Data Communications Fundamentals: Data Transmission and Coding

Cristian S. Calude Clark Thomborson

July 2010
Version 1.1: 24 July 2010, to use "k" rather than "K" as the SI prefix for kilo.

Thanks to

Nevil Brownlee and Ulrich Speidel for stimulating discussions and critical comments.

Goals for this week

- "C" students are fluent with basic terms and concepts in data transmission and coding. They can accurately answer simple questions, using appropriate terminology, when given technical information. They can perform simple analyses, if a similar problem has been solved in a lecture, in a required reading from the textbook, or in a homework assignment.
- "B" students are conversant with the basic theories of data transmission and coding: they can derive non-trivial conclusions from factual information.
- "A" students are fluent with the basic theories of data transmission and coding: they can perform novel analyses when presented with relevant information.

References

1. B. A. Forouzan. *Data Communications and Networking*, McGraw Hill, 4th edition, New York, 2007.
2. W. A. Shay. *Understanding Data Communications and Networks*, 3rd edition, Brooks/Cole, Pacific Grove, CA, 2004. **(course textbok)**

Pictures

All pictures included and not explicitly attributed have been taken from the instructor's documents accompanying Forouzan and Shay textbooks.

Factors determining data transmission

- cost of a connection
- amount of information transmitted per unit of time (bit rate)
- immunity to outside interference (noise)
- security (susceptibility to unauthorised "listening", modification, interruption, or channel usage)
- logistics (organising the wiring, power, and other physical requirements of a data connection)
- mobility (moving the station)

Analog and digital signals

Connected devices have to "understand" each other to be able to communicate.

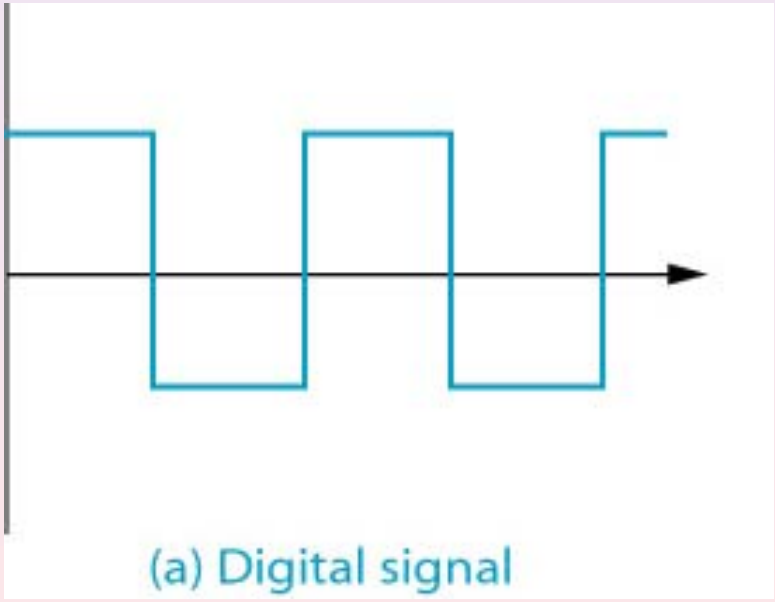
Communication standards assure that communicating devices represent and send information in a "compatible way".

There are two types of ways to transmit data:

- via **digital signals**, which can be represented either electronically (by sequences of specified voltage levels) or optically,
- via **analog signals**, which are formed by continuously varying voltage levels.

Digital signals 1

Digital signals are graphically represented as a square wave: the horizontal axis represents time and the vertical axis represents the voltage level.



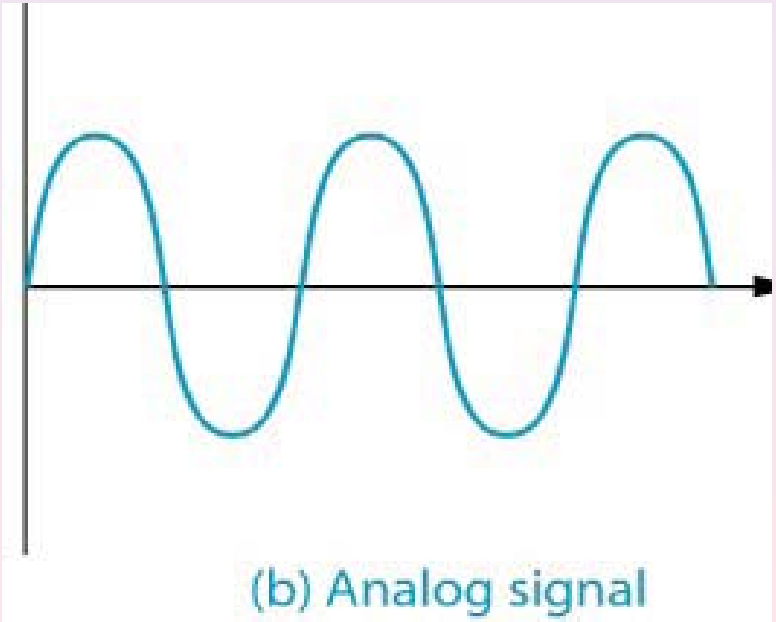
Digital signals 2

The alternating high and low voltage levels may be symbolically represented by 0s and 1s. This is the simplest way to represent a binary string (bit-string).

Each 0 or 1 is called a **bit**. Various codes combine bits to represent information stored in a computer.

Analog signals

PCs often communicate via modems over telephone lines using **analog signals** which are formed by continuously varying voltage levels:



How signals travel?

There are three types of transmission media, each with many variations:

- *conductive metal*, like copper or iron, that carries both digital and analog signals; coaxial cable and twisted wire pairs are examples,
- *transparent glass strand or optical fibre* that transmits data using light waves,
- *no physical connection* that transmits data using electromagnetic waves (as those used in TV or radio broadcast).

Costs

There are two forms of costs:

- costs of wires, cables, devices, etc.,
- number of bits transmitted per unit of time.

Bytes 1

For example, the storage capacity can be expressed in:

- B (bytes, 1 B = 8 b (bits)),
- kB (10^3 bytes),
- MB (10^6 bytes),
- GB (10^9 bytes),
- TB (10^{12} bytes),
- ...

Bytes 2

More about quantities of bytes

A **kibibyte** (**ki**lo **bi**nary **byte**) is a unit of information established by the International Electrotechnical Commission (2000).

Name (SI symbol)	Standard SI	Name (binary symbol)	Value
kilobyte (kB)	$10^3 = 1000^1$	**kibibyte (KiB)**	2^{10}
megabyte (MB)	$10^6 = 1000^2$	mebibyte (MiB)	2^{20}
gigabyte (GB)	$10^9 = 1000^3$	gibibyte (GiB)	2^{30}
terabyte (TB)	$10^{12} = 1000^4$	tebibyte (TiB)	2^{40}
petabyte (PB)	$10^{15} = 1000^5$	pebibyte (PiB)	2^{50}
exabyte (EB)	$10^{18} = 1000^6$	exbibyte (EiB)	2^{60}
zettabyte (ZB)	$10^{21} = 1000^7$	zebibyte (ZiB)	2^{70}
yottabyte (YB)	$10^{24} = 1000^8$	yobibyte (YiB)	2^{80}

http://www.bipm.org/utils/common/pdf/si_brochure_8_en.pdf, p. 121.

Bit rate

The **bit rate** is the number of bits transmitted per unit of time. The typical unit is *bits per second (b/s)*.

- b/s (bits per second),
- kb/s (10^3 bits per second),
- Mb/s (10^6 bits per second),
- Gb/s (10^9 bits per second),
- Tb/s (10^{12} bits per second).

Depending on the medium and the application, bit rates vary from a few hundred b/s to gigabits per second and pushing into terabits per second.

Note: "Kb" and "KB" are ambiguous. Some authors use "K" to denote 2^{10}, and some authors are careless about the capitalisation of the SI prefix "k".

Bit rates vary according to:

- downstream vs. upstream,
- the time of day — usually between 4pm to midnight speeds are likely to be slower,
- the web-sites you visit — some web-sites limit the speed at which they send out information,
- how many computers share the connection,
- the application(s) used,
- viruses and spyware on your computer.

Bandwidth and latency 1

Broadband is associated with high-speed connection, but speed is determined by various factors, among them bandwidth and latency.

- Bandwidth describes how much data can be sent over the network.
- Latency is the elapsed time for a single byte (or packet) to travel from one host to another. There are propagation, transmission and processing delays.
- Errors appear due to interference, busy routers (which drop packets), and link failures.

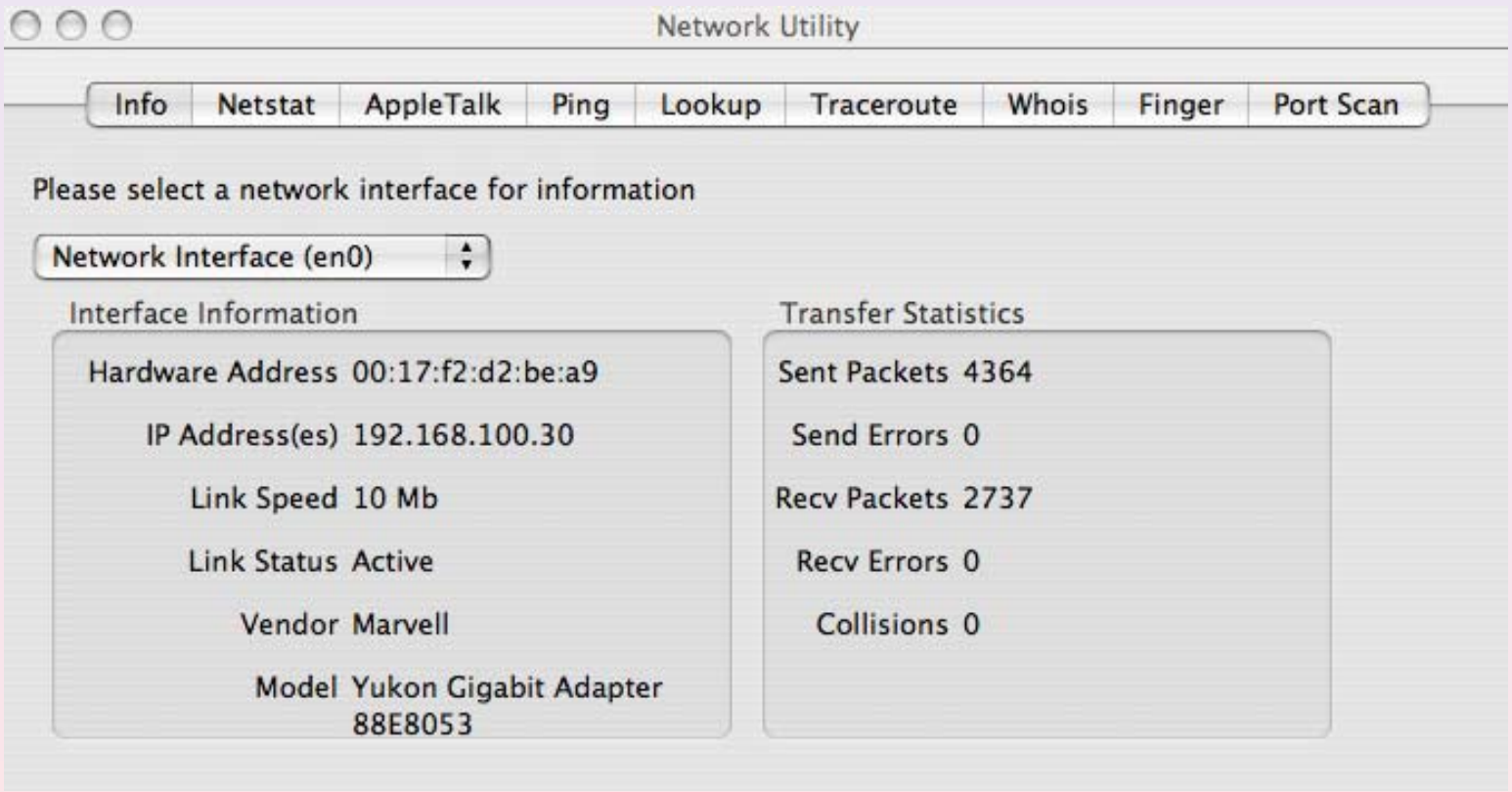
Bandwidth is limited to the poorest link, but latency and network errors are cumulative. The farther data has to travel between hosts, the more traffic it must compete with, and the more resources it uses.

Bandwidth and latency 2

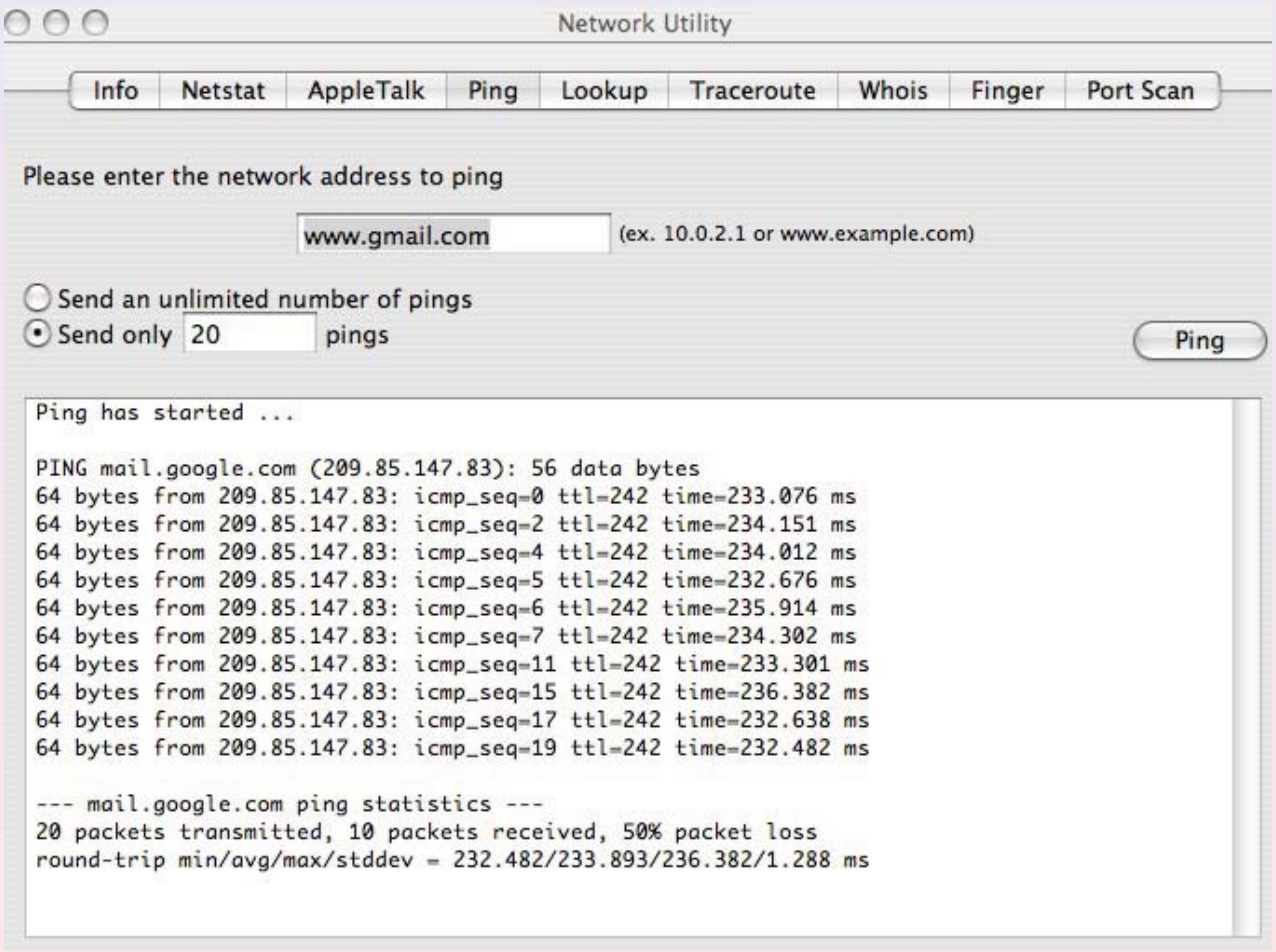
	Bandwidth (down/up)	Latency	Errors
28.8 Analog Modem	3.0 kB/s	120 ms	med.
56K Analog Modem	6.0 kB/s / 4.4 kB/s	100 ms	med.
ISDN	8 kB/s	20 ms	low
ADSL	0.2-1 MB/s / 0.1-0.2 MB/s	10 ms	low
ADSL2+	3 MB/s / 0.1 MB/s	10 ms	low
ADSL2+M	3 MB/s / 0.4 MB/s	10 ms	low
Ethernet (1 hop)	1-100 MB/s	$<$ 1 ms	low
Ethernet (multihop)	1-10 MB/s	1-100 ms	var.
Internet	0-100 MB/s	10-1000 ms	var.

Common network connections.

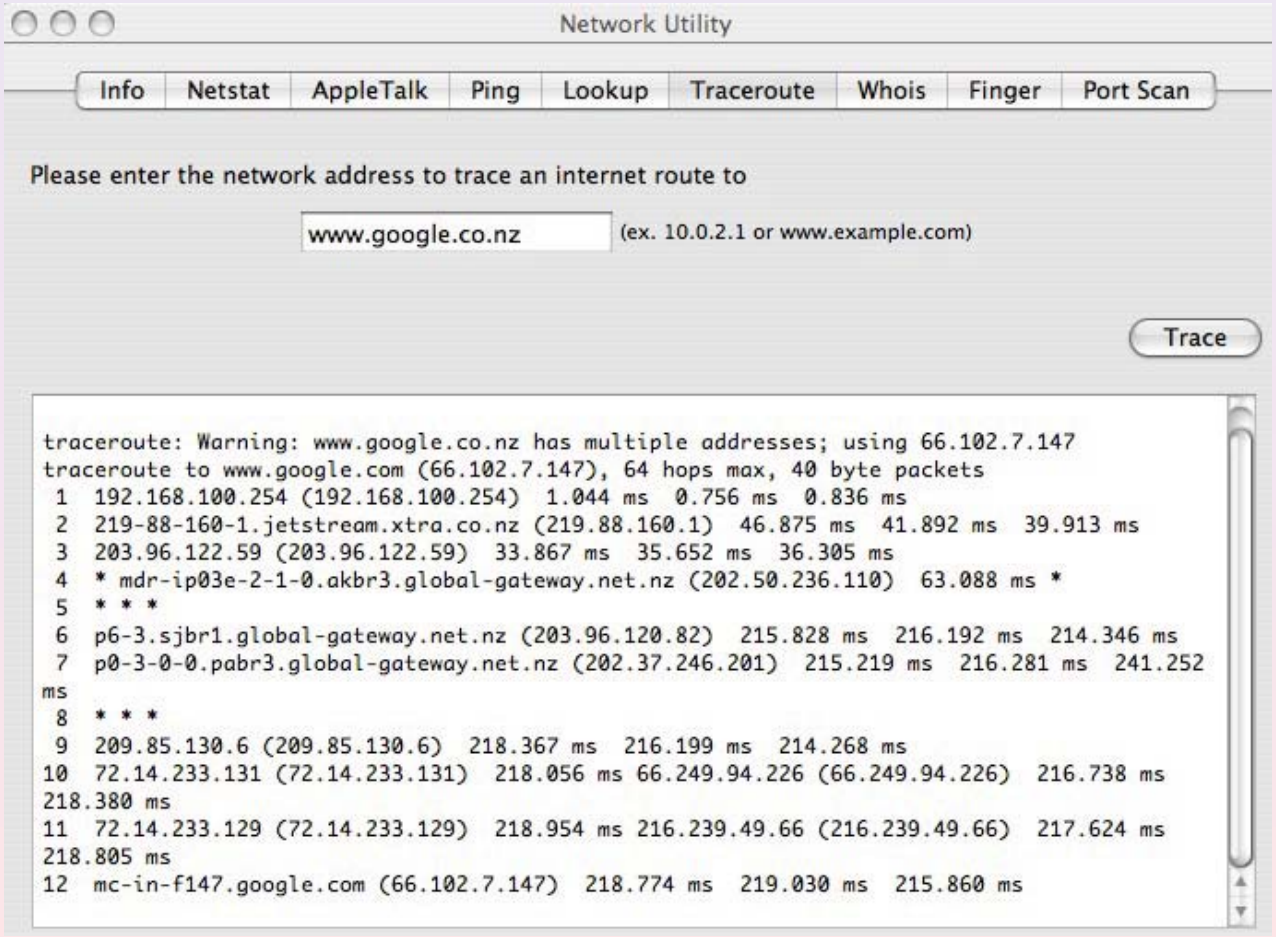
Network utility



Ping


Network Utility
Info
Netstat
AppleTalk
Ping
Lookup
Traceroute
Whois
Finger
Port Scan
Please enter the network address to ping
www.gmail.com
(ex. 10.0.2.1 or www.example.com)
Send an unlimited number of pings
Send only 20 pings
Ping
Ping has started ...
PING mail.google.com (209.85.147.83): 56 data bytes
64 bytes from 209.85.147.83: icmp_seq=0 ttl=242 time=233.076 ms
64 bytes from 209.85.147.83: icmp_seq=2 ttl=242 time=234.151 ms
64 bytes from 209.85.147.83: icmp_seq=4 ttl=242 time=234.012 ms
64 bytes from 209.85.147.83: icmp_seq=5 ttl=242 time=232.676 ms
64 bytes from 209.85.147.83: icmp_seq=6 ttl=242 time=235.914 ms
64 bytes from 209.85.147.83: icmp_seq=7 ttl=242 time=234.302 ms
64 bytes from 209.85.147.83: icmp_seq=11 ttl=242 time=233.301 ms
64 bytes from 209.85.147.83: icmp_seq=15 ttl=242 time=236.382 ms
64 bytes from 209.85.147.83: icmp_seq=17 ttl=242 time=232.638 ms
64 bytes from 209.85.147.83: icmp_seq=19 ttl=242 time=232.482 ms
--- mail.google.com ping statistics ---
20 packets transmitted, 10 packets received, 50% packet loss
round-trip min/avg/max/stddev = 232.482/233.893/236.382/1.288 ms

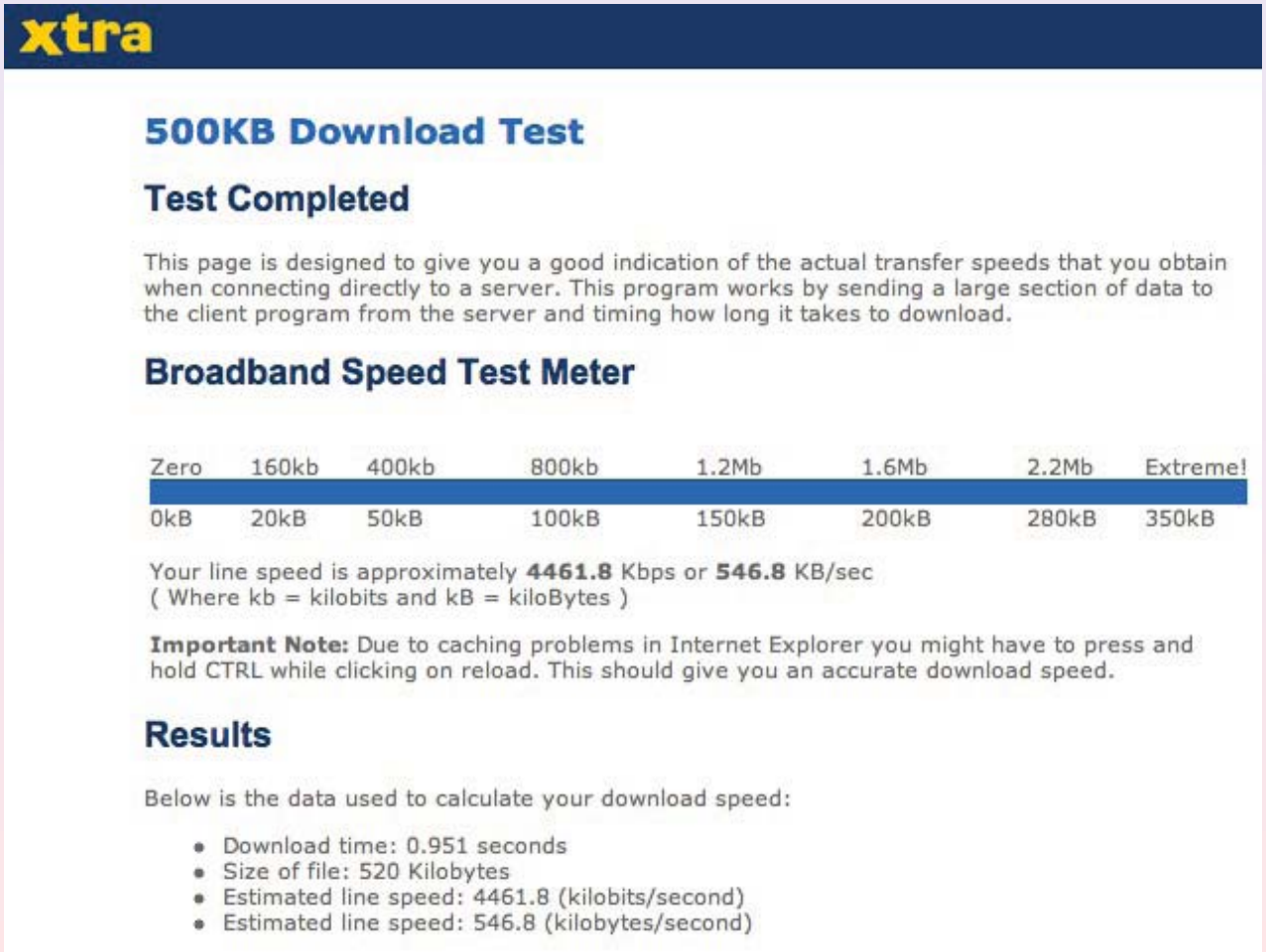
Trace route


Network Utility
Info
Netstat
AppleTalk
Ping
Lookup
Traceroute
Whois
Finger
Port Scan
Please enter the network address to trace an internet route to
www.google.co.nz
(ex. 10.0.2.1 or www.example.com)
Trace
traceroute: Warning: www.google.co.nz has multiple addresses; using 66.102.7.147
traceroute to www.google.com (66.102.7.147), 64 hops max, 40 byte packets
1 192.168.100.254 (192.168.100.254) 1.044 ms 0.756 ms 0.836 ms
2 219-88-160-1.jetstream.xtra.co.nz (219.88.160.1) 46.875 ms 41.892 ms 39.913 ms
3 203.96.122.59 (203.96.122.59) 33.867 ms 35.652 ms 36.305 ms
4 * mdr-ip03e-2-1-0.akbr3.global-gateway.net.nz (202.50.236.110) 63.088 ms *
5 * * *
6 p6-3.sjbr1.global-gateway.net.nz (203.96.120.82) 215.828 ms 216.192 ms 214.346 ms
7 p0-3-0-0.pabr3.global-gateway.net.nz (202.37.246.201) 215.219 ms 216.281 ms 241.252
ms
8 * * *
9 209.85.130.6 (209.85.130.6) 218.367 ms 216.199 ms 214.268 ms
10 72.14.233.131 (72.14.233.131) 218.056 ms 66.249.94.226 (66.249.94.226) 216.738 ms
218.380 ms
11 72.14.233.129 (72.14.233.129) 218.954 ms 216.239.49.66 (216.239.49.66) 217.624 ms
218.805 ms
12 mc-in-f147.google.com (66.102.7.147) 218.774 ms 219.030 ms 215.860 ms

Commercial bit rates in NZ (2007)

- Dial-up: [up to] 56 kb/s
- Telecom (ADSL): "Maximum speed - as fast as your phone line allows."
- Telstra (ADSL): 2 Mb/s downstream and upstream
- Woosh (wireless): [up to] "40 times faster than dial-up"

Bit rates: xtra test (Ethernet, Glendowie) 1



xtra

500KB Download Test

Test Completed

This page is designed to give you a good indication of the actual transfer speeds that you obtain when connecting directly to a server. This program works by sending a large section of data to the client program from the server and timing how long it takes to download.

Broadband Speed Test Meter

Zero	160kb	400kb	800kb	1.2Mb	1.6Mb	2.2Mb	Extreme!
0kB	20kB	50kB	100kB	150kB	200kB	280kB	350kB

Your line speed is approximately **4461.8** Kbps or **546.8** KB/sec
(Where kb = kilobits and kB = kiloBytes)

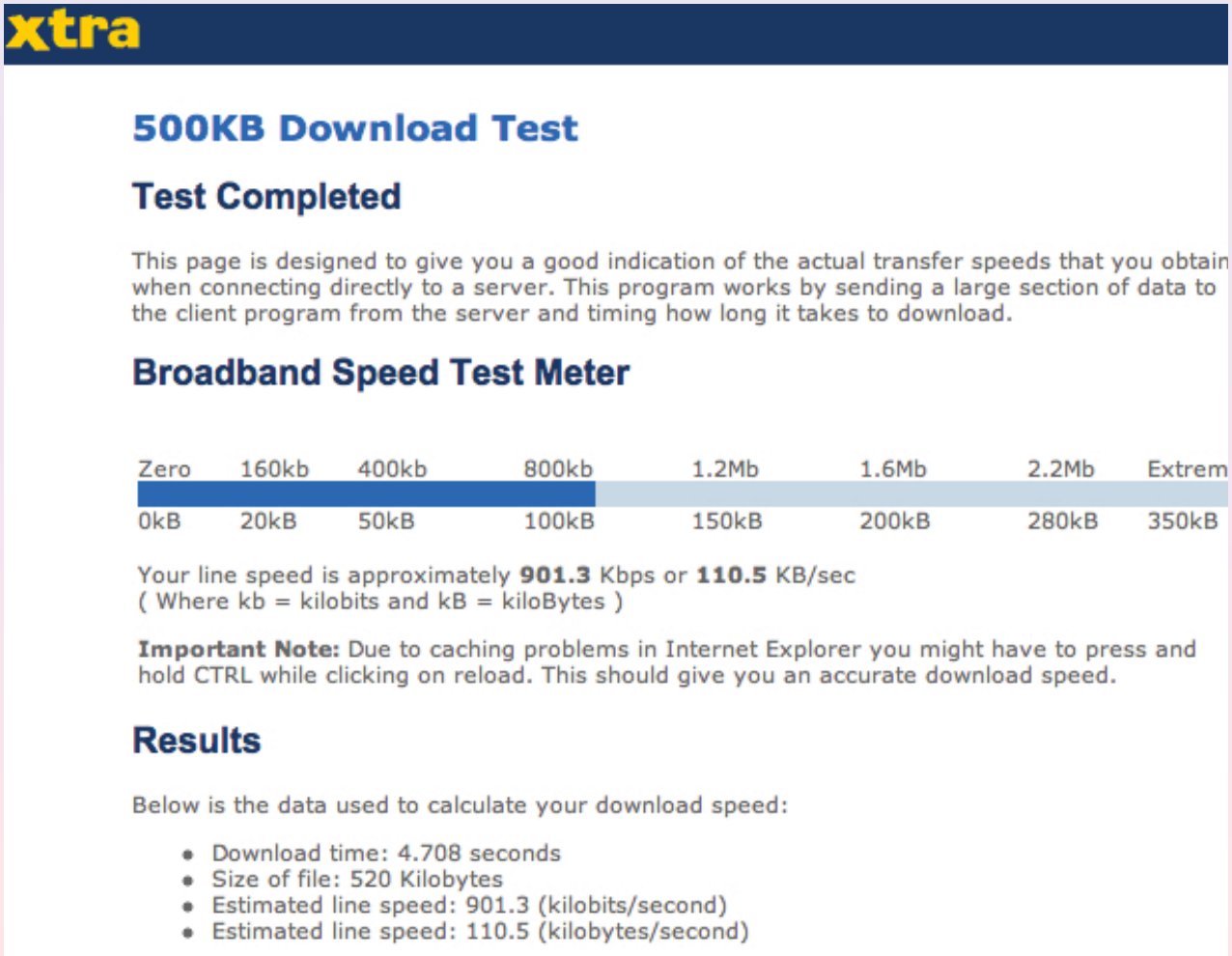
Important Note: Due to caching problems in Internet Explorer you might have to press and hold CTRL while clicking on reload. This should give you an accurate download speed.

Results

Below is the data used to calculate your download speed:

- Download time: 0.951 seconds
- Size of file: 520 Kilobytes
- Estimated line speed: 4461.8 (kilobits/second)
- Estimated line speed: 546.8 (kilobytes/second)

Bit rates: xtra test (Ethernet, Mt. Eden) 2



xtra

500KB Download Test

Test Completed

This page is designed to give you a good indication of the actual transfer speeds that you obtain when connecting directly to a server. This program works by sending a large section of data to the client program from the server and timing how long it takes to download.

Broadband Speed Test Meter

Zero	160kb	400kb	800kb	1.2Mb	1.6Mb	2.2Mb	Extrem
0kB	20kB	50kB	100kB	150kB	200kB	280kB	350kB

Your line speed is approximately **901.3** Kbps or **110.5** KB/sec
(Where kb = kilobits and kB = kiloBytes)

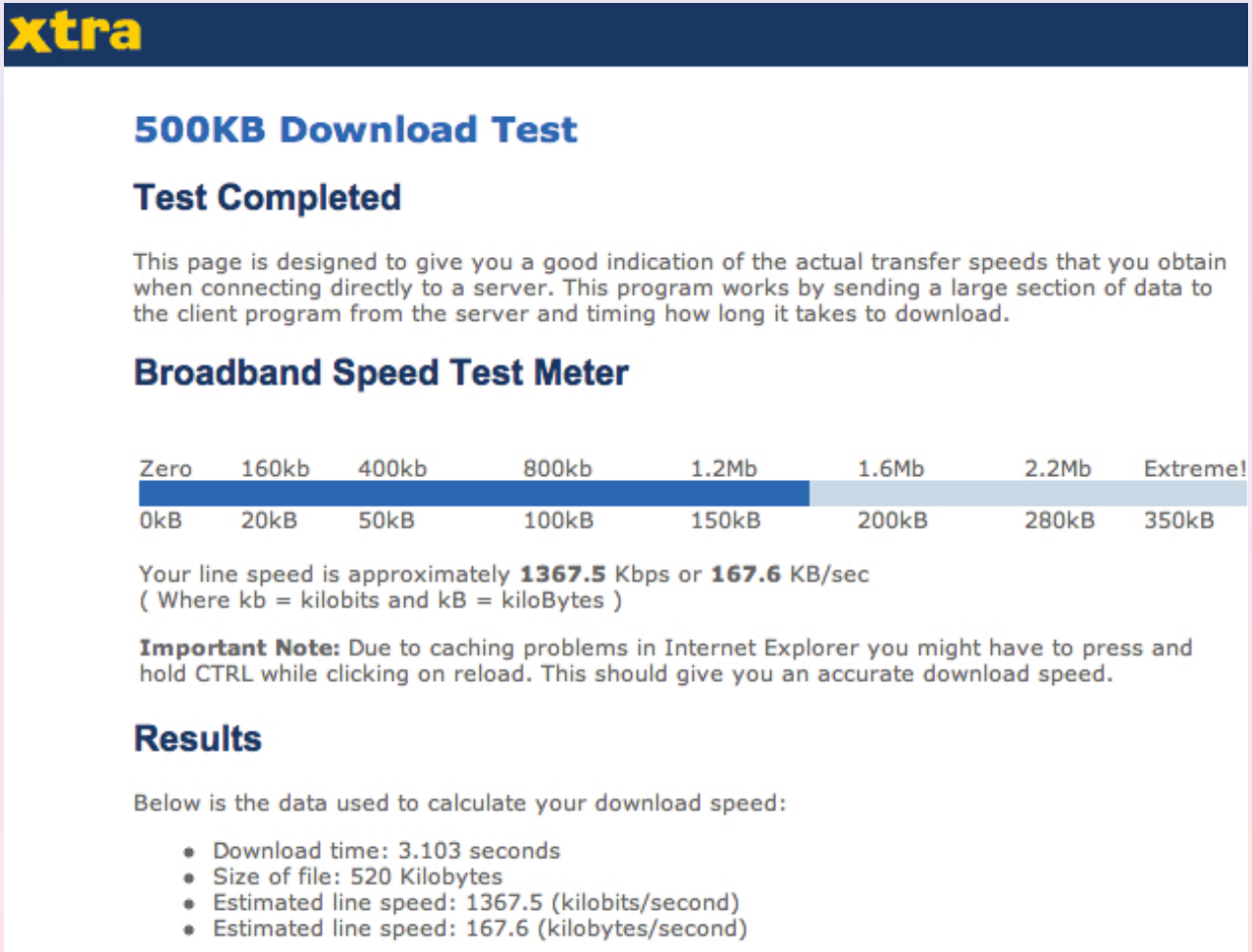
Important Note: Due to caching problems in Internet Explorer you might have to press and hold CTRL while clicking on reload. This should give you an accurate download speed.

Results

Below is the data used to calculate your download speed:

- Download time: 4.708 seconds
- Size of file: 520 Kilobytes
- Estimated line speed: 901.3 (kilobits/second)
- Estimated line speed: 110.5 (kilobytes/second)

Bit rates: xtra test (wireless, Mt. Eden) 3



xtra

500KB Download Test

Test Completed

This page is designed to give you a good indication of the actual transfer speeds that you obtain when connecting directly to a server. This program works by sending a large section of data to the client program from the server and timing how long it takes to download.

Broadband Speed Test Meter

Zero	160kb	400kb	800kb	1.2Mb	1.6Mb	2.2Mb	Extreme!
0kB	20kB	50kB	100kB	150kB	200kB	280kB	350kB

Your line speed is approximately **1367.5** Kbps or **167.6** KB/sec
(Where kb = kilobits and kB = kiloBytes)

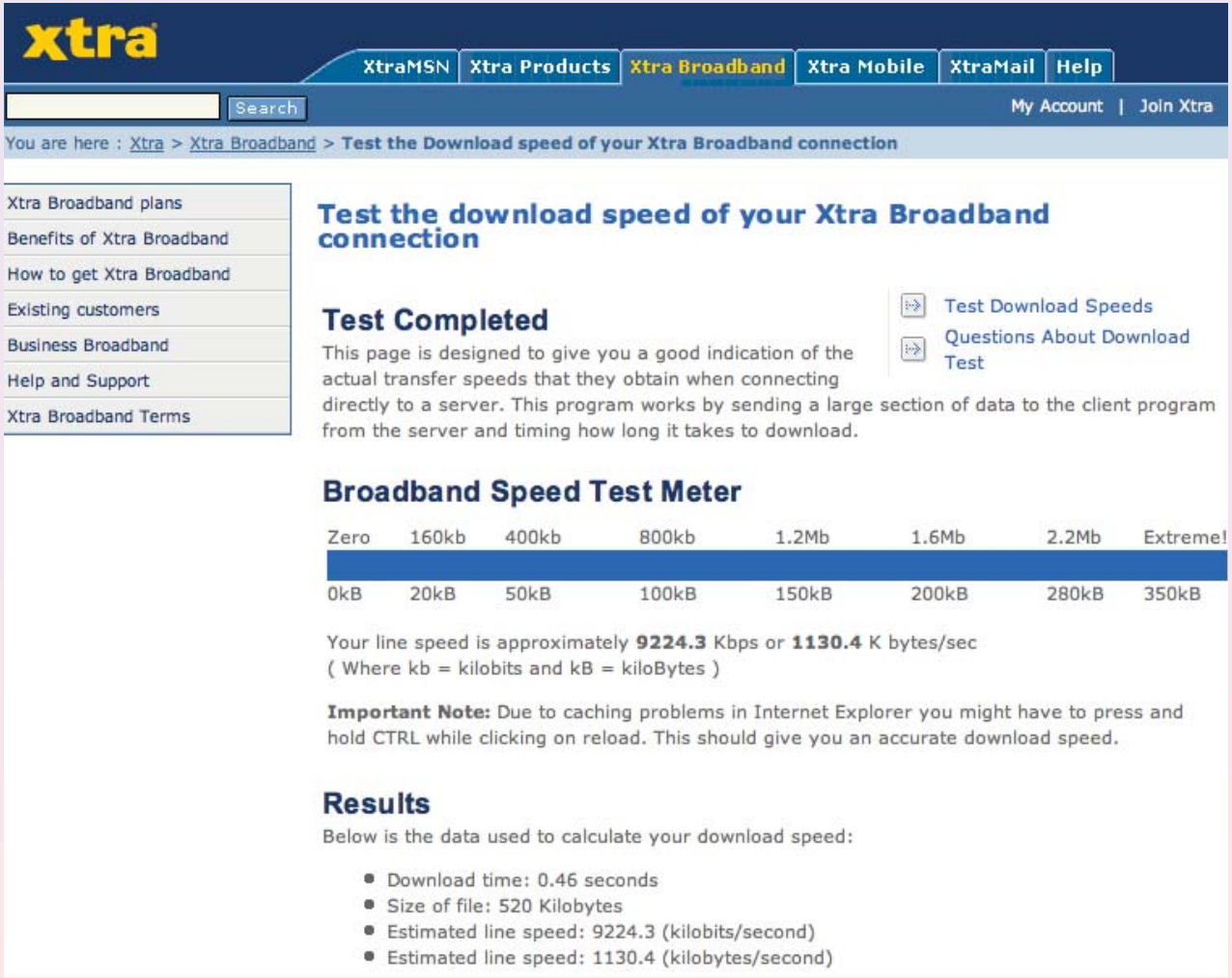
Important Note: Due to caching problems in Internet Explorer you might have to press and hold CTRL while clicking on reload. This should give you an accurate download speed.

Results

Below is the data used to calculate your download speed:

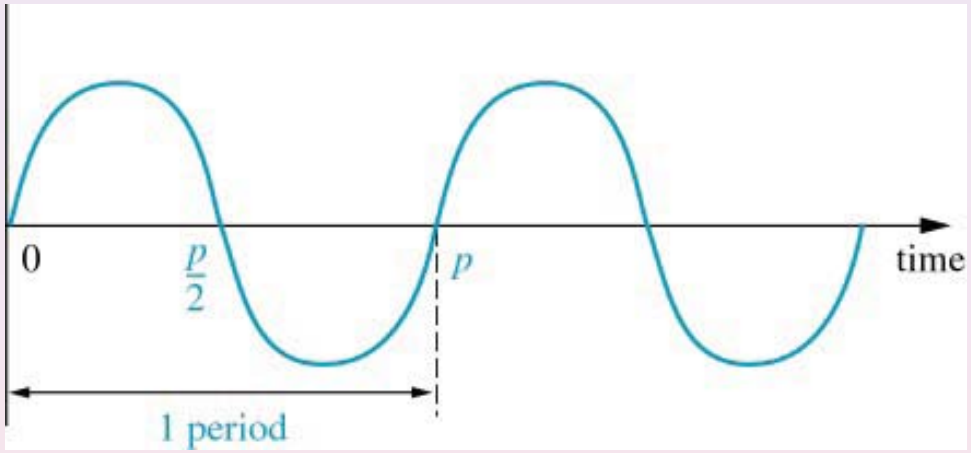
- Download time: 3.103 seconds
- Size of file: 520 Kilobytes
- Estimated line speed: 1367.5 (kilobits/second)
- Estimated line speed: 167.6 (kilobytes/second)

Bit rates: xtra test (Ethernet, UoA) 4



Bandwidth 1

Some analog signals



are **periodic**, i.e. they repeat a pattern/cycle continuously. The **period** of a signal is the time required by the signal to complete one cycle. The signal above has period p.

Bandwidth 2

A signal's frequency, f, is the number of cycles through which the signal can oscillate in a second.

The frequency and period are related as follows:

$$f = \frac{1}{p}.$$

The unit of measurement is "cycles per second", or **hertz (Hz)**.

Bandwidth 3

Suppose the period is

$$p = 0.5 \text{ microsecond } (\mu\text{s}) = 0.5 \times 10^{-6} \text{ s}$$

Then, the frequency is

$$f = \frac{1}{0.5 \times 10^{-6}} = 2 \times 10^{6} \text{ Hz } = 2 \text{ MHz}.$$

Bandwidth 4

The **bandwidth** is equal to the difference between the highest and lowest frequencies that can be transmitted.

- A telephone signal can handle frequencies in the range 300 to 3,300 Hz, so its bandwidth is equal to 3,300 - 300 = 3,000 Hz; very high (low) pitched sounds cannot pass through the telephone.
- Sometimes the term bandwidth is used for the amount of traffic between a web site and the rest of the Internet.

How much is?

- How much is 200MB?
  About 40 music files (average of 5MB per song) or stream 80 minutes of **news, sport and entertainment** (NSE).
- But 1GB?
  About 200 music files or stream 400 minutes of NSE.
- But 5GB (5,000MB)?
  About 1000 music files or stream 2000 minutes NSE.
- But 10GB (10,000MB)?
  About 2000 music files or stream 4000 minutes of NSE.
- But 20GB (20,000MB)?
  About 4000 music files or stream 8000 minutes of NSE.

How is information coded?

Whether the medium uses light, electricity, or microwaves, we must answer perhaps the most basic of all communications questions:

How is information coded in a format suitable for transmission?

Bits

Regardless of implementation, all switches are in one of two states: open or closed, symbolically, 0 and 1.

Bits can store only two distinct pieces of information. Grouping them, allows for many combinations:

- two bits allow $2^2 = 4$ unique combinations: $00, 01, 10, 11$
- three bits allow for $2^3 = 8$ combinations,
- ten bits allow for $2^{10} = 1,024$ combinations,
- fifty bits allow for $2^{50} = 1,125,899,906,842,624$ combinations,
- n bits allow for 2^n combinations.

From bits to codes

Grouping bits allows one to associate certain combinations with specific items such as characters, numbers, pictures. Loosely speaking, this association is called a **code**. Not every association is a code as we shall soon learn.

A difficult problem in communications is to establish communications between devices that operate with different codes. There are standards, but not all standards are compatible!

> *The nice thing about standards is that you have so many to choose from.*
> *– Tanenbaum, Computer Networks (2nd Ed.), 1988*

Early codes: Morse 1

Originally created for Morse's electric telegraph in 1838, by the American inventor Samuel Morse, the **Morse code** was also extensively used for early radio communication beginning in the 1890s.

The telegraph required a human operator at each end. The sender would tap out messages in Morse code which would be transmitted down the telegraph wire to a human decoder translating them back into ordinary characters.

Early codes: Morse 2

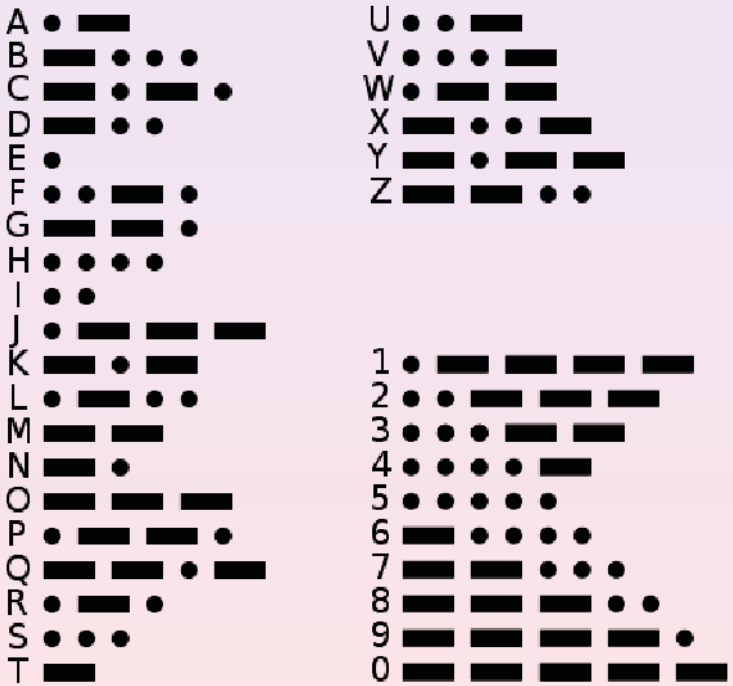
Morse code is transmitted using just two states — on (1) and off (0) — so it was an early form of a digital code.
International Morse code is composed of six elements:

- **short mark, dot or 'dit' (·)** – 1
- **longer mark, dash or 'dah' (-)** – 111
- intra-character gap (between the dots and dashes within a character) – 0
- short gap (between letters) – 000
- medium gap (between words) – 0000000

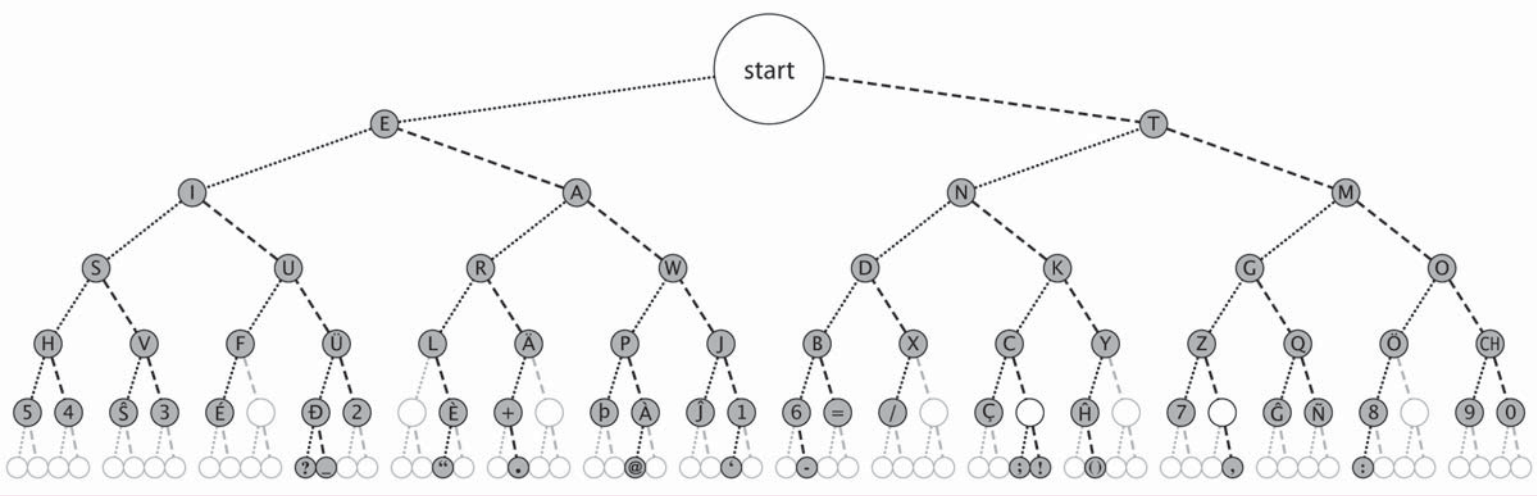
Early codes: Morse 3

International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.



Code Tree for Morse 4



http://commons.wikimedia.org/wiki/File:Morse_code_tree3.png

Early codes: Morse 5

Morse code is a **variable-length code**:

- letter codes have different lengths; the letter E code is a single dot (1000), the letter H code has four dots (1010101000);
- the code (0000000) for an inter-word gap (the 'space' character) is of length 7;

Reason: more frequent letters are assigned shorter codes, so messages can be sent quickly.

Questions:

- Does this code discriminate against Gaelic, Welsh, and other languages with letter frequencies that are very dissimilar to English?
- Why do digits have varying lengths in Morse code? (Do you think '5' is much more frequent than '1'?)

Early codes: Morse 6

Morse code is still in use today by radio amateurs and until rather recently was used in shipping.

An experienced operator can handle about 30 words per minute (standard word is 5 characters as in “Paris”) and some higher than that. This is faster than most people can hand-write.

Early codes: Baudot code 1

The **Baudot code**—also known as **International Telegraph Alphabet No 2 (ITA2)**—is named after its French inventor Émile Baudot. ITA2 is a fix-length code using 5 bits for each character (digits and letters). This code was developed around 1874.

With 5-bit codes we can name $2^5 = 32$ different objects, but we have 36 letters and digits (plus special characters) to code!

For example, the letter Q and digit 1 have the same code: 10111. In fact each digit's code duplicates that of some letter.

Early codes: Baudot code 2

00	01	02	03	04	05	06	07
NUL	E 3	LF	A -	SP	S '	I 8	U 7
08	09	0A	0B	0C	0D	0E	0F
CR	D ENQ	R 4	J BEL	N ,	F !	C :	K (
10	11	12	13	14	15	16	17
T 5	Z +	L)	W 2	H £	Y 6	P 0	Q 1
18	19	1A	1B	1C	1D	1E	1F
O 9	B ?	G &	FIGS	M .	X /	V ;	LTRS
Letters			Figures		Control Chars.		

Early codes: Baudot code 3

Do you think we have got a problem?
More precisely, how can we tell a digit from a letter?

Answer: using the same principle that allows a keyboard key to represent two different characters. On the keyboard we use the Shift key; the Baudot code uses the extra information

11111 (shift down) and 11011 (shift up)

to determine how to interpret a 5-bit code. Upon receiving a shift down, the receiver decodes all codes as letters till a shift up is received, and so on.

Early codes: Baudot code 4

Here is an example.
ABC123, is coded from left to right as follows:

11111 00011 11001 01110 11011 10111 10011 00001

Early codes: BCD, BCDIC, ASCII codes

- BCD stands for **binary-coded decimal**, a code developed by IBM for its mainframe computers using 6-bit codes;
- BCDIC stands for **binary-coded decimal interchange code**, an expansion of BCD including codes also for non-numeric data;
- ASCII (pronounced ['æski]) stands for the **American Standard Code for Information Interchange**; it is a 7-bit code that assigns a unique combination to every keyboard character and to some special functions.

ASCII code (decimal, binary, hexadecimal) 1

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ASCII code (decimal, binary, hexadecimal) 2

Each code corresponds to a printable or unprintable character.

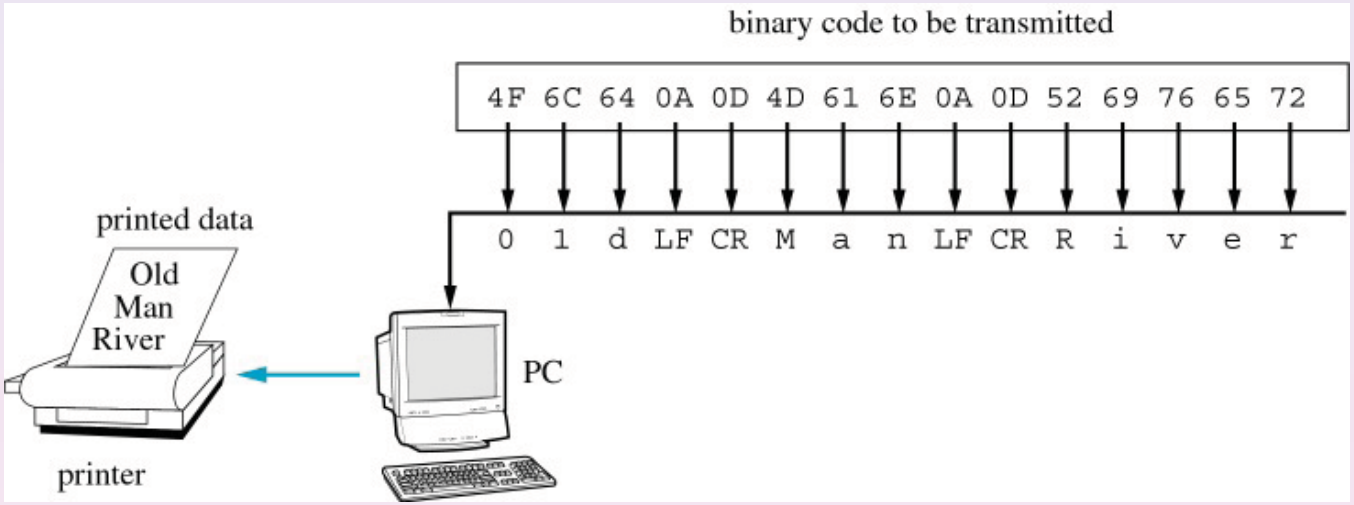
Printable characters include letters, digits, and special punctuation (commas, brackets, question marks).

Unprintable characters are special functions (e.g. line feed, tab, carriage return, BEL, DC1/XON/ctrl-Q, DC3/XOFF/ctrl-S).

Standard ASCII has 128 different characters.

Extended ASCII codes (e.g. ISO-8859-1, Mac OS Roman, ...) have an additional 128 characters.

ASCII code 3



If codes are sent with the leftmost first, as the printer receives each code, it analyses and takes some action: for 4F, 6C and 64 it prints 0, 1, and d. The next two codes, OA and OD, denote unprintable characters (LF = line feed, CR = carriage return). When OA is received, nothing is printed, but the mechanism to advance to the next line is activated.

Unicode 1

UTF-32 is the fixed-length "Unicode Transformation Format":

- 17 code planes (characters for most modern languages are in plane 0);
- up to 65,536 code points per plane (ASCII uses 128 code points, and Extended ASCII uses 256 code points);
- Four bytes per character.

UTF-8 is a variable-length encoding, with 1 to 4 bytes per character:

- The 128 characters of ASCII are encoded in one byte (with a leading 0)
- Another 1920 characters are encoded in two bytes: Latin letters with diacritics, Greek, Cyrillic, Coptic, Armenian, Hebrew, Arabic, Syriac, Tāna.

Unicode 2

The graphical rendering of a Unicode character string is system-dependent.

- Multiple UTF characters may be rendered as a single glyph (e.g. “f” followed by “i” may be rendered as a ligature “fi”).
- Distinct UTF characters may be rendered with the same glyph, or with a very similar one, so following a Unicode hyperlink can be dangerous [Fu 2006, 10.1145/1143120.1143132].

fi → ﬁ

fl → ﬂ

s	s	s	S	S	S	ƨ
0073	FF53	0455	10BD	FF33	0405	03E8
o	o	o	o	º	0	o
006F	03BF	043E	FF4F	00BA	FFB7	047B
u	u	u	U	U	IJ	ų
0075	2294	03C5	22C3	222A	0132	1E75
p	p	p	p	ρ	ƿ	P
0070	0440	FF50	01BF	03C1	05E7	0420

Figure 1. Characters similar to “s”, “o”, “u”, and “p” (in Arial Unicode MS Font).

What is a code? 1

We can now ask the important question:

What is a code?

Here is an example. The character "=" is represented by the binary code word "0111101". Why do we need the leading zero? Surely "111101" means the same thing because both "09" and "9" mean nine?

All ASCII codes have the same length. This ensures that an important property—called the prefix property—holds true for the ASCII code.

What is a code? 2

A **code** is the assignment of a unique string of characters (a **codeword**) to each character in an alphabet.

A code in which the codewords contain only zeroes and ones is called a **binary code**.

The encoding of a string of characters from an alphabet (the cleartext) is the concatenation of the codewords corresponding to the characters of the cleartext, in order, from left to right. A code is **uniquely decodable** if the encoding of every possible cleartext using that code is unique.

What is a code? 3

For example, here are two possible binary codes for the alphabet $\{a, c, j, l, p, s, v\}$:

	code 1	code 2
a	1	010
c	01	01
j	001	001
l	0001	10
p	00001	0
s	000001	1
v	0000001	101

What is a code? 4

Both code 1 and code 2 satisfy the definition of a code. However,

- code 1 is uniquely decodable, but
- code 2 is not uniquely decodable; for example, the encodings of the cleartexts “pascal” and “java” are both

001010101010

Prefix codes 1

A **prefix code** is a code with the "prefix property":

> *no codeword is a (proper) prefix of any other codeword in the set.*

The code $\{0, 10, 11\}$ has the prefix property; the code $\{0, 1, 10, 11\}$ does not, because "1" is a prefix of both "10" and "11".

Code 1 is a prefix code, but code 2 is not.

> *Why is the prefix property important?*

Because prefix codes are uniquely decodable.

Prefix codes 2

Every fixed-length code is a prefix code.

There can be no prefixes in the code table, because no codeword is any longer or shorter than any other.

Therefore, ASCII is a prefix code.

Prefix codes 3

Is Morse a prefix code?

- Consider the dit-dah table. The codeword for A (dit-dah) is a prefix of the codeword for J (dit-dah-dit-dit).
- Consider a binary representation where characters begin with a short-gap (000). The codeword for A (00010111) is a prefix of the codeword for J (000101110101).
- Consider a binary representation where characters end with a short-gap (000). The codeword for A (10111000) is *not* a prefix of the codeword for J (101110101000).

Do you think prefix codes are less efficient than non-prefix codes, i.e. do they take longer to transmit? (Hint: Kraft's Theorem.)

Do you think it is easier to learn a prefix code?