the same source and destination addresses.

Tom: So when my computer sends a file, it transmits a thousand packets straight away?

Niels: With the earliest versions of TCP, you might almost have done that. The TCP protocol included flow control from the start; this mechanism lets the receiver tell the sender how much data the receiver can handle, and the sender must not send more than that. In road traffic analogy, you tell the sender that you have free parking space for only ten trucks, and then the sender will load at most ten trucks and send them your way. When the trucks have arrived, been unloaded, and left again, you tell the receiver that you have new parking space available.

Tom: So if the receiver has space for a thousand packets, I send that?

Niels: That could cause problems for the network. As soon as the capacity of the computers connected to the network outgrow the capacity of the routers, the network suffered "congestion collapse", with drastically reduced performance.[11]. It became clear that flow control was not sufficient. It prevents overloading of the receiver, but not overloading of intermediate routers. Congestion control is needed too.

Tom: So how does that work?

Niels: The sender maintains a limit, called the *congestion windows*, on the amount of packets in-flight in the network, i.e., packets that have been sent but not yet acknowledged. With a fixed limit, packet with new data would be sent only in response to an acknowledgement of an older packet.

Then TCP also have rules for the adjustment of this window size: When the connection is new, the window size is increased by one packet for each received ACK, meaning that you can send two new packets for each received ACK. When a packet is lost, meaning that the network is getting congested, the window size is cut in half, followed by an increase of one packet per roundtrip time.

This style of congestion control was introduced in TCP around 1988, and it seems to work fairly well.

(FIXME: Say anything more about aimd? How to finish.)

Tom: Hmm. I think I get the idea, but I have to think it over to understand why it works. One other thing: You have been talking about IP and IP-addresses. But you

[11] **(FIXME: Quote Konstantin Avrachenkov?)** See RFC 896 for a contemporary source

never use IP-addresses, do you? At least all addresses I use are human-readable, e.g., `wikipedia.org`. Are they related in some way?

Niels: That's the domain name system (DNS), which is a huge distributed database. But it's getting late, so if you'd like me to try to explain how that works, let's do that over dinner.