Concurrent Computing (Computer Networks)

Daniel Page

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB. UK.
⟨Daniel.Page@bristol.ac.uk⟩

March 14, 2016

Keep in mind there are *two* PDFs available (of which this is the latter):

1. a PDF of examinable material used as lecture slides, and
2. a PDF of non-examinable, extra material:
   - the associated notes page may be pre-populated with extra, written explaination of material covered in lecture(s), plus
   - anything with a "grey'ed out" header/footer represents extra material which is useful and/or interesting but out of scope (and hence not covered).

Notes:

Notes:

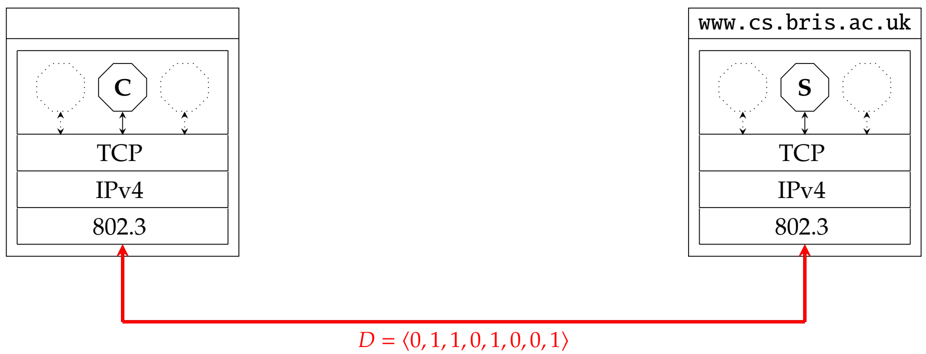
- **Goal**: investigate the **physical layer**, e.g.,

  1. communication media and signals,
  2. encoding and/or modulation,
  3. multiplexing, and
  4. efficiency metrics (and limits)

  st. we can transmit (unstructured) **bit-sequences** along a physical connection between two end-points.

Notes:

- Looking at the remit of the physical layer, one *could* view it roughly as the combination of a medium-independent sub-layer (e.g., the line or block coding scheme) and a medium-dependent sub-layer (e.g., sampling the medium itself). So, in a sense the goal is to form an (abstract), medium-independent using a (concrete) medium.
- The basic challenge here is as follows:
  - Within a micro-processor, the distance a signal needs to be propagated is *very* small; even for electronics vs. *micro*-electronics, distances are small.
  - At this scale, direct electrical connection (along a wire) can be used to drive voltage levels (e.g., and wlog. GND and V_{dd} to represent 0 and 1). Apart from at extreme distances or propagation speeds, there is seldom a need to consider analogue imperfections in the signal (e.g., noise): we sort of assume the wire communicates digital signals, even though in reality they are (somewhat) analogue.
  - At larger scales however, these issues *are* important and need to be addressed: the danger, in short, is that one cannot distinguish the signalling levels (e.g., GND and V_{dd}, so hence 0 and 1) from each other reliably.
- For simplicity, we focus on a high-level overview of transmission alone: it's vital to at least keep in mind that *decoding* and/or *de*modulation, plus *de*multiplexing will be required by the receiver (and some schemes are designed and optimised explicitly to simplify such tasks).



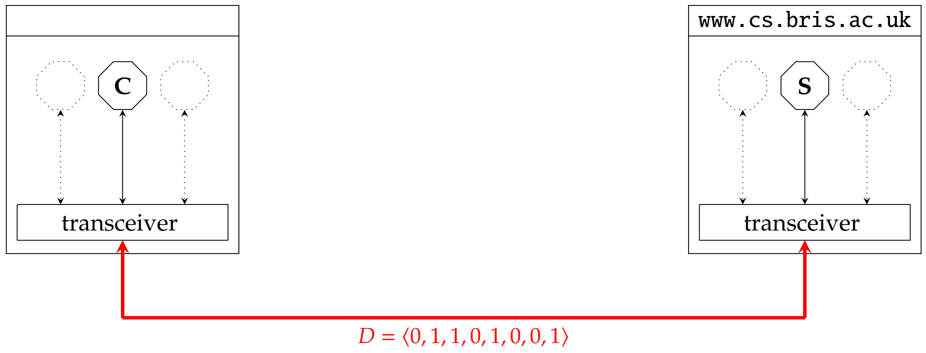
- **Goal**: investigate the **physical layer**, e.g.,

  1. communication media and signals,
  2. encoding and/or modulation,
  3. multiplexing, and
  4. efficiency metrics (and limits)

  st. we can transmit (unstructured) **bit-sequences** along a physical connection between two end-points.

Notes:

- Looking at the remit of the physical layer, one *could* view it roughly as the combination of a medium-independent sub-layer (e.g., the line or block coding scheme) and a medium-dependent sub-layer (e.g., sampling the medium itself). So, in a sense the goal is to form an (abstract), medium-independent using a (concrete) medium.
- The basic challenge here is as follows:
  - Within a micro-processor, the distance a signal needs to be propagated is *very* small; even for electronics vs. *micro*-electronics, distances are small.
  - At this scale, direct electrical connection (along a wire) can be used to drive voltage levels (e.g., and wlog. GND and V_{dd} to represent 0 and 1). Apart from at extreme distances or propagation speeds, there is seldom a need to consider analogue imperfections in the signal (e.g., noise): we sort of assume the wire communicates digital signals, even though in reality they are (somewhat) analogue.
  - At larger scales however, these issues *are* important and need to be addressed: the danger, in short, is that one cannot distinguish the signalling levels (e.g., GND and V_{dd}, so hence 0 and 1) from each other reliably.
- For simplicity, we focus on a high-level overview of transmission alone: it's vital to at least keep in mind that *decoding* and/or *de*modulation, plus *de*multiplexing will be required by the receiver (and some schemes are designed and optimised explicitly to simplify such tasks).



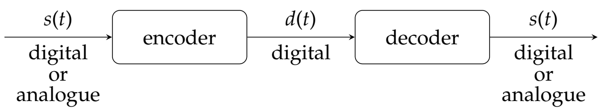
- Goal: investigate the **physical layer**, e.g.,
  1. communication media and signals,
  2. encoding and/or modulation,
  3. multiplexing, and
  4. efficiency metrics (and limits)

  st. we can transmit (unstructured) **bit-sequences** along a physical connection between two end-points.

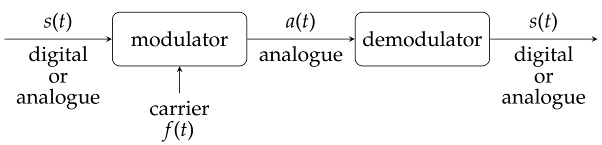
Notes:

- Looking at the remit of the physical layer, one *could* view it roughly as the combination of a medium-independent sub-layer (e.g., the line or block coding scheme) and a medium-dependent sub-layer (e.g., sampling the medium itself). So, in a sense the goal is to form an (abstract), medium-independent using a (concrete) medium.
- The basic challenge here is as follows:
  - Within a micro-processor, the distance a signal needs to be propagated is *very* small; even for electronics vs. *micro*-electronics, distances are small.
  - At this scale, direct electrical connection (along a wire) can be used to drive voltage levels (e.g., and wlog. GND and V_{dd} to represent 0 and 1). Apart from at extreme distances or propagation speeds, there is seldom a need to consider analogue imperfections in the signal (e.g., noise): we sort of assume the wire communicates digital signals, even though in reality they are (somewhat) analogue.
  - At larger scales however, these issues *are* important and need to be addressed: the danger, in short, is that one cannot distinguish the signalling levels (e.g., GND and V_{dd}, so hence 0 and 1) from each other reliably.
- For simplicity, we focus on a high-level overview of transmission alone: it's vital to at least keep in mind that *decoding* and/or *de*modulation, plus *de*multiplexing will be required by the receiver (and some schemes are designed and optimised explicitly to simplify such tasks).

- Idea:
  - we have some digital input (i.e., our data), so can
    1. directly **encode** it, i.e.,



       via a digital signalling scheme, or
    2. use it to **modulate** a **carrier signal**, i.e.,



       via an analogue signalling scheme

    to produce an (digital or analogue) output signal,
  - so can then transmit that signal along a communication medium (e.g., a wire), and
  - the resulting behaviour has a clear theoretical basis.

Notes:

- There are four cases to consider
- digital input, digital signal,
- digital input, analogue signal,
- analogue input, digital signal, and
- analogue input, analogue signal.
  Although we deal with the first two only (we want to communicate a sequence of bits), it is important to carefully distinguish between digital from analogue signals: the latter allows continuous variation of the signal level over time, whereas (from an idealised perspective, i.e., allowing instantaneous changes) the former allows only discrete amplitude levels.
- Although the terms are somewhat interchangable, strictly speaking
  - **encoding** translates an input (either digital *or* analogue) into a digital output (and vice versa for **decode**),
  - **modulation** translates an input (either digital *or* analogue) into a analogue output (and vice versa for **demodulate**).
- Within the same context, the terms **baseband** and **passband** also appear; these can be quite confusing.
  - A baseband signal requires a wide range of frequencies, i.e., lower-bound 0 to upper-bound f_{max}; this is often the case for a digital input, since the "sharp" transitions between 0 and 1 demand more frequencies for a reliable representation. As such, a baseband modulation scheme represents data via use of (relatively) more bandwidth.
  - A passband signal requires a narrow range of frequencies, i.e., lower-bound f_{min} to upper-bound f_{max}; the term stems from *the* passband between high- and low-pass filters (i.e., the range of frequencies allowed to propagate without attenuation). As such, a passband modulation scheme represents data via use of (relatively) more bandwidth.
- The act of **bandlimiting** is essentially filtering an input (i.e., applying a **bandpass filter**) st. frequencies outside the passband are eliminated.
- The choice between options depends somewhat on the context: in some cases we might rather transmit the digital input directly (cf. a sequence of low and high voltage pulses to represent 0 and 1 in TIA-232-F), in others it may be perferable to transmit a representation of the encoded baseband signal (even if it uses more bandwidth) to avoid having to form an equivalent passband representation.

“Communication Theory in 10 minutes” (1)

Definition (**sinusoid**)

The sinusoidal function

$$s(t) = A \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

is periodic, and parameterised by

1. an **amplitude** A (which is the maximum deviation of $s(t)$ from 0),
2. a **frequency** f (which is inversely proportional to the period, which is normally termed the **wavelength** λ), and
3. a **phase** (or offset) φ (which basically dictates where in the cycle the wave is at time $t = 0$).

Notes:

- The fact such a function is periodic means that $s(t) = s(t + n \cdot \lambda)$ for any integer n; since $f = \frac{1}{\lambda}$, it follows that larger frequency implies smaller wavelength and hence shorter period.
- It is common to additionally see **angular frequency** $\omega = 2 \cdot \pi \cdot f$ used as a parameter instead: this is measured in radians.

“Communication Theory in 10 minutes” (1)

Definition (**sinusoid**)

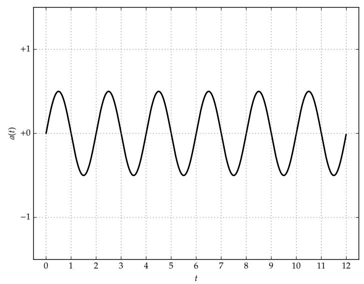
The sinusoidal function

$$s(t) = A \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

is periodic, and parameterised by

1. an **amplitude** A (which is the maximum deviation of $s(t)$ from 0),
2. a **frequency** f (which is inversely proportional to the period, which is normally termed the **wavelength** λ), and
3. a **phase** (or offset) φ (which basically dictates where in the cycle the wave is at time $t = 0$).

By evaluating over a time period (i.e., over a range of t), such a wave can be visualised as a **waveform**, e.g.,



where $A = 0.5, f = 0.5, \varphi = 0$.

Notes:

- The fact such a function is periodic means that $s(t) = s(t + n \cdot \lambda)$ for any integer n; since $f = \frac{1}{\lambda}$, it follows that larger frequency implies smaller wavelength and hence shorter period.
- It is common to additionally see **angular frequency** $\omega = 2 \cdot \pi \cdot f$ used as a parameter instead: this is measured in radians.

"Communication Theory in 10 minutes" (1)

Definition (**sinusoid**)

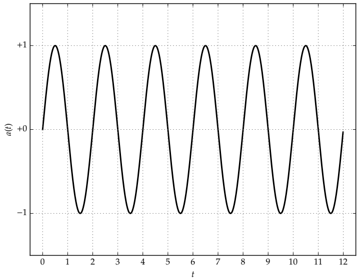
The sinusoidal function

$$s(t) = A \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

is periodic, and parameterised by

1. an **amplitude** A (which is the maximum deviation of $s(t)$ from 0),
2. a **frequency** f (which is inversely proportional to the period, which is normally termed the **wavelength** λ), and
3. a **phase** (or offset) φ (which basically dictates where in the cycle the wave is at time $t = 0$).

By evaluating over a time period (i.e., over a range of t), such a wave can be visualised as a **waveform**, e.g.,



where $A = 1.0, f = 0.5, \varphi = 0$.

Notes:

- The fact such a function is periodic means that $s(t) = s(t + n \cdot \lambda)$ for any integer n; since $f = \frac{1}{\lambda}$, it follows that larger frequency implies smaller wavelength and hence shorter period.
- It is common to additionally see **angular frequency** $\omega = 2 \cdot \pi \cdot f$ used as a parameter instead: this is measured in radians.

"Communication Theory in 10 minutes" (1)

Definition (**sinusoid**)

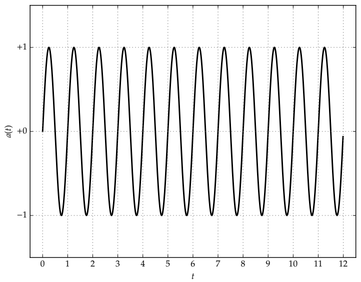
The sinusoidal function

$$s(t) = A \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

is periodic, and parameterised by

1. an **amplitude** A (which is the maximum deviation of $s(t)$ from 0),
2. a **frequency** f (which is inversely proportional to the period, which is normally termed the **wavelength** λ), and
3. a **phase** (or offset) φ (which basically dictates where in the cycle the wave is at time $t = 0$).

By evaluating over a time period (i.e., over a range of t), such a wave can be visualised as a **waveform**, e.g.,



where $A = 1.0, f = 1.0, \varphi = 0$.

Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks)
git # 3627080 @ 2016-03-11
University of BRISTOL

Notes:

- The fact such a function is periodic means that $s(t) = s(t + n \cdot \lambda)$ for any integer n; since $f = \frac{1}{\lambda}$, it follows that larger frequency implies smaller wavelength and hence shorter period.
- It is common to additionally see **angular frequency** $\omega = 2 \cdot \pi \cdot f$ used as a parameter instead: this is measured in radians.

Definition (**sinusoid**)

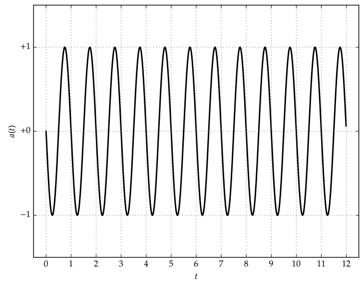
The sinusoidal function

$$s(t) = A \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

is periodic, and parameterised by

1. an **amplitude** A (which is the maximum deviation of $s(t)$ from 0),
2. a **frequency** f (which is inversely proportional to the period, which is normally termed the **wavelength** λ), and
3. a **phase** (or offset) φ (which basically dictates where in the cycle the wave is at time $t = 0$).

By evaluating over a time period (i.e., over a range of t), such a wave can be visualised as a **waveform**, e.g.,



where $A = 1.0$, $f = 1.0$, $\varphi = \pi$.

Notes:

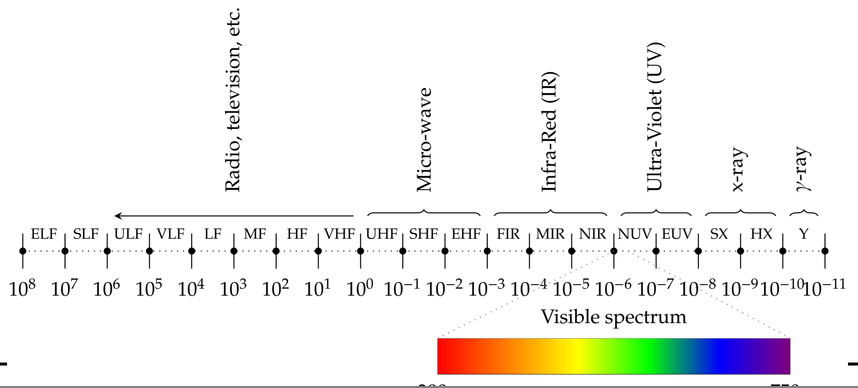
- The fact such a function is periodic means that $s(t) = s(t + n \cdot \lambda)$ for any integer n; since $f = \frac{1}{\lambda}$, it follows that larger frequency implies smaller wavelength and hence shorter period.
- It is common to additionally see **angular frequency** $\omega = 2 \cdot \pi \cdot f$ used as a parameter instead: this is measured in radians.

Definition (**Electro-Magnetic (EM) radiation**)

Electro-Magnetic (EM) radiation behaves like a wave: it

1. has orthogonal electric and magnetic components,
2. has a direction of **propagation**,
3. propagates through a given medium at a speed of $v = f \cdot \lambda$ where f and λ are the frequency and wavelength, and
4. interacts somehow with the medium as it does so.

One can visualise the **EM spectrum** by inspecting various values of the wavelength λ, e.g.,



Notes:

- Although we have implicitly focused on wires, other mediums are clearly also important. A simple classification might divide them into three, i.e.,
  1. wired media,
  2. wireless (or radio) media, and
  3. fibre optic media.

  It is vital to remember that each of their characteristics differ, and this will influence other design decisions. For example, whether or not a signal can be transmitted in a chosen direction (or "aimed") will differ between types of medium: both **directed** (or **guided**, e.g., wired) and also **undirected** (or **unguided**, e.g., wireless) cases clearly exist, and may or may not be useful in a specific context.

Definition (Fourier analysis [4])

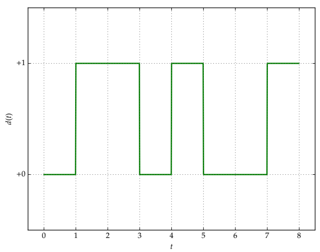
Fourier analysis allows us to represent a signal as an (infinite) sum of sinusoids:

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n)$$

The resulting **Fourier series** (or **Fourier expansion**) $s(t)$ typically makes use of **Fourier coefficients** a_n and b_n for $1 \leq n < N$ wrt. some (finite) bound N.

Notes:

- Eh?!



Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

  where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

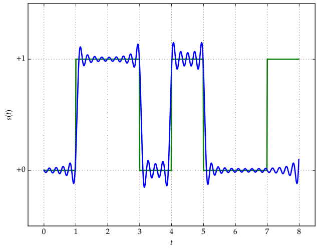
  and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

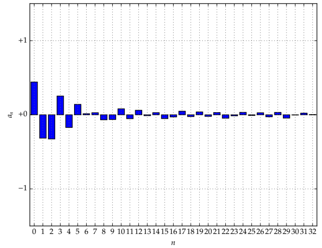
- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

"Communication Theory in 10 minutes" (4)

- Eh?!



$\equiv$



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

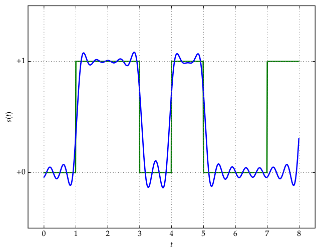
and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

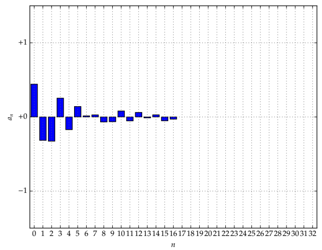
- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or "bias") to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is "good enough" to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

"Communication Theory in 10 minutes" (4)

- Eh?!



$\equiv$



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

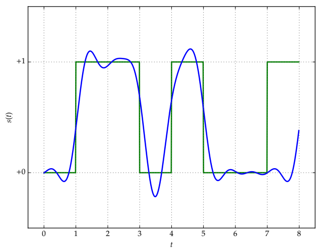
$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

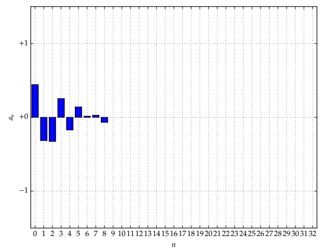
$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or "bias") to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is "good enough" to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

- Eh?!



$\equiv$



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{rcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

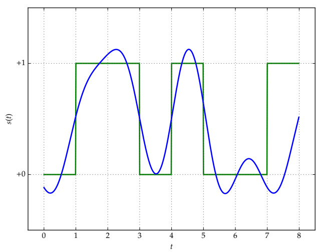
and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

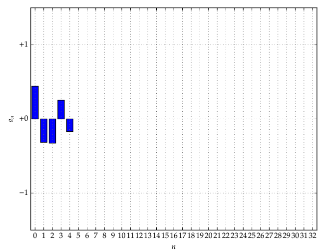
- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or "bias") to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is "good enough" to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

"Communication Theory in 10 minutes" (4)

- Eh?!



$\equiv$



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{rcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

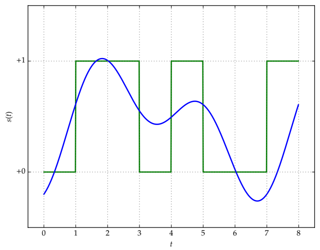
$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or "bias") to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is "good enough" to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

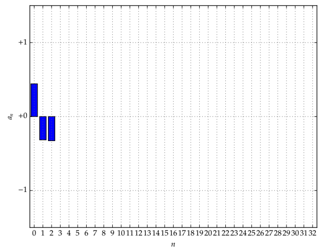
Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11 University of BRISTOL

“Communication Theory in 10 minutes” (4)

- Eh?!



$\equiv$



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

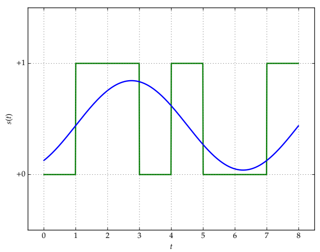
and using $\omega = \frac{2\cdot\pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

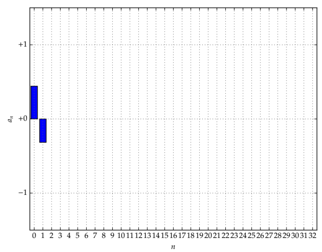
- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

“Communication Theory in 10 minutes” (4)

- Eh?!



$\equiv$



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

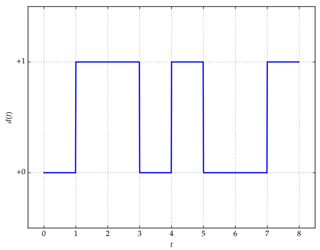
and using $\omega = \frac{2\cdot\pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

“Communication Theory in 10 minutes” (4)

- Eh?!



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation,
- *but*, when transmitted, the signal will still be
  1. **delayed**, since it takes time to propagate,
  2. **attenuated**, meaning it becomes weaker,
  3. subjected to bandwidth degradation, and
  4. subjected to the influence of **noise**.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

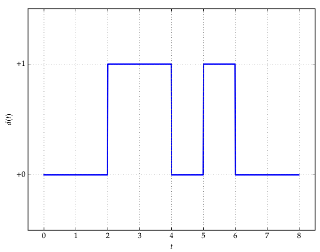
and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

“Communication Theory in 10 minutes” (4)

- Eh?!



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation,
- *but*, when transmitted, the signal will still be
  1. **delayed**, since it takes time to propagate,
  2. **attenuated**, meaning it becomes weaker,
  3. subjected to bandwidth degradation, and
  4. subjected to the influence of **noise**.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

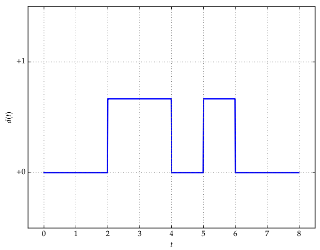
and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

Daniel Page (daniel.page@bristol.ac.uk)
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11
University of BRISTOL

- Eh?!



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation,
- *but*, when transmitted, the signal will still be
  1. **delayed**, since it takes time to propagate,
  2. **attenuated**, meaning it becomes weaker,
  3. subjected to bandwidth degradation, and
  4. subjected to the influence of **noise**.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

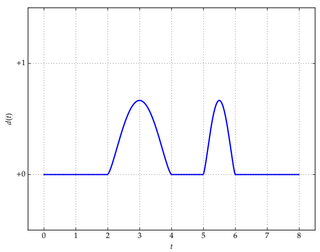
and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

“Communication Theory in 10 minutes” (4)

- Eh?!



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation,
- *but*, when transmitted, the signal will still be
  1. **delayed**, since it takes time to propagate,
  2. **attenuated**, meaning it becomes weaker,
  3. subjected to bandwidth degradation, and
  4. subjected to the influence of **noise**.

Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11 University of BRISTOL

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

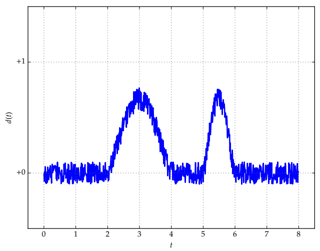
and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or “bias”) to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is “good enough” to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

"Communication Theory in 10 minutes" (4)

- Eh?!



- larger **bandwidth** (i.e., wider range of frequencies) normally allows a higher **fidelity** representation,
- *but*, when transmitted, the signal will still be
  1. **delayed**, since it takes time to propagate,
  2. **attenuated**, meaning it becomes weaker,
  3. subjected to bandwidth degradation, and
  4. subjected to the influence of **noise**.

Notes:

- Use of n as an index is slightly annoying, but also standard: probably it is used to avoid i, and hence confusion wrt. complex numbers.
- The representation shown is the end result of manipulating a simpler starting point

$$s(t) = A_0 + \sum_{n=1}^{\infty} A_n \cdot \sin(\omega \cdot t \cdot n + \phi_n)$$

where ω is the angular frequency of $s(t)$, and ϕ_n is the phase (or offset) of the n-th term. By setting

$$\begin{array}{lcl} a_n & = & A_n \cdot \sin(\phi_n) \\ b_n & = & A_n \cdot \cos(\phi_n) \end{array}$$

and using $\omega = \frac{2 \cdot \pi}{\lambda}$ and $f = \frac{1}{\lambda}$ we end up with

$$s(t) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(2\pi \cdot f \cdot t \cdot n) + b_n \cdot \sin(2\pi \cdot f \cdot t \cdot n).$$

- The n-th term is called the **n-th harmonic**, with the special-case of $n = 1$ called the **fundamental harmonic**,
- The leading term $\frac{a_0}{2}$ is a special-case which captures the mean amplitude of $s(t)$ over the period; this effectively acts as a fixed offset (or "bias") to the representation.
- The reason for bandwidth degradation is that frequencies above some threshold will be *highly* attenuated: this effectively means they are not present, or equivalently that their amplitudes are low enough to be viewed as zero.
- Interestingly, there comes a point where an N-term Fourier series is "good enough" to represent a given signal: using $N' > N$ could then be viewed as wasteful wrt. bandwidth, because the additional terms add no extra fidelity.

"Communication Theory in 10 minutes" (5)

Definition (signalling levels)

When sampled at a given instance in time, a signal will take one of l **signalling levels**; this means each **symbol** transmitted will take one of l values. Note that $l > 2$ implies the ability to transmit *more* than 1 bit of information per symbol.

Definition (modulation rate)

The **modulation rate** measures how quickly (i.e., how often per unit of time) the channel can change (or transition, which may be termed a **signalling event**) between signalling levels; this of course determines the (minimum) **symbol period**.

Notes:

Definition (**signalling rate**)

The **signalling rate** (or **symbol rate**, or **baud rate**) of a channel measures how many symbols it can transmit per unit of time. The associated **data signalling rate** (or just **data rate**, or **gross bit rate**) measures this in bits per second.

Notes:

- If the signalling rate is r, then the data rate will be
$$r \cdot \log_2(l) \text{bit s}^{-1}$$
which, for $l = 2$ signalling levels implies that the signalling and data rate are the same (which should make sense: with 2 symbols, the symbols are bits and hence the signalling rate will already be captured in the same units as the data rate).

Definition (**Signal-to-Noise ratio (SNR)**)

The **Signal-to-Noise ratio (SNR)** is a measure of how strong a signal is relative to the noise level: this determines the number of signalling levels we can reliably distinguish between.

Definition (**Nyquist limit** [5] and **Shannon capacity** [6])

Consider a communication channel whose bandwidth, signal and noise strengths are denoted H, S and N. The **Nyquist limit** ignores noise: it tells us that

- the maximum signalling rate is $2 \cdot H$, so
- if there are l signalling levels, then we can write the sigalling rate (i.e., gross bit rate) as $2 \cdot H \cdot \log_2(l) \text{bit s}^{-1}$.

The **Shannon capacity** of the same channel now *with* (Gaussian) noise, is

$$H \cdot \log_2(1 + \frac{S}{N}) \text{bit s}^{-1},$$

meaning the lower the SNR (e.g., the higher the noise) the fewer bits per second we can reliably communicate.

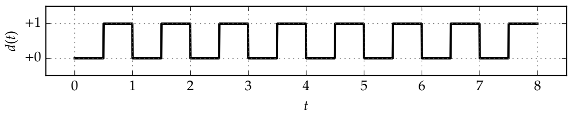
Notes:

Encoding/Modulation (1) – Digital signalling

Original data

0 1 1 0 1 0 0 1

Clock signal



Notes:

- The acronyms used (RZ, NRZ, and NRZ-I etc.) are somewhat archaic, and so quite unfriendly. The one exception is Manchester encoding, whose meaning is much less opaque: the scheme was invented in around 1949 at the University of Manchester (for use within their Mark 1 computer); you may alternatively see it described as **Manchester Phase Encoding (MPE)**, which hints at the fact it is sort of a digital version of the analogue PSK scheme introduced later.
- The original description of Manchester encoding used

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  rather than

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 0 \\ 0 \rightarrow 1 & \mapsto & 1 \end{array}$$

  The latter is adopted by 802.3, so we use it here.
- This list of example isn't a limit by any means: there are *many* other schemes. For example, it is easy to imagine other **differential** cases: a Manchester encoding where

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  is replaced by

$$\begin{array}{rcc} \text{transition wrt. previous bit} & \mapsto & 1 \\ \text{no transition wrt. previous bit} & \mapsto & 0 \end{array}$$

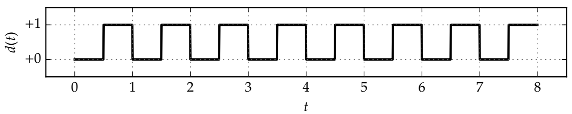
  is obviously possible, for instance. Note that by (no) transition we mean at the first clock half-period; there is *still* a transmission at the second half-period to ensure the same property of not needing a clock.
- RZ can *sort* of be described as having two signalling levels, with the "rest" level (normally denoted as 0, hence the name) implicitly between the them.

Encoding/Modulation (1) – Digital signalling

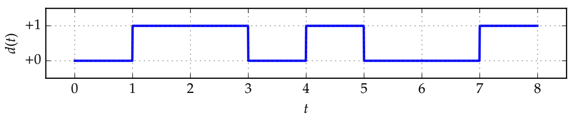
Original data

0 1 1 0 1 0 0 1

Clock signal



Non-Return to Zero (NRZ)



Notes:

- The acronyms used (RZ, NRZ, and NRZ-I etc.) are somewhat archaic, and so quite unfriendly. The one exception is Manchester encoding, whose meaning is much less opaque: the scheme was invented in around 1949 at the University of Manchester (for use within their Mark 1 computer); you may alternatively see it described as **Manchester Phase Encoding (MPE)**, which hints at the fact it is sort of a digital version of the analogue PSK scheme introduced later.
- The original description of Manchester encoding used

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  rather than

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 0 \\ 0 \rightarrow 1 & \mapsto & 1 \end{array}$$

  The latter is adopted by 802.3, so we use it here.
- This list of example isn't a limit by any means: there are *many* other schemes. For example, it is easy to imagine other **differential** cases: a Manchester encoding where

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  is replaced by

$$\begin{array}{rcc} \text{transition wrt. previous bit} & \mapsto & 1 \\ \text{no transition wrt. previous bit} & \mapsto & 0 \end{array}$$

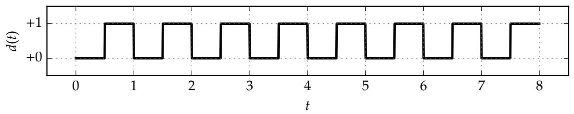
  is obviously possible, for instance. Note that by (no) transition we mean at the first clock half-period; there is *still* a transmission at the second half-period to ensure the same property of not needing a clock.
- RZ can *sort* of be described as having two signalling levels, with the "rest" level (normally denoted as 0, hence the name) implicitly between the them.

Encoding/Modulation (1) – Digital signalling

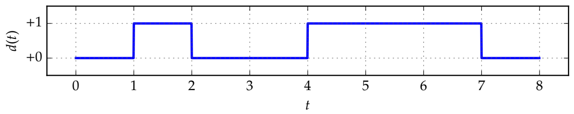
Original data

0 1 1 0 1 0 0 1

Clock signal



Non-Return to Zero Inverted (NRZ-I)



Notes:

- The acronyms used (RZ, NRZ, and NRZ-I etc.) are somewhat archaic, and so quite unfriendly. The one exception is Manchester encoding, whose meaning is much less opaque: the scheme was invented in around 1949 at the University of Manchester (for use within their Mark 1 computer); you may alternatively see it described as **Manchester Phase Encoding (MPE)**, which hints at the fact it is sort of a digital version of the analogue PSK scheme introduced later.
- The original description of Manchester encoding used

$$\begin{array}{lcl} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  rather than

$$\begin{array}{lcl} 1 \rightarrow 0 & \mapsto & 0 \\ 0 \rightarrow 1 & \mapsto & 1 \end{array}$$

  The latter is adopted by 802.3, so we use it here.
- This list of example isn't a limit by any means: there are *many* other schemes. For example, it is easy to imagine other **differential** cases: a Manchester encoding where

$$\begin{array}{lcl} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  is replaced by

$$\begin{array}{rcl} \text{transition wrt. previous bit} & \mapsto & 1 \\ \text{no transition wrt. previous bit} & \mapsto & 0 \end{array}$$

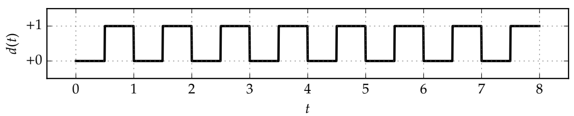
  is obviously possible, for instance. Note that by (no) transition we mean at the first clock half-period; there is *still* a transmission at the second half-period to ensure the same property of not needing a clock.
- RZ can *sort* of be described as having two signalling levels, with the "rest" level (normally denoted as 0, hence the name) implicitly between the them.

Encoding/Modulation (1) – Digital signalling

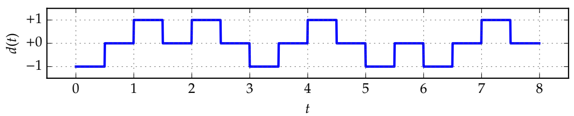
Original data

0 1 1 0 1 0 0 1

Clock signal



Return to Zero (RZ)



Notes:

- The acronyms used (RZ, NRZ, and NRZ-I etc.) are somewhat archaic, and so quite unfriendly. The one exception is Manchester encoding, whose meaning is much less opaque: the scheme was invented in around 1949 at the University of Manchester (for use within their Mark 1 computer); you may alternatively see it described as **Manchester Phase Encoding (MPE)**, which hints at the fact it is sort of a digital version of the analogue PSK scheme introduced later.
- The original description of Manchester encoding used

$$\begin{array}{lcl} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  rather than

$$\begin{array}{lcl} 1 \rightarrow 0 & \mapsto & 0 \\ 0 \rightarrow 1 & \mapsto & 1 \end{array}$$

  The latter is adopted by 802.3, so we use it here.
- This list of example isn't a limit by any means: there are *many* other schemes. For example, it is easy to imagine other **differential** cases: a Manchester encoding where

$$\begin{array}{lcl} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  is replaced by

$$\begin{array}{rcl} \text{transition wrt. previous bit} & \mapsto & 1 \\ \text{no transition wrt. previous bit} & \mapsto & 0 \end{array}$$

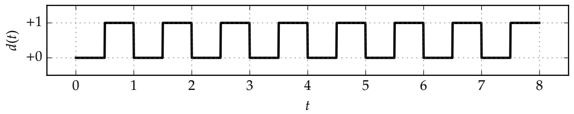
  is obviously possible, for instance. Note that by (no) transition we mean at the first clock half-period; there is *still* a transmission at the second half-period to ensure the same property of not needing a clock.
- RZ can *sort* of be described as having two signalling levels, with the "rest" level (normally denoted as 0, hence the name) implicitly between the them.

Encoding/Modulation (1) – Digital signalling

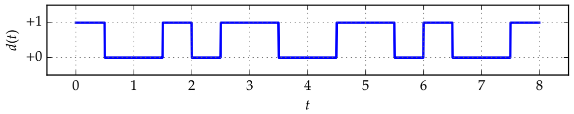
Original data

0 1 1 0 1 0 0 1

Clock signal



Manchester (per 802.3)



Notes:

- The acronyms used (RZ, NRZ, and NRZ-I etc.) are somewhat archaic, and so quite unfriendly. The one exception is Manchester encoding, whose meaning is much less opaque: the scheme was invented in around 1949 at the University of Manchester (for use within their Mark 1 computer); you may alternatively see it described as **Manchester Phase Encoding (MPE)**, which hints at the fact it is sort of a digital version of the analogue PSK scheme introduced later.
- The original description of Manchester encoding used

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  rather than

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 0 \\ 0 \rightarrow 1 & \mapsto & 1 \end{array}$$

  The latter is adopted by 802.3, so we use it here.
- This list of example isn't a limit by any means: there are *many* other schemes. For example, it is easy to imagine other **differential** cases: a Manchester encoding where

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  is replaced by

$$\begin{array}{rcc} \text{transition wrt. previous bit} & \mapsto & 1 \\ \text{no transition wrt. previous bit} & \mapsto & 0 \end{array}$$

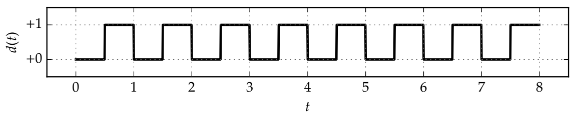
  is obviously possible, for instance. Note that by (no) transition we mean at the first clock half-period; there is *still* a transmission at the second half-period to ensure the same property of not needing a clock.
- RZ can *sort* of be described as having two signalling levels, with the "rest" level (normally denoted as 0, hence the name) implicitly between the them.

Encoding/Modulation (1) – Digital signalling

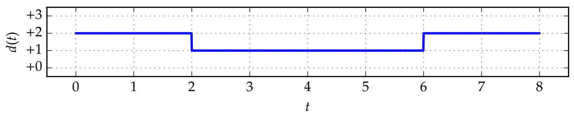
Original data

0 1 1 0 1 0 0 1

Clock signal



l-ary (for $l = 4$)



Notes:

- The acronyms used (RZ, NRZ, and NRZ-I etc.) are somewhat archaic, and so quite unfriendly. The one exception is Manchester encoding, whose meaning is much less opaque: the scheme was invented in around 1949 at the University of Manchester (for use within their Mark 1 computer); you may alternatively see it described as **Manchester Phase Encoding (MPE)**, which hints at the fact it is sort of a digital version of the analogue PSK scheme introduced later.
- The original description of Manchester encoding used

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  rather than

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 0 \\ 0 \rightarrow 1 & \mapsto & 1 \end{array}$$

  The latter is adopted by 802.3, so we use it here.
- This list of example isn't a limit by any means: there are *many* other schemes. For example, it is easy to imagine other **differential** cases: a Manchester encoding where

$$\begin{array}{ccc} 1 \rightarrow 0 & \mapsto & 1 \\ 0 \rightarrow 1 & \mapsto & 0 \end{array}$$

  is replaced by

$$\begin{array}{rcc} \text{transition wrt. previous bit} & \mapsto & 1 \\ \text{no transition wrt. previous bit} & \mapsto & 0 \end{array}$$

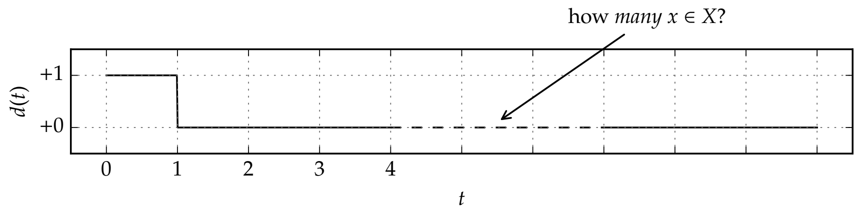
  is obviously possible, for instance. Note that by (no) transition we mean at the first clock half-period; there is *still* a transmission at the second half-period to ensure the same property of not needing a clock.
- RZ can *sort* of be described as having two signalling levels, with the "rest" level (normally denoted as 0, hence the name) implicitly between the them.

Encoding/Modulation (2) – Digital signalling

- ... or, to summarise, we have something like the following

Scheme	Signalling levels	Modulation rate	Self clocked?	Differential ?	Runs of $x \in X$
NRZ	2	r	$\times$	$\times$	$X = \{0, 1\}$
NRZ-I	2	r	$\times$	$\checkmark$	$X = \{0\}$
RZ	2(ish)	$r \cdot 2$	$\checkmark$	$\times$	$X = \emptyset$
Manchester	2	$r \cdot 2$	$\checkmark$	$\times$	$X = \emptyset$
l-ary	l	$r / \log_2(l)$	$\times$	$\times$	$X = \{0, 1, \ldots, n-1\}$

where the last column suggests a problem, i.e.,



Notes:

- The idea here is that if we have a (very) long sequence of repeated signals, then a) this becomes hard to distinguish from a "dead" (or disconnected) channel, and b) even with a clock, synchronisation becomes hard (e.g., due to drift).

Encoding/Modulation (3) – Digital signalling

- (A) solution:
  1. *pre*-encode the data, e.g., by using a **block code** such as **4B/5B** where

4-bit data word	5-bit code word
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

  2. the combination of say 4B/5B plus NRZ-I means
     - there is never a long sequence of transmitted 0 or 1, *and*
     - the overhead is *lower* than using alternatives such as Manchester.

Notes:

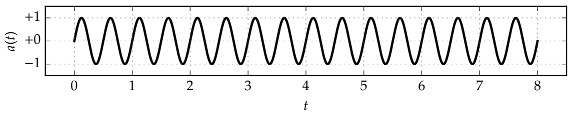
- 4B/5B is one among *many* solutions to this problem; others include alternative block coding schemes (e.g., 6B/8B), or "scrambling" which basically XORs the data with a pseudo-random bit sequence (which, since it is random-ish, ensures bits are 0 or 1 roughly with probability 1/2).

Encoding/Modulation (4) – Analogue signalling

Original data

0 1 1 0 1 0 0 1

Carrier signal



Notes:

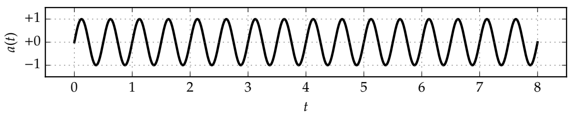
- In reality, passband modulation is *much* more complex than the high-level overview presented here: although this enough to capture the intuition, a more accurate overview probably demands a dedicated unit (Wikipedia [?] provides a good starting point for those whose are interested).
- Each of the schemes generalises to some extent. For example, it is more accurate to talk about n-PSK where 1-PSK (or BPSK) relates to two signalling levels (or in this cases phases) versus 4-PSK (or QPSK) which has more.

Encoding/Modulation (4) – Analogue signalling

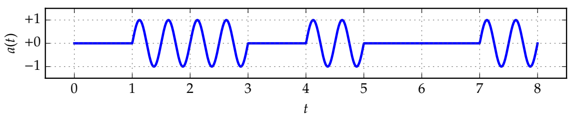
Original data

0 1 1 0 1 0 0 1

Carrier signal



Amplitude-Shift Keying (ASK)



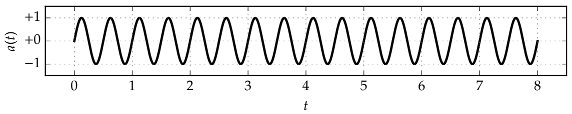
Notes:

- In reality, passband modulation is *much* more complex than the high-level overview presented here: although this enough to capture the intuition, a more accurate overview probably demands a dedicated unit (Wikipedia [?] provides a good starting point for those whose are interested).
- Each of the schemes generalises to some extent. For example, it is more accurate to talk about n-PSK where 1-PSK (or BPSK) relates to two signalling levels (or in this cases phases) versus 4-PSK (or QPSK) which has more.

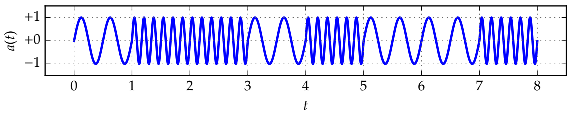
Encoding/Modulation (4) – Analogue signalling

Original data: 0 1 1 0 1 0 0 1

Carrier signal



Frequency-Shift Keying (FSK)



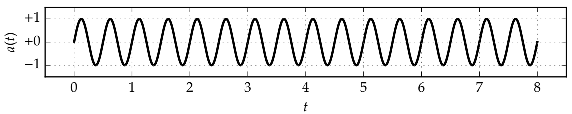
Notes:

- In reality, passband modulation is *much* more complex than the high-level overview presented here: although this enough to capture the intuition, a more accurate overview probably demands a dedicated unit (Wikipedia [?] provides a good starting point for those whose are interested).
- Each of the schemes generalises to some extent. For example, it is more accurate to talk about n-PSK where 1-PSK (or BPSK) relates to two signalling levels (or in this cases phases) versus 4-PSK (or QPSK) which has more.

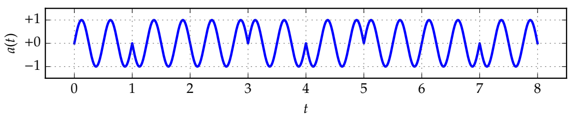
Encoding/Modulation (4) – Analogue signalling

Original data: 0 1 1 0 1 0 0 1

Carrier signal



Phase-Shift Keying (PSK)



Notes:

- In reality, passband modulation is *much* more complex than the high-level overview presented here: although this enough to capture the intuition, a more accurate overview probably demands a dedicated unit (Wikipedia [?] provides a good starting point for those whose are interested).
- Each of the schemes generalises to some extent. For example, it is more accurate to talk about n-PSK where 1-PSK (or BPSK) relates to two signalling levels (or in this cases phases) versus 4-PSK (or QPSK) which has more.

Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11 University of BRISTOL

Multiplexing (1)

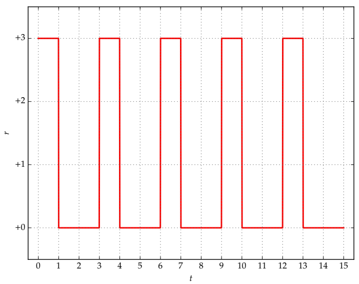
- Problem: what if we have 1 communication medium, and m streams of data (each of n symbols say)?
- (A) solution: statically **(de)multiplex** the streams, via
  1. **Time-Division Multiplexing (TDM)**, st. time is divided into "slots" and then allocated on a round-robin basis to the streams, or
  2. **Frequency-Division Multiplexing (FDM)**, where each stream is "shifted" into a different range of frequencies.

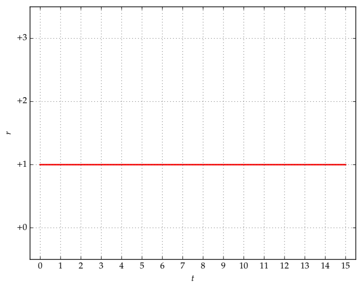
  or, from the perspective of a given stream ...

Notes:

Multiplexing (2)

- ... we can visualise utilisation as





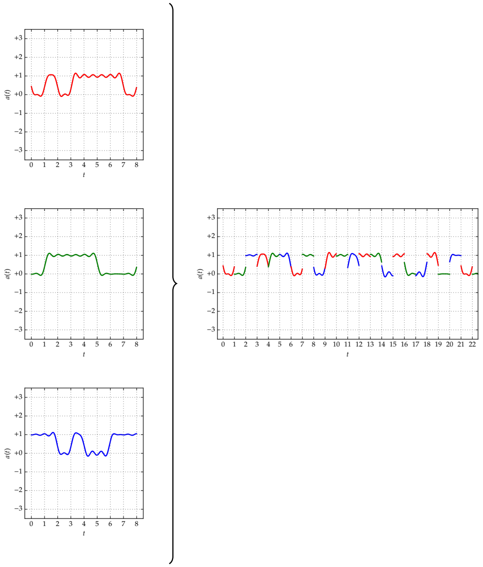
st. the stream either

1. has access to *all* of the available bandwidth, but for *part* of the time, or
2. has access to *part* of the available bandwidth, but for *all* of the time.

Notes:

- At face value, TDM and FDM both provide the same *overall* bandwidth for a given stream: the area under either graphs (i.e., the bandwidth used over time) will be the same. However, some more subtle trade-offs are evident: on one hand TDM may be more complex to realise (we need the transmitted and reciever to be synchronised, so they know which time slot is active), but, on the other hand, when a given stream is transmitted it has access to a higher bandwidth (so might be subject to less of a delay).

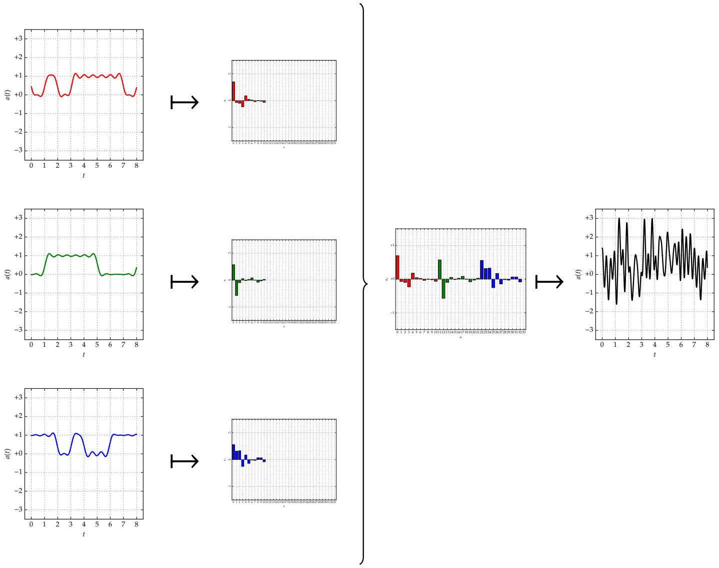
Multiplexing (3) – TDM



Notes:

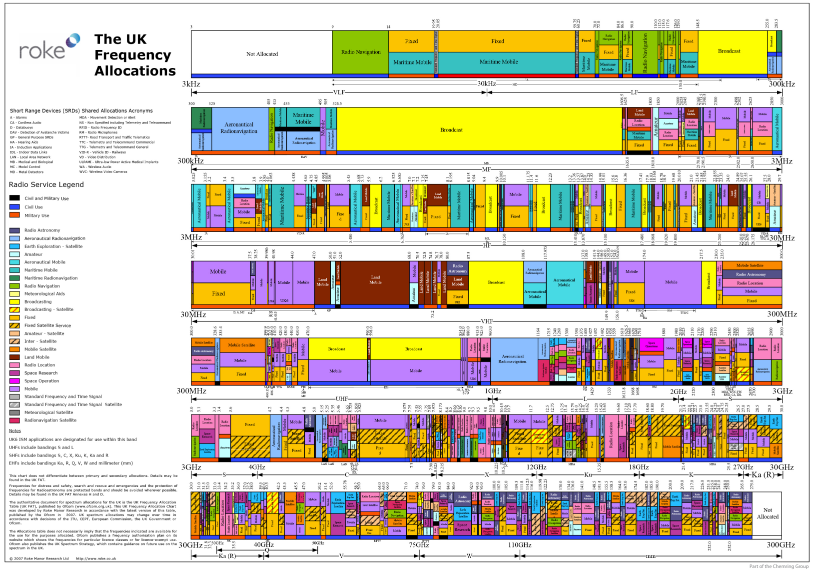
- Although they are not shown, it is common to separate each time slot using a **guard period** (where nothing is transmitted) so demultiplexing is easier.

Multiplexing (4) – FDM



Notes:

Multiplexing (5) – FDM



http://www.roke.co.uk/download/datasheets/uk-frequency-allocations.pdf

Notes:

Metrics (1)

- **Goal**: assess a communication channel using a suitable quality metric.

Notes:

- A (non-exhaustive) selection of characteristics we could use to describe a given channel include:

width	$\Longrightarrow$	serial, parallel
direction	$\Longrightarrow$	simplex, half-duplex full-duplex
timing	$\Longrightarrow$	synchronous, asynchronous
rate	$\Longrightarrow$	signalling rate, modulation rate
multiplicity	$\Longrightarrow$	unicast, multicast, broadcast
efficiency	$\Longrightarrow$	latency, bandwidth, throughput
reliability	$\Longrightarrow$	noise/error rate and type

  You may see the term **channel model** used: this is essentially a more abstract description of the channel, in terms of similar characteristics (e.g., noise).

- If communication between two end-points is described as **duplex**, it can occur in *both* directions; **simplex** communication is one-way only. Further to this, one can define

  1. **full-duplex**, i.e., both directions simultaneously, or
  2. **half-duplex**, i.e., one direction at a time.

- **Unicast**, **multicast** and **broadcast** channels might be better described as supporting one-to-one, one-to-some (i.e., to a selected subset of all possible receivers), and one-to-many (often *all* possible receivers) communication.
- It's important to see that the metrics covered can apply to *various* layers, not just the physical layer: it's important to know which layer a measurement is relative to.
- Among the metrics we *don't* explicitly cover, the level of noise (or a model for that noise) is a key example; excluding this is justified by the fact we don't cover techniques for error detection or correction in this unit.

- Goal: assess a communication channel using a suitable quality metric.
  1. a channel can be described using numerous characteristics,

Notes:

- A (non-exhaustive) selection of characteristics we could use to describe a given channel include:

width	$\Longrightarrow$	serial, parallel
direction	$\Longrightarrow$	simplex, half-duplex full-duplex
timing	$\Longrightarrow$	synchronous, asynchronous
rate	$\Longrightarrow$	signalling rate, modulation rate
multiplicity	$\Longrightarrow$	unicast, multicast, broadcast
efficiency	$\Longrightarrow$	latency, bandwidth, throughput
reliability	$\Longrightarrow$	noise/error rate and type

  You may see the term **channel model** used: this is essentially a more abstract description of the channel, in terms of similar characteristics (e.g., noise).
- If communication between two end-points is described as **duplex**, it can occur in *both* directions; **simplex** communication is one-way only. Further to this, one can define
  1. **full-duplex**, i.e., both directions simultaneously, or
  2. **half-duplex**, i.e., one direction at a time.
- **Unicast**, **multicast** and **broadcast** channels might be better described as supporting one-to-one, one-to-some (i.e., to a selected subset of all possible receivers), and one-to-many (often *all* possible receivers) communication.
- It's important to see that the metrics covered can apply to *various* layers, not just the physical layer: it's important to know which layer a measurement is relative to.
- Among the metrics we *don't* explicitly cover, the level of noise (or a model for that noise) is a key example; excluding this is justified by the fact we don't cover techniques for error detection or correction in this unit.

- Goal: assess a communication channel using a suitable quality metric.
  1. a channel can be described using numerous characteristics,
  2. their units aren't very consistent, e.g.,
     - base-10 for countable quantities, e.g.,

$$\left.\begin{array}{lcl} \text{kilo} & \equiv & 10^3 \\ \text{mega} & \equiv & 10^6 \\ \text{giga} & \equiv & 10^9 \end{array}\right\} 25\text{Mbit}\,\text{s}^{-1} = 25 \cdot 10^6 \text{bit}\,\text{s}^{-1}$$

     - base-2 for data or storage size, e.g.,

$$\left.\begin{array}{lcl} \text{kilo} & \equiv & 2^{10} \\ \text{mega} & \equiv & 2^{20} \\ \text{giga} & \equiv & 2^{30} \end{array}\right\} 25\text{MB} = 25 \cdot 2^{20} \cdot 8\text{bit}$$

Notes:

- A (non-exhaustive) selection of characteristics we could use to describe a given channel include:

width	$\Longrightarrow$	serial, parallel
direction	$\Longrightarrow$	simplex, half-duplex full-duplex
timing	$\Longrightarrow$	synchronous, asynchronous
rate	$\Longrightarrow$	signalling rate, modulation rate
multiplicity	$\Longrightarrow$	unicast, multicast, broadcast
efficiency	$\Longrightarrow$	latency, bandwidth, throughput
reliability	$\Longrightarrow$	noise/error rate and type

  You may see the term **channel model** used: this is essentially a more abstract description of the channel, in terms of similar characteristics (e.g., noise).
- If communication between two end-points is described as **duplex**, it can occur in *both* directions; **simplex** communication is one-way only. Further to this, one can define
  1. **full-duplex**, i.e., both directions simultaneously, or
  2. **half-duplex**, i.e., one direction at a time.
- **Unicast**, **multicast** and **broadcast** channels might be better described as supporting one-to-one, one-to-some (i.e., to a selected subset of all possible receivers), and one-to-many (often *all* possible receivers) communication.
- It's important to see that the metrics covered can apply to *various* layers, not just the physical layer: it's important to know which layer a measurement is relative to.
- Among the metrics we *don't* explicitly cover, the level of noise (or a model for that noise) is a key example; excluding this is justified by the fact we don't cover techniques for error detection or correction in this unit.

Metrics (1)

- Goal: assess a communication channel using a suitable quality metric.
  1. a channel can be described using numerous characteristics,
  2. their units aren't very consistent, e.g.,
     - base-10 for countable quantities, e.g.,

$$\left.\begin{array}{lcl} \text{kilo} & \equiv & 10^3 \\ \text{mega} & \equiv & 10^6 \\ \text{giga} & \equiv & 10^9 \end{array}\right\} 25\text{Mbit}\,\text{s}^{-1} = 25 \cdot 10^6 \text{bit}\,\text{s}^{-1}$$

     - base-2 for data or storage size, e.g.,

$$\left.\begin{array}{lcl} \text{kilo} & \equiv & 2^{10} \\ \text{mega} & \equiv & 2^{20} \\ \text{giga} & \equiv & 2^{30} \end{array}\right\} 25\text{MB} = 25 \cdot 2^{20} \cdot 8\text{bit}$$

  3. their utility depends (a *lot*) on the application, and

Notes:

- A (non-exhaustive) selection of characteristics we could use to describe a given channel include:

width	$\Longrightarrow$	serial, parallel
direction	$\Longrightarrow$	simplex, half-duplex full-duplex
timing	$\Longrightarrow$	synchronous, asynchronous
rate	$\Longrightarrow$	signalling rate, modulation rate
multiplicity	$\Longrightarrow$	unicast, multicast, broadcast
efficiency	$\Longrightarrow$	latency, bandwidth, throughput
reliability	$\Longrightarrow$	noise/error rate and type

  You may see the term **channel model** used: this is essentially a more abstract description of the channel, in terms of similar characteristics (e.g., noise).
- If communication between two end-points is described as **duplex**, it can occur in *both* directions; **simplex** communication is one-way only. Further to this, one can define
  1. **full-duplex**, i.e., both directions simultaneously, or
  2. **half-duplex**, i.e., one direction at a time.
- **Unicast**, **multicast** and **broadcast** channels might be better described as supporting one-to-one, one-to-some (i.e., to a selected subset of all possible receivers), and one-to-many (often *all* possible receivers) communication.
- It's important to see that the metrics covered can apply to *various* layers, not just the physical layer: it's important to know which layer a measurement is relative to.
- Among the metrics we *don't* explicitly cover, the level of noise (or a model for that noise) is a key example; excluding this is justified by the fact we don't cover techniques for error detection or correction in this unit.

Metrics (1)

- Goal: assess a communication channel using a suitable quality metric.
  1. a channel can be described using numerous characteristics,
  2. their units aren't very consistent, e.g.,
     - base-10 for countable quantities, e.g.,

$$\left.\begin{array}{lcl} \text{kilo} & \equiv & 10^3 \\ \text{mega} & \equiv & 10^6 \\ \text{giga} & \equiv & 10^9 \end{array}\right\} 25\text{Mbit}\,\text{s}^{-1} = 25 \cdot 10^6 \text{bit}\,\text{s}^{-1}$$

     - base-2 for data or storage size, e.g.,

$$\left.\begin{array}{lcl} \text{kilo} & \equiv & 2^{10} \\ \text{mega} & \equiv & 2^{20} \\ \text{giga} & \equiv & 2^{30} \end{array}\right\} 25\text{MB} = 25 \cdot 2^{20} \cdot 8\text{bit}$$

  3. their utility depends (a *lot*) on the application, and
  4. they have different meanings depending on the context (e.g., bandwidth in CS vs. EE)!

Notes:

- A (non-exhaustive) selection of characteristics we could use to describe a given channel include:

width	$\Longrightarrow$	serial, parallel
direction	$\Longrightarrow$	simplex, half-duplex full-duplex
timing	$\Longrightarrow$	synchronous, asynchronous
rate	$\Longrightarrow$	signalling rate, modulation rate
multiplicity	$\Longrightarrow$	unicast, multicast, broadcast
efficiency	$\Longrightarrow$	latency, bandwidth, throughput
reliability	$\Longrightarrow$	noise/error rate and type

  You may see the term **channel model** used: this is essentially a more abstract description of the channel, in terms of similar characteristics (e.g., noise).
- If communication between two end-points is described as **duplex**, it can occur in *both* directions; **simplex** communication is one-way only. Further to this, one can define
  1. **full-duplex**, i.e., both directions simultaneously, or
  2. **half-duplex**, i.e., one direction at a time.
- **Unicast**, **multicast** and **broadcast** channels might be better described as supporting one-to-one, one-to-some (i.e., to a selected subset of all possible receivers), and one-to-many (often *all* possible receivers) communication.
- It's important to see that the metrics covered can apply to *various* layers, not just the physical layer: it's important to know which layer a measurement is relative to.
- Among the metrics we *don't* explicitly cover, the level of noise (or a model for that noise) is a key example; excluding this is justified by the fact we don't cover techniques for error detection or correction in this unit.

Metrics (2)

Definition (**bandwidth**)

The **bandwidth** of a communication channel is the number of symbols which can be transmitted per unit of time; this is sometimes referred to as the **channel capacity**, and often measured in bits per second (which is then the **bit rate**).

It is common to contrast total available bandwidth, with that achievable in practice; the latter is termed **throughput**, st.

$$\text{bandwidth} \geq \text{throughput} + \text{overhead}.$$

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11 University of BRISTOL

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (latency)

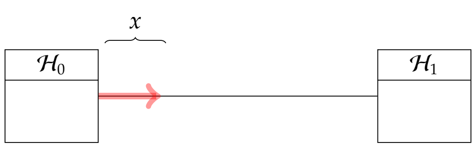
The latency of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the transmission delay, and
- d is the propagation delay

given n symbols and a bandwidth of r symbols per unit of time. Note that

- One-Way Delay (OWD) measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- Round-Trip Time (RTT) measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, plus the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks)
git # 3627080 @ 2016-03-11
University of BRISTOL

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support asynchronous bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \mathrm{bit\,s}^{-1}$ may not in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term channel efficiency measures throughput as a percentage of available bandwidth; channel utilisation measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the transmission delay is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

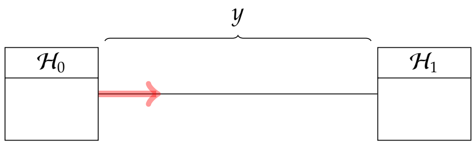
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of 1Gbit s^{-1}.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

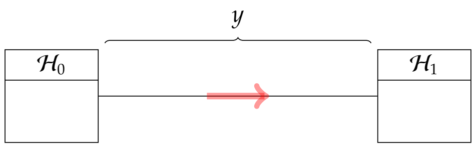
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of 1Gbit s^{-1}.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Definition (latency)

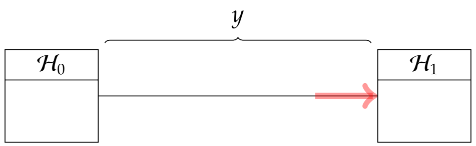
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (latency)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,

st. latency $=$ $x + y$ $=$ transmission delay + propagation delay.

Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks)
git # 3627080 @ 2016-03-11
University of BRISTOL

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Definition (**latency**)

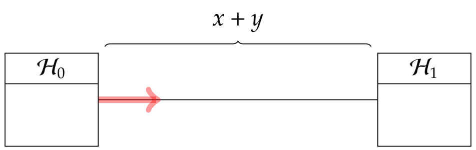
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (latency)

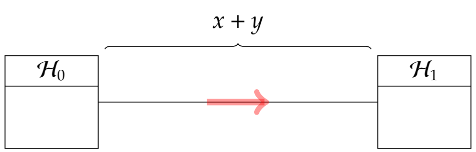
The latency of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the transmission delay, and
- d is the propagation delay

given n symbols and a bandwidth of r symbols per unit of time. Note that

- One-Way Delay (OWD) measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- Round-Trip Time (RTT) measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, plus the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11
University of BRISTOL

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support asynchronous bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit}\,\text{s}^{-1}$ may not in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term channel efficiency measures throughput as a percentage of available bandwidth; channel utilisation measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the transmission delay is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

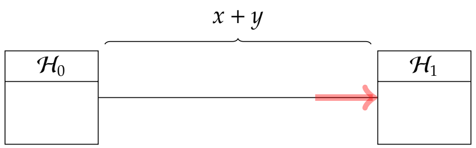
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1 \mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x \mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Daniel Page ⟨Daniel.Page@bristol.ac.uk⟩
Concurrent Computing (Computer Networks) git # 3627080 @ 2016-03-11
University of BRISTOL

Definition (**latency**)

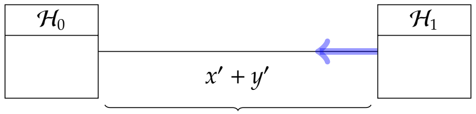
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (latency)

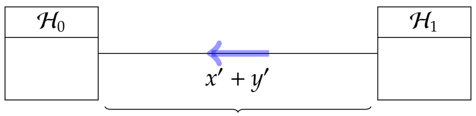
The latency of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the transmission delay, and
- d is the propagation delay

given n symbols and a bandwidth of r symbols per unit of time. Note that

- One-Way Delay (OWD) measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- Round-Trip Time (RTT) measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Daniel Page (Daniel.Page@bristol.ac.uk)
Concurrent Computing (Computer Networks)
git # 3627080 @ 2016-03-11
University of BRISTOL

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\mathrm{Gbit\,s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\mathrm{bit\,s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term channel efficiency measures throughput as a percentage of available bandwidth; channel utilisation measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the transmission delay is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Definition (**latency**)

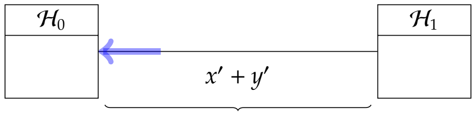
The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**latency**)

The **latency** of a connection relates to the (total) time required to transmit data between two end-points (e.g., between $\mathcal{H}_0$ and $\mathcal{H}_1$). This is typically expressed as $n/r + d$, where

- n/r is the **transmission delay**, and
- d is the **propagation delay**

given n symbols and a bandwidth of r symbols per unit of time. Note that

- **One-Way Delay (OWD)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, whereas
- **Round-Trip Time (RTT)** measures the latency of $\mathcal{H}_0$ transmitting data to $\mathcal{H}_1$, *plus* the latency of $\mathcal{H}_1$ transmitting an associated response back to $\mathcal{H}_0$.

i.e.,

st. $\text{RTT} = (x + y) + (x' + y') \geq 2 \cdot \text{OWD}$.

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\,\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\,\text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (**bandwidth-latency product**)

Imagine that a given channel is a pipe, whose diameter is defined by bandwidth and length by latency. The "volume" of the pipe is termed the **bandwidth-latency product** (or **bandwidth-delay product**), and captures the amount of data "in-flight".

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\,\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\,\text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Definition (**bandwidth-latency product**)

Imagine that a given channel is a pipe, whose diameter is defined by bandwidth and length by latency. The "volume" of the pipe is termed the **bandwidth-latency product** (or **bandwidth-delay product**), and captures the amount of data "in-flight".

i.e.,

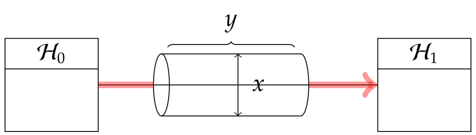
Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of 1Gbit s^{-1}.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x$bit s^{-1} may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Definition (bandwidth-latency product)

Imagine that a given channel is a pipe, whose diameter is defined by bandwidth and length by latency. The "volume" of the pipe is termed the bandwidth-latency product (or bandwidth-delay product), and captures the amount of data "in-flight".

i.e.,



Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of 1Gbit s^{-1}.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x$bit s^{-1} may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (2)

Definition (bandwidth-latency product)

Imagine that a given channel is a pipe, whose diameter is defined by bandwidth and length by latency. The "volume" of the pipe is termed the **bandwidth-latency product** (or **bandwidth-delay product**), and captures the amount of data "in-flight".

i.e.,

st. bandwidth-latency product $= x \cdot y =$ bandwidth $\cdot$ latency

Notes:

- Bandwidth often forms part of the name for a given communication technology; for example, gigabit Ethernet implies a bandwidth of $1\text{Gbit}\,\text{s}^{-1}$.
- It is not uncommon for a channel to support *asynchronous* bandwidths, depending on whether data is being transmitted (i.e., via the up-link) or received (i.e., via the down-link).
- It is important to remember the "gap" between available bandwidth (resp. maximum throughout) vs. actual throughout. Put simply, it implies that a channel described as $x\text{bit}\,\text{s}^{-1}$ may *not* in fact be able to transmit data at that rate. Some obvious overheads relating to this gap include
  - extra data added to support encapsulation (or framing),
  - shared access to the channel, whereby use by one entity impacts on the throughput of another,
  - any artificial limits (e.g., bandwidth cap or bandwidth throttling as used by some ISPs,
  - physical constraints such as error rate.
- The term **channel efficiency** measures throughput as a percentage of available bandwidth; **channel utilisation** measures the proportion of time the channel is used to communicate useful payloads vs. non-use and communication overhead (e.g., due to encapsulation).
- Put more simply, the **transmission delay** is the time taken to transmit data comprising n symbols if the transmission rate is r: you could say it takes $\frac{n}{r}$ units of time to take the data and "put it" onto the communication medium. Then, it takes d units of time for that data to propagate along the medium to the other end.
- As soon as the connection is indirect, propagation delay becomes much more complicated to model. For instance, it must capture delay associated with a store-and-forward approach (e.g., queuing and processing delays) in any intermediate nodes.
- Note that RTT is particularly relevant in TCP/IP-based networks, since a) an acknowledgement is required for (more or less) each transmitted packet, and b) routing decisions are often made based on RTT (e.g., one route may be preferred over another if it has a lower RTT).
- Put another way, in TCP/IP-based networks (where latency is typically RTT) the bandwidth-latency product is basically the amount of data transmitted but not (yet) acknowledged.

Metrics (3)

Example

Consider the following communication mediums:

1. a MODEM-based channel where

$$\begin{array}{rcl} d &=& 5\text{ms} \\ r &=& 56\text{kbit}\,\text{s}^{-1} \\ n &=& 1250\text{B} \end{array} \qquad \begin{array}{rcl} \text{latency} &=& \frac{1250\cdot 8}{56\cdot 10^3} + 5\cdot 10^{-3} \\ &\simeq& 184\cdot 10^{-3}\text{s} \end{array}$$

2. an ADSL-based channel where

$$\begin{array}{rcl} d &=& 50\text{ms} \\ r &=& 8\text{Mbit}\,\text{s}^{-1} \\ n &=& 1250\text{B} \end{array} \qquad \begin{array}{rcl} \text{latency} &=& \frac{1250\cdot 8}{8\cdot 10^6} + 50\cdot 10^{-3} \\ &\simeq& 51\cdot 10^{-3}\text{s} \end{array}$$

3. a fiber-based channel where

$$\begin{array}{rcl} d &=& 28\text{ms} \\ r &=& 171\text{Gbit}\,\text{s}^{-1} \end{array} \qquad \begin{array}{rcl} \text{bandwidth-latency product} &=& 28\cdot 10^{-3}\cdot 171\cdot 10^9 \\ &\simeq& 48\cdot 10^8\text{bit} \\ &\simeq& 60\cdot 10^7\text{B} \end{array}$$

- **Conclusion(s)**:
  - *either* term may dominate the overall latency (which may not be obvious initially),
  - a "long and fat enough" pipe can contain a *lot* of in-flight data: potentially this needs to be considered in higher layers, and
  - latency (resp. bandwidth) probably matters more for smaller (resp. larger) amounts of data.

Notes:

- In the first (resp. second) example, the dominant term is transmission (resp. propagation) delay. Although somewhat obvious (since latency is the sum of these terms), this illustrates the fact that *either* a "long" channel *or* a "slow" channel can yield high latency.
- For the fiber-based channel, we assume a distance of 5280 miles which is roughly that between Bristol and San Francisco; based on the fact we know the speed of light is $1.86\cdot 10^5\text{mi}\,\text{s}^{-1}$, we get a delay of $28\cdot 10^{-3}$s.

Conclusions

- Take away points:
  - Digital and analogue signal processing is a *big* topic; this is a light-weight introduction only!
  - There are *many* of possible ways to address the initial (fairly simple) goal ...
  - ... on one hand this is great (because we can match a choice to our needs); on the other, it's not so great (since making a *good* choice is harder).
  - Keep in mind that
    1. a given approach is often underpinned by theory, but
    2. choices are often made with lower-level, Engineering requirements in mind,
    3. we can only make *good* choices by understanding how the channel is used.

Notes:

References

[1] Wikipedia: Block code.
http://en.wikipedia.org/wiki/Block_code.

[2] Wikipedia: Data transmission.
http://en.wikipedia.org/wiki/Data_transmission.

[3] Wikipedia: Line code.
http://en.wikipedia.org/wiki/Line_code.

[4] G. Hall.
Chapter 12: Introduction to Fourier series.
In *Modern Engineering Mathematics*. Prentice Hall, 4th edition, 2010.

[5] H. Nyquist.
Certain topics in telegraph transmission theory.
Transactions of the AIEE, 47:617–644, 1928.

[6] C.E. Shannon.
Communication in the presence of noise.
Proceedings of the IRE, 37(1):10–21, 1949.

[7] W. Stallings.
Chapter 4: Data transmission.
In *Data and Computer Communications* [10].

Notes:

References

[8] W. Stallings.
Chapter 5: Transmission media.
In *Data and Computer Communications* [10].

[9] W. Stallings.
Chapter 6: Signal encoding techniques.
In *Data and Computer Communications* [10].

[10] W. Stallings.
Data and Computer Communications.
Pearson, 9th edition, 2010.

Notes: