Ddc (4343201) - Problem Solver

Antigravity AI

June 4, 2026

Contents

CHAPTER 1

Introduction to Digital and Data Communication System

This chapter focuses on the fundamental computational techniques used in digital and data communication systems. It covers the calculation of basic signal properties, data rate limitations over noiseless and noisy channels, signal-to-noise ratio measurements, and network performance metrics.

1.1 Signal Properties: Frequency Period and Wavelength

Understanding the basic mathematical relationships between frequency, period, and wavelength is essential for analyzing signals in a communication medium.

Methodology:

Calculation of Frequency and Period The fundamental formulas relating period (T), frequency (f), and wavelength (λ) are:

1. **Period and Frequency**: The period is the inverse of the frequency.

$$T = \frac{1}{f} \quad \text{and} \quad f = \frac{1}{T}$$

   Where T is the period in seconds (s) and f is the frequency in Hertz (Hz).

2. **Wavelength**: The distance a simple signal can travel in one period.

$$\lambda = \frac{v}{f}$$

   Where λ is the wavelength in meters (m), v is the propagation speed in the medium in meters per second (m/s), and f is the frequency in Hertz (Hz). For electromagnetic waves in a vacuum, $v = c = 3 \times 10^8$ m/s.

Worked Example:

Calculate the period of a signal with a frequency of 100 MHz. **Step 1: Identify the given values.** Frequency $f = 100 \text{ MHz} = 100 \times 10^6 \text{ Hz} = 10^8 \text{ Hz}$.

Step 2: Apply the period formula.

$$T = \frac{1}{f}$$

Step 3: Calculate the period.

$$T = \frac{1}{10^8 \text{ Hz}} = 10^{-8} \text{ s} = 10 \text{ ns (nanoseconds)}$$

Conclusion: The period of the signal is 10 ns.

Worked Example:

Find the wavelength of a 2.4 GHz Wi-Fi signal traveling through the air. **Step 1: Identify the given values.** Frequency $f = 2.4$ GHz $= 2.4 \times 10^9$ Hz. Assume the speed of the signal in air is approximately the speed of light in a vacuum, $v = 3 \times 10^8$ m/s.
Step 2: Apply the wavelength formula.

$$\lambda = \frac{v}{f}$$

Step 3: Calculate the wavelength.

$$\lambda = \frac{3 \times 10^8}{2.4 \times 10^9} = \frac{3}{24} = 0.125 \text{ m} = 12.5 \text{ cm}$$

Conclusion: The wavelength of the Wi-Fi signal is 12.5 cm.

1.2 Bit Rate and Baud Rate

In digital transmission, it is crucial to distinguish between the bit rate (data rate) and the baud rate (signaling rate).

Methodology:

Bit Rate and Baud Rate Conversion

1. **Bit Rate (R)**: The number of bits transmitted per second, measured in bits per second (bps).
2. **Baud Rate (S)**: The number of signal elements transmitted per second, measured in baud.
3. **Signal Levels (L)**: The number of distinct signal states used to represent data.
4. **Relationship**: The number of bits carried by each signal element is $r = \log_2(L)$. The relationship between Bit Rate and Baud Rate is:

$$R = S \times \log_2(L)$$

$$S = \frac{R}{\log_2(L)}$$

Worked Example:

An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second find the bit rate. **Step 1: Identify the given values.** Number of bits per signal element $r = \log_2(L) = 4$ bits. Baud rate $S = 1000$ baud.
Step 2: Apply the bit rate formula.

$$R = S \times r$$

Step 3: Calculate the bit rate.

$$R = 1000 \times 4 = 4000 \text{ bps}$$

Conclusion: The bit rate of the signal is 4000 bps.

Worked Example:

Calculate the required baud rate to transmit data at a rate of 100 kbps using 16-QAM (Quadrature Amplitude Modulation which uses 16 signal levels). **Step 1: Identify the given values.** Bit rate $R = 100$ kbps $= 100,000$ bps. Number of signal levels $L = 16$.
Step 2: Calculate the number of bits per signal element.

$$r = \log_2(L) = \log_2(16) = 4 \text{ bits/baud}$$

Step 3: Apply the baud rate formula.

$$S = \frac{R}{r}$$

Step 4: Calculate the baud rate.

$$S = \frac{100,000}{4} = 25,000 \text{ baud}$$

Conclusion: The required baud rate is $25,000$ baud.

1.3 Signal-to-Noise Ratio (SNR)

The Signal-to-Noise Ratio quantifies how much a signal has been corrupted by noise. It is typically expressed in decibels (dB) because signals and noise can vary over a wide range of magnitudes.

Methodology:

Signal to Noise Ratio Calculation

1. **SNR (Power Ratio)**: The absolute ratio of signal power to noise power.

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

Where P_{signal} is the average signal power and P_{noise} is the average noise power. Both must be in the same units (e.g., Watts or milliWatts).

2. **SNR in Decibels (SNR_{dB})**:

$$\text{SNR}_{\text{dB}} = 10\log_{10}(\text{SNR}) = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right)$$

3. **Converting SNR_{dB} back to SNR (Ratio)**:

$$\text{SNR} = 10^{\frac{\text{SNR}_{\text{dB}}}{10}}$$

Worked Example:

The power of a signal is 10 mW and the power of the noise is 1 µW. Calculate the SNR and SNR_{dB}. **Step 1: Identify and convert to consistent units.** $P_{\text{signal}} = 10 \text{ mW} = 10 \times 10^{-3}$ W. $P_{\text{noise}} = 1\ \mu\text{W} = 1 \times 10^{-6}$ W.
Step 2: Calculate the absolute SNR.

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \frac{10 \times 10^{-3}}{1 \times 10^{-6}} = 10,000$$

Step 3: Calculate the SNR_{dB}.

$$\text{SNR}_{\text{dB}} = 10\log_{10}(10,000) = 10 \times 4 = 40 \text{ dB}$$

Conclusion: The SNR is 10,000 and the SNR_{dB} is 40 dB.

Worked Example:

A communication channel has an SNR_{dB} of 36 dB. Determine the absolute Signal-to-Noise ratio. **Step 1: Identify the given value.** $\text{SNR}_{\text{dB}} = 36$ dB.
Step 2: Apply the conversion formula.

$$\text{SNR} = 10^{\frac{\text{SNR}_{\text{dB}}}{10}}$$

Step 3: Calculate the SNR.

$$\text{SNR} = 10^{\frac{36}{10}} = 10^{3.6} \approx 3981$$

Conclusion: The absolute Signal-to-Noise ratio is approximately 3981.

1.4 Data Rate Limits

Data rate limits define the maximum speed at which data can be transmitted over a channel. The limit depends on whether the channel is considered noiseless or noisy.

Methodology:

Nyquist Bit Rate for Noiseless Channels For a theoretical channel where no noise is present, the maximum bit rate is determined by the Nyquist theorem.

1. **Nyquist Formula**:

$$C = 2 \times B \times \log_2(L)$$

   Where C is the maximum channel capacity (or bit rate) in bits per second (bps), B is the bandwidth of the channel in Hertz (Hz), and L is the number of signal levels used to represent data.

2. To find the required number of signal levels L for a desired bit rate C:

$$\log_2(L) = \frac{C}{2B} \implies L = 2^{\frac{C}{2B}}$$

Worked Example:

Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with 2 signal levels. What is the maximum bit rate? **Step 1: Identify the given values.** Bandwidth $B = 3000$ Hz. Number of signal levels $L = 2$.

Step 2: Apply the Nyquist formula.

$$C = 2 \times B \times \log_2(L)$$

Step 3: Calculate the maximum bit rate.

$$C = 2 \times 3000 \times \log_2(2)$$

$$C = 6000 \times 1 = 6000 \text{ bps}$$

Conclusion: The maximum bit rate is 6000 bps.

Worked Example:

We need to send data at 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels are required? **Step 1: Identify the given values.** Bit rate $C = 265,000$ bps. Bandwidth $B = 20,000$ Hz.

Step 2: Rearrange the Nyquist formula to solve for L.

$$C = 2 \times B \times \log_2(L)$$

$$265,000 = 2 \times 20,000 \times \log_2(L)$$

Step 3: Calculate the number of levels.

$$\log_2(L) = \frac{265,000}{40,000} = 6.625$$

$$L = 2^{6.625} \approx 98.7$$

Conclusion: Since the number of levels must be an integer (and typically a power of 2), we would need 128 signal levels (2^7) to comfortably support this data rate.

Methodology:

Shannon Capacity for Noisy Channels In reality, all channels have some noise. The Shannon capacity determines the theoretical highest data rate for a noisy channel.

1. **Shannon Capacity Formula**:
$$C = B \times \log_2(1 + \text{SNR})$$
Where C is the channel capacity in bits per second (bps), B is the bandwidth of the channel in Hertz (Hz), and SNR is the absolute Signal-to-Noise Ratio (not in dB).

2. **Important Note**: If the SNR is given in dB, you must first convert it to the absolute SNR ratio before applying the Shannon formula.

Worked Example:

Calculate the theoretical highest bit rate of a standard telephone line which has a bandwidth of 3000 Hz and a Signal-to-Noise ratio of 3162. **Step 1: Identify the given values.** Bandwidth $B = 3000$ Hz. $\text{SNR} = 3162$.
Step 2: Apply the Shannon capacity formula.

$$C = B \times \log_2(1 + \text{SNR})$$

Step 3: Calculate the capacity.

$$C = 3000 \times \log_2(1 + 3162)$$

$$C = 3000 \times \log_2(3163)$$

Using the property $\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$:

$$\log_2(3163) \approx \frac{3.500}{0.301} \approx 11.62$$

$$C = 3000 \times 11.62 = 34,860 \text{ bps}$$

Conclusion: The theoretical maximum capacity of the telephone line is approximately $34,860$ bps.

Worked Example:

A channel has a bandwidth of 1 MHz and an SNR_{dB} of 63 dB. What is the appropriate bit rate and signal levels if we combine both Shannon and Nyquist limits? **Step 1: Calculate the absolute SNR from SNR_{dB}.** $\text{SNR}_{\text{dB}} = 63$.

$$\text{SNR} = 10^{\frac{63}{10}} = 10^{6.3} \approx 1,995,262$$

Step 2: Apply the Shannon capacity formula to find the upper limit. $B = 1,000,000$ Hz.

$$C = B \times \log_2(1 + \text{SNR})$$

$$C = 10^6 \times \log_2(1 + 1,995,262) \approx 10^6 \times \log_2(1,995,263)$$

$$\log_2(1,995,263) \approx 20.92$$

$$C \approx 10^6 \times 20.92 = 20.92 \text{ Mbps}$$

The Shannon capacity tells us we cannot transmit faster than 20.92 Mbps.

Step 3: Use the Nyquist formula to find the required signal levels for a practical bit rate. Let us aim for a practical bit rate slightly below the Shannon limit, for instance, 20 Mbps.

$$C_{\text{Nyquist}} = 2 \times B \times \log_2(L)$$

$$20,000,000 = 2 \times 1,000,000 \times \log_2(L)$$

$$20 = 2 \times \log_2(L)$$

$$10 = \log_2(L) \implies L = 2^{10} = 1024$$

Conclusion: The theoretical limit is 20.92 Mbps. To transmit at a practical speed of 20 Mbps we need a signal with 1024 levels.

1.5 Network Performance Metrics: Latency (Delay)

Latency is the total time it takes for an entire message to completely arrive at the destination from the time the first bit is sent. It consists of multiple components.

Methodology:

Calculation of Transmission and Propagation Delay The total latency is given by:

$$\text{Latency} = \text{Transmission Delay} + \text{Propagation Delay} + \text{Queuing Delay} + \text{Processing Delay}$$

In most computational problems, queuing and processing delays are negligible or given, focusing the calculation on transmission and propagation:

1. **Transmission Delay**: The time required to push all the bits of a message onto the wire.

$$\text{Transmission Delay} = \frac{\text{Message Size}}{\text{Bandwidth}}$$

   Where Message Size is in bits and Bandwidth is in bits per second (bps).

2. **Propagation Delay**: The time required for a bit to travel from the source to the destination.

$$\text{Propagation Delay} = \frac{\text{Distance}}{\text{Propagation Speed}}$$

   Where Distance is in meters and Propagation Speed is in meters per second (m/s).

Worked Example:

Calculate the transmission delay for a 5 MB (Megabyte) file sent over a 10 Mbps (Megabits per second) link. **Step 1: Identify and convert given values to bits.** Message Size $= 5$ MB $= 5 \times 10^6$ Bytes $= 5 \times 10^6 \times 8$ bits $= 40 \times 10^6$ bits. Bandwidth $= 10$ Mbps $= 10 \times 10^6$ bps.
Step 2: Apply the transmission delay formula.

$$\text{Transmission Delay} = \frac{\text{Message Size}}{\text{Bandwidth}}$$

Step 3: Calculate the delay.

$$\text{Transmission Delay} = \frac{40 \times 10^6 \text{ bits}}{10 \times 10^6 \text{ bps}} = 4 \text{ seconds}$$

Conclusion: It takes 4 seconds to push the entire file onto the network link.

Worked Example:

What is the propagation delay if two hosts are separated by 12000 km and the signal travels through the cable at 2.4×10^8 m/s? **Step 1: Identify and convert given values to standard units.** Distance $d = 12,000 \text{ km} = 12,000 \times 10^3 \text{ m} = 1.2 \times 10^7 \text{ m}$. Propagation Speed $v = 2.4 \times 10^8$ m/s.

Step 2: Apply the propagation delay formula.

$$\text{Propagation Delay} = \frac{d}{v}$$

Step 3: Calculate the delay.

$$\text{Propagation Delay} = \frac{1.2 \times 10^7}{2.4 \times 10^8} = \frac{1}{20} \text{ seconds} = 0.05 \text{ seconds} = 50 \text{ ms}$$

Conclusion: The propagation delay is 50 milliseconds.

Worked Example:

A 1000-byte packet is sent over a 1 Gbps link. The distance between the sender and receiver is 10 km and the propagation speed is 2×10^8 m/s. Calculate the total latency assuming queuing and processing delays are zero. **Step 1: Calculate the Transmission Delay.** Message Size = 1000 bytes × 8 bits/byte = 8000 bits. Bandwidth = 1 Gbps = 10^9 bps.

$$\text{Transmission Delay} = \frac{8000}{10^9} = 8 \times 10^{-6} \text{ s} = 8 \text{ µs}$$

Step 2: Calculate the Propagation Delay. Distance = 10 km = $10,000$ m = 10^4 m. Speed = 2×10^8 m/s.

$$\text{Propagation Delay} = \frac{10^4}{2 \times 10^8} = \frac{1}{2 \times 10^4} = 0.5 \times 10^{-4} \text{ s} = 50 \text{ µs}$$

Step 3: Calculate Total Latency.

$$\text{Total Latency} = \text{Transmission Delay} + \text{Propagation Delay}$$

$$\text{Total Latency} = 8 \text{ µs} + 50 \text{ µs} = 58 \text{ µs}$$

Conclusion: The total time from when the first bit is sent to when the last bit arrives is 58 µs.

CHAPTER 2

Digital Modulation Techniques

2.1 Bit Rate and Baud Rate

In digital modulation, it is essential to distinguish between bit rate and baud rate (signal rate). Bit rate is the number of bits transmitted per second, while baud rate is the number of signal elements transmitted per second.

Methodology:

Calculating Bit Rate and Baud Rate The relationship between bit rate (N) and baud rate (S) is given by the formula:

1. Identify the number of bits per signal element r. If the number of signal levels or states is L, then $r = \log_2(L)$.
2. Use the formula to calculate the baud rate:

$$S = \frac{N}{r}$$

   where:

   - S = Baud rate (symbols per second or bauds)
   - N = Bit rate (bits per second or bps)
   - r = Number of data bits per signal element

3. Alternatively if baud rate is given compute bit rate as:

$$N = S \times r$$

Worked Example:

Bit Rate and Baud Rate Calculation An analog signal carries 4 bits per signal element. If 1000 signal elements are sent per second, find the bit rate.
Solution:

1. **Identify given parameters:** Baud rate $S = 1000$ bauds. Bits per signal element $r = 4$.
2. **Calculate Bit Rate (N):**

$$N = S \times r$$

$$N = 1000 \times 4 = 4000 \text{ bps}$$

The bit rate is 4000 bps or 4 kbps.

Worked Example:

Determining Baud Rate for QAM A 16-QAM signal has a bit rate of 24000 bps. Calculate its baud rate.
Solution:

1. **Identify given parameters:** Bit rate $N = 24000$ bps. For 16-QAM the number of signal levels $L = 16$.
2. **Determine bits per symbol (r):**
$$r = \log_2(L) = \log_2(16) = 4 \text{ bits/symbol}$$
3. **Calculate Baud Rate (S):**
$$S = \frac{N}{r} = \frac{24000}{4} = 6000 \text{ bauds}$$

The baud rate is 6000 bauds.

2.2 Amplitude Shift Keying (ASK)

In Amplitude Shift Keying, the amplitude of the carrier signal is varied to represent digital data.

Methodology:

ASK Bandwidth Calculation The bandwidth of an ASK signal depends on the signal rate and the modulation technique.

1. Determine the baud rate (S) of the signal. If bit rate (N) and bits per symbol (r) are given, use $S = N/r$. For binary ASK (BASK), $r = 1$, so $S = N$.
2. Calculate the bandwidth (B) using the formula:
$$B = (1 + d) \times S$$
where:
   - B = Bandwidth (in Hz)
   - d = Modulation factor or roll-off factor ($0 \leq d \leq 1$)
   - S = Baud rate
3. If d is not specified assume the worst-case scenario $d = 1$, which gives $B = 2S$. The minimum bandwidth occurs at $d = 0$, giving $B = S$.

Worked Example:

Bandwidth of BASK Signal Calculate the minimum bandwidth required for a Binary Amplitude Shift Keying (BASK) signal transmitting at 5 kbps in half-duplex mode.
Solution:

1. **Identify given parameters:** Bit rate $N = 5000$ bps. For BASK $L = 2$ so $r = 1$.
2. **Calculate Baud Rate (S):**
$$S = \frac{N}{r} = \frac{5000}{1} = 5000 \text{ bauds}$$
3. **Calculate Minimum Bandwidth:** For minimum bandwidth $d = 0$.
$$B = (1 + 0) \times S = 1 \times 5000 = 5000 \text{ Hz}$$

The minimum bandwidth required is 5 kHz.

2.3 Frequency Shift Keying (FSK)

In Frequency Shift Keying, the frequency of the carrier signal is varied to represent digital data.

Methodology:

FSK Bandwidth Calculation The bandwidth of an FSK signal includes the shift in frequencies and the bandwidth of the signal itself.

1. Determine the baud rate S. For Binary FSK (BFSK), $S = N$.
2. Identify the two carrier frequencies f_1 (for logic 1) and f_2 (for logic 0).
3. Calculate the frequency shift $2\Delta f = |f_1 - f_2|$.
4. Calculate the bandwidth (B) using:

$$B = (1 + d) \times S + 2\Delta f$$

   where d is the roll-off factor ($0 \leq d \leq 1$).

5. For minimum bandwidth ($d = 0$), the formula simplifies to:

$$B_{min} = S + 2\Delta f$$

Worked Example:

Bandwidth of BFSK Signal A binary FSK system transmits data at a bit rate of 2000 bps. The two carrier frequencies used are 50 kHz and 55 kHz. Calculate the minimum bandwidth required.
Solution:

1. **Identify given parameters:** Bit rate $N = 2000$ bps. Since it is binary FSK $r = 1$ and $S = N = 2000$ bauds. Carrier frequencies $f_1 = 55$ kHz and $f_2 = 50$ kHz.
2. **Calculate Frequency Shift ($2\Delta f$):**

$$2\Delta f = 55000 - 50000 = 5000 \text{ Hz}$$

3. **Calculate Minimum Bandwidth:** Using $d = 0$ for minimum bandwidth:

$$B_{min} = S + 2\Delta f$$

$$B_{min} = 2000 + 5000 = 7000 \text{ Hz}$$

The minimum bandwidth required is 7 kHz.

2.4 Phase Shift Keying (PSK)

In Phase Shift Keying, the phase of the carrier signal is varied to represent data. Variations include Binary PSK (BPSK), Quadrature PSK (QPSK), and higher-order PSK like 8-PSK.

Methodology:

PSK Bandwidth and Phase Shifts

1. Determine the number of signal states L based on the modulation type (e.g. $L = 2$ for BPSK $L = 4$ for QPSK $L = 8$ for 8-PSK).
2. Compute the bits per symbol $r = \log_2(L)$ and baud rate $S = N/r$.

3. Calculate the phase difference between adjacent symbols:

$$\Delta\phi = \frac{360^\circ}{L}$$

4. Calculate the bandwidth using the same formula as ASK:

$$B = (1 + d) \times S$$

where d is the roll-off factor. The bandwidth efficiency improves with higher-order PSK since S decreases for a given N.

Worked Example:

Bandwidth and Phase Difference of QPSK A QPSK system transmits data at 10 Mbps. Assuming a roll-off factor of $d = 0.5$, calculate the baud rate, the required bandwidth, and the phase shift between consecutive symbols.

Solution:

1. **Determine Baud Rate (S):** For QPSK $L = 4$, so $r = \log_2(4) = 2$ bits/symbol. Bit rate $N = 10 \times 10^6$ bps.

$$S = \frac{N}{r} = \frac{10 \times 10^6}{2} = 5 \times 10^6 \text{ bauds} = 5 \text{ Mbaud}$$

2. **Calculate Bandwidth (B):**

$$B = (1 + d) \times S$$

$$B = (1 + 0.5) \times 5 \times 10^6 = 1.5 \times 5 \times 10^6 = 7.5 \times 10^6 \text{ Hz} = 7.5 \text{ MHz}$$

3. **Calculate Phase Difference ($\Delta\phi$):**

$$\Delta\phi = \frac{360^\circ}{4} = 90^\circ$$

The baud rate is 5 Mbaud, bandwidth is 7.5 MHz, and phase shift is 90°.

2.5 Quadrature Amplitude Modulation (QAM)

QAM is a combination of ASK and PSK, varying both amplitude and phase to increase data throughput without increasing bandwidth.

Methodology:

QAM Performance Metrics

1. Identify the modulation order L (e.g. 16 for 16-QAM 64 for 64-QAM).
2. Calculate bits per symbol $r = \log_2(L)$.
3. Calculate Baud Rate $S = N/r$.
4. Calculate the Bandwidth B:

$$B = (1 + d) \times S$$

5. Note that QAM uses both phase and amplitude shifts so the constellation diagram will have L distinct points.

Worked Example:

Comparing 64 QAM and 256 QAM A channel has a bandwidth of 6 MHz and uses a roll-off factor $d = 0$. Compare the maximum bit rates achievable using 64-QAM and 256-QAM.

Solution:

1. **Calculate Baud Rate (S):** Since $d = 0$, $B = S$. $S = 6$ MHz $= 6 \times 10^6$ bauds.
2. **Calculate Bit Rate for 64-QAM:** $L = 64$, so $r = \log_2(64) = 6$ bits/symbol.

$$N_{64} = S \times r = 6 \times 10^6 \times 6 = 36 \times 10^6 \text{ bps} = 36 \text{ Mbps}$$

3. **Calculate Bit Rate for 256-QAM:** $L = 256$, so $r = \log_2(256) = 8$ bits/symbol.

$$N_{256} = S \times r = 6 \times 10^6 \times 8 = 48 \times 10^6 \text{ bps} = 48 \text{ Mbps}$$

Using 64-QAM yields a bit rate of 36 Mbps, whereas 256-QAM yields 48 Mbps over the same bandwidth.

2.6 Summary of Modulation Performance

When comparing different digital modulation techniques for a fixed bit rate N, we observe distinct bandwidth requirements.

Modulation Type	**Bits per Symbol (r)**	**Baud Rate (S)**	**Minimum Bandwidth ($d = 0$)**
BASK / BPSK	1	N	N
BFSK	1	N	$N + 2\Delta f$
QPSK	2	$N/2$	$N/2$
8-PSK	3	$N/3$	$N/3$
16-QAM	4	$N/4$	$N/4$
64-QAM	6	$N/6$	$N/6$

CHAPTER 3

Information Theory and Coding

3.1 Information and Entropy

Methodology:

Information Entropy and Information Rate **1. Amount of Information (I):** The amount of information carried by a discrete message x_i with probability $P(x_i)$ is:

$$I(x_i) = \log_2\left(\frac{1}{P(x_i)}\right) \text{ bits}$$

2. Entropy (H): Entropy is the average information per message (or symbol) of a source emitting M messages:

$$H = \sum_{i=1}^{M} P(x_i)\log_2\left(\frac{1}{P(x_i)}\right) \text{ bits/symbol}$$

3. Information Rate (R): If symbols are emitted at a rate of r symbols per second, the information rate is:

$$R = r \times H \text{ bits/second}$$

Note: $\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$ or $\frac{\ln(x)}{\ln(2)}$.

Worked Example:

Calculate Information Entropy and Rate A discrete memoryless source emits five symbols with probabilities 0.4, 0.2, 0.2, 0.1, and 0.1. Calculate the amount of information in the first symbol, the entropy of the source, and the information rate if the source emits 2000 symbols per second.

Solution: Step 1: Calculate the amount of information in the first symbol. Given $P(x_1) = 0.4$.

$$I(x_1) = \log_2\left(\frac{1}{0.4}\right) = \frac{\log_{10}(2.5)}{\log_{10}(2)} \approx 1.322 \text{ bits}$$

Step 2: Calculate the entropy of the source. The probabilities are $P = \{0.4, 0.2, 0.2, 0.1, 0.1\}$.

$$H = 0.4\log_2\left(\frac{1}{0.4}\right) + 2 \times 0.2\log_2\left(\frac{1}{0.2}\right) + 2 \times 0.1\log_2\left(\frac{1}{0.1}\right)$$

$$H = 0.4(1.322) + 0.4(2.322) + 0.2(3.322)$$

$$H = 0.5288 + 0.9288 + 0.6644 = 2.122 \text{ bits/symbol}$$

Step 3: Calculate the information rate. Given $r = 2000$ symbols/sec.

$$R = r \times H = 2000 \times 2.122 = 4244 \text{ bits/sec}$$

3.2 Shannon Fano Coding

Methodology:

Shannon Fano Coding Algorithm Shannon Fano coding is a top-down approach for variable-length source coding.

1. **Sort:** List the source symbols in descending order of their probabilities.
2. **Partition:** Divide the list into two subsets such that the sum of probabilities in each subset is as close to equal as possible.
3. **Assign:** Assign a binary '0' to all symbols in the upper subset and a binary '1' to all symbols in the lower subset.
4. **Repeat:** Recursively apply steps 2 and 3 to each subset until each subset contains only one symbol.
5. **Calculate Parameters:**
   - Average codeword length: $L = \sum P_i n_i$ (where n_i is the number of bits in the codeword for symbol i).
   - Coding efficiency: $\eta = \frac{H}{L} \times 100\%$
   - Redundancy: $R = 1 - \frac{H}{L}$

Worked Example:

Shannon Fano Encoding for discrete source Apply Shannon Fano coding for a source emitting symbols with probabilities $\{0.30, 0.25, 0.20, 0.15, 0.10\}$. Calculate the average length and coding efficiency.

Solution: Step 1: Arrange probabilities in descending order. The probabilities are already sorted: $0.30, 0.25, 0.20, 0.15, 0.10$.

Step 2 & 3: Partition and assign bits.

- First partition: Group 1: $\{0.30, 0.25\}$ (sum $= 0.55$), Group 2: $\{0.20, 0.15, 0.10\}$ (sum $= 0.45$). Assign '0' to Group 1, '1' to Group 2.
- Sub-partition Group 1: Subgroup 1a: $\{0.30\}$ (assign '0'), Subgroup 1b: $\{0.25\}$ (assign '1').
- Sub-partition Group 2: Subgroup 2a: $\{0.20\}$ (assign '0'), Subgroup 2b: $\{0.15, 0.10\}$ (assign '1').
- Sub-partition Subgroup 2b: Subgroup 2b1: $\{0.15\}$ (assign '0'), Subgroup 2b2: $\{0.10\}$ (assign '1').

Codewords:

Probability	**Codeword**	**Length (n_i)**
0.30	00	2
0.25	01	2
0.20	10	2
0.15	110	3
0.10	111	3

Step 4: Calculate Average Length (L).

$$L = \sum P_i n_i = 0.30(2) + 0.25(2) + 0.20(2) + 0.15(3) + 0.10(3)$$

$$L = 0.60 + 0.50 + 0.40 + 0.45 + 0.30 = 2.25 \text{ bits/symbol}$$

Step 5: Calculate Entropy (H) and Efficiency (η).

$$H = 0.3\log_2\left(\frac{1}{0.3}\right) + 0.25\log_2\left(\frac{1}{0.25}\right) + 0.2\log_2\left(\frac{1}{0.2}\right) + 0.15\log_2\left(\frac{1}{0.15}\right) + 0.1\log_2\left(\frac{1}{0.1}\right)$$

$$H \approx 0.521 + 0.500 + 0.464 + 0.411 + 0.332 = 2.228 \text{ bits/symbol}$$

$$\eta = \frac{H}{L} \times 100\% = \frac{2.228}{2.25} \times 100\% \approx 99.02\%$$

3.3 Huffman Coding

Methodology:

Huffman Coding Algorithm Huffman coding is a bottom-up approach that guarantees the optimal prefix code.

1. **Sort:** List the source symbols in descending order of their probabilities.
2. **Combine:** Add the two lowest probabilities to form a new composite symbol.
3. **Resort:** Insert this new composite probability back into the list, maintaining the descending order.
4. **Repeat:** Repeat steps 2 and 3 until only one composite probability of 1.0 remains. This forms a tree structure.
5. **Trace back:** Starting from the root of the tree, assign a '0' to the upper branch and a '1' to the lower branch (or vice-versa). Trace the path from the root to each original symbol to determine its codeword.
6. **Calculate Parameters:** Compute average length $L = \sum P_i n_i$ and coding efficiency $\eta = \frac{H}{L} \times 100\%$.

Worked Example:

Huffman Encoding for discrete messages Design a Huffman code for a source emitting symbols with probabilities 0.4, 0.2, 0.2, 0.1, and 0.1. Find the average codeword length.

Solution: Step 1 & 2 & 3: Sort and combine.

- Initial: $0.4, 0.2, 0.2, 0.1, 0.1$
- Combine two lowest $(0.1 + 0.1 = 0.2)$. New list: $0.4, 0.2, 0.2, 0.2$ (combining the last two symbols).
- Combine two lowest $(0.2 + 0.2 = 0.4)$. New list: $0.4, 0.4, 0.2$.
- Combine two lowest $(0.4 + 0.2 = 0.6)$. New list: $0.6, 0.4$.
- Combine final two $(0.6 + 0.4 = 1.0)$.

Step 4: Trace back and assign bits (0 for upper, 1 for lower).

- Symbol 1 (0.4): Path from root directly to top node. Codeword: 1.
- Symbol 2 (0.2): Path: root $\rightarrow$ 0.6 node $\rightarrow$ 0.2 node. Codeword: 00.
- Symbol 3 (0.2): Path: root $\rightarrow$ 0.6 node $\rightarrow$ 0.4 node $\rightarrow$ 0.2 node. Codeword: 011.
- Symbol 4 (0.1): Path: root $\rightarrow$ 0.6 node $\rightarrow$ 0.4 node $\rightarrow$ 0.2 node $\rightarrow$ 0.1 node. Codeword: 0100.
- Symbol 5 (0.1): Path: root $\rightarrow$ 0.6 node $\rightarrow$ 0.4 node $\rightarrow$ 0.2 node $\rightarrow$ 0.1 node. Codeword: 0101.

(Note: Different bit assignment conventions might yield different codewords, but the lengths will remain optimal).

Codewords:

Probability	**Codeword**	**Length (n_i)**
0.4	1	1
0.2	00	2
0.2	011	3
0.1	0100	4
0.1	0101	4

Step 5: Calculate Average Length (L).

$$L = 0.4(1) + 0.2(2) + 0.2(3) + 0.1(4) + 0.1(4)$$

$$L = 0.4 + 0.4 + 0.6 + 0.4 + 0.4 = 2.2 \text{ bits/symbol}$$

3.4 Channel Capacity Theorem

Methodology:

Shannon Hartley Channel Capacity The theoretical maximum continuous data rate that can be transmitted over a noisy channel without errors is given by the Shannon-Hartley theorem.

Formula:

$$C = B\log_2\left(1+\frac{S}{N}\right)$$

Where:

- C = Channel capacity in bits per second (bps)
- B = Channel bandwidth in Hertz (Hz)
- S = Total signal power (in Watts)
- N = Total noise power (in Watts)
- S/N = Signal-to-Noise Ratio (linear scale, NOT in dB)

If S/N is given in decibels (dB), convert it to linear scale first:

$$(S/N)_{\text{linear}} = 10^{\frac{(S/N)_{\text{dB}}}{10}}$$

Worked Example:

Calculate Channel Capacity A communication channel has a bandwidth of 3000 Hz. Calculate the channel capacity if the Signal-to-Noise Ratio (SNR) is 20 dB.

Solution: Step 1: Convert SNR from dB to linear ratio. Given $(S/N)_{\text{dB}} = 20$.

$$20 = 10\log_{10}(S/N)$$

$$\log_{10}(S/N) = 2 \implies S/N = 10^2 = 100$$

Step 2: Apply the Shannon-Hartley formula. Given $B = 3000$ Hz and $S/N = 100$.

$$C = B\log_2\left(1+\frac{S}{N}\right)$$

$$C = 3000\log_2(1+100) = 3000\log_2(101)$$

Step 3: Compute the final value.

$$\log_2(101) = \frac{\log_{10}(101)}{\log_{10}(2)} \approx \frac{2.0043}{0.3010} \approx 6.658$$

$$C = 3000 \times 6.658 = 19974 \text{ bps} \approx 19.97 \text{ kbps}$$

3.5 Hamming Code for Error Correction

Methodology:

Generation of Hamming Code Hamming code is a block code used for single-bit error correction.

1. **Determine number of parity bits (p):** For m data bits, find the minimum p that satisfies:

$$2^p \geq m + p + 1$$

2. **Determine bit positions:** Number the bits starting from 1 (left to right). Positions that are powers

of 2 (1, 2, 4, 8...) are reserved for parity bits ($P_1, P_2, P_4, \ldots$). The remaining positions are for data bits ($D_3, D_5, D_6, D_7, \ldots$).

3. **Calculate parity bits (Even Parity):**
   - P_1 checks positions 1, 3, 5, 7, 9, 11... (bits with 1 in the least significant bit of their binary position index).
   - P_2 checks positions 2, 3, 6, 7, 10, 11... (bits with 1 in the second LSB of their index).
   - P_4 checks positions 4, 5, 6, 7, 12, 13, 14, 15... (bits with 1 in the third LSB).

   Set each parity bit so that the total number of '1's in its checked group is even.

Worked Example:

Generate a 7 Bit Hamming Code Generate the 7-bit even parity Hamming code for the 4-bit data message 1011.

Solution: Step 1: Determine number of parity bits. Given $m = 4$. We need $2^p \geq 4 + p + 1 = p + 5$. For $p = 3$, $2^3 \geq 3 + 5 \implies 8 \geq 8$. So, $p = 3$ parity bits are required. Total code length $= m + p = 4 + 3 = 7$ bits.

Step 2: Position the data and parity bits. Data bits are $D_1 = 1, D_2 = 0, D_3 = 1, D_4 = 1$.

Position	1	2	3	4	5	6	7
Type	P_1	P_2	D_1	P_4	D_2	D_3	D_4
Value	P_1	P_2	1	P_4	0	1	1

Step 3: Calculate parity bits for even parity.

- P_1 **checks positions 1, 3, 5, 7:** Values are $P_1, 1, 0, 1$. Number of 1s in data $= 2$ (which is even). So, $P_1 = 0$.
- P_2 **checks positions 2, 3, 6, 7:** Values are $P_2, 1, 1, 1$. Number of 1s in data $= 3$ (which is odd). So, $P_2 = 1$ to make the total even.
- P_4 **checks positions 4, 5, 6, 7:** Values are $P_4, 0, 1, 1$. Number of 1s in data $= 2$ (which is even). So, $P_4 = 0$.

Step 4: Write the final codeword. The transmitted codeword is 0110011.

Methodology:

Error Detection and Correction using Hamming Code Upon receiving a Hamming code sequence:

1. **Recalculate parity checks (Syndrome calculation):** Evaluate the parity for each parity group including the parity bit itself.
   - C_1: Parity of positions 1, 3, 5, 7, 9...
   - C_2: Parity of positions 2, 3, 6, 7, 10...
   - C_4: Parity of positions 4, 5, 6, 7, 12...

   For even parity, $C_i = 0$ if the group has an even number of 1s, and $C_i = 1$ if odd.
2. **Determine the error position:** The binary word $C_4C_2C_1$ forms the syndrome. Its decimal equivalent indicates the exact bit position that is in error.
3. **Correct the error:** Flip the bit at the error position (change 0 to 1, or 1 to 0). If the syndrome is 0, there is no single-bit error.

Worked Example:

Detect and Correct Single Bit Error A 7-bit Hamming code sequence is received as 0110111. Assuming even parity was used, check if there is an error. If yes, correct it and extract the original 4-bit data.

Solution: Step 1: Write down the received bits by position.

Position	1	2	3	4	5	6	7
Received	0	1	1	0	1	1	1

Step 2: Calculate the syndrome bits (C_1, C_2, C_4).

- C_1 **(Positions 1, 3, 5, 7):** Bits are $0, 1, 1, 1$. Number of 1s $= 3$ (Odd). So, $C_1 = 1$.
- C_2 **(Positions 2, 3, 6, 7):** Bits are $1, 1, 1, 1$. Number of 1s $= 4$ (Even). So, $C_2 = 0$.
- C_4 **(Positions 4, 5, 6, 7):** Bits are $0, 1, 1, 1$. Number of 1s $= 3$ (Odd). So, $C_4 = 1$.

Step 3: Find the error position. The syndrome word is $C_4C_2C_1 = 101_2$. Convert binary 101 to decimal: $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5$. The error is at position 5.

Step 4: Correct the error and extract data. The received bit at position 5 is '1'. Change it to '0'. The corrected 7-bit sequence is 0110011. The original 4-bit data is located at positions 3, 5, 6, and 7. Extracted data: $1, 0, 1, 1$.

CHAPTER 4

Data Communication Techniques and Standards

This chapter covers the computational aspects of data communication techniques, including multiplexing, spread spectrum, and switching methods. These techniques are essential for efficiently utilizing available bandwidth and ensuring reliable transmission in digital networks.

4.1 Multiplexing Techniques

Multiplexing allows multiple signals to share a single data link. The most common types are Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM).

4.1.1 Frequency Division Multiplexing (FDM)

Methodology:

FDM Bandwidth Calculation FDM combines multiple analog signals into a single composite signal by modulating them onto different carrier frequencies. Guard bands are inserted between channels to prevent interference.

Formulas:

1. Total Bandwidth (B_{total}):

$$B_{\text{total}} = (N \times B_{\text{channel}}) + ((N-1) \times B_{\text{guard}})$$

Where:

- N = Number of channels
- B_{channel} = Bandwidth of a single channel
- B_{guard} = Bandwidth of the guard band between adjacent channels

Worked Example:

FDM Bandwidth Requirement Assume that a voice channel occupies a bandwidth of 4 kHz. We need to multiplex 10 voice channels with guard bands of 500 Hz using FDM. Calculate the required total bandwidth.

Solution:

1. **Identify the given parameters:**
   - Number of channels, $N = 10$
   - Bandwidth of one channel, $B_{\text{channel}} = 4$ kHz $= 4000$ Hz
   - Guard band, $B_{\text{guard}} = 500$ Hz
2. **Calculate the total bandwidth for the channels:**

$$B_{\text{channels}} = 10 \times 4000 = 40,000 \text{ Hz} = 40 \text{ kHz}$$

3. **Calculate the total guard band bandwidth:** Since there are 10 channels, there will be $(10-1) = 9$ guard bands.
$$B_{\text{guards}} = 9 \times 500 = 4500 \text{ Hz} = 4.5 \text{ kHz}$$
4. **Calculate the total required bandwidth:**
$$B_{\text{total}} = 40 \text{ kHz} + 4.5 \text{ kHz} = 44.5 \text{ kHz}$$

The required total bandwidth is 44.5 kHz.

4.1.2 Time Division Multiplexing (TDM)

Methodology:

Synchronous TDM Calculations In Synchronous TDM, each connection occupies a fixed time slot in every frame.
Formulas:

1. Bit Rate of Link (R_{link}):
$$R_{\text{link}} = N \times R_{\text{source}}$$
(assuming no framing bits and equal source rates)
2. Frame Rate:
$$\text{Frame Rate} = \frac{\text{Bit Rate of Source}}{\text{Bits per Source per Frame}}$$
3. Frame Duration:
$$\text{Frame Duration} = \frac{1}{\text{Frame Rate}}$$
4. Frame Size (in bits):
$$\text{Frame Size} = (N \times \text{Bits per Source}) + \text{Framing Bits}$$
5. Total Data Rate of Link:
$$\text{Total Data Rate} = \text{Frame Rate} \times \text{Frame Size}$$

Worked Example:

Synchronous TDM Data Rate Four channels, each with a bit rate of 100 kbps, are multiplexed using synchronous TDM. Each time slot carries 1 byte from each channel, and 1 extra bit is added per frame for synchronization. Calculate the frame size, frame rate, frame duration, and total data rate of the link.
Solution:

1. **Identify the parameters:**
   - Number of channels, $N = 4$
   - Channel bit rate, $R_{\text{source}} = 100$ kbps
   - Data per channel per frame = 1 byte = 8 bits
   - Framing bit = 1 bit/frame
2. **Calculate Frame Size:**
$$\text{Frame Size} = (4 \times 8 \text{ bits}) + 1 \text{ bit} = 32 + 1 = 33 \text{ bits}$$

3. **Calculate Frame Rate:** Each frame carries 8 bits from a single channel. Thus, the frame rate must be sufficient to carry $100,000$ bits/sec per channel.

$$\text{Frame Rate} = \frac{100,000 \text{ bits/sec}}{8 \text{ bits/frame}} = 12,500 \text{ frames/sec}$$

4. **Calculate Frame Duration:**

$$\text{Frame Duration} = \frac{1}{\text{Frame Rate}} = \frac{1}{12,500} = 80\text{ţs}$$

5. **Calculate Total Data Rate of the Link:**

$$\text{Total Data Rate} = \text{Frame Rate} \times \text{Frame Size} = 12,500 \times 33 = 412,500 \text{ bps} = 412.5 \text{ kbps}$$

The frame size is 33 bits, frame rate is 12,500 frames/s, frame duration is 80ţs, and total link data rate is 412.5 kbps.

Worked Example:

TDM Interleaving with Different Bit Rates Two channels of 100 kbps and one channel of 200 kbps are to be multiplexed using TDM. How can this be done, and what is the frame size and bit rate if 1 framing bit is used? Assume slot size is 1 bit.

Solution:

1. **Allocate slots based on bit rates:** Since the 200 kbps channel has twice the bit rate of the 100 kbps channels, we allocate it two slots per frame, while the 100 kbps channels get one slot each.

2. **Determine Frame Structure:** Let channels be $C1$ (100 kbps), $C2$ (100 kbps), and $C3$ (200 kbps). Frame pattern: $C1 - C3 - C2 - C3$ (4 data bits per frame).

3. **Calculate Frame Size:**

$$\text{Frame Size} = 4 \text{ data bits} + 1 \text{ framing bit} = 5 \text{ bits/frame}$$

4. **Calculate Frame Rate:** The 100 kbps channels send 1 bit per frame, so the link must transmit frames at a rate equal to the 100 kbps channel's bit rate.

$$\text{Frame Rate} = 100,000 \text{ frames/sec}$$

5. **Calculate Total Bit Rate:**

$$\text{Link Bit Rate} = \text{Frame Rate} \times \text{Frame Size} = 100,000 \times 5 = 500 \text{ kbps}$$

By allocating two slots to the faster channel, the multiplexed link operates at 500 kbps with a frame size of 5 bits.

4.2 Spread Spectrum Techniques

Spread spectrum intentionally spreads the signal over a wider bandwidth to prevent jamming, mitigate multipath fading, and provide secure communication.

4.2.1 Frequency Hopping Spread Spectrum (FHSS)

Methodology:

FHSS Bandwidth Calculations In FHSS, the carrier signal hops from one frequency to another based on a pseudorandom sequence.
Formulas:

1. Total Spread Bandwidth (B_{ss}):
$$B_{ss} \approx N \times B_d$$
Where:
   - N = Number of distinct hopping frequencies
   - B_d = Modulation bandwidth (bandwidth of the original modulated signal)

Worked Example:

FHSS Total Bandwidth An FHSS system utilizes 64 distinct hopping frequencies. If the original data signal requires a bandwidth of 100 kHz, calculate the total bandwidth required for the FHSS system.
Solution:

1. **Identify the parameters:**
   - Number of hopping frequencies, $N = 64$
   - Modulation bandwidth, $B_d = 100$ kHz
2. **Calculate Total Bandwidth:**
$$B_{ss} = N \times B_d = 64 \times 100 \text{ kHz} = 6400 \text{ kHz} = 6.4 \text{ MHz}$$

The total bandwidth required is 6.4 MHz.

4.2.2 Direct Sequence Spread Spectrum (DSSS)

Methodology:

DSSS Chip Rate and Bandwidth In DSSS, each data bit is replaced by a sequence of N chips using a spreading code.
Formulas:

1. Chip Rate (R_c):
$$R_c = N \times R_b$$
Where:
   - R_b = Data bit rate (bps)
   - N = Number of chips per bit (length of chipping code)
2. Spread Bandwidth (B_{ss}):
$$B_{ss} \approx N \times B_d$$
Where B_d is the bandwidth of the original data signal.

Worked Example:

DSSS Chip Rate Calculation A DSSS system transmits data at 1 Mbps. It uses an 11-bit Barker code for spreading. Calculate the chip rate and the total required bandwidth if the original signal requires a bandwidth of 1 MHz.

Solution:

1. **Identify the parameters:**
   - Data rate, $R_b = 1$ Mbps $= 10^6$ bps
   - Number of chips per bit, $N = 11$
   - Original signal bandwidth, $B_d = 1$ MHz
2. **Calculate the Chip Rate:**
$$R_c = N \times R_b = 11 \times 1 \text{ Mbps} = 11 \text{ Mcps (Mega chips per second)}$$
3. **Calculate the Spread Bandwidth:**
$$B_{ss} \approx N \times B_d = 11 \times 1 \text{ MHz} = 11 \text{ MHz}$$

The chip rate is 11 Mcps, and the spread bandwidth is 11 MHz.

4.3 Switching Techniques

Network switching techniques dictate how data paths are established and how packets are forwarded. The choice of switching affects total transmission delay.

Methodology:

Switching Delay Calculation The total delivery time over a network depends on the switching technique (Circuit Switching vs. Packet Switching).

Formulas:

1. Transmission Time (T_t): Time to push bits onto the link.
$$T_t = \frac{\text{Message Size in bits}}{\text{Data Rate in bps}}$$
2. Propagation Delay (T_p): Time for one bit to travel through the physical link.
$$T_p = \frac{\text{Distance}}{\text{Propagation Speed}}$$
3. Circuit Switching Total Time:
$$T_{\text{circuit}} = T_{\text{setup}} + T_t + (K \times T_p) + T_{\text{teardown}}$$
   (where K is the number of hops).
4. Packet Switching Total Time (assuming pipelining across K hops, N packets):
$$T_{\text{packet}} = (K \times T_p) + T_{t\_packet} \times (K + N - 1) + \text{Processing Delays}$$
   (Assuming all links have identical data rates and ignore processing delays if not given).

Worked Example:

Circuit Switching versus Packet Switching A file of size 1 MB is to be sent over a 3-hop path. Each link has a data rate of 1 Mbps and a propagation delay of 2 ms. For circuit switching, setup time is 200 ms and teardown time is 50 ms. For packet switching, the file is broken into packets of 1 KB (including header). Ignore processing delays. Compare the total delivery time for both methods. (1 MB $= 8 \times 10^6$ bits, 1 KB $= 8000$ bits).

Solution:

1. **Circuit Switching Time:**
   - Message Transmission Time $(T_t) = \frac{8\times10^6 \text{ bits}}{1\times10^6 \text{ bps}} = 8$ seconds $= 8000$ ms
   - Total Propagation Delay $(3 \times T_p) = 3 \times 2$ ms $= 6$ ms
   - $T_{\text{circuit}} = T_{\text{setup}} + T_t + (3 \times T_p) + T_{\text{teardown}}$
   - $T_{\text{circuit}} = 200 + 8000 + 6 + 50 = 8256$ ms $= 8.256$ seconds
2. **Packet Switching Time:**
   - Number of packets $(N) = \frac{8\times10^6 \text{ bits}}{8000 \text{ bits/packet}} = 1000$ packets
   - Transmission time per packet $(T_{t\_packet}) = \frac{8000 \text{ bits}}{1\times10^6 \text{ bps}} = 0.008$ s $= 8$ ms
   - Time for the first packet to reach the destination $= (3 \times T_p) + (3 \times T_{t\_packet}) = (3 \times 2) + (3 \times 8) = 6 + 24 = 30$ ms.
   - After the first packet arrives, the remaining 999 packets arrive one after another, pipelined every 8 ms.
   - Total Time = Time for 1st packet + $(N-1) \times T_{t\_packet}$
   - Total Time $= 30$ ms $+ (999 \times 8$ ms$) = 30 + 7992 = 8022$ ms $= 8.022$ seconds

In this scenario, packet switching (8.022 s) is slightly faster than circuit switching (8.256 s) due to the absence of setup/teardown times and the effective use of link pipelining.

CHAPTER 5

Emerging Trends in Data Communication

In modern data communication, emerging technologies such as 5G/6G Networks, Internet of Things (IoT), Edge Computing, and Software-Defined Networking (SDN) are transforming how data is transmitted and processed. This chapter focuses on the computational aspects of these technologies, including throughput estimation, battery life calculation for IoT devices, latency analysis in edge networks, and flow table memory requirements in SDN.

5.1 5G and MIMO Data Rate Calculations

Methodology:

5G Data Rate Calculation In 5G networks, the use of Multiple-Input Multiple-Output (MIMO) technology significantly increases the data rate. The theoretical maximum data rate can be calculated by scaling the Shannon capacity with the number of MIMO spatial streams.

Formulas:

1. **Single Stream Capacity:** $C = B \times \log_2(1 + SNR)$
2. **MIMO Data Rate:** $R_{MIMO} = N_s \times C$

Where:

- B = Channel Bandwidth (in Hz)
- SNR = Signal-to-Noise Ratio (linear, not in dB. If given in dB: $SNR_{linear} = 10^{\frac{SNR_{dB}}{10}}$)
- N_s = Number of spatial streams (MIMO order, e.g. 4 for 4×4 MIMO)
- C = Capacity of a single stream (in bps)
- R_{MIMO} = Total MIMO Data Rate (in bps)

Worked Example:

Calculating 5G MIMO Data Rate A 5G cellular network operates with a channel bandwidth of 100 MHz. The measured Signal-to-Noise Ratio (SNR) is 30 dB. If the base station and user equipment use a 4×4 MIMO configuration (4 spatial streams), calculate the maximum theoretical data rate.

Step 1: Convert SNR from dB to linear scale. Given $SNR_{dB} = 30$.

$$SNR = 10^{\frac{30}{10}} = 10^3 = 1000$$

Step 2: Calculate the capacity of a single spatial stream. Given $B = 100$ MHz $= 100 \times 10^6$ Hz.

$$C = B \times \log_2(1 + SNR)$$

$$C = 100 \times 10^6 \times \log_2(1 + 1000)$$

$$C = 100 \times 10^6 \times \log_2(1001)$$

Using $\log_2(1001) \approx 9.967$:

$$C \approx 100 \times 10^6 \times 9.967 = 996.7 \times 10^6 \text{ bps} \approx 996.7 \text{ Mbps}$$

Step 3: Calculate the total MIMO data rate. For a 4×4 MIMO, $N_s = 4$.

$$R_{MIMO} = N_s \times C = 4 \times 996.7 \text{ Mbps}$$

$$R_{MIMO} = 3986.8 \text{ Mbps} \approx 3.98 \text{ Gbps}$$

Answer: The maximum theoretical data rate for the 5G connection is approximately 3.98 Gbps.

5.2 IoT Device Battery Life Calculation

Methodology:

IoT Battery Life Estimation Internet of Things (IoT) devices often run on batteries and spend most of their time in a low-power "sleep" mode, waking up briefly to transmit data.
Formulas:

1. **Cycle Time:** $T_{cycle} = T_{active} + T_{sleep}$
2. **Average Current:**
$$I_{avg} = \frac{(I_{active} \times T_{active}) + (I_{sleep} \times T_{sleep})}{T_{cycle}}$$
3. **Battery Life (in hours):**
$$\text{Life}_{hours} = \frac{\text{Battery Capacity (in mAh)}}{I_{avg} \text{ (in mA)}}$$
4. **Battery Life (in days/years):**
$$\text{Life}_{days} = \frac{\text{Life}_{hours}}{24}, \quad \text{Life}_{years} = \frac{\text{Life}_{days}}{365}$$

Where:

- I_{active} and I_{sleep} = Current drawn during active and sleep modes (in mA)
- T_{active} and T_{sleep} = Time spent in active and sleep modes per cycle

Worked Example:

Estimating IoT Sensor Battery Life An IoT temperature sensor is powered by a 2500 mAh battery. It wakes up every 10 minutes to transmit data. The transmission takes 2 seconds and draws 50 mA. In sleep mode, it draws 0.02 mA. Calculate the expected battery life in years.
Step 1: Determine cycle times. Total cycle time is 10 minutes = 600 seconds.

$$T_{active} = 2 \text{ seconds}$$

$$T_{sleep} = 600 - 2 = 598 \text{ seconds}$$

Step 2: Calculate the average current consumption.

$$I_{avg} = \frac{(I_{active} \times T_{active}) + (I_{sleep} \times T_{sleep})}{T_{active} + T_{sleep}}$$

$$I_{avg} = \frac{(50 \text{ mA} \times 2 \text{ s}) + (0.02 \text{ mA} \times 598 \text{ s})}{600 \text{ s}}$$

$$I_{avg} = \frac{100 + 11.96}{600} = \frac{111.96}{600} \approx 0.1866 \text{ mA}$$

Step 3: Calculate battery life in hours.

$$\text{Life}_{hours} = \frac{\text{Capacity}}{I_{avg}} = \frac{2500 \text{ mAh}}{0.1866 \text{ mA}} \approx 13397.6 \text{ hours}$$

Step 4: Convert to years.

$$\text{Life}_{days} = \frac{13397.6}{24} \approx 558.2 \text{ days}$$

$$\text{Life}_{years} = \frac{558.2}{365} \approx 1.53 \text{ years}$$

Answer: The expected battery life of the IoT sensor is approximately 1.53 years.

5.3 Edge Computing Latency

Methodology:

Latency Analysis in Edge Networks Edge computing brings processing closer to the data source, reducing propagation delay compared to traditional cloud computing. Total latency includes transmission delay, propagation delay, and processing delay.

Formulas:

1. **Transmission Delay:** $D_{trans} = \frac{L}{R}$
2. **Propagation Delay:** $D_{prop} = \frac{d}{v}$
3. **Total Delay (One-way):** $D_{total} = D_{trans} + D_{prop} + D_{proc}$

Where:

- L = Data payload size (in bits)
- R = Link data rate (in bps)
- d = Distance to server (in meters)
- v = Speed of signal in medium (usually $\approx 2 \times 10^8$ m/s in fiber)
- D_{proc} = Processing delay at the server

Worked Example:

Comparing Cloud and Edge Latency An autonomous vehicle sends a 5 MB sensor payload for hazard detection. The link data rate is 100 Mbps. The speed of signal in the fiber is 2×10^8 m/s.

- **Cloud Server:** Distance is 1000 km. Processing time is 5 ms.
- **Edge Server:** Distance is 10 km. Processing time is 12 ms.

Calculate the total one-way delay for both the Cloud and the Edge server and determine which is faster.

Step 1: Calculate Transmission Delay (same for both). Data size $L = 5 \text{ MB} = 5 \times 8 \times 10^6$ bits $= 40 \times 10^6$ bits. Data rate $R = 100 \text{ Mbps} = 100 \times 10^6$ bps.

$$D_{trans} = \frac{L}{R} = \frac{40 \times 10^6}{100 \times 10^6} = 0.4 \text{ seconds} = 400 \text{ ms}$$

Step 2: Calculate delay for Cloud Server. Distance $d_{cloud} = 1000 \text{ km} = 10^6$ m.

$$D_{prop\_cloud} = \frac{d_{cloud}}{v} = \frac{10^6}{2 \times 10^8} = 0.005 \text{ s} = 5 \text{ ms}$$

$$D_{total\_cloud} = D_{trans} + D_{prop\_cloud} + D_{proc\_cloud}$$

$$D_{total\_cloud} = 400 \text{ ms} + 5 \text{ ms} + 5 \text{ ms} = 410 \text{ ms}$$

Step 3: Calculate delay for Edge Server. Distance $d_{edge} = 10 \text{ km} = 10^4 \text{ m}$.

$$D_{prop\_edge} = \frac{d_{edge}}{v} = \frac{10^4}{2 \times 10^8} = 0.00005 \text{ s} = 0.05 \text{ ms}$$

$$D_{total\_edge} = D_{trans} + D_{prop\_edge} + D_{proc\_edge}$$

$$D_{total\_edge} = 400 \text{ ms} + 0.05 \text{ ms} + 12 \text{ ms} = 412.05 \text{ ms}$$

Answer: Total delay for Cloud is 410 ms and for Edge is 412.05 ms. In this specific scenario, the Cloud is faster because the faster processing time at the cloud server outweighs the additional propagation delay.

5.4 Software-Defined Networking Flow Table Analysis

Methodology:

SDN Flow Table Memory Calculation In Software-Defined Networking (SDN), switches rely on flow tables to route packets. These tables are often implemented using Ternary Content-Addressable Memory (TCAM), which allows extremely fast lookups but has limited capacity and high cost.

Formulas:

1. **Total Memory Required:** $M = N_{rules} \times S_{rule}$
2. **TCAM Utilization Ratio:** $U = \left(\frac{M}{M_{total}}\right) \times 100\%$

Where:

- N_{rules} = Number of active flow rules
- S_{rule} = Size of a single flow rule (in Bytes)
- M_{total} = Total available TCAM memory in the switch

Worked Example:

Calculating SDN Switch TCAM Requirement An SDN OpenFlow switch must store 20000 active flow rules. Each rule consists of a 256-bit match field, a 64-bit instruction field, and a 32-bit counter. The switch has 1 MB of TCAM available. Calculate the size of one rule in Bytes, the total memory required, and the TCAM utilization percentage.

Step 1: Calculate the size of a single flow rule.

$$S_{rule\ (bits)} = \text{Match field} + \text{Instruction field} + \text{Counter}$$

$$S_{rule\ (bits)} = 256 + 64 + 32 = 352 \text{ bits}$$

Convert to Bytes (since 1 Byte = 8 bits):

$$S_{rule} = \frac{352}{8} = 44 \text{ Bytes}$$

Step 2: Calculate the total memory required.

$$N_{rules} = 20000$$

$$M = N_{rules} \times S_{rule} = 20000 \times 44 \text{ Bytes} = 880000 \text{ Bytes}$$

Step 3: Calculate the TCAM utilization. Total available TCAM $M_{total} = 1 \text{ MB} = 1048576$ Bytes (using base-2 for memory).

$$U = \left(\frac{M}{M_{total}}\right) \times 100\% = \left(\frac{880000}{1048576}\right) \times 100\% \approx 83.92\%$$

Answer: The size of one rule is 44 Bytes. Total memory required is 880000 Bytes. The TCAM utilization is 83.92%.

Appendix: Key Formulae

This appendix provides a quick reference to the general formulae and mathematical relationships discussed in this book, categorized by unit.

Unit 1: Data Communication Fundamentals

- **Period and Frequency:**

$$T = \frac{1}{f} \quad \text{and} \quad f = \frac{1}{T}$$

  where T is the period (in seconds) and f is the frequency (in Hz).

- **Wavelength:**

$$\lambda = \frac{v}{f}$$

  where λ is wavelength, v is propagation speed, and f is frequency.

- **Signal-to-Noise Ratio (SNR):**

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}}$$

$$\text{SNR}_{\text{dB}} = 10 \log_{10}(\text{SNR})$$

- **Nyquist Bit Rate (Noiseless Channel):**

$$C_{\text{Nyquist}} = 2 \times B \times \log_2(L)$$

  where C is capacity in bps, B is bandwidth in Hz, and L is the number of signal levels.

- **Shannon Capacity (Noisy Channel):**

$$C = B \log_2(1 + \text{SNR})$$

- **Bit Rate and Baud Rate:**

$$r = \log_2(L)$$

$$R = S \times r$$

  where R is bit rate (bps), S is baud rate (baud), L is number of signal elements, and r is bits per signal element.

- **Network Latency (Delay):**

$$\text{Transmission Delay} = \frac{\text{Message Size}}{\text{Bandwidth}}$$

$$\text{Propagation Delay} = \frac{\text{Distance}}{\text{Propagation Speed}}$$

$$\begin{aligned}\text{Total Latency} = {} & \text{Transmission Delay} + \text{Propagation Delay} \\ & + \text{Queuing Delay} + \text{Processing Delay}\end{aligned}$$

Unit 2: Digital Modulation and Transmission

- **Phase Shift in PSK:**

$$\Delta\phi = \frac{360^\circ}{L}$$

where L is the number of phases.

- **Bandwidth for Digital Modulation (ASK, PSK, QAM):**

$$B = (1 + d) \times S$$

where B is bandwidth, d is a factor related to modulation scheme (between 0 and 1), and S is baud rate.

- **Bandwidth for Frequency Shift Keying (FSK):**

$$B_{\min} = S + 2\Delta f$$

where Δf is the frequency shift from the carrier.

Unit 3: Information Theory and Coding

- **Information Content:**

$$I(x_i) = \log_2\left(\frac{1}{P(x_i)}\right) \text{ bits}$$

where $P(x_i)$ is the probability of occurrence of message x_i.

- **Entropy (Average Information):**

$$H = \sum_{i=1}^{M} P(x_i)\log_2\left(\frac{1}{P(x_i)}\right) \text{ bits/symbol}$$

- **Information Rate:**

$$R = r_s \times H$$

where R is the information rate in bits/second and r_s is the symbol rate (symbols/second).

- **Average Code Length:**

$$L = \sum_{i=1}^{M} P(x_i)n_i$$

where n_i is the number of bits for the i-th symbol.

- **Coding Efficiency:**

$$\eta = \frac{H}{L} \times 100\%$$

- **Hamming Code Parity Bits Calculation:**

$$2^p \geq m + p + 1$$

where p is the number of parity bits and m is the number of data bits.

Unit 4: Multiplexing and Switching

- **Frequency Division Multiplexing (FDM) Bandwidth:**
$$B_{\text{total}} = (N \times B_{\text{channel}}) + ((N-1) \times B_{\text{guard}})$$
where N is the number of channels.

- **Time Division Multiplexing (TDM):**
$$\text{Frame Size} = (N \times \text{Bits per Source}) + \text{Framing Bits}$$
$$\text{Frame Rate} = \frac{\text{Bit Rate of Source}}{\text{Bits per Source per Frame}}$$
$$\text{Frame Duration} = \frac{1}{\text{Frame Rate}}$$
$$\text{Total Data Rate} = \text{Frame Rate} \times \text{Frame Size}$$

- **Spread Spectrum and CDMA:**
$$B_{ss} \approx N \times B_d$$
where B_{ss} is spread spectrum bandwidth and B_d is the data bandwidth.
$$R_c = N \times R_b$$
where R_c is chip rate and R_b is bit rate.

- **Switching Delays:**
$$T_{\text{circuit}} = T_{\text{setup}} + T_t + (K \times T_p) + T_{\text{teardown}}$$
$$T_{\text{packet}} = (K \times T_p) + T_{t\_packet} \times (K + N_{pkt} - 1) + \text{Processing Delays}$$
where K is the number of links, T_t is transmission time, T_p is propagation time, and N_{pkt} is the number of packets.

Unit 5: Advanced Topics (IoT, SDN, Edge/Cloud)

- **IoT Device Power Consumption:**
$$I_{\text{avg}} = \frac{(I_{\text{active}} \times T_{\text{active}}) + (I_{\text{sleep}} \times T_{\text{sleep}})}{T_{\text{active}} + T_{\text{sleep}}}$$
$$\text{Life}_{\text{hours}} = \frac{\text{Battery Capacity}}{I_{\text{avg}}}$$

- **Software-Defined Networking (SDN):**
$$S_{\text{rule}} = \text{Match field} + \text{Instruction field} + \text{Counter}$$
$$M = N_{\text{rules}} \times S_{\text{rule}}$$
$$\text{Utilization} = \left(\frac{M}{M_{\text{total}}}\right) \times 100\%$$
where M is total memory used for rules.

- **Edge and Cloud Computing Delays:**
$$D_{\text{total}} = D_{\text{trans}} + D_{\text{prop}} + D_{\text{proc}}$$

- **MIMO Throughput:**
$$R_{\text{MIMO}} = N_s \times C$$
where N_s is the number of spatial streams and C is the capacity of a single stream.