

## Introduction to Data Communication

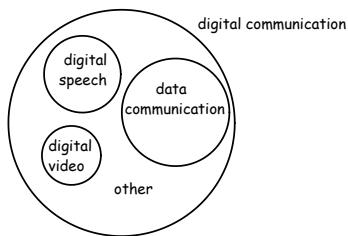
This lecture introduces some basic terminology, describes a model for communication systems and networks and describes some characteristics of data.

After this lecture you should be able to: define the terms introduced this lecture; compute information, entropy, bit, symbol, bit error and frame error rates; compute throughput; convert between characters, Unicode code points and their UTF-8 encodings; convert numbers between different number bases and bit and byte orders.

### Digital Communications

Digital communication systems communicate information encoded as binary numbers. Binary numbers are sequences of binary digits which are called bits.

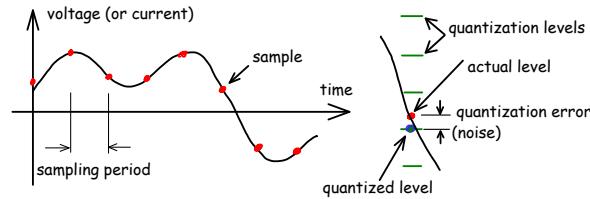
Digital communication systems can be used to transmit speech or video as well as more abstract information such as text or computer software, usually called “data”.



The terms “data communications” and “digital communications” are often used interchangeably because today the same communication systems carry all types of digital information.

### Sampled Waveforms

To represent speech or video waveforms as binary data, the analog signal – a voltage that is continuous in time and level – is sampled (measured) at a regular rate called the “sampling rate.” Each sample is then quantized into a binary number with a fixed number of bits. Thus the signal becomes discrete in both time and level. In later courses you will learn why the sampling rate for a signal must be twice the highest bandwidth and that quantizing a signal to  $B$  bits introduces “quantization noise” with a power about  $6B$  dB less than the signal power.



A binary number may be “unsigned” (non-negative) or signed, typically in two’s-complement format. You should be familiar with these encodings from previous courses.

Binary numbers are often written in hexadecimal notation 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). The prefix, “0x” taken from the C programming language, is often used to indicate hexadecimal notation.

### Character Encodings

Data often represents printable characters or “glyphs”.

A standard called **Unicode** assigns a unique number called a “code point” to over 100,000 of the characters used by over 100 languages and scripts. Unicode is used by most operating systems and Internet applications.

**Exercise 1:** How many bits would be required to uniquely identify 100,000 different characters? (Hint:  $2^{16} = 65536$ ).

We could represent each character using 32 bits (4 bytes). However, the UTF-8 (Unicode Transformation Format - 8 bits) format is more widely used since it can represent many documents using fewer bytes.

ASCII (American Standard Code for Information Interchange) was an earlier character encoding that used 7 bits to encode letters from the English alphabet, numbers and the most common punctuation

**Table 3-6.** UTF-8 Bit Distribution

Scalar Value	First Byte	Second Byte	Third Byte	Fourth Byte
00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
000uuuuu zzzzyyyy yyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

symbols. The UTF-8 encoding is the same as the ASCII encoding for these first 127 characters. This means that ASCII-encoded documents are already encoded as UTF-8.

The table below shows the ASCII table which is the first code chart from 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.

ASCII also includes some non-printable control codes (values 0 to 31) that were used to control printers. For example the line feed (LF) character would move the paper in the printer up one line.

Other Unicode characters require between 2 and 4 bytes according to the rules summarized in Table 3-6 of the Unicode standard shown above. Code points between 128 and 2047 include most characters from European languages and can be encoded in two bytes. Code points from 2048 to 65535 include most CJK (Chinese, Japanese and Korean) characters and require three bytes. Some rarely-used symbols (e.g. emoticons or Mahjong tiles) have four-byte encodings.

To encode a code point into UTF-8 format, first convert it to a binary number and find the first row in the table that can represent that number. Then copy the bits indicated by x, y, z and u from the binary number into the corresponding locations in the 1, 2 or 3 bytes of the UTF-8 encoding.

**Exercise 2:** The Chinese character for “Rice” (the grain) is “米” with Unicode value (code point) U+7C73. What is the UTF-8 encoding for this character?

### Text versus Binary Data

It's important to understand the difference between binary data and text that represents a number. For example, the 8-bit binary number 00110001 (0x31) is the ASCII encoding of the character ‘1’. Numbers can be stored in files or transmitted over communication systems in either binary or text formats. Numbers in text format can be more easily interpreted since they are sequences of printable characters but they require more bits to transmit.

**Exercise 3:**

Convert the decimal number 525 to a 16-bit (two-byte) binary number. How would you write this in hexadecimal notation?

Find the ASCII codes for the characters ‘525’. Write out the bits of the sequence that would be transmitted assuming each character is encoded in UTF-8. Hint: the UTF-8 character code for a digit is 0x30 plus the value of the digit.

Which of these two sequences of bits is the text format and which is the binary format? How many more bits would need to be stored or transmitted for the text format?

### Words, Byte and Bit Orders

Bits are often organized into “words”. Words of 8 bits are called bytes (or “octets” in many standards documents). Words composed of multiple bytes (2, 4 and 8 bytes) are also common.

It is important that communication systems preserve the order of the bytes and the order of the bits within the bytes.

Values stored in computer memory can only be addressed by byte. Thus byte order (but not bit order) is also a concern when multi-byte words are stored in memory or in files.

The bytes in a word can be ordered as the bytes are written, from “most significant byte” (MSB) to “least

significant byte” (LSB). This is sometimes called “big endian” order. The reverse order is called “little endian”. Most Internet protocols use “big endian” byte order which is sometimes called “network order.”

Data is often transmitted over a communication system as individual bits. In this case the bits in a byte can be transmitted least-significant bit first (lsb-first) or most-significant-bit first (msb-first).

Many communication systems (e.g. Ethernet) transmit data with bytes ordered MSB-first but with the bits ordered lsb-first.

**Exercise 4:** Write the sequence of bits that would be transmitted if the 16-bit value 525 was transmitted with the bytes in little-endian order and the bits lsb-first. Write the sequence of bits that would be transmitted in “network order” and the bits msb-first.

## Model of a Communication System

A model for a communication system includes the following<sup>1</sup>:

- information 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 destination - accepts the information (sometimes called a data “sink”)

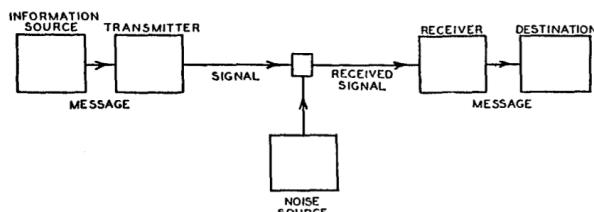
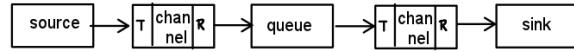


Fig. 1—Schematic diagram of a general communication system.

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.

<sup>1</sup>The diagram is from Claude Shannon's fundamental paper, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, July, October, 1948.



In this course we will study the basic principles underlying digital communication systems and networks.

**Exercise 5:** For each of the following communication systems identify the source, sink and the channel(s) involved: a cell phone call; watching a YouTube video, uploading your lab report to D2L. Which of these involve networks? Come up with your own examples of communication systems and identify these components.

## Characteristics of Data Sources, Channels and Sinks

### Information Rate

We can model sources as generating one of a limited number of messages. We define the information that is transmitted by a message that occurs with a probability  $P$  as:

$$I = -\log_2(P) \text{ bits}$$

For example, a message with a probability of  $\frac{1}{2}$  conveys 1 bit of information. While one with a probability of  $\frac{1}{4}$  carries 2 bits of information. Thus, less likely messages carry more information.

The information rate (also known as the “entropy”) of a source in units of bits per message can be computed as the average information generated by the source:

$$H = \sum_i (-\log_2(P_i) \times P_i) \text{ bits/message}$$

where  $P_i$  is the probability of the  $i$ 'th message.

**Exercise 6:** A source generates four different messages. The first three have probabilities 0.125, 0.125, 0.25. What is the probability of the fourth message? How much information is transmitted by each message? What is the entropy of the source? What is the average information rate if 100 messages are generated every second? What if there were four equally-likely messages?

We will see later in the course that there is a limit, called the “capacity,” ( $C$ ) for the information rate that can be transmitted over a given channel.

## Data Rates

The rate at which the source generates the data that is sent over the channel, the “data rate”, is also specified in bits per second (bps or b/s).

Unfortunately, particularly in consumer-facing data communications applications, the same acronym is often used for “bytes per second”. To avoid confusion it’s best to spell out the units if the meaning is not clear from the context.

Some people use the convention that a capital ‘B’ indicates bytes and a lower-case ‘b’ indicates bits, but you should not assume this convention is universally understood.

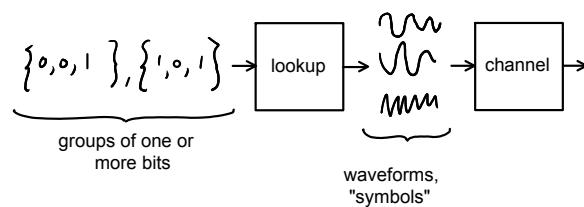
In this course “bps” or “b/s” will always mean “bits per second”. This convention is used in almost all technical documents (equipment specifications, interoperability standards, data sheets, etc.).

Computer storage is often measured in units with prefixes that are powers of 2 (e.g. kilo means  $2^{10}$  or 1024 rather than 1000 or  $10^3$ ). However, communication system data rates always use units that are powers of 10 (e.g. 1 kb/s is 1000 b/s).

**Exercise 7:** How long will it take to transfer 1 MByte at a rate of 10 kb/s?

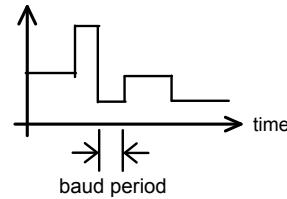
## Symbol Rate

A waveform is a voltage or current that varies with time. The purpose of a transmitter is to convert data, in the form of bits, to a signal, a waveform that carries information, for transmission over the channel. Often the transmitter uses a group 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?*).



**Exercise 8:** In the example above each group of 3 bits is used to select a symbol. How many different symbols do we need? If the bit rate is 12 kbps, what is the symbol rate?

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<sup>2</sup>”. The baud rate can be lower or higher than the bit rate depending on the design of the transmitter.



**Exercise 9:** One system encodes each bit using two pulses of opposite polarity (H-L for 0 and L-H for 1). A second system encodes bits using one pulse per bit (H for 0 and L for 1). A third system encodes two bits per pulse by using four different pulse levels (-3V for 00, -1V for 01, +1V for 10 and +3V for 11). Assuming each system transmits at 1000 bits per second, what are the baud rates in each case? How many different symbols are used by each system? What are the symbol rates?

## Error Rates

The bit error rate (BER,  $P_e$ ) is the average fraction of bits that are received incorrectly.

When these bits are grouped into “frames” we are often interested in the average fraction of the frames that contain one or more errors. This is known as the FER (Frame Error Rate). Sometimes frames include additional bits that allow us to detect most, but not all, errors. We often want the UEP (Undetected Error Probability) to be very small.

**Exercise 10:** You receive 1 million frames, each of which contains 100 bits. By comparing the received frames to the transmitted ones you find that 56 frames had errors. Of these, 40 frames had one bit in error, 15 had two bit errors and one had three errors. What was the FER? The BER?

## Throughput

In addition to the data rate at the transmitter, we are often interested in the average data rate at the destination. This is called the “throughput”. The average rate at which data arrives at the destination can be different than the instantaneous rate at which the transmitter sends data to the channel because:

<sup>2</sup>This is the definition found in the IEEE LAN standards. However, some people consider baud rate to be a synonym for symbol rate.

- the channel may have to be shared between different users
- the transmitter may add (and the receiver remove) “overhead” bits for addressing, error detection, etc
- incorrectly received data may have to be retransmitted
- there may be gaps between frames

**Exercise 11:** A system transmits data at an (instantaneous) rate of 1 Mb/s in frames of 256 bytes. 200 of these bytes are data and the rest are overhead. The time available for transmission over the channel is shared equally between four users. A  $200\ \mu s$  gap must be left between each packet. What throughput does each user see? Now assume 10% of the frames are lost due to errors. What is the new throughput per user?

## Compression

Sometimes data is not completely random and we can make use of the redundancy to reduce the amount of data that needs to be transmitted. Both lossless and lossy compression are examples of “source coding”.

**Lossless.** Some types of data contains redundancy such as sequences of bits or bytes that occur more often than others. This type of data can be compressed before transmission and then decompressed at the receiver without loss of information. An example of this “lossless” compression is the ‘zip’ compression used for computer files.

Another way of defining the information rate is that it is the theoretical minimum data rate, assuming the best possible lossless compression has been applied. Lossless compression cannot reduce the information rate of the source but it can reduce the bit rate that needs to be transmitted over the channel.

**Lossy.** Data representing speech and video 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.

## Data Rate Variability

The data rate of a source can be:

- constant: “isochronous” sources generate data at a constant bit rate (CBR) and are typical of regularly sampled waveforms such as (uncompressed) audio or video sources.
- variable: variable bit rate (VBR) sources are typical of compressed speech and video because different parts of the speech or video signal have different amounts of redundancy and can be compressed to different bit rates.
- bursty: bursty data sources generate large amounts of information at instants of time in-between pauses where no information is generated. This is typical of systems involving human-machine interaction such as web surfing.

**Exercise 12:** Plot some sample data rate versus time curves for these three types of sources. What characteristics of a video source might result in a variable bit rate when it is compressed? (*Hint: what types of redundancy are there in video?*).

## Tolerance To Impairments

Data sinks vary in their tolerance to channel impairments such as errors, delay and variability of delay (delay “jitter”).

For example, computer data transmission systems usually require very low undetected error rates (e.g. one undetected error in decades) 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 13:** For each of the following communication systems identify the tolerance it is likely to have to errors and delay: a phone call between two people, “texting”, downloading a computer program, streaming a video over a computer network. What do you think might be the maximum tolerable delay for each?

**Exercise 14:** Highlight or underline each term where it is defined in these lecture notes.

	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	*	:	J 004A	Z 005A	j 006A	z 007A
B	[VT] 000B	[ESC] 001B	+	;	K 004B	[ 005B	k 006B	{ 007B
C	[FF] 000C	[FS] 001C	,	<	L 004C	\ 005C	l 006C	 007C
D	[CR] 000D	[GS] 001D	-	=	M 004D	] 005D	m 006D	} 007D
E	[SO] 000E	[RS] 001E	.	>	N 004E	^ 005E	n 006E	~ 007E
F	[SI] 000F	[US] 001F	/	?	O 004F	_ 005F	o 006F	[DEL] 007F