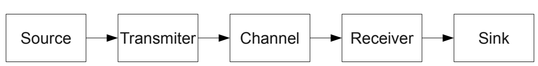
Introduction to Data Communication

This lecture introduces some basic terminology, describes a simple model of a communication link and a network and describes some characteristics of data.
After this lecture you should be able to: define the terms introduced this lecture, be able to convert numbers between different number bases and bit/byte orders and convert between ASCII characters and their character codes.

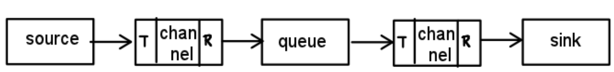
Communication Systems Model

The standard model for a communication system includes the following:

- data source - generates the information
- transmitter - converts the information into an electrical signal that can be transmitted over the channel
- channel - distorts the signal and adds noise
- receiver - attempts to recover the information that was transmitted
- data sink - accepts the information



In many cases information travels over a network consisting of multiple channels and their associated transmitters and receivers. In some cases the information is stored ("queued") before being forwarded over the next channel.



In this course we will study the implementation of digital communication systems and networks in detail.

Exercise 1: For each of the following digital communication services identify the source, sink and the channel(s) involved: (1) the Ethernet connection between a computer and a router; (2) a cell phone call ; (3) watching a YouTube video at home.

Digital and Data Communications

Digital communication systems communicate information that is in the form of digits, almost always binary digits or "bits".

Digital communication systems include those used to transmit digitized speech or video waveforms as well more abstract information such as text, images, or computer software (often called "data").

However, today the terms "data communications" and "digital communications" are often used interchangeably because the same networks are used to carry all types of digital information.

Characteristics of Data Sources and Sinks

Data sources are often compressed to reduce the amount of data that needs to be transmitted.

Data representing speech and data can often be compressed with little degradation because humans cannot perceive certain details of sounds and images. These details can be removed resulting in lower data rates. Examples of these "lossy" compression techniques include "MP3" for compressing audio and MPEG-4 for video.

Some types of data contains redundancy such as sequences of bits or bytes that occur more often than others. These type of data can be compressed and the decompressed without loss of information. An example of "lossless" compression is the 'zip' compression used for computer files.

Both lossless and lossy compression are examples of "source coding".

One characteristic of a data source is whether the rate at which the data is generated is:

- constant - constant bit rate (CBR) or "isochronous" sources are typical of regularly sampled waveforms such as audio which are not compressed

- variable - variable bit rate (VBR) sources are typical of compressed speech and video (*why?*)
- bursty - bursty data sources are typical of systems involving human interaction such as web surfing

Data sinks vary in their tolerance to channel impairments such as: data loss, errors, delay and variability of delay (delay "jitter").

For example, computer data transmission systems usually provide very low undetected error rates (e.g. once per several years) but can often tolerate high delay and delay variability (seconds). On the other hand telephone systems can tolerate loss of a small percentage of the speech waveform but become difficult to use if delays exceed a significant fraction of a second.

Exercise 2: What units would be used to specify error rate, delay, and delay variability? For each of the following data sources/sinks identify (1) the relative data rate variability, (2) the tolerance to errors, (3) the tolerance to delay: (a) a phone call between two people, (b) downloading a computer program, (c) streaming a video over a computer network. What units would each be measured in? Try to estimate typical values and for each quantity.

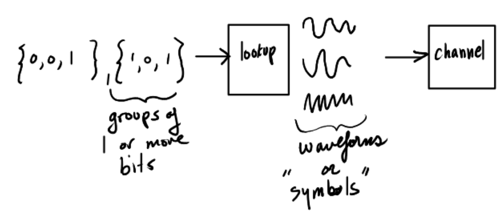
Data Rates

The rate at which the source generates data, the "data rate", is specified in bits per second (bps or b/s).

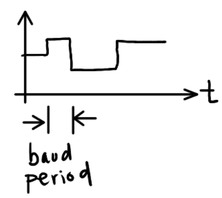
Sometimes, particularly in user-facing data communications applications, the same acronym is used for "byte per second". To avoid confusion it's best to spell out the units if the meaning is not clear from the context. In this course "bps" will always mean "bits per second".

Sometimes the data is not completely random and can be compressed. The theoretical minimum data rate, assuming the best possible compression has been applied, is called the "information rate".

A waveform is a voltage or current that varies with time. The transmitter converts the data to a signal, a waveform that carries information, for transmission over the channel. Often the transmitter uses a number of bits to select from one of several possible waveforms, or "symbols", to transmit. The rate at which these waveforms are transmitted is called the "symbol rate". The symbol rate is equal to or lower than the bit rate(*why?*).



One possible set of signaling waveforms is voltages of different levels. The maximum rate at which these levels can change is called the "baud rate". The baud rate can be lower or higher than the bit rate depending on the design of the transmitter.



Throughput

In addition to the data rate at the source, we can measure the average data rate at the receiver. This is called the "throughput". This can be different than the rate at the transmitter because the transmitter may need to:

- share the channel with other transmitters
- add "overhead" bits for addressing, error detection, etc
- retransmit data that was not received correctly

The term "goodput" is sometimes used to refer to the throughput as measured by a computer application.

Bit and Byte Order

The bits generated by a data source are usually organized into "words". Words of 8 bits are called bytes (or "octets" in some standards documents). Words of 4 bits are often called nibbles (or nybbles). Words composed of other even multiples of 8 bits (16, 32, 48 and 64 bits) are also common.

It is important that the order of the bits be preserved between the source and sink.

If the bits in a word represent a binary number they can be ordered from "most significant bit" (MSB) to "least significant bit" (LSB). This is sometimes called "big endian" order. The reverse order is called "little endian".

Often the bits (in either bit order) are part of bytes which themselves are concatenated to form words. The bytes in each word can also be ordered MSB(yte) first (big-endian) or LSB(yte) first (little endian).

Most Internet protocols use big-endian bit and byte order which is sometimes called "network order". In network order the bits and bytes are transmitted in the order they are written.

Exercise 3: Convert the decimal number 3525 to a 16-bit (two-byte) binary number. Write the sequence of bits that would be transmitted if both the bytes and bits were transmitted in little-endian order. Write the sequence of bits that would be transmitted in "network order".

Notation

When collections of bits are interpreted as numbers they can be written using digits from various bases.

The most common notation is hexadecimal because it allows 8-bit values (bytes) to be written using two hexadecimal digits. Hexadecimal digits are 0 to 9 and A through F (representing values from 10 through 15). Typically a special prefix of non-numeric character(s) is used to indicate that the number is written in binary, octal or hexadecimal notation. Typical prefixes for hexadecimal notation include "$", "#", "0x", and "0H".

Octal numbers (representing 3 bits per digits) are occasionally used as well.

Exercise 4: Write the 16-bit number above in hexadecimal notation.

Character Codes

Data can also represent printable characters ("glyphs").

A standard, called Unicode, has been developed to assign a unique number to each of the many thousands of characters used by the world's various languages.

The first 127 characters of Unicode correspond to an earlier character encoding called ASCII (American Standard Code for Information Interchange).

UTF-8 is the most common way of representing Unicode characters as a sequence of bytes. The UTF-8 encoding is the same as the ASCII encoding for the first 127 characters. This makes existing ASCII files compatible with software that expects UTF-8 encoded Unicode. Other Unicode characters require between 2 and 6 bytes.

ASCII also includes some non-printable control codes that will be discussed later in the course.

The next page shows the ASCII table as given in version 6.2 of the Unicode standard. The columns are labelled with the most significant (first) hex digit and the rows with the least-significant (second) hex digit of the numerical value of each character.

Exercise 5: Find the ASCII codes for the characters '3525'. Write out the first 16 bits of the sequence that would be transmitted assuming each character is encoded using 8 bits per character and little-endian bit order. *Hint: the character code for a digit is 0x30 plus the value of the digit.*

Exercise 6: Highlight or underline each term where it is defined in these lecture notes.

The Telecom Ecosystem

The following are the main types of companies and organizations involved in the data communications industry:

Component Manufacturers make components such as ICs, capacitors, cables, etc

Distributors buy components from manufacturers

Equipment Manufacturers buy components and build equipment

Retailers buy equipment and sell to customers

Service Providers buy equipment and provide services to customers

Standards Organizations write and publish standards so that equipment from different manufacturers will interoperate

Regulators publish and enforce regulations to ensure efficient use of public resources and protect the safety of the public

Exercise 7: Draw a diagram showing the flow of of goods and services between these various entities. Classify the following: Intel, Xilinx, Analog Devices, Belden, Avnet, DigiKey, Samsung, Apple, Cisco, Ericsson, Telus, Netflix, Walmart, Amazon, IEEE-SA, IETF, Industry Canada, CRTC, FCC. Look these up if you're not familiar with them.

	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	SOH 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	STX 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT 0004	DC4 0014	$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074
5	ENQ 0005	NAK 0015	% 0025	5 0035	E 0045	U 0055	e 0065	u 0075
6	ACK 0006	SYN 0016	& 0026	6 0036	F 0046	V 0056	f 0066	v 0076
7	BEL 0007	ETB 0017	' 0027	7 0037	G 0047	W 0057	g 0067	w 0077
8	BS 0008	CAN 0018	(0028	8 0038	H 0048	X 0058	h 0068	x 0078
9	HT 0009	EM 0019	) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079
A	LF 000A	SUB 001A	* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A
B	VT 000B	ESC 001B	+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B
C	FF 000C	FS 001C	, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C
D	CR 000D	GS 001D	- 002D	= 003D	M 004D	] 005D	m 006D	} 007D
E	SO 000E	RS 001E	. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E
F	SI 000F	US 001F	/ 002F	? 003F	O 004F	_ 005F	o 006F	DEL 007F