1 Introduction

A communication network is an interconnection of devices designed to carry information from various sources to their respective destinations. To execute this task of carrying information, a number of protocols (algorithms) have to be developed to convert the information to bits and transport these bits reliably over the network. The first part of this book deals with the development of mathematical models which will be used to design the protocols used by communication networks. To understand the scope of the book, it is useful first to understand the architecture of a communication network.

The sources (also called end hosts) that generate information (also called data) first convert the data into bits (0s and 1s) which are then collected into groups called packets. We will not discuss the process of converting data into packets in this book, but simply assume that the data are generated in the form of packets. Let us consider the problem of sending a stream of packets from a source S to destination D, and assume for the moment that there are no other entities (such as other sources or destinations or intermediate nodes) in the network. The source and destination must be connected by some communication medium, such as a coaxial cable, telephone wire, or optical fiber, or they have to communicate in a wireless fashion. In either case, we can imagine that S and D are connected by a communication link, although the link is virtual in the case of wireless communication. The protocols that ensure reliable transfer of data over such a single link are called the *link layer* protocols or simply the link layer. The link layer includes algorithms for converting groups of bits within a packet into waveforms that are appropriate for transmission over the communication medium, adding error correction to the bits to ensure that data are received reliably at the destination, and dividing the bits into groups called frames (which may be smaller or larger than packets) before converting them to waveforms for transmission. The process of converting groups of bits into waveforms is called modulation, and the process of recovering the original bits from the waveform is called demodulation. The protocols used for modulation, demodulation, and error correction are often grouped together and called the *physical layer* set of protocols. In this book, we assume that the physical layer and link layer protocols are given, and that they transfer data over a single link reliably.

Once the link layer has been designed, the next task is one of interconnecting links to form a network. To transfer data over a network, the entities in the network must be given addresses, and protocols must be designed to route packets from each source to their destination via intermediate nodes using the addresses of the destination and the intermediate nodes. This task is performed by a set of protocols called the *network layer*. In the Internet, the network layer is called the Internet Protocol (IP) layer. Note that the network layer protocols can be designed independently of the link layer, once we make the assumption that the link layer protocols have been designed to ensure reliable data transfer over each link. This concept of independence among the design of protocols at each layer is called

layering and is fundamental to the design of large communication networks. This allows engineers who develop protocols at one layer to abstract the functionalities of the protocols at other layers and concentrate on designing efficient protocols at just one layer.

Next, we assume that the network layer has been well designed and that it somehow generates routes for packets from each possible source to each possible destination in the network. Recall that the network is just an interconnection of links. Each link in the network has a limited capacity, i.e., the rate at which it can transfer data as measured in bits per second (bps). Since the communication network is composed of links, the sources producing data cannot send packets at arbitrarily high rates since the end-to-end data transfer rate between a source and its destination is limited by the capacities of the links on the route between the source and the destination. Further, when multiple source-destination (S-D) pairs transfer data over a network, the network capacity has to be shared by these S-D pairs. Thus, a set of protocols has to be designed to ensure fair sharing of resources between the various S-D pairs. The set of protocols that ensures such fair sharing of resources is called the *transport* layer. Transport layer protocols ensure that, most of the time, the total rate at which packets enter a link is less than or equal to the link capacity. However, occasionally the packet arrival rate at a link may exceed the link capacity since perfectly efficient transport layer protocol design is impossible in a large communication network. During such instances, packets may be dropped by a link and such packet losses will be detected by the destinations. The destinations then inform the sources of these packet losses, and the transport layer protocols may retransmit packets if necessary. Thus, in addition to fair resource sharing and congestion control functionalities, transport layer protocols may also have end-to-end (source-destination) error recovery functionalities as well.

The final set of protocols used to communicate information over a network is called the *application* layer. Application layer protocols are specific to applications that use the network. Examples of applications include file transfer, real-time video transmission, video or voice calls, stored-video transmission, fetching and displaying web pages, etc. The application layer calls upon transport protocols that are appropriate for their respective applications. For example, for interactive communication, occasional packet losses may be tolerated, whereas a file transfer requires that all packets reach the destination. Thus, the former may use a transport protocol that does not use retransmissions to guarantee reliable delivery of every packet to the destination, while the latter will use a transport protocol that ensures end-to-end reliable transmission of every packet.

In addition to the protocol layers mentioned above, in the case of wireless communications, signal propagation over one link may cause interference at another link. Thus, a special set of protocols called Medium Access Control (MAC) protocols are designed to arbitrate the contention between the links for access to the wireless medium. The MAC layer can be viewed as a sublayer of the link layer that further ensures reliable operation of the wireless "links" so that the network layer continues to see the links as reliable carriers of data. A schematic of the layered architecture of a communication network is provided in Figure 1.1. To ensure proper operation of a communication network, a packet generated by an application will not only contain data, but also contain other information called the *header*. The header may contain information such as the transport protocol to be used and the address of the destination for routing purposes.

The above description of the layered architecture of a communication network is an abstraction. In real communication networks, layering may not be as strict as defined

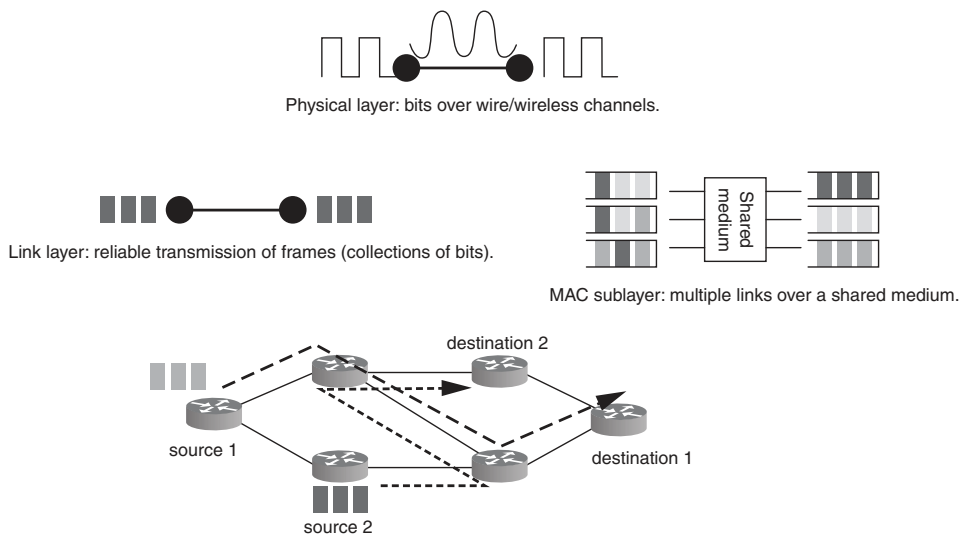

Figure 1.1 Schematic of the layered architecture of a communication network.

above. Some protocols may have functionalities that cut across more than one layer. Such cross-layer protocols may be designed for ease of implementation or to improve the efficiency of the communication network. Nevertheless, the abstraction of a layered architecture is useful conceptually, and in practice, for the design of communication networks.

Having described the layers of a communication network, we now discuss the scope of this book. In Part I, we are interested in the design of protocols for the transport, network, and MAC sublayers. We first develop a mathematical formulation of the problem of resource sharing in a large communication network accessed by many sources. We show how transport layer algorithms can be designed to solve this problem. We then drill deeper

into the communication network, and understand the operation of a single link and how temporary overload is handled at a link. Next, we discuss the problem of interconnecting links through a router in the Internet and the problem of contention resolution between multiple links in a wireless network. The algorithms that resolve contention in wireless links form the MAC sublayer. As we will see, the algorithms that are used to interconnect links within a wireline router share a lot of similarities with wireless MAC algorithms. We devote a separate chapter to network protocols, where we discuss the actual protocols used in the Internet and wireless networks, and relate them to the theory and algorithms developed in the earlier chapters. Part I concludes with an introduction to a particular set of application layer protocols called peer-to-peer networks. Traditional applications deliver data from a single source to a destination or a group of destinations. They simply use the lower layer protocols in a straightforward manner to perform their tasks. In Peer-to-Peer (P2P) networks, many users of the network (called peers) are interested in the same data, but do not necessarily download these data from a single destination. Instead, peers download different pieces of the data and share these pieces among themselves. This type of sharing of information make P2P systems interesting to study in their own right. Therefore, we devote a separate chapter to the design of these types of applications in Part I.

Part II is a collection of mathematical tools that can be used for performance analysis once a protocol or a set of protocols have been designed. The chapters in this part are not organized by functionalities within a communication network, but are organized by the commonality of the mathematical tools used. We will introduce the reader to tools from queueing theory, heavy-traffic methods, large deviations, and models of wireless networks where nodes are viewed as random points on a plane. Throughout, we will apply these mathematical tools to analyze the performance of various components of a communication network.

2 Mathematics of Internet architecture

In this chapter, we will develop a mathematical formulation of the problem of resource allocation in the Internet. A large communication network such as the Internet can be viewed as a collection of communication links shared by many sources. Congestion control algorithms are protocols that allocate the available network resources in a fair, distributed, and stable manner among the sources. In this chapter, we will introduce the network utility maximization formulation for resource allocation in the Internet, where each source is associated with a utility function $U_r(x_r)$, and x_r is the transmission rate allocated to source r. The goal of fair resource allocation is to maximize the net utility $\sum_r U_r(x_r)$ subject to resource constraints. We will derive distributed, congestion control algorithms that solve the network utility maximization problem. In a later chapter, we will discuss the relationship between the mathematical models developed in this chapter to transport layer protocols used in the Internet. Optimality and stability of the congestion control algorithms will be established using convex optimization and control theory. We will also introduce a game-theoretical view of network utility maximization and study the impact of strategic users on the efficiency of network utility maximization. Finally, routing and IP addressing will be discussed. The following key questions will be answered in this chapter.

- *What is fair resource allocation?*
- *How do we use convex optimization and duality to design distributed resource allocation algorithms to achieve a fair and stable resource allocation?*
- *What are the game-theoretic implications of fair resource allocation?*

2.1 Mathematical background: convex optimization

In this section, we present some basic results from convex optimization which we will find useful in the rest of the chapter. Often, the results will be presented without proofs, but some concepts will be illustrated with figures to provide an intuitive feel for the results.

2.1.1 Convex sets and convex functions

We first introduce the basic concepts from optimization theory, including the definitions of convex sets and convex functions.

Definition 2.1.1 (Convex set) A set $\mathcal{S} \subseteq \mathcal{R}^n$ is convex if $\alpha x + (1-\alpha)y \in \mathcal{S}$ whenever $x, y \in \mathcal{S}$ and $\alpha \in [0,1]$. Since $\alpha x + (1-\alpha)y$, for $\alpha \in [0,1]$, describes the line segment

between x and y, a convex set can be pictorially depicted as in Figure 2.1: given any two points $x, y \in \mathcal{S}$, the line segment between x and y lies entirely in $\mathcal{S}$. □

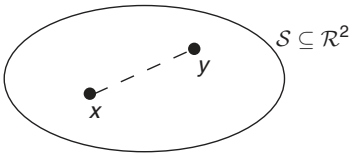

Figure 2.1 A convex set, $\mathcal{S} \subseteq \mathcal{R}^2$.

Definition 2.1.2 (Convex hull) The convex hull of set $\mathcal{S}$, denoted by $Co(\mathcal{S})$, is the smallest convex set that contains $\mathcal{S}$, and contains all convex combinations of points in $\mathcal{S}$, i.e.,

$$Co(\mathcal{S}) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \middle| x_i \in \mathcal{S}, \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1 \right\}.$$

See Figure 2.2 for an example. □

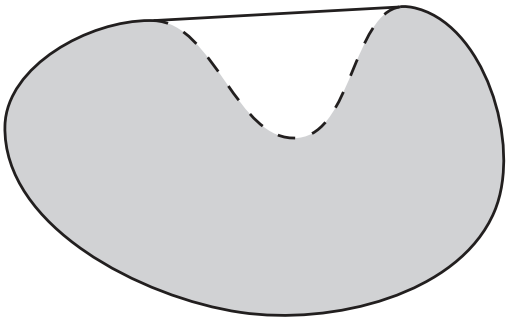

Figure 2.2 The solid line forms the boundary of the convex hull of the shaded set.

Definition 2.1.3 (Convex function) A function $f(x) : \mathcal{S} \subseteq \mathcal{R}^n \to \mathcal{R}$ is a convex function if $\mathcal{S}$ is a convex set and the following inequality holds for any $x, y \in \mathcal{S}$ and $\alpha \in [0, 1]$:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha) f(y);$$

$f(x)$ is strictly convex if the above inequality is strict for all $\alpha \in (0, 1)$ and $x \neq y$. Pictorially, $f(x)$ looks like a bowl, as shown in Figure 2.3. □

Definition 2.1.4 (Concave function) A function $f(x) : \mathcal{S} \subseteq \mathcal{R}^n \to \mathcal{R}$ is a concave function (strictly concave) if $-f$ is a convex (strictly convex) function. Pictorially, $f(x)$ looks like an inverted bowl, as shown in Figure 2.4. □

The line segment connecting the two points $(x, f(x))$ and $(y, f(y))$ lies "above" the plot of $f(x)$.

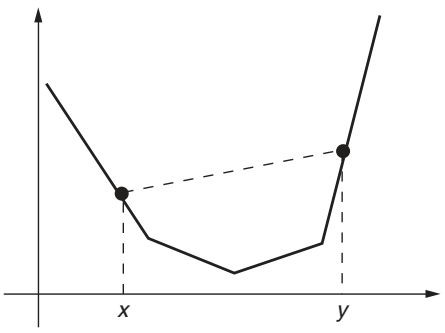

Figure 2.3 Pictorial description of a convex function in $\mathcal{R}^2$.

The line segment connecting the two points $(x, f(x))$ and $(y, f(y))$ lies "below" the plot of $f(x)$.

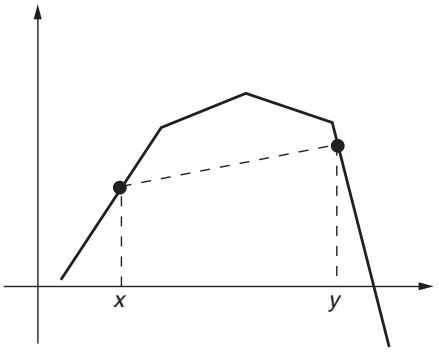

Figure 2.4 Pictorial description of a concave function in $\mathcal{R}^2$.

Definition 2.1.5 (Affine function) A function $f(x) : \mathcal{R}^n \to \mathcal{R}^m$ is an affine function if it is a sum of a linear function and a constant, i.e., there exist $\alpha \in \mathcal{R}^{m\times n}$ and $a \in \mathcal{R}^m$ such that

$$f(x) = \alpha x + a.$$ □

The convexity of a function may be hard to verify from the definition given above. Therefore, next we present several conditions that can be used to verify the convexity of a function. The proofs are omitted here, and can be found in most textbooks on convex analysis or convex optimization.

Result 2.1.1 (First-order condition I) Let $f : \mathcal{S} \subset \mathcal{R} \to \mathcal{R}$ be a function defined over a convex set $\mathcal{S}$. If f is differentiable and the derivative $f'(x)$ is non-decreasing (increasing) in $\mathcal{S}$, $f(x)$ is convex (strictly convex) over $\mathcal{S}$. □

Result 2.1.2 (First-order condition II) Let $f : \mathcal{S} \subset \mathcal{R}^n \to \mathcal{R}$ be a differentiable function defined over a convex set $\mathcal{S}$. Then f is a convex function if and only if

$$f(y) \geq f(x) + \nabla f(x)(y - x), \qquad \forall x, y \in \mathcal{S}, \tag{2.1}$$

where

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right)$$

and x_i is the ith component of vector x. Pictorially, if x is one-dimensional, this condition implies that the tangent of the function at any point lies below the function, as shown in Figure 2.5.

Note that $f(x)$ is strictly convex if the inequality above is strict for any $x \neq y$. □

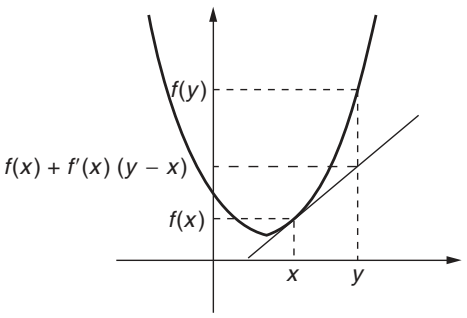

Figure 2.5 Pictorial description of inequality (2.1) in one-dimensional space.

Result 2.1.3 (Second-order condition) Let $f : \mathcal{S} \subset \mathcal{R}^n \to \mathcal{R}$ be a twice differentiable function defined over the convex set $\mathcal{S}$. Then, f is a convex (strictly convex) function if the Hessian matrix $\mathbf{H}$ with

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x)$$

is positive semidefinite (positive definite) over $\mathcal{S}$. □

Result 2.1.4 (Strict separation theorem) Let $\mathcal{S} \subset \mathcal{R}^n$ be a convex set and x be a point that is not contained in $\mathcal{S}$. Then there exists a vector $\beta \in \mathcal{R}^n$, $\beta \neq 0$, and constant $\delta > 0$ such that

$$\sum_{i=1}^{n} \beta_i y_i \leq \sum_{i=1}^{n} \beta_i x_i - \delta$$

holds for any $y \in \mathcal{S}$. □

2.1.2 Convex optimization

We first consider the following unconstrained optimization problem:

$$\max_{x\in\mathcal{S}} f(x), \tag{2.2}$$

and present some important results without proofs.

Definition 2.1.6 (Local maximizer and global maximizer) For any function $f(x)$ over $\mathcal{S} \subseteq \mathcal{R}^n$, x^* is said to be a *local* maximizer or *local* optimal point if there exists an $\epsilon > 0$ such that

$$f(x^* + \delta x) \leq f(x^*)$$

for δx such that $\|\delta x\| \leq \epsilon$ and $x + \delta x \in \mathcal{S}$, where $\|\cdot\|$ can be any norm; x^* is said to be a *global* maximizer or *global* optimal point if

$$f(x) \leq f(x^*)$$

for any $x \in \mathcal{S}$. When not specified, maximizer refers to global maximizer in this book. □

Result 2.1.5 If $f(x)$ is a continuous function over a compact set $\mathcal{S}$ (i.e., $\mathcal{S}$ is closed and bounded if $\mathcal{S} \subseteq \mathcal{R}^n$), then $f(x)$ achieves its maximum over this set, i.e., $\max_{x\in\mathcal{S}} f(x)$ exists. □

Result 2.1.6 If $f(x)$ is differentiable, then any local maximizer x^* in the interior of $\mathcal{S} \subseteq \mathcal{R}^n$ satisfies

$$\nabla f(x^*) = 0. \tag{2.3}$$

If $f(x)$ is a concave function over $\mathcal{S}$, condition (2.3) is also sufficient for x^* to be a local maximizer. □

Result 2.1.7 If $f(x)$ is concave, then a local maximizer is also a global maximizer. In general, multiple global maximizers may exist. If $f(x)$ is strictly concave, the global maximizer x^* is unique. □

Result 2.1.8 Results 2.1.6 and 2.1.7 hold for convex functions if the max in the optimization problem (2.2) is replaced by min, and maximizer is replaced by minimizer in Results 2.1.6 and 2.1.7. □

Result 2.1.9 If $f(x)$ is a differentiable function over set $\mathcal{S}$ and x^* is a maximizer of the function, then

$$\nabla f(x^*)dx \leq 0$$

for any feasible direction dx, where a non-zero vector dx is called a feasible direction if there exists α such that $x + adx \in \mathcal{S}$ for any $0 \leq a \leq \alpha$.

Further, if $f(x)$ is a concave function, then x^* is a maximizer if and only if

$$\nabla f(x^*)\delta x \leq 0$$

for any δx such that $x^* + \delta x \in \mathcal{S}$. □

Next, we consider an optimization problem with equality and inequality constraints as follows:

$$\max_{x\in\mathcal{S}} f(x), \tag{2.4}$$

subject to

$$h_i(x) \leq 0, i = 1, 2, ..., I, \tag{2.5}$$

$$g_j(x) = 0, j = 1, 2, ..., J. \tag{2.6}$$

A vector x is said to be *feasible* if $x \in \mathcal{S}$, $h_i(x) \leq 0$ for all i, and $g_j(x) = 0$ for all j. While (2.5) and (2.6) are inequality and equality constraints, respectively, the set $\mathcal{S}$ in the above problem captures any other constraints that are not in equality or inequality form.

A key concept that we will exploit later in the chapter is called Lagrangian duality. Duality refers to the fact that the above maximization problem, also called the *primal* problem, is closely related to an associated problem called the *dual* problem. Given the constrained optimization problem in (2.4)–(2.6), the *Lagrangian* of this optimization problem is defined to be

$$L(x, \lambda, \mu) = f(x) - \sum_{i=1}^{I} \lambda_i h_i(x) + \sum_{j=1}^{J} \mu_j g_j(x), \qquad \lambda_i \geq 0 \ \forall i.$$

The constants $\lambda_i \geq 0$ and μ_j are called Lagrange multipliers. The *Lagrangian dual function* is defined to be

$$D(\lambda, \mu) = \sup_{x\in\mathcal{S}} L(x, \lambda, \mu).$$

Let f^* be the maximum of the optimization problem (2.4), i.e., $f^* = \max_{x\in\mathcal{S}} f(x)$. Then, we have the following theorem.

Theorem 2.1.1 $D(\lambda, \mu)$ is a convex function and $D(\lambda, \mu) \geq f^*$.

Proof The convexity comes from a known fact that the pointwise supremum of affine functions is convex (see Figure 2.6). To prove the bound, note that $h_i(x) \leq 0$ and $g_j(x) = 0$ for any feasible x, so the following inequality holds for any feasible x:

$$L(x, \lambda, \mu) \geq f(x).$$

This inequality further implies that

$$\sup_{\substack{x\in\mathcal{S} \\ h(x)\leq 0 \\ g(x)=0}} L(x, \lambda, \mu) \geq \sup_{\substack{x\in\mathcal{S} \\ h(x)\leq 0 \\ g(x)=0}} f(x) = f^*.$$