Data

This lecture describes a simple model of a communication system and then describes some characteristics of data.

Communication Systems Model

We can view a communication system as the sequence of:

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

Digital communication systems communicate information that is in the form of digits, typically binary digits or "bits". The term "data" sometimes means information other than digitally encoded speech and video. However, today the terms "data communications" and "digital communications" are often used interchangeably.

Exercise 1: Identify some digital communication systems you use every day. For each of these, identify the five components listed above.

Characteristics of Data Sources and Sinks

Data can come from various sources. In many cases the data represents digitized speech or video waveforms. In other cases it represents more abstract information such as text, images, or computer software.

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. This type of compression is called "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
- variable - variable bit rate (VBR) sources are typical of compressed speech and video (*why?*)
- bursty - bursty 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: For each of the systems in the previous exercise identify the data rate variability and the delay and error tolerance.

Data Rates

The rate at which the source generates data 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 data rate after the best possible compression has been applied is called the "information rate".

The transmitter converts the data to a waveform (the signal) for transmission over the channel. Often the transmitter uses groups of bits to select from one of several possible waveforms 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.

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. 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.

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.