1 Introduction

To state intuitively, the question investigated in this book is the following:

How does one communicate one or more source signals over a network from nodes (servers) that observe/supply the sources to a set of sink nodes (clients) to realize the best possible reconstruction of the signals at the clients?

The above question sets the unifying theme for the problems studied in this book, and, as will be made clear in this introduction, contains some of the most important and fundamental problems in information and network communication theory.

1.1 Network representation of source coding problems

Let's start with the observation that even the simplest source coding problems have (perhaps trivial) network representations. Figure 1.1, for instance, shows network representations of the three arguably most fundamental source coding problems. Here, the goal is to communicate a single source signal X, from a single server node s to one or more sinks. Each link of the network has a "capacity" assigned to it, which, when properly normalized, indicates the number of bits that can be communicated over that link, without errors, for every source symbol emitted from X.

Figure 1.1(a) is the simplest source coding problem. The receiver node t receives an R_1 bits encoding of X from the source node s. If X admits a rate-distortion function $D_X(R)$, the reconstruction error at t is at best $d_t = D_X(R_1)$. Of course if X is defined on a finite alphabet, then for large enough R_1, there is a possibility to communicate X losslessly, or without distortion (i.e., $d_t = 0$).

Figure 1.1(b), on the other hand, is the network representation of the progressive source coding problem, when $R_1 > R_2$, while Fig. 1.1(c) represents the network of two-description multiple-description source coding problem. In the latter case, sink node t_1 receives a description of X with rate R_1, while t_2 receives another description of rate R_2. Node t_3 receives both descriptions, and in the terminology of multiple-description coding (MDC), it acts as the *joint decoder*.

These examples correspond to the case of a single source signal X. Same is true for problems involving multiple source signals observed by multiple, usually non-communicating, encoders; problems often called distributed source coding. The problem becomes particularly interesting when the multiple signals in question are correlated. Figure 1.1(a) is the simplest setting of a distributed source coding problem in which two correlated sources X and Y are encoded at two separate (non-communicating) nodes s_1 and s_2, and are communicated to a single receiver node (or decoder), t.

This problem is called Slepian-Wolf distributed source coding [1] when the two sources are on finite alphabets and the goal is lossless communication of both of them to the sink t. Slepian and Wolf found necessary and sufficient conditions on the rates R_1 and R_2 for which lossless communication of both sources X and Y to t is feasible. In the special case where $R_2 = \infty$, i.e., the source Y is losslessly present at the decoder only, the problem of distributed source coding is usually called the Wyner-Ziv problem [69].

There is a large body of literature on theoretical and practical aspects of source coding problems with simple network representations as in Figs. 1.1 and 1.2. These include the complete characterization of achievable rate-distortion regions for many classes of important signals [2], [1] as well as powerful practical coding approaches that

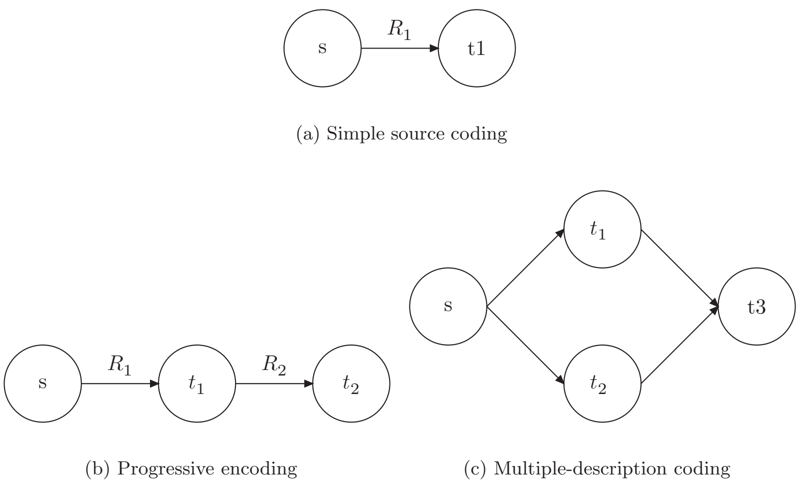

Figure 1.1 Network representation of fundamental source coding problems

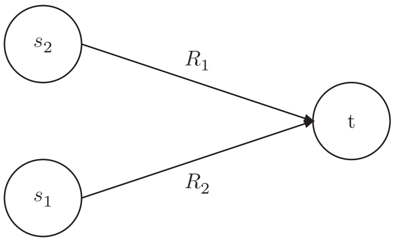

Figure 1.2 The network representation of Slepian-Wolf, Wyner, and Wyner-Ziv problems

perform close to these theoretical limits, some of which will be reviewed throughout this book.

1.2 Source coding and communication in networks with more complex topologies

For networks discussed so far, network structures are very simple. Since the communication links are assumed to be error-free, the network communication aspect of the source coding-and-networking problem is trivial. In particular, once the source or sources are encoded, the encodings are simply passed over the corresponding links. Recent advances in network information flow (e.g., network coding), however, suggest that there is much more to network communication than simple information relay.

This book intends to not only cover some of the above mentioned special cases of source coding in networks, but to go beyond, by exploring problems with networks of complex topologies. Figure 1.3 shows the two simplest network information flow scenarios. In both scenarios, there is a single source to communicate from *s* to one or more sink nodes, over an arbitrarily complex network. For these arbitrary networks, as schematically depicted in Fig. 1.3, a new dimension enters the problem – that of on-route information processing. As will become clear shortly, in most scenarios, source coding is accompanied by network coding and routing (an aspect that we call on-route processing) in a nontrivial fashion and, thus, in general, the two have to be considered jointly. In other words, optimal utilization of network resources requires a joint consideration of both source coding and on-route processing.

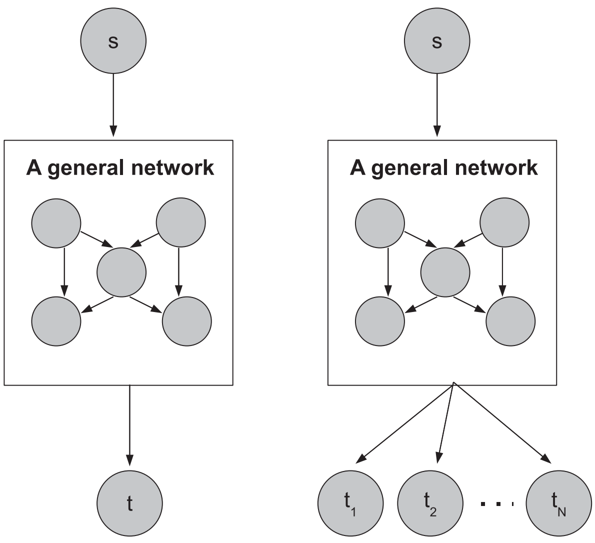

Figure 1.3 Unicast vs. multicast source communication

1.3 Separability of source coding and on-route processing

Before getting to more general scenarios, we first review some of the few cases in which source coding and on-route processing can in fact be separately performed without loss of optimality.

1.3.1 Lossless communication of a single source

If X is defined on a finite alphabet, and the goal is a lossless communication of X from s to all the sink nodes, then source coding and communication can be broken down into two steps without loss of optimality, (1) a source coding step and, (2) a network communication step. The network communication step, however, is fundamentally different in the unicast and multicast scenarios.

Suppose the entropy of the source signal X is $H(X)$. Then, in the unicast scenario, X can be communicated to the single source node t losslessly if and only if the max-flow from s to t is at least $H(X)$. Furthermore, it suffices to perform only routing (in fact simple relaying) in the network communication step. For the multicast scenario, on the other hand, the source X can be communicated from s to sinks, if and only if the max-flow from s to every sink node is at least $H(X)$. Furthermore, network communication, in general, requires re-encoding of information at relay nodes (i.e., network coding).

1.3.2 Single source communication to sinks with equal max-flow

Another special case in which source coding and network communication can be separated without loss of optimality is when a single source X has to be communicated from s to sink nodes with equal max-flow h. In that case, one can separately encode X to rate h and then use network coding (in the multicast scenario) or simple routing (in the unicast scenario) to communicate the source encoding to the sink nodes. If the source admits a rate-distortion function $D_X(R)$, the distortion $D_X(h)$ is achievable at all sink nodes. It is easy to verify that this is the smallest achievable distortion, given that the max-flow to each sink node is h. These results are reviewed in more detail in Chapter 2.

1.4 More general scenarios

It turns out that source coding and network communication are non-separable in most other scenarios, some important cases of which are reviewed next.

1.4.1 Distributed source coding in arbitrary networks

The Slepian-Wolf network settings in Fig. 1.2 can be generalized to the case of an arbitrary network, as in Fig. 1.4. If X and Y are on finite alphabets, necessary and sufficient conditions under which they can be losslessly communicated to the sink nodes $t_1, t_2, \ldots, t_n$ have been found recently [3].

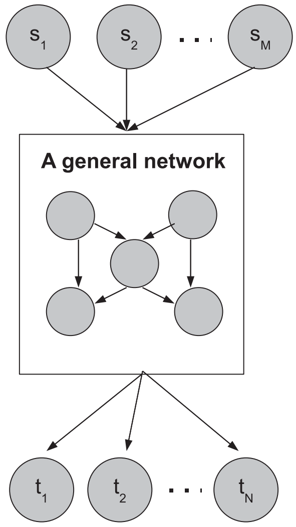

Figure 1.4 Slepian-Wolf coding in general networks

One important question discussed in [4] is whether distributed source coding can be separated from network communication for Slepian-Wolf problem in arbitrary networks. What happens if one encodes X to R_1 bits, and multicasts it from s_1 to all receivers and then encodes Y into R_2 bits and multicasts it to all sinks from s_2? It turns out that this separation strategy is suboptimal in general. More precisely, for many cases, it is impossible to losslessly communicate X and Y to all the sinks unless distributed source coding and network coding are done jointly. This warrants the study of a new class of codes, which can be called joint network source codes (JNSC).

1.4.2 Lossy communication of a single source in arbitrary networks

As we saw earlier, when there is only a single sink, or when the max-flow to all sink nodes is the same, source coding and on-route processing can be separated without loss of optimality.

However, the nature of the problem changes drastically when the set of sink nodes has heterogeneous flow properties (i.e., they don't have equal max-flows). In this case, a joint consideration of source coding and on-route processing is necessary even when only a single source is transmitted in the network. The scenarios in Fig. 1.5 illustrate how source coding and on-route processing (here in particular network coding) can become entangled in a complex way, as explained below.

Figure 1.5(a) illustrates the case where the source X has to be communicated only to the nodes 5, 6. The max-flow into both nodes 5 and 6 is 2. Thus, as stated before, one can optimally encode X into a source code stream of rate 2, break the stream into two sub streams a and b, each of rate 1, and communicate them to nodes 5 and 6. Note that network coding (i.e., bitwise XOR operation on streams a and b) is necessary at node 3. Methods and results in network coding are briefly reviewed in Part I of this book.

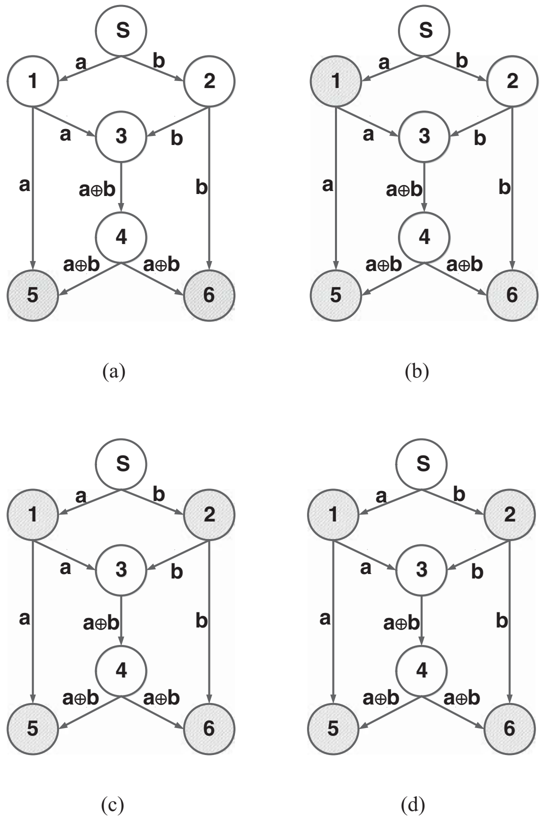

Figure 1.5 The Butterfly network for which all links have capacity one: When the sink nodes are, (a) 5 and 6, (b) 1, 5, and 6, (c) 1, 2, 5, and 6, (d) 4, 5, and 6

But, what if nodes 1, 5, and 6 constitute the set of sink nodes, as in Fig. 1.5(b)? In this case, the max-flow is 1 for node 1 and is 2 for nodes 5 and 6. The optimal strategy is now to progressively encode X into a stream of rate 2. Take the first portion of this stream to make another stream a of rate 1 and make the rest of the stream into another stream named b, again of rate 1. Network coding should be used to get both a and b to nodes 5 and 6 and stream a to node 1. Note that node 2 gets the stream b only. Since this is the second portion of a progressively encoded source code stream, node 2 will not be able to reconstruct X at all (but this is OK, since node 2 is not a sink node). This "layered" coding and communication strategy is reviewed in Chapter 9.

Figure 1.5(c) is yet another scenario, in which nodes 1, 2, 5, and 6 are sink nodes. In this case, the most general strategy is to encode X into two multiple-description source code streams each of rate 1 (name them streams a and b). Then use network coding to communicate both a and b to nodes 5, 6. Node 1 will receive only the description a and node 2 will only receive b. The use of multiple-description codes for efficient source multicast is reviewed in Chapters 5 and 6, while practical methods for designing multiple-description code streams are reviewed in Chapter 8.

As Fig. 1.5(a) to Fig. 1.5(c) suggest, sometimes, it is possible to break (without the loss of optimality) the task of source coding and on-route processing into a proper concatenation of well understood source coding operations (e.g., progressive coding or MDC), followed by network communication techniques (e.g., network coding or routing). This breakdown, however, cannot be done blindly. In other words, the choice of source encoding and network communication strategies should be made jointly.

In most other cases, however, full, joint consideration of source coding and on-route processing is required. When nodes 4, 5, and 6 are sinks, for instance, none of the above strategies is necessarily optimal (Fig. 1.5(d)).

The above examples suggest that a separate, formal treatment of the problem of source communication in networks is required. In this book, we will gather all these problems under the same umbrella, that of *Network-aware Source Coding and Communication* or NASCC. We will review a wide spectrum of new and old results related to different instances of the NASCC problem.

1.5 Applications and motivations

Our model of networks adopted in this text, in many ways, is an abstraction of computer networks, and is consistent with layered design of today's network protocols. The network is modeled as a graph of interconnected nodes that can communicate with rates constrained to the capacity of network links, i.e., the topological structure of the network is explicitly taken into account.

Our model is particularly relevant to the Internet at the router level and to overlay Peer-to-Peer (P2P) networks. As such, the immediate and by far the largest application domain of this research is real-time multimedia streaming over the Internet. But all the results of this book are valid in any other application area in which a data source (e.g., a physical measurement) has to be relayed to one or more receivers over an underlying network, an important instance of which is signal communication in sensor networks. For the clarity and concreteness of the presentations, this book will limit its discussions on applications to networked multimedia communications.

Real-time multimedia communication spans a wide range of applications, including digital TV and radio broadcasts over the Internet (e.g., IP-TV [5]), video conferencing, video on demand (VoD), distant education, telemedicine, voice over IP (VoIP), online computer games, virtual whiteboards, security and surveillance modules, and many others. Current estimates show that real-time multimedia traffic generated by real-time streaming and VoIP only, accounts for more than 21 percent of the overall Internet traffic in Europe [6], a share that is expected to increase dramatically with the advent of IP-TV technologies. Multicast applications are arguably the most resource-intensive multimedia applications on the web. Most radio stations as well as hundreds of TV channels now stream their live programs on the web. In February 2006, 148 million users listened to radio stations streamed through Shoutcast.com [7] alone.

The NASCC problem studied in this book investigates optimal utilization of bandwidth resources for multimedia multicast applications. On the theoretical side, the study of the NASCC problem, we believe, can fundamentally change our view of signal communication in networks, in much the same way as network coding (discussed in the next chapter) has changed our view of network information flow in the past few years.

1.6 Network-aware source coding and communication: a formal definition

The network model considered in this book is similar to now standard models in network information flow theory [8]. In particular, the model of a network is very close to one's intuition of a computer network: an interconnected set of nodes, each capable of processing and making decisions, which can reliably communicate over their connections provided that the capacities of all connection links are respected.

We are interested in designing a networked communication system to communicate a source signal from a set of source nodes (servers) to a set of sink nodes (clients), so that the source signal can be reconstructed with the best average quality at all the sink nodes. Unlike most frameworks of network information flow, the reconstruction of the source does not need to be perfect. In fact, for most real valued multimedia signals, perfect reconstruction is not necessary or even possible (the digitization process is already lossy in nature). Another major difference is that the quality of the source reconstruction and the input data used for such reconstruction do not need to be the same for all the sinks.

In the version of the problem discussed in this book, it is assumed that the source has been compressed off-line and has been deployed at the server nodes in advance. Again, this formulation is from a networked multimedia application point of view, where a multimedia content (e.g., a video clip) is encoded off-line and is deposited at one or more server nodes in the network before the communication starts. Such an assumption is of course not necessary when there is only a single server node in the network, a case that in fact includes some of the most interesting scenarios, such as live media streaming.

At the time of presentation, the content is *streamed* to one or more users. The communication capacity of the links limits the amount of information that can be communicated from node to node and hence the quality of the reconstruction of the source signal at the sink nodes. We call the problem of finding the strategy that maximizes the overall quality of the signal reconstruction at sink nodes, Network-aware Source Coding and Communication, and it is formally defined next.

Problem formulation:
Formally, the Network-aware Source Coding and Communication (NASCC) problem is defined by the following elements:

- A directed graph $G\langle V, E\rangle$ with nodes set V and edge set $E \subset V \times V$.
- A number of, possibly correlated, sources $X_1, X_2, \ldots, X_K$ over some common alphabet Γ. Of particular interest is the case where $\Gamma = \mathbb{R}^N$, for some N.

- A function $R = E \to \mathbb{R}^+$ that assigns a capacity $R(e)$ to each link $e \in E$. Bandwidths are normalized with the source bandwidth; therefore, $R(e)$ is expressed in units of bits per source symbol.
- $S_i, T_i \subseteq V$, for $i = 1, 2, \ldots, K$ that denote the set of server and sink nodes respectively for source X_i. We let $S = \cup_i S_i, T = \cup_i T_i$ denote the set of all server and sink nodes. The server nodes observe, encode, and communicate X_k in the network. Server nodes are not able to directly communicate (or collaborate) in the encoding process.

Throughout, we assume each source X admits a rate-distortion function $D_X(\cdot)$ under some family of distortion measures. For the most part, we also assume the source X is progressively refinable.

Nodes can communicate with neighbor nodes at a rate specified by the capacity of the corresponding link. $R(e)$, therefore, specifies that an average of $R(e)$ bits can be successfully communicated over link e per source symbol emitted from X.

The task is to communicate the source X_i from the server nodes in S_i and reconstruct X_i at the sink nodes in T_i. Throughout this book, $|\cdot|$ denotes the cardinality of a finite set. Distortion vectors $\mathbf{d}_k = (d_t, t \in T_k) \in \mathbb{R}^{|T_k|}$ for $K = 1, 2, \ldots, K$ are said to be simultaneously achievable if X_k can be reconstructed with an average distortion of d_t at sink node $t \in T_k$ by using a coding scheme that respects the capacity constraints on the links, i.e., the rate of information per source symbol communicated over e is no greater than $R(e)$.

Unlike classical point to point, or even multi-terminal information theory, it proves extremely hard to completely characterize the most general class of possible codes. Therefore, just as in [8], we need to leave the details of the code unspecified.

A number of considerations about the above formulation are due. For clarity, let's assume that there is only one source X, with one set of receivers T.

- A theoretically intriguing problem is how to characterize the set of all achievable distortion vectors $\mathbf{d} \in \mathbb{R}^{|T|}$. Note that this problem includes the usual lossless network coding problem as its special case, if the source alphabet Γ is finite.
- An equivalent problem, which is more relevant practically, is that of finding a coding scheme to minimize a weighted average distortion $\overline{d}(\mathbf{p}) = \sum_t p_t d_t$ for a weighting vector of Lagrangian multipliers $\mathbf{p} = (p_t; t \in T)$. This formulation is mostly considered in this book.

The remainder of this book is a systematic review of the known results and recent developments in dealing with the NASCC problem according to the taxonomy presented in the introduction.

1.7 Organization of this book

Finding the most general source coding-network communication strategy remains an open problem, with little hope for a solution. Even some of its simplest special cases

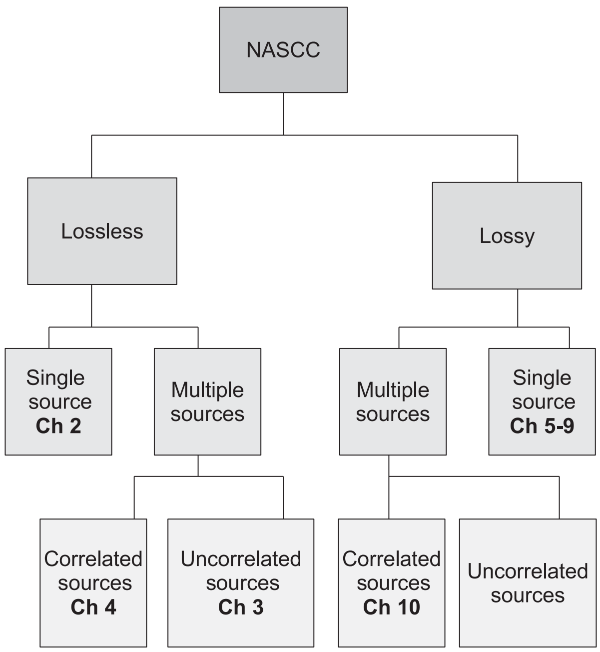

Figure 1.6 A taxonomy of source coding and communication problems in networks and their relation to the chapters of this book

(e.g., n-description source encoding) are known to be notoriously hard problems. In this book, however, we adopt a pragmatic approach. We investigate the solutions to this problem that use well-understood source coding techniques (e.g., progressive encoding, MDC, Slepian-Wolf, or Wyner-Ziv coding) along with optimized network communication strategies (e.g., optimized routing, relaying, or network coding), to come up with strong and *practical* solutions.

Figure 1.6 is a taxonomy of NASCC problems and their relationship to the chapters of this book. A brief description of the chapters is given below.

- Chapter 2: Part I of this book, which starts with Chapter 2, deals with lossless communication of sources in networks. Chapter 2 is concerned with the case with only one source node. We will consider scenarios where network coding is and is not allowed. There are several excellent tutorials and textbooks already available on network coding. As such, this chapter is intended to review recent results with an emphasis on algorithmic and complexity perspectives of designing optimal information delivery mechanisms.
- Chapters 3 and 4: In these chapters, we extend the discussions from Chapter 2 to the case of multiple sources. We consider the cases where information sources are correlated and uncorrelated, as well as unicast and multicast scenarios. Chapter 3 covers the case of independent source signals. Results and algorithms when network coding is and is not allowed are reviewed. Discussions will include Li and Li's conjectures