ETSN10: Network Architecture and Performance - Reference Sheet

Karl Hallsby

Last Edited: January 23, 2020

Contents

List of Theorems

1 Networking Review

This course assumes that you already know about basic networking terms and models. However, this section is meant as a refresher for people who have already taken that particular set of courses, or as a quick introduction for those who have not taken those courses yet.

1.1 Networking Stack

Defn 1 (Networking Stack)**.** The *networking stack* is a set of layers that are built on top of another that are used to implement inter-computer communication. There are 5 layers:

1. Physical Layer
2. Data-Link Layer
3. Network Layer
4. Transport Layer
5. Application Layer

Each of these fulfill different roles, allowing for different implementations to be used interchangeably. Additionally, not all points in the network graph require the entire networking stack. For example, a router or switch does not require the Transport Layer or the Application Layer, since the job of this hardware is to just move packets around.

Each layer of the networking stack add a header and possibly a footer. Information in headers and footers includes source and destination addresses, checksums, packet size, and protocol identifiers.

Defn 2 (Encapsulation)**.** The process of adding headers and footers is called *encapsulation*.

Defn 3 (Decapsulation)**.** The process of removing the headers and footers of a Packet at the other end is called *decapsulation*.

1.2 Physical Layer

Defn 4 (Physical Layer)**.** The *physical layer* in the Networking Stack is the lowest layer in the stack. It consists of the physical connection that is made between computers and the modulation and coding to represent data as a signal on that connection. For obvious reasons, all devices **must** implement this layer.

Some examples of this layer are:

- Copper twisted-pair cables
- Fiber optic wires
- Radio transmission

Remark 4.1. The design of different Physical Layers is more a hardware design question than that of a software implementation issue. So, it is not discussed in this course, beyond the use of cellular networks.

1.3 Data-Link Layer

Defn 5 (Data-Link Layer)**.** The *data-link layer* is the next layer in the Networking Stack. All devices must implement this layer. The data-link layer is responsible for getting data between two devices that have a Physical Layer connection between them. It is also known as the *Medium Access Control (MAC) layer*. It controls when each device should use the link, and may also include error correction and retransmission.

Some examples of these are:

- Ethernet
- Point-to-Point Protocol (PPP)
- 802.11 (WiFi)
- Bluetooth

1.4 Network Layer

Defn 6 (Network Layer)**.** The *network layer* is the third layer in the Networking Stack. This is the last layer that every device must have. It is responsible for end-to-end data delivery between two devices (hosts) anywhere on the network. It is responsible for routing and forwarding.

Some examples of this layer are:

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)

- Routing Protocols

Defn 7 (Routing). *Routing* is the process of determining and selecting the best end-to-end path through a network.

There are algorithms designed to discover the other participants in the network by analyzing the grpah that is created when each participant is considered a node in this graph.

Defn 8 (Forwarding). *Forwarding* is the process of selecting the next hop for the given packet.

Defn 9 (Circuit Switching). In *circuit switching*, dedicated resources are allocated end-to-end for a particular flow of data. No other flows may use those resources while the connection is maintained. Meaning, that during a connection, there is a single dedicated circuit between the host and the receiver.

Defn 10 (Packet Switching). In *packet switching*, no resources are allocated to a flow, and each flow's data is broken up into discrete packets. Each Packet is routed and forwarded independently.

Defn 11 (Packet). A connection from one device to another is a stream. However, to effectively implement many things in the Networking Stack, the stream must be broken into several discrete *packets*.

1.4.1 Discovering Routing Information

1.4.1.1 Dijkstra's Algorithm Dijkstra's Algorithm starts at the source and builds up the shortest path step-by-step.

There are 2 main problems with Dijkstra's Algorithm:

1. The weights along the edges of the graph cannot be negative.
   - If this were to occur, then we would have an infinite cycle going through the negative edge.
2. We **must** know the entire network's topology before beginning the algorithm.
   - Typically, this is not possible on vast networks, like the Internet.

Algorithm 1.1: Dijkstra's Algorithm

Input : A graph with weights on each edge representing "distance" between the nodes.
Output: The path through the graph with the least weight from some starting node.

Assign tentative distance estimate to each node: 0 for the initial node, ∞ for all other nodes.
Mark all nodes except the initial node as unvisited.
Set the initial node as current node.
for *The current node's unvisited neighbours* **do**
  Calculate their distance.
  Compare this new distance against the current distance.
  Take whichever distance is smaller and use that as the new value.
  Mark the current node as visited.
  if *Reached target node* **OR** *No more unvisited nodes with distance* $< \infty$ **then**
    Stop and terminate the algorithm.
  Set the unvisited node with the smallest distance as the new current node.

1.4.1.2 Bellman-Ford Algorithm The Bellman-Ford algorithm works with graphs that have negative weight values on edges. It also **does not** require us to know the entire network's topology before beginning. However, it is much slower than

Dijkstra's Algorithm.

Algorithm 1.2: Bellman-Ford Algorithm

Input : A graph with weights on each edge representing "distance" between the nodes.
Output: The path through the graph with the least weight from some starting node.

Set the distance for the source node to 0 and for all other nodes to ∞.
Set the predecessor of all nodes to null.
for $n \in N$ *where* N *is the number of nodes* **AND** $n \leq N - 1$ **do**
  We repeat $N - 1$ times, because $N - 1$ is the maximum length of a non-cyclic path.
  for *Each edge* (u, v) **do**
    if *distance to* u*, plus the edge weight* $<$ *current distance to* v **then**
      We have discovered a shorter path to v.
      Set the distance to v to this new value and the predecessor of v to u.
for *each edge* (u, v) **do**
  if *distance to* u *plus the edge weight* $<$ *the distance to* v **then**
    The graph contains a negative-weight cycle.
    Terminate.
  We have found the shortest paths to each node from the source.
  Terminate.

1.4.2 Routing Information

Defn 12 (Routing Protocol)**.** A *routing protocol* is used to construct a Routing Table in each node.

Defn 13 (Routing Table)**.** The *routing table* is a table that contains entries on the shortest distances through the current network graph. This is used to help determine where to next forward packets onto.

Each node has a routing table that contains the cost to the destination and the next hop for each destination. For a valid routing table, we need to have:

1. Which destination the packets are going to?
2. What is the next hop after the destination?
3. The "metric" or weight of that particular path through the network graph?
   - We need this term so we can update the routing table as we discover new routes.
   - This way if a new route with a lower cost presents itself, we can replace the higher-cost one with the lower one.

There are 2 ways to collect the routing information necessary to properly route packets through a network. These 2 routing tables are:

1. Static Routing Tables
2. Dynamic Routing Tables

1.4.2.1 Static Routing Tables

Defn 14 (Static Routing Table)**.** *Static routing table*s are manually configured by a user or a program before the system begins routing the information.

This means that there is no overhead to attempt to figure out the network's topology. The information needed has already been given to the device's Routing Table, so that step is completed.

1.4.2.2 Dynamic Routing Tables

Defn 15 (Dynamic Routing Table)**.** A *dynamic routing table* is a Routing Table that is built automatically by the device. To do this, a Routing Protocol is used to build the table while the device is running. This is done using either:

1. Distance Vector Routing Protocol
2. Link State Routing Protocol

Defn 16 (Distance Vector Routing Protocol)**.** The *distance vector routing algorithm* is a Routing Protocol that uses Bellman-Ford Algorithm to construct a Routing Table.

Each node only begins with knowledge of their immediate neighbours and the costs to reach them.

- Nodes then send this information (the routing table) to their neighbours.

- If a neighbour sends us a route that is shorter than one we already have, update our table to reflect this.
- After updating, send the new table to our neighbours.
- If a node goes down, discard any lines in the routing table that have it as the next hop and follow the above to find a new route.

Defn 17 (Link State Routing Protocol)**.** The *link state routing protocol* is a Routing Protocol that uses Dijkstra's Algorithm to construct a Routing Table. Thus, we need to know what the whole network looks like.

- Each node floods the network with the list of nodes it can connect to and the costs to them.
- Every node builds up a picture of the entire network, then can use Dijkstra's Algorithm to determine the shortest path to each destination.
- The Routing Table is then constructed based on the computed shortest paths.

However, the problem with both Distance Vector Routing Protocol and Link State Routing Protocol is that they do not scale well. However, to help with this, both protocols are contained within a single Autonomous System.

Defn 18 (Autonomous System)**.** An *autonomous system* is a smaller network, like a business or home, that performs a Routing Protocol upon itself. It is done this way because both the Distance Vector Routing Protocol and Link State Routing Protocol do not scale to Internet-sized networks well.

To combat this, each of the autonomous systems has a Speaker Node that speaks to the outside world. This speaker node is used in the Path Vector Routing Protocol.

Defn 19 (Speaker Node)**.** A *speaker node* is a specially designated node within a single Autonomous System that communicates **only with other speaker nodes**. Then, the Path Vector Routing Protocol is performed upon all the speaker nodes to construct a wider network graph consisting only of the Autonomous Systems that the speaker nodes belong to.

Defn 20 (Path Vector Routing Protocol)**.** Between Autonomous Systems, we use a variant of Distance Vector Routing Protocol called *path vector routing protocol*. Each Autonomous System has its own Speaker Node. Only the Speaker Nodes can communicate across the Autonomous System boundary, and exchange information about which destinations they can reach and the paths to them.

1.4.3 IP Addresses

Defn 21 (IP Address)**.** *IP Address*es are the unique end-point routing identifiers used on Packets to deliver their data. A natural analogy is that of apartment numbers on apartment buildings.

IPv4 uses 32 bits, split up into 4 8-bit chunks. Each chunk is read as its decimal equivalent, for humans. IPv6 uses 128 bits, where every 16 bits are interpreted as a hexadecimal number.

Defn 22 (Subnet)**.** All hosts that are in the subnet will have IP Addresses that match the number of bits that are present in the subnet. Keeping with the apartment analogy, the subnet is like the address of the apartment building.

For example, in `192.168.2.0/24`, the `24` means the first 24 bits of the subnet are 1's (`255.255.255.0`). So, the first 24 bits, or 3 IP Address blocks (`192.168.2`), will remain the same for every client in that subnet.

Defn 23 (Classless Inter-Domain Routing)**.** *Classless Inter-Domain Routing* or *CIDR* (pronounced "cider") is a hierarchical way to organize IP Addresses.

This allowed:

- Subnets to be of any length.
- Destinations in a router's routing and forwarding tables may be full IP Addresses or Subnets.
  - A destination IP Address will then be matched to the most specific destination in the table when making forwarding decisions.

1.5 Transport Layer

Defn 24 (Transport Layer)**.** The *transport layer* is built where the Network Layer has delivered all the data to the end host.

This means that only the source and destination hosts have this layer. So, routers and switches do not have this, but your phone, laptop, and the server you're connecting to do.

This layer may be responsible for:

- Connection-Oriented Communication
- Multiplexing Data Flows
- Reliable data delivery
- Data flow control
- Data congestion control

Some implementation of this layer do not handle all of these functions, but all **must** handle the multiplexing of different data flows.

There are 2 main transport layers in-use today:

1. Transmission Control Protocol
2. User Datagram Protocol

Defn 25 (Connection-Oriented Communication)**.** A *connection-oriented communication* system means that a connection must be established before any data is transferred. In addition, this connection must be maintained throughout the transmission of data.

Remark 25.1 (Connectionless Communication). In a *connectionless communication* system, hosts can send data at any time without prior connection being made.

1.5.1 Transport Protocols

Defn 26 (User Datagram Protocol)**.** *User Datagraph Protocol (UDP)* is the simplest transport protocol available. It performs multiplexing and error checking, **but nothing else**.

When a host wants to send data to another host, it just sends it, making UDP a Connectionless Communication protocol. If the data gets lost, too bad.

Defn 27 (Transmission Control Protocol)**.** *Transmission Control Protocol (TCP)* is a Connection-Oriented Communication protocol. It performs:

- Multiplexing
- Reliable data delivery
- Error detection
- Ordered data delivery
- Flow control
- Congestion control

Transmission Control Protocol uses a 3-way hand shake to establish a connection.

Defn 28 (Three-Way Handshake)**.** Transmission Control Protocol uses a *Three-Way Handshake* to establish a connection.

The client sends a `SYN` message to the recipient. The receiver sends back a `SYN_ACK` message to acknowledge the receipt of the original `SYN` message. The client then sends another `ACK` to the receiver to acknowledge the receipt of the `SYN_ACK` message.

Because this handshake is asymmetrical, there is a difference between servers and clients. A server must have a Port open, and be listening for client connections.

Defn 29 (Four-Way Handshake)**.** Transmission Control Protocol uses a *four-way handshake* to close a connection. Each host can close its side of the connection independently.

1.5.2 Multiplexing Data Flows

Defn 30 (Port)**.** A *port* is a single instance of a data flow. There are many different flows of data. These may be from different applications or different instances of the same application. In both Transmission Control Protocol and User Datagram Protocol, flows are given unique *port numbers*.

Some of these are standard for particular applications, e.g. port 80 for HTTP (web), port 25 for SMTP (email). The transport protocol uses the port number to deliver data to the correct application.

1.5.3 Data Delivery

1.5.3.1 Data Delivery in Transmission Control Protocol

Defn 31 (Sequence Number)**.** Transmission Control Protocol uses *sequence number*s to make sure data is complete and in order when it is delivered. It also includes error detection, and segments with errors are retransmitted.

1.5.4 Flow Control

Defn 32 (Flow Control)**.** *Flow control* refers to signalling between the sender and receiver to ensure the sender does not send data faster than the receiver can process.

A receiving host may have limited buffer space for incoming messages and takes time to process each message.

1.5.4.1 Flow Control in Transmission Control Protocol

Defn 33 (Sliding Window)**.** Flow Control in Transmission Control Protocol uses a *sliding window* mechanism.

1.5.5 Congestion Control

Defn 34 (Congestion Control)**.** *Congestion control* refers to mechanisms for detecting and reducing Congestion.

Defn 35 (Congestion)**.** *Congestion* in a network occurs when there is too much data being sent and the network is unable to deliver it all. This can result in data loss or long delays.

1.5.5.1 Congestion Control in Transmission Control Protocol

Defn 36 (Congestion Window)**.** In Transmission Control Protocol, lost data is considered a sign of Congestion and the sender should reduce its rate. The sender has a *congestion window*, which refers to the maximum number of unacknowledged segments that may be in transit at a time.

A Transmission Control Protocol connection begins in **slow start**, where the Congestion Window is doubled every round trip. When a threshold is reached, it changes to *congestion avoidance*, where the congestion window increases by 1 maximum segment size each time. When packet loss is detected, the congestion window is halved.

$$\text{BW}_{Max} = \frac{\text{MSS} \times \sqrt{\frac{3}{2}}}{\text{RTT} \times \sqrt{p}} \tag{1.1}$$

- BW_{Max}: Maximum bandwidth.
- MSS: Maximum Transmission Control Protocol Packet size.
- RTT: Round trip time.
- p: Packet loss probability.

1.6 Application Layer

Defn 37 (Application Layer)**.** The *application layer* is the last and highest layer in the Networking Stack. This layer is created by the actual applications that interface with the user, their email client, web browser, games, etc. This is the layer where data is actually generated and consumed by any application that communicates over the Internet.

Some examples of item in this layer are:

- Hypertext Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (SMTP)
- Extensible Messaging and Presence Protocol (XMPP)
- Skype

2 Probability Review

This section is meant to quick review and introduce the equations that will be used throughout this course. It is not meant to be comprehensive and/or in-depth. For more information about the topic of probability and statistics, refer to the Math 374 - Probability and Statistics document.

2.1 Axioms of Probability

Defn 38 (Sample Space)**.** The *sample space* is the set of all possible outcomes in a random experiment. It is denoted with the capital Greek omega.

$$\Omega \tag{2.1}$$

Defn 39 (Event)**.** An *event* is a subset of the Sample Space that we are interested in. These are generally denoted with capital letters.

$$A \subseteq \Omega \tag{2.2}$$

Defn 40 (Mutually Exclusive)**.** Any two Events are *mutually exclusive* if the equation below holds.

$$\text{P}(A \cup B) = \text{P}(A) + \text{P}(B) \tag{2.3}$$

Laws that follow from the above definitions (Definitions 38 to 40).

1. The conjugate of the Event occurring, i.e. the Event **not** occurring is:

$$\mathrm{P}(\bar{A}) = 1 - \mathrm{P}(A) \tag{2.4}$$

2. The probability of the union of 2 Events is:

$$\mathrm{P}(A \cap B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \cup B) \tag{2.5}$$

   - If A and B are Mutually Exclusive, then $\mathrm{P}(A \cup B) = 0$.

2.2 Conditional Probability

Defn 41 (Conditional Probability)**.** *Conditional probability* is the probability of an Event occurring when it is known that another Event occurred.

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)} \tag{2.6}$$

Defn 42 (Independent)**.** Events are *independent* if the probability of the events' intersection is the same as their probabilities multipled together.

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\,\mathrm{P}(B) \tag{2.7}$$

Remark 42.1 (Conditional Probability and Independent Events). If A and B are Events and are Independent, then

$$\begin{aligned} \mathrm{P}(A \mid B) &= \mathrm{P}(A) \\ \mathrm{P}(B \mid A) &= \mathrm{P}(B) \end{aligned} \tag{2.8}$$

2.3 Random Variables

Defn 43 (Random Variable)**.** A *random variable* is a mapping from an Event's outcome to a real number.

There are 2 types of random variables, based on what the mapping ends up with:

1. Discrete Random Variables are mapped to integers, $\mathbb{Z}$.
2. Continuous Random Variables are mapped to the real numbers, $\mathbb{R}$.

2.3.1 Discrete Random Variables

Defn 44 (Discrete Random Variable)**.** A *Discrete Random Variable* is one whose values are mapped from an Event's outcome to the integer numbers ($\mathbb{Z}$). These Random Variables are drawn from outcomes that are finite (sides on a die) or countably infinite.

The probability of a single value of the discrete random variable is denoted differently here than in the course material. The subscript refers to which discrete random variable we are working with (in this case X) and the variable in parentheses is the value we are calculating for (in this case $x \in X$).

$$p_X(x) \tag{2.9}$$

The sum of all probabilities for values that the discrete random variable can take **must** sum to 1.

$$\sum_{x \in X} p_X(x) = 1 \tag{2.10}$$

The mean or expected value of a discrete random variables is shown below:

$$\begin{aligned} \mu &= \sum_{x \in X} x p_X(x) \\ \mathbb{E}[X] &= \sum_{x \in X} x p_X(x) \end{aligned} \tag{2.11}$$

The variance of a discrete random variable is how "off" a value from the random variable is from the mean/expected value.

$$\begin{aligned} \sigma^2 &= \sum_{x \in X} (x - \mu)^2 p_X(x) \\ \mathrm{VAR}[X] &= \sum_{x \in X} \big(x - \mathbb{E}[X]\big)^2 p_X(x) \end{aligned} \tag{2.12}$$

The standard deviation is the square root of the variance.

$$\begin{aligned}\sigma = \sqrt{\sigma^2} &= \sqrt{\sum_{x \in X} (x-\mu)^2 p_X(x)} \\ \mathrm{STD}[X] = \sqrt{\mathrm{VAR}[X]} &= \sqrt{\sum_{x \in X} \left(x - \mathbb{E}[X]\right)^2 p_X(x)}\end{aligned} \tag{2.13}$$

There are 5 different Discrete Random Variable distributions that we will be heavily utilizing in this course.

2.3.1.1 Uniform Random Variable

Defn 45 (Uniform Random Variable)**.** The *uniform random variable* is a Discrete Random Variable whose probabilities for each outcome is equal.

For a Discrete Random Variable X, which has $|X|$ possible values,

$$p_X(x) = \frac{1}{|X|} \tag{2.14}$$

Example 2.1: Uniform Random Variable. Lecture 1

For example, the roll of a die is typically modelled as a uniform random variable. Find the probability distribution function, the expected value, and the variance.

Let's assume this is a 6-sided die. And let's map each side's number to a value in the range of $X \in [1, 6]$.
Using Equation (2.14), we can find the probability distribution easily.

$$p_X(x) = \begin{cases} \frac{1}{6} & x = 1 \\ \frac{1}{6} & x = 2 \\ \frac{1}{6} & x = 3 \\ \frac{1}{6} & x = 4 \\ \frac{1}{6} & x = 5 \\ \frac{1}{6} & x = 6 \end{cases}$$

Using Equation (2.11), we can find the the expected value/mean.

$$\begin{aligned}\mu = \mathbb{E}[X] &= \sum_{x=1}^{6} x p_X(x) \\ &= \frac{1}{6} + \frac{2}{6} + \frac{3}{6} + \frac{4}{6} + \frac{5}{6} + \frac{6}{6} \\ &= \frac{1+2+3+4+5+6}{6} \\ &= \frac{21}{6} = 3.5\end{aligned}$$

Using Equation (2.12), we can find the variance.

$$\begin{aligned}\sigma^2 = \mathrm{VAR}[X] &= \sum_{x=1}^{6} \left(x - \mathbb{E}[X]\right)^2 p_X(x) \\ &= \sum_{x=1}^{6} (x-3.5)^2 \left(\frac{1}{6}\right) \\ &= 2.91667\end{aligned}$$

Using Equation (2.13), we can find the standard deviation.

$$\begin{aligned}
\sigma = \text{STD}[X] &= \sqrt{\sum_{x=1}^{6} \left(x - \mathbb{E}[X]\right)^2 p_X(x)} \\
&= \sqrt{\sum_{x=1}^{6} (x - 3.5)^2 \left(\frac{1}{6}\right)} \\
&= \sqrt{2.91667} \\
&= 1.70783
\end{aligned}$$

2.3.1.2 Bernoulli Random Variable

Defn 46 (Bernoulli Random Variable)**.** The *Bernoulli random variable* is one where **only one** test occurs, and there are only 2 outcomes.

The probability of success is denoted

$$p_X(\text{success}) = p \tag{2.15}$$

The probability of failure is denoted

$$p_X(\text{failure}) = 1 - p \tag{2.16}$$

The mean/expected value is:

$$\mu = \mathbb{E}[X] = p \tag{2.17}$$

The variance is:

$$\sigma^2 = \text{VAR}[X] = (1 - p)p \tag{2.18}$$

2.3.1.3 Binomial Random Variable

Defn 47 (Binomial Random Variable)**.** The *binomial random variable* is one where n trials are run with no stops for a success, where the Random Variable in each run is a Bernoulli Random Variable.

The probability of k successes with n trials is

$$\binom{n}{k} p^k (1 - p)^{n-k} = \frac{n!}{k!(n-k)!}(1 - p)^{n-k} \tag{2.19}$$

The mean/expected value after n trials is

$$\mu = \mathbb{E}[X] = np \tag{2.20}$$

The variance after n trials is

$$\sigma^2 = \text{VAR}[X] = np(1 - p) \tag{2.21}$$

2.3.1.4 Geometric Random Variable

Defn 48 (Geometric Random Variable)**.** The *geometric random variable* is one where n trials are run, where the nth trial is a success, meaning there are $n - 1$ previous failures. The Random Variable in each run is a Bernoulli Random Variable.

This means **each trial** has a probability of success of

$$p_X(\text{success}) = p \tag{2.22}$$

And **each trial** has a probability of failure of

$$p_X(\text{failure}) = 1 - p \tag{2.23}$$

The mean/expected value is

$$\mu = \mathbb{E}[X] = \frac{1}{p} \tag{2.24}$$

The variance is

$$\sigma^2 = \text{VAR}[X] = \frac{1 - p}{p^2} \tag{2.25}$$

2.3.1.5 Poisson Random Variable

Defn 49 (Poisson Random Variable)**.** The *Poisson random variable* is used to model the number of independent events that occur over a given period of time.

The Poisson random variable has one parameter,

$$\lambda \tag{2.26}$$

λ is the average number of events per unit of time.

The probability function for the value of $x \in X$ of this random variable is

$$p_X(x) = e^{-\lambda}\left(\frac{\lambda^x}{k!}\right) \tag{2.27}$$

The mean/expected value is

$$\mu = \mathbb{E}[X] = \lambda \tag{2.28}$$

The variance is

$$\sigma^2 = \text{VAR}[X] = \lambda \tag{2.29}$$

2.3.2 Continuous Random Variables

Defn 50 (Continuous Random Variable)**.** A *Continuous Random Variable* is one whose values are mapped from an Event's outcome to the real numbers ($\mathbb{R}$).

Continuous random variables cannot be calculated at single points, but must be integrated over a range. This is called the Cumulative Distribution Function (CDF). The subscript refers to the random variable we are using, and x is the value being calculated for.

$$F_X(x) = \text{P}(X \leq x) \tag{2.30}$$

Continuous random variables have a Probability Density Function (PDF), which is the derivative of the CDF.

$$f_X(x) = \frac{d}{dx}F_X(x) \tag{2.31}$$

This PDF must integrate to 1

$$\int\limits_{x \in X} f_X(x) = 1 \tag{2.32}$$

The expected value/mean is

$$\begin{aligned} \mu &= \int\limits_{x \in X} x f_X(x)\, dx \\ \mathbb{E}[X] &= \int\limits_{x \in X} x f_X(x)\, dx \end{aligned} \tag{2.33}$$

The variance is

$$\begin{aligned} \sigma^2 &= \int\limits_{x \in X} (x-\mu)^2 f_X(x)\, dx \\ \text{VAR}[X] &= \int\limits_{x \in X} \left(x - \mathbb{E}[X]\right)^2 f_X(x)\, dx \end{aligned} \tag{2.34}$$

2.3.2.1 Negative Exponential Random Variable

Defn 51 (Negative Exponential Random Variable)**.** The *negative exponential random variable*, sometimes shortened to exponential random variable, is a Continuous Random Variable that has 1 parameter μ, the rate of decay.

The negative exponential random variable has a PDF of

$$f_X(x) = \mu e^{-\mu x} \tag{2.35}$$

The negative exponential random variable has a CDF of

$$F_X(x) = 1 - e^{-\mu x} \tag{2.36}$$

Its mean/expected value is

$$\begin{aligned}\mu &= \frac{1}{\mu} \\ \mathbb{E}[X] &= \frac{1}{\mu}\end{aligned} \tag{2.37}$$

Its variance is

$$\begin{aligned}\sigma^2 &= \left(\frac{1}{\mu}\right)^2 \\ \mathrm{VAR}[X] &= \left(\frac{1}{\mu}\right)^2\end{aligned} \tag{2.38}$$

Its standard deviation is

$$\begin{aligned}\sigma = \sqrt{\sigma^2} &= \frac{1}{\mu} \\ \mathrm{STD}[X] = \sqrt{\mathrm{VAR}[X]} &= \frac{1}{\mu}\end{aligned} \tag{2.39}$$

2.3.2.2 Gaussian Random Variable

Defn 52 (Gaussian Random Variable)**.** The *Gaussian random variable*, sometimes called the normal random variable, has 2 parameters, μ and σ.

The Probability Density Function (PDF) is

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x-\mu^2}{2\sigma^2}} \tag{2.40}$$

The mean/expected value is

$$\begin{aligned}\mu &= \mu \\ \mathbb{E}[X] &= \mu\end{aligned} \tag{2.41}$$

The variance is

$$\begin{aligned}\sigma^2 &= \sigma^2 \\ \mathrm{VAR}[X] &= \sigma^2\end{aligned} \tag{2.42}$$

The standard deviation is

$$\begin{aligned}\sigma = \sqrt{\sigma^2} &= \sigma \\ \mathrm{STD}[X] = \sqrt{\mathrm{VAR}[X]} &= \sigma\end{aligned} \tag{2.43}$$

2.4 Multiple Random Variables

Defn 53 (Joint Probability Function)**.** If X, Y are Discrete Random Variables, the probability of the joint event occurring is

$$p_{X,Y}(x, y) \tag{2.44}$$

where the subscripted X, Y refers to the Discrete Random Variables in use and x, y refers to the values being calculated for.

$$\mathrm{P}(x, y) = \mathrm{P}(X = x, Y = y) \tag{2.45}$$

Defn 54 (Joint Cumulative Distribution Function)**.** If X, Y are Continuous Random Variables, then their *joint cumulative distribution function* is

$$F_{X,Y}(x, y) = \mathrm{P}(X \leq x, Y \leq y) \tag{2.46}$$

Remark 54.1 (Independent Multiple Random Variables). X, Y are independent if and only if

$$F_{X,Y}(x, y) = F_X(x)F_Y(y) \tag{2.47}$$

Defn 55 (Joint Probability Density Function)**.** If X, Y are Continuous Random Variables, then their *joint probability density function* is

$$f_{X,Y}(x, y) = \frac{\partial^2}{\partial x \partial y} F_{X,Y}(x, y) \tag{2.48}$$

2.5 Properties of Expected Value

(i) Expected Value obeys the laws of linearity.

$$\mathbb{E}[aX + b] = a\,\mathbb{E}[X] + b \tag{2.49}$$

(ii) For any Random Variables X, Y

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y] \tag{2.50}$$

(iii) For any Independent Multiple Random Variables X, Y,

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y] \tag{2.51}$$

2.6 Properties of Variance

(i) Variance does not obey the laws of linearity.

$$\mathrm{VAR}[aX + b] = a^2\,\mathrm{VAR}[X] \tag{2.52}$$

(ii) For any Independent Multiple Random Variables X, Y,

$$\mathrm{VAR}[X + Y] = \mathrm{VAR}[X] + \mathrm{VAR}[Y] \tag{2.53}$$

2.7 Covariance

Defn 56 (Covariance)**.** The *covariance* of two Random Variables is defined as

$$\begin{aligned} \mathrm{Cov}[X, Y] &= \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)\big(Y - \mathbb{E}[Y]\big)\Big] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y] \end{aligned} \tag{2.54}$$

Remark 56.1 (Relationship Between Covariance and Variance). Variance is actually a case of Covariance of a Random Variable with itself.

$$\mathrm{VAR}[X] = \mathrm{Cov}[X, X] \tag{2.55}$$

The Covariance of two Random Variables can have 3 possible values:

1. Positive: If one Random Variable increases, the other does too.
2. Negative: If one Random Variable increases, the other decreases.
3. Zero: If one Random Variable increases, the other does nothing.

2.8 Correlation Coefficient

Defn 57 (Correlation)**.** The *correlation of X and Y* is defined as the $1, 1$ moment.

$$\mathbb{E}\left[X^1 Y^1\right] \tag{2.56}$$

Defn 58 (Correlation Coefficient)**.** The *correlation coefficient of X and Y* is a measure of the **LINEAR relationship** between X and Y. It does not say anything about nonlinear dependence. It is defined as

$$\begin{aligned} \rho_{X,Y} &= \frac{\mathrm{Cov}[X, Y]}{\sigma_X \sigma_Y} \\ &= \frac{\mathrm{Cov}[X, Y]}{\mathrm{STD}[X]\,\mathrm{STD}[Y]} \end{aligned} \tag{2.57}$$

Remark 58.1. $\rho_{X,Y}$ only ranges from $-1 \leq \rho_{X,Y} \leq 1$.

Remark 58.2. The closer $\rho_{X,Y}$ are to $+1$, the closer X and Y are to having a positive linear relationship. The closer $\rho_{X,Y}$ are to -1, the closer X and Y are to having a negative linear relationship. The closer $\rho_{X,Y}$ are to 0, the closer X and Y are to being *uncorrelated.*

If $\rho_{X,Y} = 0$, then

- If X, Y are Independent Multiple Random Variables, then they are uncorrelated.
- If X, Y are uncorrelated ($\rho_{X,Y} = 0$), then they may still **not** be independent.

2.9 Stochastic Processes

Defn 59 (Stochastic Process)**.** A *stochastic process* or random process $\mathbf{x}(t)$ has 2 meanings:

1. For every time instant, $x(t)$ is a Random Variable.
2. For every point (sample) in an outcome space Ω, $x(t)$ is a real-valued function of time.

There are 4 different possible types:

1. Discrete-Time and discrete-value
2. Discrete-Time and continuous-value
3. Continuous-Time and discrete-value
   - Packet arrival to destination over time.
4. Continuous-Time and continuous-value
   - End-to-end delay in a network.

Defn 60 (Stationary Process)**.** A Stochastic Process $x(t)$ is called (weakly) stationary if:

- $\mathbb{E}[x(t)]$ is a constant, it is independent of t.
- The Correlation or Covariance of the process at 2 points in time $x(t_1)$ and $x(t_2)$, is a function of the difference $t_2 - t_1$ **only**.

2.10 The Poisson Process

Defn 61 (The Poisson Process)**.** This is a continuous-time, discrete-value Stochastic Process. It is also a Stationary Process. This process in **memoryless**, the previous time instant's values do not affect this time instant's value.

This is very commonly used to describe the arrivals into a queueing system or network. There is a parameter λ that is the average rate (packets per second, in this case).

There are 3 different definitions for this process:

1. Behaviour for a very small interval of time.
   - Approximately a Bernoulli Random Variable.
   - The time interval is small enough such that only 1 event occurs with probability $\lambda\Delta t + o(\Delta t)$, which becomes $\lambda\Delta t$ for small Δt.
   - 0 events occur with probability $1 - \lambda\Delta t + o(\Delta t)$, which becomes $1 - \lambda\Delta t$ for small Δt.
   - Probabilities between non-overlapping intervals are independent
2. Behaviour over a longer period of time.
   - Receive multiple events, k.
   - Probability of k events is $p_X(k) = e^{-\lambda}\frac{\lambda^k}{k!}$
3. Behaviour between events.
   - Approximately a Negative Exponential Random Variable.
   - Time t between events, $p(t) = \lambda e^{-\lambda t}$

Poisson Process Definition 1/2 Equivalence. Find the probability of k arrivals in time t. Divide the period t into N small periods of Δt each such that as $N \to \infty$, $\Delta t \to 0$.

Approximate the sum of N independent Bernoulli Random Variables with probability $\frac{\lambda t}{N}$ each.

The probability that k out of N will be 1 and the rest 0 is:

$$\frac{N!}{k!(N-k)!}\left(\lambda\frac{t}{N}\right)^k\left(1-\lambda\frac{t}{N}\right)^{N-k}$$

When N is large, 3 things happen.

1. $$\frac{N!}{(N-k)!} \approx N^k \tag{2.58}$$
2. $$\left(1-\lambda\frac{t}{N}\right)^N \approx e^{-\lambda t} \tag{2.59}$$
3. $$\left(1-\lambda\frac{t}{N}\right)^{-k} \approx 1 \tag{2.60}$$

So, when $N \to \infty$, we obtain

$$\mathrm{P}(k \text{ arrivals in time } t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}$$

■

Poisson Process Definition 2/3 Equivalence. Since

$$\mathrm{P}(k \text{ arrivals in time } t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}$$

The probability that time between arrivals is more than t is the same as the probability of 0 arrivals during time $t = e^{-\lambda t}$. So, this is exponentially distributed as required. ■

Poisson Process Definition 3/1 Equivalence. Suppose that the last arrival was time t ago.

A priori probability that the next arrival comes after more than t from the previous arrival is $e^{-\lambda t}$.
A priori probability this next arrival is after more than $t + \Delta t = e^{-\lambda(t+\Delta t)}$.
A priori probability that this next arrival happens within the next $\Delta t = e^{-\lambda t} - e^{-\lambda(t+\Delta t)}$.

Conditioning this on knowing that there were no other arrivals in the last t time, and using a Taylor expansion:

$$\frac{e^{-\lambda t} - e^{-\lambda(t+\Delta t)}}{e^{-\lambda t}} = 1 - e^{-\lambda \Delta t} \qquad \approx \lambda \Delta t - \frac{(\lambda \Delta t)^2}{2} + \frac{(\lambda \Delta t)^3}{6} - \dots$$

Since this result is independent of t, this satisfies the requirement of the independence from previously arrived messages.

■

2.10.1 Sums of the Poisson Processes

The superposition of several different Poisson processes.

- There are m independent Poisson Processes, with rates λ_i for $i = 1, 2, \dots, m$
- Sum is also a Poisson Process: $\lambda = \sum_{i=1}^{m} \lambda_i$

All of this comes together for the equation below. It models the number of things k that have arrived over a certain period of time t.

$$\mathrm{P}(\mathbf{x}(t) = k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \tag{2.61}$$

3 Performance Evaluation

We do this to:

- Evaluate existing systems
- Design new network systems
- Predict system behaviours under different conditions

3.1 Performance Measures

How do we measure the performance of a large complex network?

- Data transfer speed
- Reliability:
  - Guaranteed throughput
  - Guarantee of any other performance measurement
  - Integrity of data
  - Predictability of errors
  - Uptime/Downtime/Availability
- Security
- User satisfaction
- Sustainability
- Maintainability
- Throughput/Goodput
- Delay/Latency
- Energy Efficiency
- Jitter (Delay variance)
- Packet Loss

3.2 Performance Evaluation

How can we evaluate the performance of a large complex network?

- Analysis: Mathematical modelling, calculations.
- Simulation: Software implementation of system model.
- Real-World Experimentation: Testing the actual system.

Analysis	**Simulation**	**Experimentation**
— Requires detailed understanding of system properties	+ Only requires modelling the environment with a straightforward implementation	++ No modelling or understanding of how the system required
— Usually requires approximations and simplifying assumptions.	+ Possible to implement complex details of system without approximation	++ Captures complete behaviour of system and environment without approximation.
++ Allows for deep insight for a broad range of scenarios.	+ Allows insight to broad range of scenarios.	— Requires deployment of every scenarios tested and may be difficult to reproduce.
+ Rare events and boundary cases are included.	+ Study of rare events is tricky, but possible.	— Rare events may be impossible to study.

Table 3.1: Performance Evaluation Pros and Cons

3.3 Statistical Data Analysis

Only analysis produces exact results. Simulation and experimentation produce samples from some underlying random distribution. This means we need to perform statistical analysis of these resuilts.

3.3.1 Sampling

We assume a random variable Z with an unknown probability distribution, but we can assume a distribution to start with. We estimate the key distribution metrics:

- Mean (1st moment)
- Variance (2nd moment)
- Variance of the variance (3rd moment)

We obtain n **independent** samples, $z_1, z_2, \ldots, z_n$. To estimate the mean/expected value, we use the equation below.

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{3.1}$$

Where $\bar{z}$ is also a Random Variable. So, we can perform an expected value calculation on $\bar{z}$.

$$\mathbb{E}[\bar{z}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} z_i\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[z_i] \tag{3.2}$$

So, as $n \to \infty$, $\mathbb{E}[\bar{z}] \to \mu$.

A Complex Numbers

Complex numbers are numbers that have both a real part and an imaginary part.

$$z = a \pm bi \tag{A.1}$$

where

$$i = \sqrt{-1} \tag{A.2}$$

Remark (i vs. j for Imaginary Numbers). Complex numbers are generally denoted with either i or j. Since this is an appendix section, I will denote complex numbers with i, to make it more general. However, electrical engineering regularly makes use of j as the imaginary value. This is because alternating current i is already taken, so j is used as the imaginary value instad.

$$Ae^{-ix} = A\left[\cos(x) + i\sin(x)\right] \tag{A.3}$$

A.1 Complex Conjugates

If we have a complex number as shown below,

$$z = a \pm bi$$

then, the conjugate is denoted and calculated as shown below.

$$\overline{z} = a \mp bi \tag{A.4}$$

Defn A.1.1 (Complex Conjugate)**.** The conjugate of a complex number is called its *complex conjugate.* The complex conjugate of a complex number is the number with an equal real part and an imaginary part equal in magnitude but opposite in sign.

The complex conjugate can also be denoted with an asterisk ($*$). This is generally done for complex functions, rather than single variables.

$$z^* = \overline{z} \tag{A.5}$$

A.1.1 Complex Conjugates of Exponentials

$$\overline{e^z} = e^{\overline{z}} \tag{A.6}$$

$$\overline{\log(z)} = \log(\overline{z}) \tag{A.7}$$

A.1.2 Complex Conjugates of Sinusoids

Since sinusoids can be represented by complex exponentials, as shown in Appendix B.2, we could calculate their complex conjugate.

$$\begin{aligned}\overline{\cos(x)} &= \cos(x) \\ &= \frac{1}{2}\left(e^{ix} + e^{-ix}\right)\end{aligned} \tag{A.8}$$

$$\begin{aligned}\overline{\sin(x)} &= \sin(x) \\ &= \frac{1}{2i}\left(e^{ix} - e^{-ix}\right)\end{aligned} \tag{A.9}$$

B Trigonometry

B.1 Trigonometric Formulas

$$\sin(\alpha) + \sin(\beta) = 2\sin\left(\frac{\alpha+\beta}{2}\right)\cos\left(\frac{\alpha-\beta}{2}\right) \tag{B.1}$$

$$\cos(\theta)\sin(\theta) = \frac{1}{2}\sin(2\theta) \tag{B.2}$$

B.2 Euler Equivalents of Trigonometric Functions

$$e^{\pm j\alpha} = \cos(\alpha) \pm j\sin(\alpha) \tag{B.3}$$

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2} \tag{B.4}$$

$$\sin(x) = \frac{e^{jx} - e^{-jx}}{2j} \tag{B.5}$$

$$\sinh(x) = \frac{e^{x} - e^{-x}}{2} \tag{B.6}$$

$$\cosh(x) = \frac{e^{x} + e^{-x}}{2} \tag{B.7}$$

B.3 Angle Sum and Difference Identities

$$\sin(\alpha \pm \beta) = \sin(\alpha)\cos(\beta) \pm \cos(\alpha)\sin(\beta) \tag{B.8}$$

$$\cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta) \tag{B.9}$$

B.4 Double-Angle Formulae

$$\sin(2\alpha) = 2\sin(\alpha)\cos(\alpha) \tag{B.10}$$

$$\cos(2\alpha) = \cos^2(\alpha) - \sin^2(\alpha) \tag{B.11}$$

B.5 Half-Angle Formulae

$$\sin\left(\frac{\alpha}{2}\right) = \sqrt{\frac{1-\cos(\alpha)}{2}} \tag{B.12}$$

$$\cos\left(\frac{\alpha}{2}\right) = \sqrt{\frac{1+\cos(\alpha)}{2}} \tag{B.13}$$

B.6 Exponent Reduction Formulae

$$\sin^2(\alpha) = \frac{1-\cos(2\alpha)}{2} \tag{B.14}$$

$$\cos^2(\alpha) = \frac{1+\cos(2\alpha)}{2} \tag{B.15}$$

B.7 Product-to-Sum Identities

$$2\cos(\alpha)\cos(\beta) = \cos(\alpha-\beta) + \cos(\alpha+\beta) \tag{B.16}$$

$$2\sin(\alpha)\sin(\beta) = \cos(\alpha-\beta) - \cos(\alpha+\beta) \tag{B.17}$$

$$2\sin(\alpha)\cos(\beta) = \sin(\alpha+\beta) + \sin(\alpha-\beta) \tag{B.18}$$

$$2\cos(\alpha)\sin(\beta) = \sin(\alpha+\beta) - \sin(\alpha-\beta) \tag{B.19}$$

B.8 Sum-to-Product Identities

$$\sin(\alpha) \pm \sin(\beta) = 2\sin\left(\frac{\alpha \pm \beta}{2}\right)\cos\left(\frac{\alpha \mp \beta}{2}\right) \tag{B.20}$$

$$\cos(\alpha) + \cos(\beta) = 2\cos\left(\frac{\alpha + \beta}{2}\right)\cos\left(\frac{\alpha - \beta}{2}\right) \tag{B.21}$$

$$\cos(\alpha) - \cos(\beta) = -2\sin\left(\frac{\alpha + \beta}{2}\right)\sin\left(\frac{\alpha - \beta}{2}\right) \tag{B.22}$$

B.9 Pythagorean Theorem for Trig

$$\cos^2(\alpha) + \sin^2(\alpha) = 1^2 \tag{B.23}$$

B.10 Rectangular to Polar

$$a + jb = \sqrt{a^2 + b^2}e^{j\theta} = re^{j\theta} \tag{B.24}$$

$$\theta = \begin{cases} \arctan\left(\frac{b}{a}\right) & a > 0 \\ \pi - \arctan\left(\frac{b}{a}\right) & a < 0 \end{cases} \tag{B.25}$$

B.11 Polar to Rectangular

$$re^{j\theta} = r\cos(\theta) + jr\sin(\theta) \tag{B.26}$$

C Calculus

C.1 Fundamental Theorems of Calculus

Defn C.1.1 (First Fundamental Theorem of Calculus)**.** The *first fundamental theorem of calculus* states that, if f is continuous on the closed interval $[a, b]$ and F is the indefinite integral of f on $[a, b]$, then

$$\int_a^b f(x)\, dx = F(b) - F(a) \tag{C.1}$$

Defn C.1.2 (Second Fundamental Theorem of Calculus)**.** The *second fundamental theorem of calculus* holds for f a continuous function on an open interval I and a any point in I, and states that if F is defined by

$$F(x) = \int_a^x f(t)\, dt,$$

then

$$\begin{aligned} \frac{d}{dx}\int_a^x f(t)\, dt &= f(x) \\ F'(x) &= f(x) \end{aligned} \tag{C.2}$$

Defn C.1.3 (argmax)**.** The arguments to the *argmax* function are to be maximized by using their derivatives. You must take the derivative of the function, find critical points, then determine if that critical point is a global maxima. This is denoted as

$$\operatorname*{argmax}_x$$

C.2 Rules of Calculus

C.2.1 Chain Rule

Defn C.2.1 (Chain Rule)**.** The *chain rule* is a way to differentiate a function that has 2 functions multiplied together.

If

$$f(x) = g(x) \cdot h(x)$$

then,

$$\begin{aligned} f'(x) &= g'(x) \cdot h(x) + g(x) \cdot h'(x) \\ \frac{df(x)}{dx} &= \frac{dg(x)}{dx} \cdot g(x) + g(x) \cdot \frac{dh(x)}{dx} \end{aligned} \tag{C.3}$$

D Laplace Transform

Defn D.0.1 (Laplace Transform)**.** The *Laplace transformation* operation is denoted as $\mathcal{L}\{x(t)\}$ and is defined as

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt \tag{D.1}$$