Many treatments of wireless communications focus on a wireless link as an isolated entity. Our concern here is with networks that support all multimedia services (including data, graphics, audio, images, and video) for tetherless (not physically wired to the network), nomadic (able to access the network from many locations), and mobile (accessing the network while moving) users. Such a network is termed an integrated-services multimedia network with wireless access. Wireless access is a key component of tetherless and mobile access in particular. Important components of such a multimedia network include a (typically broadband) backbone network, wireless access links to that backbone, terminals associated with each user (where some terminals are tetherless and others are not), and centralized data and computational servers. It is expected that an integrated-services multimedia network will serve a large and heterogeneous mix of applications. Overall, in this large and complex system, the fact that there is wireless access should have broad implications to all the system components, not just to the wireless access link. Conversely, design issues in the remainder of the system impact the wireless access link design. A primary objective of this chapter is to identify these cross-cutting issues. As such, the issues explored in this chapter are complementary to many of those discussed in the preceding chapters of this volume.
Most of our attention is focused on the signal processing technologies in a multimedia network, including compression, modulation, forward error-correction coding, and encryption, as well as limited attention to other elements that interact with signal processing (such as protocols). Additional compression and coding issues are explored in detail in Chapter 7. From a networking perspective, we define as signal processing those functions that modify or hide basic syntactical and semantic components of a bit stream payload, as opposed to those functions that are oblivious to the payload bits (such as protocols, routing, etc.).

In treating these issues, it is important to identify the objectives that are to be achieved. We can list those objectives relevant to this chapter as follows:

- For continuous-media services like audio and video, as well as graphics, the relevant “quality” criterion is subjective.
- As discussed further below, the most critical objective performance criteria are low interactive delay and high traffic capacity for wireless access links.
- Privacy by end-to-end encryption will be important for some users; it also is one aspect of intellectual property protection and authorized access control.
- Applications, terminal capabilities (from desktop to various flavors of portable terminals with different processing and resolution), and transport media (especially wireless and fiber backbones) will be heterogeneous. It is important to support both desktop and portable terminals with a common set of applications. It is also important that applications deploy seamlessly to the network without the application developer needing to deal explicitly with a diversity of transport and terminal capabilities.
- Not only point-to-point connections, but also point-to-multipoint and multipoint-to-point connections will exist [1, 2].
- Propagation characteristics will vary widely, depending on assumptions about carrier frequency, bandwidth, propagation characteristics (especially in-building vs. wide-area networks), terminal velocity, etc. This chapter is primarily focused on broadband in-building wireless networks, where we expect relatively slow terminal velocity and hence relatively slowly varying channel characteristics. However, most considerations discussed in this chapter apply to more general situations, and we will mention the impact of less ideal channel characteristics.

It is quite challenging to meet all these objectives simultaneously. To have any hope requires a carefully crafted architecture. One clear conclusion is that the wireless access link typically will be the limiting factor in achieving good subjective quality, as it is inherently unreliable and typically has limited bandwidth resources relative to backbone networks. Thus, any architectural constructs should first and foremost be aimed at achieving the best subjective quality as limited by the wire-
less access link, making compromises in other components (terminals, servers, and backbone network) as necessary. The wireless access link should not be considered just an "add on" to an existing backbone infrastructure, as is unfortunately the most common design philosophy today.

Another important consideration is complexity management [4]. The internet protocols have managed to contain complexity by partitioning most functionality within the terminals and keeping the internet layer relatively simple and state-free. As a result, a rich suite of applications has been deployed swiftly. In contrast, the public telephone network, with a relatively limited set of services based on 64 kb/s circuits and a centralized control model, is straining at the limits of complexity within the switching-node software. Because the central-control telephone model is not extensible to achieving the flexibility required in future multimedia networks, distributed "intelligent networking" approaches to control are being deployed [5] and even more sophisticated approaches are being considered [6, 7]. The internet model also becomes considerably more complicated when extended to continuous media (CM) services, due to the need for resource reservations and multicast connections [2]. Careful attention should be paid to complexity management from the outset.

In this chapter, we propose some architectural principles and discuss their implications to the constituent signal processing technologies listed above. These principles suggest many new requirements for the signal processing and present opportunities to signal processing researchers and developers for years to come. A more general perspective on the convergence of communications and computing as embodied in multimedia networking is available [8, 9], as is a treatise on some of the societal impacts [10].

6.1 Basic Considerations

In this section, we describe some of the constituent technologies and their relationship to multimedia networking.

6.1.1 High-Level Network Architecture

One group has made a proposal for an architecture for the future Global Information Infrastructure (GII), and for consistency, we draw upon their architecture [3]. As in [9], we use the terminology shown in Figure 6.1, which differs slightly from that in [3]. Applications draw upon the services layer (termed transport services in [3]), which calls upon the bitway layer (called bearer services in [3]). The bitway layer establishes

---

1Actually, [3] adds a fourth layer, middleware services, which we delete here because it is generally unrelated to signal processing functions of concern in this chapter.
connections between endpoints, carries data between the endpoints, and monitors its own performance. The services layer provides a set of common generic capabilities that are available to all applications; examples include reliable streams (supporting applications like file transfer), reliable transactions, electronic payments, directory services, and audio or video transport (for multimedia applications). One of the functionalities in the services layers is the conditioning of data for the bitway (for example, the compression of audio or video) and compensating for impairments in the bitway (for example, resequencing of packets, as in the Transport Control Protocol (TCP), or retransmission of lost packets, as in TCP, or resynchronization of audio and video streams, as in the MPEG-2 transport stream [11, 12]).

A concrete example of the functional groupings of these layers is shown in Figure 6.2 for a video application. The application presents a stream of raw
(uncompressed) video frames to the services layer, where the quality attributes
describing the video service to the application include resolution, frame rate, pixel
depth, compression fidelity, etc. The services layer does video compression, as well
as perhaps encryption, and presents to the bitway a compressed video stream (to
save bandwidth on the transport). The service is described to the bitway by its rate
attributes (average and peak rate), and the bitway to the service by its quality-of-
service (QoS) attributes including loss, corruption, and delay characteristics.

6.1.2 Signal Processing Functions and Constraints

In this section, we discuss briefly and qualitatively some of the interactions
between signal processing functions and the CM systems and network architecture
within which they are embedded.

Compression removes signal redundancy as well as signal components that are
subjectively unimportant, so as to increase the traffic-carrying capacity of transmis-
sion links within the bitway layer. Compression typically offers a trade-off between
a signal’s decoded fidelity and its transmitted bandwidth and often has the side ef-
effect of increasing the reliability requirements (loss and corruption) for an acceptable
subjective quality. Compression can be divided into two classes: signal semantics
based (such as conventional video and audio compression), and lossless, which
processes a bit stream without cognizance of the underlying signal semantics. The
compression typically has to make some assumptions about the bitway characteris-
tics, such as the relative importance of rate and reliability (see Section 6.1.5).

Encryption reversibly transforms one bit stream into another such that a rea-
sonable facsimile of the original bit stream is unavailable to a receiving terminal
without knowledge of appropriate keys [13]. Encryption is one component of a
conditional access system, with which a service provider can choose whether and
when any individual receiver can access the provided service, and is also useful in
ensuring privacy. It precludes any processing of a bit stream because it hides the
underlying syntactical and semantic components, except in a secure server that has
keys available to it. It also increases susceptibility to bit errors and synchronization
failures, as discussed in Section 6.3.1.

Forward error-correction coding (FEC) adds a controlled redundancy so that
transmission impairments such as packet loss or bit errors can be reversed. We dis-
tinguish binary FEC techniques from signal-space techniques. Binary FEC is applied
to a bit stream and produces a bit stream; examples include Reed-Solomon coding
and convolutional coding. Binary FEC has the virtue of flexibility, as it can be
applied on a network end-to-end basis in a manner transparent to the individual
links. FEC can be combined with other techniques, such as interleaving (to change
the temporal pattern of errors) and retransmission (to repeat lost or corrupted
information), if the temporal characteristics or error mechanisms are known. Signal-space coding, on the other hand, is tightly coupled with the modulation method (used to encode bits onto waveforms) and the physical characteristics of the medium and is often accompanied by soft decoding. Examples include lattice and trellis coding. It is usually custom tailored to the physical characteristics of each link and, as a result, can offer significantly higher performance. Wireless link modulation methods often incorporate power control and temporal and spatial diversity reception as well, as discussed in earlier chapters.

Figure 6.3 illustrates some fundamental syntactical constraints that we should keep in mind while designing a network architecture for CM services:

- Compression must precede encryption, and decryption must precede decompression. Encryption would hide basic statistical characteristics of an uncompressed audio or video signal, such as spatial and temporal correlations, that are heavily exploited by compression algorithms.

- Compression must precede FEC, and decompression must follow FEC since there is no point to “correcting” the benign and desired changes in a bit stream due to compression and decompression.

The relationship between encryption and error-correction coding is more complicated. Since encryption, like binary coding, transforms one bit stream into another, it can precede or follow binary error-correction coding. However, since a

![Diagram](image_url)

**Figure 6.3** Illustration of some fundamental syntactical constraints on signal processing functions.
signal-space code generates an output in the real-number field, it cannot precede
encryption and signal-space decoding cannot follow decryption. The purpose of
binary coding before encryption is to attempt to correct postdecryption errors. The
purpose of binary or signal-space coding after encryption is to prevent errors in the
transport of the encrypted bit stream, which will indirectly prevent postdecryption
errors.

6.1.3 Bitway Architecture

A bitway is the layer of a CM system responsible for transmitting data bits from
one place (and time, for storage applications) to another. As part of this task, the
bitway commonly carries out:

- Routing, establishment of a path from one communication endpoint to
  another;
- QoS establishment, the negotiation of service bit-rate characteristics and bit-
  way impairment characteristics;
- Resource reservation, the assignment of resources to a particular connection
to ensure compliance with the “QoS contract”; and
- Monitoring of network component status, source rate behavior, and QoS of
  active connections.

CM services commonly rely on a combination of several underlying trans-
mission sublinks. This is especially true for wireless-based services, e.g., paging
and cellular telephony. Heterogeneous sublinks certainly complicate the imple-
mentation of the bitway functions listed above. However, they also complicate the
design and configuration of the signal processing functions described in Section
6.1.2.

The best trade-off between signal fidelity and bandwidth appropriate for a
high-speed wired sublink may be very unreasonable for a wireless link. Fre-
quently, this motivates system designs that include transcoders within the net-
work, as with the IS-54 digital cellular system described in Section 6.1.5. Of course,
if decompression and recompression are performed at transcoders within the
network, there is a requirement for a secure server to perform decryption and
encryption.

However, it is with FEC that the existence of multiple heterogeneous subnet-
works complicates CM system design the most. It would be simplest to provision
a single end-to-end FEC system across a heterogeneous network; however, the effi-
ciencies of closely coupled error-correction coding, suitably designed modulation,
and specific subnet characteristics may be too significant to pass up in some cases. Section 6.1.5 and Section 6.4.1 discuss this further.

6.1.4 Corruption, Loss, and Delay Effects

Packet-based communications networks inevitably introduce three types of impairments. There is packet loss (failure to arrive), packet corruption (bit errors occurring within the payload), and packet delay. Packet loss can occur due to several mechanisms, such as bit errors in the header, or buffer overflow during periods of network congestion.

Data networks do not make a distinction between loss and corruption since a packet that is corrupted is useless and hence is discarded. CM services can tolerate some level of loss and corruption without undue subjective impairment, especially if appropriate masking is built into the signal decoders. This is fortunate, since absolute reliability such as afforded by data networks, requires retransmission mechanisms, which can introduce indeterminate delay, often excessive to interactive applications like telephony and video conferencing. Another distinct characteristic of CM services is that loss and corruption are different effects. Lost data must be masked, for example, in video by repeating information from a previous frame or in audio substituting a zero-level signal. Under some circumstances, it is possible to make good use of corrupted information, for example, by displaying it as if it were correct. The resulting subjective impairment may be less severe than if the corrupted data were discarded and masked.

Some CM compression standards, generally those presuming a reliable transport mechanism (such as MPEG video [14, 15, 16, 17]) discard corrupted data and attempt to mask the discarded information. Other standards—those designed for a very unreliable transport (such as the voice compression in digital cellular telephony [18] and video compression designed for multiple access wireless applications [19])—use corrupted data as if it were error free and minimize the subjective impact of the errors. An important research agenda for the future is audio and video coding algorithms that are robust to loss and corruption introduced by wireless networks, recognizing that these effects are more severe than in backbone networks.

CM services are real-time, meaning that they require transport-delay bounds. However, there is a wide variation in delay tolerance, depending on the application. For example, a video-on-demand application will be relatively tolerant of delay, whereas it is critical that transport delay be very small (on the order of 50 ms or so) for a multimedia editing or video conferencing application. Much recent attention is focused on achieving bounded delay through appropriate resource
reservation protocols [2, 20, 21]. Given this wide range of delay tolerance, it is clear that the highest traffic capacity can be obtained only by segmenting services by delay, coupled with delay-cognizant scheduling algorithms within the bitway statistical multiplexers.

Audio and video services are usually considered to be synchronous, implying that network transport jitter is removed by buffering before reconstruction of the audio or video. For the special case of voice, it is possible to change the temporal relationship of talkspurts somewhat without any noticeable effect, but video display is organized into periodic frames (at 24, 25, or 30 frames/s), and all information destined for a frame must arrive before it can be displayed. (We make an alternative proposal in Section 6.4.3.)

Packets arriving after some prescribed delay bound are usually considered to be lost, as if they did not arrive at all.\(^2\) This illustrates another important characteristic of CM services: the existence of stale information that will be discarded by the receiver if it does not arrive in timely fashion. As another example, the bitway may be working feverishly to deliver a pause-frame video when the motion suddenly resumes. Any state residing in the bitway relevant to the pause-frame will not be used at the receiver. The purging of stale information within the bitway layer will increase traffic capacity.

### 6.1.5 Joint Source/Channel Coding

Joint source/channel coding (JSCC) is a way to increase the traffic capacity of a network, subject to a subjective quality objective. While a classic “separation theorem” of Shannon states that it is possible to separate the source and channel coding without loss of performance, his result requires conditions (on channel memory and time variation) not usually satisfied on wireless channels [22, 23] and, further, takes no account of delay or complexity. In fact, substantial gains can be achieved in traffic capacity for a given subjective quality using JSCC on wireless channels, for three reasons:

- Since wireless channels typically do not satisfy the assumptions of the separation theorem, that theorem does not rule out greater performance through better source-channel coordination.
- For some interactive applications like video conferencing, we are particularly concerned about minimizing delay, which is outside the scope of the separation theorem.

\(^2\)We argue in Section 6.4.3 that this model for the reconstruction of video may not be the best in case of packet networks with substantial delay jitter.
For applications like audio, video, and graphics, the only meaningful criterion of quality is subjective. This implies that we must improve the system through experimentation; subjective quality falls outside the scope of the separation (or any other) theorem.

We can divide JSCC roughly into two classes: tightly coupled and loosely coupled. Tightly coupled JSCC, which predominates in the literature, designs the source coding, modulation, and channel coding jointly, assuming therefore that the channel coding and modulation are cognizant of the full details of the source coding, and vice versa [24, 25, 26, 27]. This approach is applicable when designing a stand-alone system, such as wireless transmission of high-definition television (HDTV) [27].

If JSCC is to be applied to an integrated services multimedia network, we have to deal with complications like the fact that a single source coder must be able to deal with a variety of transport links (broadband backbone and wireless in particular), the concatenation of heterogeneous transport links, and multicast connections with common source representations flowing over heterogeneous links in parallel. In this environment, it is appropriate to consider loosely coupled JSCC, which is the only variety we pursue in this chapter. Loosely coupled JSCC attempts to abstract those attributes of the source and the channel that are most relevant to the other and to make those attributes generic; that is, broadly applicable to all sources and channels and not tightly coupled to the specific type.

Loosely coupled JSCC is thus viewed differently from the perspective of the “source” and the “channel,” where channel is usually taken to mean a given physical-layer medium, but which we take here to mean the entire bitway network. From the perspective of the bitway, JSCC ideally adjusts the allocation of network resources (buffer space, bandwidth, power, etc.) to maximize the network traffic capacity subject to a subjective quality objective. From the perspective of the source, JSCC ideally processes the signal in such a way that bitway network impairments have minimal subjective effect, subject to maximizing the network’s traffic capacity. This suggests that the source coding must take account of how the bitway allocates resources, and the effect this allocation has on end-to-end impairments as well as on traffic capacity, and conversely the bitway needs to know the source coding strategy and the subjective impact of the bitway resource allocations. However, to embed such common knowledge would be a violation of the loosely coupled assumption, creating an unfortunate coupling of source and channel that precludes further evolution of each. Rather, we propose a model in which the source is abstracted in terms of its bitrate attributes only, and the bitway is abstracted in terms of its QoS attributes only. The benefits of JSCC can still be achieved with this limited knowledge, but only if the source and channel are
allowed to *negotiate* at session establishment. During negotiation, each source or channel is fully cognizant of its internal characteristics and can influence the other only through a give-and-take in establishing the rate and QoS attributes, taking into account some measure of cost. The substream architecture discussed in Section 6.2 will increase the effectiveness of loosely coupled JSCC.

A simple example of JSCC is compression [33]. The classical goal of compression is to minimize bit rate, which is intended to maximize the traffic capacity of the network without harming the subjective quality appreciably. However, minimizing the bit rate (say, in the average sense) is simplistic because traffic capacity typically depends on more than average bit rate. To cite several examples:

- The statistical multiplexing advantage in congestion-dominated subnets depends on the peakiness of the offered bit streams, at least for a constant loss and delay objective, and the manner in which the bit rate varies with time is an important factor in the traffic capacity.

- A side effect of compression, at least at a relatively constant subjective quality, is usually to generate a variable bit rate, and exploiting that variable bit rate through statistical multiplexing often results in packet losses under high traffic loads. This, in turn, causes subjective impairment.

- Compression normally results in an increase in the susceptibility to bit errors. On interference-dominated subnets, such as cellular radio wireless access links, it is expensive (in terms of traffic capacity) to provision consistently low error probability since doing so requires large transmitted power and hence increased interference to other users.\(^3\) Thus, the traffic capacity of such a subnet depends strongly on the reliability requirement, as well as the bit rate, and it is not automatically the case that minimizing the bit rate is equivalent to maximizing the traffic capacity [19, 29, 30, 31, 32].

Having stated our objective for JSCC in multimedia networks, let us now examine some current examples of tightly coupled JSCC and point out their shortcomings for an integrated-services network. Some systems effectively ignore the benefits of JSCC by focusing on a limited set of environments. Even standards such as MPEG targeted at widespread use commonly make specific limiting assumptions about the transport. The MPEG designers assume that uncorrected errors are infrequent enough that blocks of data with errors can be discarded and masked, with the resulting artifacts propagating until the next intraframe coded video frame. This results in error rate requirements on the order of \(10^{-9}\) to \(10^{-12}\)

\(^3\)For example, for wireless CDMA, the traffic capacity is related to the product of average bit rate and a monotonic function of bit error rate [28].
(depending on the application) [34, 35]. While this rate is feasible in storage, fiber, and broadcast wireless applications (such as terrestrial HDTV [33]), this is likely not feasible in multiple-access wireless applications.⁴ (Voice standards intended for multiple-access channels and mobile receivers with fading generally assume a worst-case error rate in the range of $10^{-2}$ to $10^{-3}$, which is more representative on these types of channels during deep fades [18].) MPEG illustrates the difficulty in designing compression standards with sufficient flexibility and scalability to accommodate a variety of transport scenarios.

MPEG-1 is limited not just to low-error-rate bitways but to low-delay-jitter bitways as well. Fortunately, this limitation was addressed during the design of MPEG-2. The MPEG-2 Real-Time Interface (RTI) permits system designers to choose the maximum delay jitter expected in their systems; given this value, the RTI specifies how decoders can handle the specified jitter. The generic nature of the RTI came about specifically because the MPEG-2 designers wanted to handle delay jitter in a variety of bitways: satellite, terrestrial, fiberoptic, cable, etc. This is an example of transport characteristics influencing compression design. See Section 6.4.2 for further discussion of MPEG.

The critical role of traffic capacity in wireless access subnets typically results in systems with intricate but inflexible schemes for JSCC, as can be illustrated by a couple of concrete recent examples. These examples also illustrate some of the pitfalls of the coupling of the CM service and the network, and they point to some opportunities to reduce this coupling.

One example is the IS-54 digital cellular telephony standard. This standard uses radio time-division multiple-access (TDMA) transport, which due to vehicular velocity is subject to rapid fading. Due to fading and also in an effort to increase traffic capacity by an aggressive cellular frequency reuse pattern, worst-case error rates on the order of $10^{-2}$ are tolerated (the error rate could be reduced at the expense of traffic capacity, of course). The speech is aggressively compressed, and as a result, the error susceptibility is increased, particularly for a subset of the bits. Therefore, the speech coder bits are divided into two groups, one of which is protected by a convolutional code and the other left unprotected. Interleaving is used to spread out errors (which are otherwise grouped at the demodulator output). What do we consider undesirable about this system design? At least a few things:

- The close coupling between the speech compression algorithm (which determines which bits require more FEC protection) and the transport (FEC and modulation) makes it impossible to modify one without the other. This situation will not be acceptable in a heterogeneous bitway.

⁴While FEC may be able to achieve such error rates, countering the worst-case error rate environment during deep fades will require very high levels of redundancy, which, because it is present even during favorable channel conditions, will severely restrict the traffic capacity [19].
Sec. 6.1  Basic Considerations

• The bitway (including FEC) is designed with a particular CM service in mind, namely, speech. It would be better if the transport were more flexible so that other services could easily be introduced.

• The transcoder (which converts from 64 kb/s speech to a compressed speech) introduces a substantial delay of about 80 ms. Two subscribers conversing via two digital cellular telephones encounter two tandem transcoders (where neither is necessary) and a round-trip delay on the order of 320 ms.

• The pulse code modulation (PCM) landline network connecting the cellular base station to another subscriber is carrying a much higher bit rate than necessary. While this bit rate does not affect the traffic capacity of the circuit-switched telephone network, a more flexible network would have a reduced traffic capacity (expressed in terms of simultaneous telephone calls) due to this mismatch of resources.

These issues are addressed in a more general context in Section 6.3.

A final example illustrates an architecture that begins to redress some of these problems. The Advanced Television Research Consortium (ATRC) proposal for terrestrial broadcast TV [36] attempts to separate the design of the video compression from the transport subsystem by defining an intermediate packet interface with fixed-length packets (cells). Above this interface is an adaptation layer that converts the video compression output byte stream into cells, and below this interface the cells are transported by error-correction coding and radio frequency (RF) modulation. Much more enlightening is the way in which a modicum of JSCC is achieved. First, the compression algorithm splits its output into two substreams, where, roughly speaking, the more subjectively important information is separated from the less subjectively important (and a reasonable rendition of the video can be obtained from the first substream).\(^5\) This separation is maintained across the packet interface and is thus visible to the bitway. The bitway transmits these two substreams via separate modulators on separate RF carriers, where the first substream is transmitted at a higher power level. The motivation for doing so illustrates another important role of JSCC on wireless access links; namely, achieving graceful degradation in quality as the transmission environment deteriorates. In this case, in the fringe reception area the quality will deteriorate because the second substream is received unreliably, but a useful picture, based on the first substream, is still available. This system illustrates some elements of an architecture that will be proposed later. Chapter 7 focuses on these and other aspects of JSCC and also proposes some specific JSCC structures.

\(^5\)Reference [36], and other work on packet video, use the term priority to distinguish between the substreams. We avoid that term here because it is usually applied to control the order of arrival or discard in congestion-dominated packet networks, and can be misleading when applied in more general contexts.
6.2 **Modularity of Services and Bitway Layers**

To allow different transmission media to work with the same source coding, and different source coders to work with different transmission media, it is especially important that we logically separate the design of source coders (in the services layer) from the transmission (in the bitway layer) as much as possible. (As discussed in Section 6.3, this separation is even more advantageous in heterogeneous transport environments.) This separation requires a careful partitioning of functionality between these layers and appropriate abstractions at their interface. This section concentrates on this partitioning and interface and describes a basic bitway model appropriate for multimedia services. See [31] for a description of the video compression problem in this heterogeneous environment.

### 6.2.1 Partitioning of Functionality

While [3] does not attempt a detailed partitioning of functions between services and bitway layers, we make a proposal here specifically with respect to signal processing functions, as shown in Figure 6.4. FEC has been placed in the bitway layer, and compression and encryption in the services layer, where we have termed the interface between these two layers the *medley gateway* [45]; the term *gateway* refers to the connection between layers, and *medley* refers to the heterogeneous sub-stream structure we envision at this gateway, as we discuss later.

Compression is inherently a "conditioning for transport" function and hence belongs in the services layer. We explicitly avoid compression, or transcoding (converting from one compression to another), within the bitway layer. The reasons for this are elaborated further in Section 6.3.

The reasons that we include encryption within the services layer are more subtle:

- Encryption must follow compression (and precede decompression) and hence cannot reside in the application layer.

<table>
<thead>
<tr>
<th>Services</th>
<th>Bitway</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medley gateway</td>
<td></td>
</tr>
<tr>
<td>Compression</td>
<td>Error-control coding</td>
</tr>
<tr>
<td>Encryption</td>
<td>Modulation</td>
</tr>
</tbody>
</table>

*Figure 6.4* Partitioning of signal-processing functions between the service and bitway layers.
- There may be two or more bitway layers in a given connection in a heterogeneous environment (Section 6.3). Including encryption in the services layer opens the possibility of doing encryption on an end-to-end basis, with resulting simplified key management and higher level of security. Encryption in the bitway layer could result in two or more encryption/decryption operations, with complications to key management and reduced security due to "in the clear" signals available at intermediate points. Also, encryption would not be under the control of the user, but rather the service provider, dramatically reducing the security from the perspective of the user.

- The proposed architecture eliminates the increased burden of error multiplication due to encryption (see Section 6.3.1) on the FEC algorithms since FEC decoding occurs prior to decryption.

The reasons we have placed FEC in the bitway layer include the following:

- The most unreliable transmission media, wireless, are also the most critical with respect to spectral efficiency. On such media, signal-space coding techniques (for example, trellis coding and multidimensional signal constellations [46]) are tightly integrated into the modulation system and hence are inherently localized to each transmission link in the connection.

- There are many error correction techniques available, such as retransmission, FEC, interleaving, etc. It is most efficient for these techniques to be tightly coupled to the transmission environment. For example, the temporal characteristics of wireless access links depend heavily on the level of mobility, and the level of interleaving (to counter error-correlation effects) and the coding techniques are best coordinated with that mobility.

- Achieving high traffic capacity on time-varying media (such as wireless channels in the presence of terminal motion) requires techniques that take account of the state of the channel, so that parameters such as FEC redundancy, transmit power, etc., are varied with time. This important class of techniques is practical to implement only within the bitway because of the close coupling to the physical layer and the need for low-latency feedback between modulation and coding and the physical layer.

- As discussed in Section 6.3.3.2, performing FEC on an end-to-end basis implies codes that deal with a variety of different loss and corruption mechanisms, such as packet loss due to congestion (erasure codes), independent errors, and correlated errors due to interference in wireless access links (interleaving). In practice, this implies that different codes would have to be concatenated to deal with every possible contingency, and the resulting multiple layers of redundancy would be carried by every link with a resultant traffic penalty.
• End-to-end FEC would require sufficient redundancy for the worst-case link, resulting in a rate penalty on links with less severe impairments. In the absence of adaptive configuration, the redundancy has to be adjusted for the global worst case.

• End-to-end acknowledgment and repetition protocols will generally impose too large a delay for critical interactive CM services like video conferencing.

While we propose that the primary responsibility for error correction fall to the bitway, there is no reason to dogmatically preclude the involvement of the service, as discussed further in Section 6.2.5. For example, in "best effort" data services without delay guarantees, services retransmission protocols (as in TCP) may be acceptable. As another example, a subset of the data in a CM service may require extraordinary reliability but be relatively insensitive to delay, as, for example, coder configuration and state information. In the latter case, relying on a reliable transport protocol may be a better solution than imposing a high reliability requirement on the bitway layer. More generally, experience has shown that:

• Turning a poor reliability channel into one with moderate reliability is best done within the physical layer and utilizing signal space or binary coding techniques with soft decoding.

• Turning a modest reliability channel into one with almost complete reliability is best done with acknowledgment and retransmission protocols. These protocols are best done on an end-to-end basis, rather than embedded into each link, because of the delay that they introduce.

Thus, the best approach depends on circumstances, but very high reliability streams will involve a combination of FEC in the bitway layer and retransmission in the services layer. This is yet another reason to place encryption in the services layer—so as to perform decryption on the most reliable representation of the bit stream and thus minimize error multiplication effects.

### 6.2.2 Abstracted View of the Bitway

To maintain flexibility and contain complexity, it is important that abstractions of both services and bitway be defined at the medley gateway. These abstractions should retain information that is relevant and critical, while hiding unnecessary details. One of our major goals is to separate, insofar as is possible, the design of the service from the bitway. Not only is this an important complexity management technique, but it is critical to our ability to deal with complex bitway entities such as concatenated heterogeneous links and multicast connections.
Since the bitway core function is to transport packets, the abstract view should focus on the fundamental packet transport impairments of corruption, loss, and delay. A basic model incorporating these three elements is shown schematically in Figure 6.5. Often, the service will be interested in the temporal properties of these impairments; that is, a characterization of whether impairments like losses, corruption, or excessive delays are likely to be bunched together, or if they are statistically spread out in time. This issue is discussed further later.

The description of the properties of the connection that the bitway provides to the service is called a flowspec [2]. The most relevant of these properties are:

- **Rate** attributes, such as average rate, peak rate, and a characterization of the temporal characteristics of the rate.
- **QoS** attributes, including loss, corruption, and delay, and the temporal characteristics of these impairments. Other QoS attributes also may be specified explicitly, for example, whether a connection guarantees to deliver packets in sequence.

Note what information is not included in the bitway model. We deliberately exclude knowledge of the detailed transmission and switching structure within the bitway. For example, we hide from the service any knowledge of whether loss and delay is caused by congestion or by FEC and interleaving techniques, etc. Similarly, knowledge of whether corruption is caused by thermal noise, or interference, or is affected by time-varying mechanisms like Ricean or Rayleigh fading, is omitted. This strategy places on the bitway modeling the burden of specifying fundamental impairments with sufficient detail that the transmission characteristics are sufficiently characterized for purposes of the service.

### 6.2.3 Abstracted View of the Service

In considering the abstraction of the service as seen by the bitway, a primary objective is to allow JSCC, in spite of our careful separation of the design of the two layers. To this end, we include in the services layer abstraction the substream structure...
shown in Figure 6.6. The stream of packets is logically divided into *substreams*, which are visible to the bitway. The integrity of substreams is maintained across multiple links (see Section 6.3). Each substream is associated with distinct QoS and rate attributes established by negotiation with the application. The QoS attributes are aggregated values from the individual links, so that each substream on each link has a potentially different QoS objective. Thus, within the bitway, each packet is identified as to its substream, which implicitly specifies the QoS objective for that packet. JSCC then takes a specific form: each source coder segments its packets according to QoS objectives and then associates that packet with the appropriate substream. The system is also cognizant of the traffic it has generated for each substream.

For example, the two-level priority schemes in video coding can be thought of as associating high-importance packets with one substream and low-importance packets with another substream. The higher-importance substream would have a QoS requirement associated with a lower loss probability than the lower-importance substream. The bitway can exploit the relaxed QoS requirement of the lower-importance substream to achieve a higher traffic capacity.

More generally, the service, knowing the QoS to be expected on the substreams, can associate packets with substreams in a way that results in acceptable subjective quality. The bitway, knowing the QoS expectations and rates, can allocate its internal resources, such as buffer capacity, power, etc., in a way that maximizes the traffic capacity. In the absence of the substream structure, the bitway would have to provide the tightest or most expensive QoS requirements to the entire stream in order to achieve the same overall subjective quality.

Fortunately, the substream model is consistent with the most important existing protocols. Substreams have been proposed in ST-II, the second-generation Internet Stream Protocol [47]. Version 6 of the Internet Protocol (IP) includes the concept of a *flow*, which is similar to our substream, by including a *flow label* in the packet header [48]. Asynchronous transfer mode (ATM) networks incorporate virtual circuits (VC), and associate QoS classifications with those VCs, where nothing precludes a single application from using multiple VCs. The notion of separating
packets into (usually two) priority classes is often proposed for video [49, 50, 51], usually with the view toward congestion networks. In particular, a two-level priority for video paired with different classes of service in the transmission has been proposed for broadcast HDTV [36]. We believe that substreams should be the universal paradigm for interconnection of services and bitways for a number of reasons elucidated below, and especially the support of wireless access and encryption. By attaching the name “medley gateway” to such an interface, we are not implying that a totally new gateway function is required. Rather, we propose this name as a common terminology applying to these disparate examples of a similar concept.

The distinction between a stream composed of a set of substreams and a set of streams with different QoS requirements is that a stream composed of substreams can have the rate and QoS descriptions of the substreams “linked together.” For example, a service could specify that the temporal rate characteristics of all of its substreams are highly correlated (or that two substreams’ rates are very negatively correlated). Also, a service could request “loss priorities” from a bitway by explicitly specifying that packets on one substream should not be discarded while packets on another substream are delivered successfully. Another example is a service that requests one substream be given a higher “delay priority” than another substream to ensure that packets on the first substream experience less delay than packets on the second.

Combining the bitway and service abstractions, the overall situation is illustrated in Figure 6.7. Each of a set of substreams receives different QoS attributes.

![Figure 6.7](image)

*Figure 6.7 The abstracted bitway for a set of substreams.*

A little thought confirms that correlations can be expected among substreams emanating from a single source. For example, in video, high- and low-motion information will typically have negatively correlated rates attributes.
and hence a quantitatively different bitway model. As discussed in Section 6.4.4, this bitway abstraction opens up some interesting new possibilities in the design of services.

### 6.2.4 Loosely Coupled Joint Source/Channel Coding

The abstractions introduced in the bitway model make opportunities in loosely coupled JSCC more transparent. The JSCC functionality is now divided between the services layer and the bitway layer. The bitway, in an effort to maximize its traffic-carrying capacity, does the following:

- Affords each packet a loss or corruption probability lower than required by the QoS specified for the substream with which it is associated.
- Takes maximum advantage of the delay flexibility afforded by the QoS on a per packet basis. This is a new opportunity in JSCC not anticipated in previous approaches and is discussed further in Section 6.4.3.

Simultaneously, the service attempts to maximize the subjective quality afforded to the application or user within the constraints of the agreed flowspec. For example, packets less sensitive to delay are associated with a substream with a relaxed delay specification.

In the absence of the substream structure, the bitway would have to provide the tightest or most expensive QoS requirements to the entire stream in order to achieve the same overall subjective quality (and the QoS needs of different packets may vary over several orders of magnitude, e.g., for MPEG video headers vs. chrominance coefficients). Thus, the bitway has the option of exploiting the substream structure to achieve more efficient resource use through JSCC. Critically, substreams are generic and not associated with any particular service (for example, audio or video or a specific audio or video coding standard).

The medley gateway model does impose one limitation on JSCC. It does not include a feedback mechanism by which information on the current conditions in the bitway layer can be fed back to affect the services layer. Nor does it allow the flowspec to be time-dependent. One can envision scenarios under which this "closed loop" feedback would be useful. One example is flow control, in which compression algorithms are adjusted to the current information-carrying capacity of a time-varying channel. Another is an adjustment of compression algorithms to the varying bit-error rate due to time-varying noise or interference effects. We do not include these capabilities because we question their practicality in the general situation outlined in Section 6.3, where the services layer implementation may be
geographically separated from the bitway entity in question, implying an unacceptably high delay in the feedback path. This does raise questions of how to deal with time-varying wireless channels. In this case, we do not preclude feedback within a bitway link, adapting various functions like FEC and power control in an attempt to maintain a fixed QoS.

6.2.5 Substream-Based Transport Protocols

Within the services layer, there is typically a transport protocol, the purpose of which is to serve as a “translation” between the characteristics of the bitway layer and the differentiated needs of the applications. An example in the Internet would be TCP, which adds, among other things, retransmission and acknowledgment to ensure reliable and in-sequence delivery of packets for data applications. TCP adds significant delay and hence may not be appropriate for critical interactive CM services, especially those that do not require reliable delivery, as discussed in Section 6.1.4. The question then arises, what is the appropriate transport protocol? Since the transport protocol by definition impacts the QoS as seen by the application, of course constrained by the QoS provisioned by the bitway, any consideration of QoS and JSCC must incorporate the transport protocols.

The multiple substream model of the medley gateway has several characteristics that may particularly require a transport protocol:

- Each substream is simply a packet delivery mechanism. There are configured QoS attributes relative to packet loss, corruption, and delay, but no guarantee that a particular packet is actually delivered, nor any guarantee that packets are delivered in the same order in which they were transmitted.
- The substreams are asynchronous at the receiver, implying that there is no predictable temporal relationship between their delivery. This asynchrony is quite deliberate, since the substreams may have different delay QoS attributes.

Should the application desire more control, for example, guaranteed packet delivery, guaranteed order of delivery, or synchronization of the substreams at the receiver, an appropriate transport protocol can be invoked. A general architecture for such a protocol in the context of a medley bitway is shown in Figure 6.8. The medley transport protocol presents a service with \(N\) substreams to the application and makes use of \(M\) medley bitway substreams. While it would be likely that \(M = N\), that is not necessarily the case, as will be illustrated by a concrete example in Section 6.4.4. The general purpose of the transport protocol is to modify the semantics of the bitway so as to ensure ordered delivery or synchronization among
substreams. The transport protocol may require a feedback stream not shown, for example, carrying acknowledgments or requests for retransmission. Generally, QoS attributes such as reliability and delay will be substantially affected by the transport protocol. For example, ordered delivery will add delay since it will be necessary to buffer packets arriving before one or more of their predecessors, and synchronization of substreams will make all substreams suffer the worst-case bitway delay.

Thus far, to our knowledge there have been no proposals for substream-based transport protocols, although of course an existing transport protocol such as User Datagram Protocol (UDP) could be used independently on each substream. Medley transport protocols should be a profitable area for research.

6.2.6 Scalability and Configurability Issues

Requiring services and bitway to be mixed and matched arbitrarily puts a much greater burden on each. A service entity that is designed to utilize any bitway entity must exhibit scalability to deal, for example, with both a broadband backbone bitway and a wireless access bitway. Similarly, the bitway must be prepared to allocate its resources differently for different rate attributes and QoS requirements, for example, to provision both an audio and a video service.

In the loosely coupled JSCC model, we envision a connection establishment flowspec negotiation between service source and sink and bitway. These three entities can iterate through multiple sets of flowspec attributes to find a set that balances service performance and connection cost goals well. For example:

- The service entity, based on subjective quality criteria requested by the application, requests a flowspec of the bitway. However, since the bitway can conceivably be anything between a broadband backbone and a wireless access link, this request may be wildly unrealistic or too expensive.
The bitway entity determines the feasibility of the flowspec, and if feasible, passes back to the service a cost\textsuperscript{7} associated with that flowspec.

- The service and the bitway exchange sets of flowspecs, choosing flowspecs that improve service performance or reduce cost. This process results in a final agreed-to flowspec.

- Both the service and the bitway configure themselves. This action implies appropriate resource allocation by the bitway to guarantee that the agreed flowspec will be achieved. This also implies that the service chooses signal processing operations and a substream decomposition to conform to the rate attributes in the flowspec to maximize subjective quality subject to the agreed-to flowspec.

During the negotiation, the bitway entity must aggregate QoS impairments and costs for all sublinks in a connection. Suitable modeling of these impairments, their costs, and their aggregation will be a big challenge.

Unfortunately, an establishment negotiation in this form is not advisable for multicast connections because it is not scalable and is likely to be overly complex. The service source would have to negotiate with an unknown number of service sinks and associated bitway entities—potentially, thousands. Further, sinks will typically be joining and leaving the multicast connection during the session, and it is not reasonable to expect that the source will reconfigure (e.g., new compression algorithm or substream decomposition) on each of these events, especially if doing so requires all other sinks to reconfigure as well. Mobility of receiving terminals raises similar issues.

To avoid this problem, we can envision a different form of configuration for multicast groups, with some likely compromise in performance, inspired by the multicast backbone (MBone) \cite{52} and the resource reservation protocol (RSVP) \cite{2}. The service source generates a substream decomposition that is designed to support a variety of bitway scenarios, unfortunately without knowing in advance their details. It also indicates to the bitway (and potential service sinks) information as to the trade-offs between QoS and subjective quality for each substream. Each new sink joining the multicast group subscribes to this static set of substreams, based on resources and subjective quality objectives, and this subscription would be propagated to the nearest feasible splitting point. The QoS up to this splitting point would be predetermined, but possibly configurable downstream to the new sink. The resulting compromise—the bitway QoS to each new sink would

\textsuperscript{7}In a commercial context, cost is likely to be expressed in monetary terms, or in other contexts, it may be expressed in other terms. In any case, an important component of the cost will be the traffic capacity implications of the requested flowspec.
be constrained by the QoS to the splitting point established by other sinks—could be mitigated by allowing a sink to request the addition of bitway resources upstream from the splitting point.

For wireless links, the ability to configure QoS depends on assumptions about the propagation environment and terminal speed. For well-controlled, indoor, wireless local-area networks, it may be relatively easy to configure reproducible QoS attributes because low terminal speeds will result in a slowly varying propagation condition due to fading. In that case, the media-access layer may be able to adaptively maintain a reasonably constant QoS over time. In contrast, in wide-area wireless networks with high terminal velocities and high carrier frequencies, fading and shadowing effects may make it extremely difficult to adaptively maintain QoS. In this case, it may be more appropriate to view the configured QoS as an objective rather than as a guarantee and to assume that there is an outage probability (possibly configurable but at least provided to the application); that is, probability that the QoS objective is violated. Many intermediate situations are surely possible.

### 6.3 Edge vs. Link Architecture for Service Layer

In Section 6.2, we addressed the problem of separating the designs of the service from the bitway while leaving open most possibilities for JS
c. Our motivation was to allow the flexibility to substitute freely the service or bitway realizations. In this section, we consider a related set of issues in the provision of CM services through two or more heterogeneous subnets. Many of the issues addressed in Section 6.2 become more important.

Consider two basic architectures, illustrated in Figure 6.9, for concatenated links, where each link corresponds to one homogeneous bitway subnet. For example, in wireless access to a broadband network, the wireless subnet would constitute one bitway link, and the broadband subnet would constitute the second link. The distinction between the link architecture and the edge architecture is whether or not a services layer is included within each subnet. The back-to-back services layers in the link architecture include, for CM services like audio and video, a decompression signal processing function followed by a compression function. These functions together constitute a transcoder.

A transcoder is functionally equivalent to introducing an analog link in the network by converting from one compressed digital representation to analog (by decompressing and D/A converting) and then converting from analog to a different compression standard (by A/D converting with a synchronous sampling clock and compressing). This virtual analog link circumvents many interoperability

---

8The term edge denotes the entry point to the first bitway link in the network.
issues, such as ensuring an allocation of the same bit rate on all network links. In some situations, such as introducing new technology into a legacy system, transcoding may be unavoidable. For example, in the telephone network, in a call from a wired to a digital cellular telephone, one voice coding technique (8 kHz sampled PCM) is used on the wired network and another vector-sum-excited linear prediction (VSELP) coding, in the case of the North American IS-54 standard) is used on the digital cellular subnet [18]. This approach is for valid and important technical reasons; namely, the desire for spectral efficiency on the digital cellular subnet, resulting in more aggressive compression (traded off against implementation cost and reduced subjective quality) and the need for JSCE between speech coder and wireless link.

In the Internet, services layers like TCP or UDP are realized at the edges. That is, the Internet uses today the edge architecture. In extensions to the Internet architecture for realizing CM services, under some limited circumstances transcoders are proposed to be included within the network<sup>5</sup>; thus, the Internet is currently proposed to move (at least to a minor extent) in the direction of the link architecture.

In designing a new infrastructure, it should be possible to avoid transcoders, and we believe very desirable as well. We argue that the edge architecture is superior and should be adopted for the future. The resulting architecture is structured as in Figure 6.10. Compression, encryption, and QoS negotiation occur in the services layer, at the network edge. FEC, modulation, and resource reservation occur in the

<sup>5</sup>Specifically, in multicast CM services, bridges incorporating transcoder functionality are allowed at the nodes of the multicast spanning tree as a method of accommodating heterogeneous downstream terminals [53]. We later propose an alternative method to solve this problem.
Figure 6.10  A proposed architecture including compression, encryption, and error-correction encoding. Encryption is performed independently on each sub-stream so that the QoS after decryption can be controlled within the bitway.

bitway layer, at each network link. In favor of this architecture, we mention five factors:

- **Privacy and security.** The link architecture is incapable of providing privacy by end-to-end encryption under user control since an encrypted signal cannot be transcoded. The best that can done is encryption on a link basis by the service provider(s), with no ability for the user to verify that encryption has been performed. In our opinion, this problem alone should preclude serious consideration of the link architecture.\(^{10}\)

- **Openness to change.** The edge architecture is open to substitution of different services layers at the network edge (user terminal or access point). This flexibility leads to an economically viable method to upgrade services over time, as well as to introduce new ones, as discussed in Section 6.3.2.

- **Performance.** The link architecture suffers from the accumulation of delay and subjective impairment through tandem compressions and decompressions of the CM signal. This problem has already become serious in digital cellular telephony, where each transcoder introduces on the order of 80 ms of delay. This delay is inherent to the compression in transcoding since compression

\(^{10}\)This issue is discussed in [54], where it is pointed out that end-to-end encryption alone allows routing information to be intercepted internal to the network. It is argued that a combination of both end-to-end and link-by-link encryption is the most secure option.
is at its heart a time-averaging process. In more complicated heterogeneous scenarios, delay could become unacceptable for delay-sensitive interactive applications.

- **Complexity.** The edge architecture has a number of challenges, as discussed later, but overall we believe it substantially reduces the complexity of establishment and configuration.

- **Mobility.** The link architecture embeds considerably more state within the network associated with the realization of a CM service, creating additional requirements for migration of state when terminals are mobile (requiring the movement or the disestablishment/establishment of multipoint connection spanning trees).

The impairment accumulation and mobility considerations are relatively straightforward; the following subsections discuss the other factors.

### 6.3.1 Privacy and Security

Encryption is an important requirement for privacy and for preventing unauthorized interception in intellectual property protection schemes. Of course, encryption is accompanied by a host of other issues, such as key management and distribution, that are beyond the scope of this chapter. Not all services will require encryption, but the network architecture has to accommodate it for those cases where it is required. One issue with encryption is whether it is applied end-to-end or only on selected links of the network (especially the wireless link). End-to-end encryption affords much greater protection to the user than does link-by-link encryption because keys are known only to the user. Since encryption deliberately hides the syntactical and semantic components of the signal, no compression can be incorporated into the network where streams may be encrypted, including the conversion from one compression standard to another.

Encryption techniques can be divided into two classes [13]. In the binary additive stream cipher, which is used, for example, to encrypt the speech signal in the Groupe Speciale Mobile (GSM) digital cellular system [41], the data is exclusive-or’ed with the same random-looking running key generator (RKG) bit sequence at the transmitter and receiver. The RKG depends on a secret key known to both encryption and decryption [42]. The stream cipher has the advantage of no error multiplication and propagation effects; however, the loss of synchronization of the RKG will be catastrophic. A block cipher algorithm applies a functional transformation to a block of data plus a secret key to yield the encrypted block, and an inverse function at the receiver can recover the data if the key is available. For example, the Data Encryption Standard (DES) applies its transformation to blocks of 64 bits by
means of a 56-bit key [43]. In fact, error propagation within the block is considered a desirable property of the cryptosystem; that is, block ciphers should on average modify an unpredictable half of the plaintext bits whenever a single ciphertext bit is changed (this behavior is called the "strict avalanche property" [44]). There are variations on block ciphers with feedforward and feedback of delayed ciphertext blocks that cause error propagation beyond a single block.

Another important issue is the impact of encryption on QoS. In a general, integrated-services multimedia network, encryption techniques with error propagation should not be used for CM services since this use will preclude strategies designed to tolerate errors rather than mask them.

Neither a stream nor block cipher is ideal: the stream cipher introduces serious synchronization issues in a packet network, while the block cipher has severe error propagation. This is a serious issue for wireless multimedia networks that should be addressed by additional research.

### 6.3.2 Openness to Change

The history of signal processing operations like compression is one of relentless improvement in performance parameters like compression ratio, subjective quality, and delay. Algorithm improvements are usually accompanied by increasing processing requirements, but fortuitously the cost/performance of electronics also advances relentlessly. It would, given this history, be unfortunate to "freeze" existing performance attributes through an architecture that discourages or precludes change.

In this regard, the argument in favor of the edge architecture is economic: it allows the latest technologies to be introduced into the network in an economically viable way. New signal processing technologies are initially more expensive than older technologies since innovation and engineering costs must be recovered and because such technologies usually require more processing power. In the edge architecture, the services signal processing is realized within the user terminal or at a user access point; that is, it is provisioned specifically for the user. Only users who are willing to pay the cost penalty of the latest technology need upgrade, and only services desired for that user need be provisioned.

In contrast, in the link architecture, service signal processing elements are embedded widely throughout the network. At each point, it is necessary to deploy all services, including the latest and highest performance. The practical result is that for *any* users to benefit from a new technology, a global upgrade throughout the network is required. If only a relatively few users are initially willing to pay the incremental cost of new technology, there is no business case for this upgrade. There is also the question of who provisions and pays for the substantial infrastructure that would be required to support transcoding in or near base stations.
Further, the link architecture also requires that, for \( N \) different performance flavors of a given service, internal nodes in the network be prepared to implement \( N(N - 1) \) distinct transcenders, and that nodes be prepared to implement all distinct services. These nodes must also implement all feasible encryption algorithms and must be cognizant of encryption keys. In contrast, in the edge architecture, the edge nodes need implement only those services desired by the local application/user and only the flavor with the highest desired performance (as well as fallback to lower-performance flavors).

Past examples of these phenomena are easy to identify. The voiceband data modem, realized on an end-to-end basis, has advanced through two orders of magnitude in performance while simultaneously coming down in price. Users desiring state-of-the-art performance must pay a cost increment, but other users need not upgrade. If a higher-performance modem encounters a less capable modem, it falls back to that mode. Realizing the older modem standards introduces only a tiny cost increment since the design costs have been amortized and the lower performance standard requires less processing power. This example provides a useful model of how a service can be incrementally upgraded over time in the edge architecture. It illustrates that each terminal does not have to implement a full suite of standards, but rather needs to include only those services and the highest performance desired by the local application or user, as well as fallback modes to all lower speed standards.\(^ {11} \) The fallback modes, which are the only concession to interoperability with other terminals, do not add appreciable cost—the lower performance standards require less processing power, and the design costs of the older standards have been previously amortized.\(^ {12} \) The total end-to-end performance will be dictated by the lowest performance at the edges.

Contrast this behavior with the circuit-switched telephone network, where the same voice coding has been entrenched since the dawn of digital transmission. This voice coding standard is heavily embedded in the network, which was originally envisioned as a voice network. Today, it would be feasible to provide a much-improved voice quality (especially in terms of bandwidth) at the same bit rate, but there is no economically viable way to introduce this technology into the network.

The ability for users or third-party vendors to add new or improved services, even without the involvement of the network provider, is perceived as one of the key features of the Internet, leading to the rapid deployment of new capabilities such as the World Wide Web (WWW). In the link architecture, the necessary involvement of network service providers in services is undoubtedly a major barrier to innovation within the services domain of functionality, such as signal compression.

\(^ {11} \)This style of progressive improvement in a standard is already evident in MPEG, where MPEG-2 decoders are required to also be MPEG-1 compliant. Numerous examples of this methodology exist in other domains, such as microprocessor architectures.

\(^ {12} \)This argument is valid for software-defined standards, and is valid today in audio applications and will be increasingly valid in video as well.
6.3.3 Performance and Efficiency

In this section we examine the relative performance and efficiency of the architectures discussed above.

6.3.3.1 Modulation

Packet loss, corruption, and delay are especially problematic in wireless communication, which is limited by low bandwidth, time-varying multipath fading and interference. Moving to higher radio frequencies may alleviate spectrum congestion, but this solution is attended by a host of other difficulties, including susceptibility to atmospheric attenuation from fog and rain. Thus, the application of physical layer signal processing to combat impairments is more important in a wireless context than in a wireline backbone network.

We can distinguish between two categories of physical layer signal processing. As discussed in earlier chapters, transmit waveform shaping, spatial and temporal diversity-combining, and equalization are commonly employed wireless physical layer techniques that strive to unilaterally improve the reliability of all information bits. These methods trade off reliability for signal processing overhead (hardware cost), delay, and reduced traffic capacity. In contrast, power control and signal-space codes (such as trellis-coded modulation and shell mapping) form a class of methods with an additional dimension: given a fixed amount of resources—transmit power in the case of the former, hardware complexity and radio spectrum in the case of the latter—these strategies can selectively allocate impairments to different information bits, thereby controlling QoS. The ability to match transmit power to loss and corruption requirements is essential for maximizing capacity in wireless cellular networks, where excessive power creates unnecessary interference to other users.

The substream abstraction (Section 6.2.3) enables this matching. As shown in Figure 6.11, each bitway link is obligated to maintain the structure of the medley gateway at its output. That is, the medley gateway is the interface between service

![Figure 6.11](image-url) Each bitway link maintains the structural integrity of the medley gateway, making the structure available to downstream bitway links.
and bitway layers and also the interface between distinct bitway entities. This is why we call it a gateway—since it serves as a common protocol interface between heterogeneous bitway subnets. The substream structure is visible to each bitway link, which is able to allocate resources and to tailor its modulation efficiently in accordance with JSCC.

6.3.3.2 Forward Error-Correcting Coding

With end-to-end FEC, the transport may provision reliability by applying binary FEC on an end-to-end basis. The FEC-encoded information bitstream may then transparently pass through multiple transport links to be FEC decoded (again at the network edge) by the sink. Priority encoding transmission (PET) [37] is an example of the end-to-end FEC approach. The goal of PET is to provide reliable transmission of compressed video over wired packet networks. The primary error mechanism in these networks is congestion, leading to excessively delayed packets or buffer overflow. PET combats congestion losses by using a form of binary FEC known as erasure coding: $B$ packets are encoded into $N$ packets, such that all $B$ packets can be recovered from any $B$ out of $N$ packets successfully received. This approach is appealing for its simplicity and, in fact, can be efficient for a homogeneous wired transport whose links have very similar characteristics.

An alternative architecture is to provision reliability by applying physical layer signal processing on a link-by-link basis: each link is made cognizant of the loss and corruption requirements of an application, then applies its own specific physical layer processing to meet these requirements. This is the architecture we prefer, for reasons we now elaborate.

The link-by-link approach to providing reliability necessitates a mechanism for QoS negotiation on each link so that it can configure itself in accordance with the requirements of a particular source stream. With binary FEC, we can do away with QoS negotiation altogether, which is certainly an advantage. However, consider the reliability requirements of the wireless link. Since the wireless link has no knowledge of the source requirements, it must be designed for a homogeneous QoS across all streams. There are two options. First, the designer can adjust the reliability for the most stringent—or most demanding—source. This conservative design approach will, for less stringent source requirements, overprovision resources such as bandwidth and power and overly restrict interference, thus reducing traffic capacity. While we don’t expect this to be a major issue on backbone networks, it may severely decrease the capacity of the bottleneck wireless access network if there is a wide variation in source QoS needs.

The second option is to design the wireless access link to be suitable for the least stringent source requirement and compensate by FEC on an end-to-end basis, as in PET. This option introduces several sources of inefficiency for heterogeneous
networks with wireless access. As noted earlier, binary erasure codes are very efficient in combatting congestion-based losses in a wired packet network. Their performance is significantly poorer in a wireless environment, where packets are likely to be corrupted due to the inherently high bit-error rate (BER). For the $10^{-2}$ uncoded BER typical of high mobility wireless and a packet size of $M = 120$ bits, the packet loss rate after applying an $(N = 8, B = 2)$ erasure code is:

$$\Pr[\text{packet error}] = \sum_{i=N-B+1}^{N} p^i (1 - p)^{N-i} = 0.26,$$

where

$$p = 1 - (1 - \text{BER}_{\text{uncoded}})^M.$$  

This performance is a modest improvement over the uncoded packet error rate of 70% but was achieved by quadrupling the bandwidth. A better way of lowering losses is to attempt to reduce the corruption rate instead, for example, by using a convolutional code. For the same bandwidth expansion as an $(8,2)$ erasure code, a convolutional code can lower the BER by two orders of magnitude [38, 39], thereby lowering the packet error rate to 1%. On a wireless link with rapid fading, this code will usually be accompanied by interleaving to turn correlated errors into quasi-independent errors. While convolutional coding may be attractive for wireless links, it is largely ineffective in a wired network, where losses are congestion-derived. Thus, it will be necessary with end-to-end FEC to concatenate different codes and interleaving designed to combat all anticipated error mechanisms, implying that the wireless link traffic will be penalized by redundancy intended for the other links in the network as well as its own.

The link-by-link architecture also permits us to apply physical layer signal processing techniques not possible in end-to-end binary FEC. In end-to-end binary FEC, one has no choice but to perform a hard decision on the information bits as they cross from one link to another. On a wireless link, we have control over the modulation and demodulation process and thus can apply soft decoding to the information bits. Hard decisions made prior to the final decoding result in an irreversible loss of information. This loss is equivalent to a 2 dB drop in the signal-to-noise ratio (SNR) [40], and the effect on loss and corruption is cumulative across multiple links. In addition, we can consider making the FEC and interleaving adaptive to the local traffic and propagation conditions on the wireless link.

Overall, active configuration of the QoS on a wireless link based on individual source requirements will substantially increase traffic capacity. The price to be paid is an infrastructure for QoS negotiation and configuration and the need to provision variable QoS in a wireless network; the latter issue is addressed further in Section 6.4.1. Fully quantifying this benefit requires further research since it
depends on the characteristics and requirements of the source traffic, as well as on
the benefits of variable QoS.

6.3.4 Complexity and Resource Allocation

Both the link and the edge architectures raise important issues in resource allocation
in session establishment. In both cases, for CM services the overriding objective is to
obtain acceptable and controllable subjective quality in the audio or video service.
Subjective quality is measured objectively by attributes such as frame rate and reso-
olution (for video), bandwidth (for audio), and delay (for both video and audio). It is
also measured by other factors more difficult to characterize, such as the perceptual
impact of artifacts introduced in the process of decompression by information cor-
ruped or discarded in the service (i.e., in the compression) and in the bitway
(packet losses), and also artifacts introduced by corruption in the bitway.

Inherently, resources belong to individual links, not to end-to-end con-
nections. However, the QoS negotiation between the services and bitway layers that
establishes each link’s resource use can be done end-to-end or link-by-link.

In the link architecture, overall subjective quality objectives must be refer-
enced back to the individual links, since each link will contribute artifacts that
impair subjective quality (such as quantization, blocking effects, error masking
effects, etc.). These artifacts will accumulate across links in a very complicated and
difficult-to-characterize way. (For example, how is a blocking or masking artifact
represented in the next compression/decompression stage?) It is relatively
straightforward to partition objective impairments like delay among the links.
Other objective attributes like frame rate, bandwidth, and resolution will be dic-
tated by the worst-case link and are thus also straightforward to characterize. Sub-
jective impairments due to loss and corruption artifacts will, however, be very
difficult, if not impossible, to characterize in a heterogeneous bitway environment.
Simple objective measures like mean-square error are fairly meaningless in the face
of complex impairments like the masking of bitway losses. Thus, as a practical
matter, it will be very difficult to predict and control end-to-end subjective quality.

The situation in the edge architecture is quite different. The first step is to gen-
erate an aggregated bitway model for all the concatenated bitway links. That is, the
loss models for the individual links must be referenced to a loss model for the over-
all connection, and similarly for corruption and delay. There are no doubt serious
complications in this aggregation, for example, correlations of loss mechanisms in
successive links due to common traffic. Nevertheless, this is a relatively straigh-
toward task susceptible to analytical modeling. Once this analysis is done, the aggre-
gate bitway model must be related back to service subjective quality, much in the
fashion of a single link in the link architecture. There is no need to characterize the
accumulation of artifacts in multiple compression/decompression stages. Accurate prediction and control of subjective quality in the edge architecture should be feasible, and this is an additional advantage over the link architecture.

### 6.3.5Multicast Connections

The problem of multicast connections is illustrated in Figure 6.12. With heterogeneous receiving terminals, or heterogeneous subnets, we may need different representations (say, with different bandwidth or resolution) of the CM service after a splitting bridge, but to conserve bitway resources we want to share a common stream before the bridge. An obstacle to this is encryption, which will hide the syntax of the originating stream. One solution is to locate transcoding at the bridge, preceded by decryption and followed by encryption, but this solution introduces all the disadvantages of the link architecture.\(^{13}\) The medley gateway provides a framework for the solution to this problem, as shown in Figure 6.13. At the point

---

\(^{13}\)A transcoding approach to multicast splitting is currently envisioned as part of the future Internet architecture [53].
where two representations are split, a (not necessarily proper) subset of the medley substreams is extracted for each downstream branch. From the perspective of the bitway, different endpoint terminals receive different subsets of substreams, with the great simplification that the bridging function can be accomplished entirely within the bitway layer. If each substream is independently encrypted, encryption does not interfere with this bridging function. Substreams in this context play a similar role to multicast groups in the MBone [52].

Support for heterogeneous terminals in the edge architecture presents to the service a well-defined design problem: perform a layered compression, such that a subset of the substreams embodies a minimal representation of the source, and the additional substreams provide additional information (higher resolution, higher sampling rate, etc.) to terminals with greater capabilities. Thus, in the edge architecture, the substream structure is used for three distinct but complementary purposes:

- **JSCC.** It allows the service to present to each bitway entity, in a generic fashion separated from particular service standards, the differing QoS requirements of different packets, thus allowing the bitway to efficiently allocate its resources.
- **Layered coding.** It allows the service to decompose its layered encoding in a way that is also generic and visible to the bitway layer, so that the splitting function required in multicast connections with heterogeneous terminals can be performed entirely within the bitway.
- **Privacy and security.** Independent encryption of the substreams allows the privacy and security of end-to-end encryption without interfering with either JSCC or multicast splitting.

### 6.4 Design Examples

JSCC for the medley gateway model has serious implications to the design of the wireless bitway, source coding, and services. In this section, we illustrate this by a few design examples.

#### 6.4.1 Variable QoS in Wireless Bitways

In Section 6.2, we discussed two design philosophies for multimedia networks: homogeneous QoS in the network with end-to-end unequal error protection (UEP), and active configuration of QoS within the individual links of the network. In the latter case, the approach is to adjust the QoS, and hence resources, of individual links in accordance with the requirements of each constituent stream. As we
believe the latter is a superior approach for wireless bitway design, we now discuss the provisioning of variable QoS. Our development complements that of Chapter 5. For generality, we consider a medley bitway (with substreams), although these results would apply equally well to the more restrictive case of QoS provisioned on a stream rather than substream granularity.

A variable QoS medley bitway has two design challenges: provide flexibility in loss/corruption/delay attributes with a substream granularity, and exploit the configured characteristics to maximize the traffic capacity. We illustrate the design issues for a wireless direct-sequence code-division multiple-access (CDMA) system. We focus on achieving variable reliability and ignore the issue of variable delay discussed elsewhere [60].

There are two handles for controlling reliability in CDMA: FEC and power control. Achieving variable reliability with FEC would require UEP. Many forms of UEP coding have been developed, notably algebraic codes for UEP and embedding asymmetric constellations in trellis-coded modulation [61]. However, the number of different levels of reliability provided by these techniques is limited, and it is difficult to apply them to hierarchical UEP. Variable rate (VR) convolutional codes have also been suggested for UEP coding of speech [62]. By adopting UEP, we can increase the reliability of a substream by adjusting the coding rate, at the expense of bandwidth expansion from the redundancy. Signal space codes such as trellis-coded modulation are particularly attractive for wireless networks because they provide redundancy without increasing bandwidth. However, it is difficult to generate (by varying the constellation size) the trellis equivalent of a variable rate convolutional code since, as Ungerboeck has shown, virtually all of the coding gain is attained by doubling the alphabet size [40].

As developed in Chapter 5, an alternative mechanism to control reliability QoS would be to adjust the signal-to-interference + noise ratio (SINR) by adjusting the transmitted power, taking into account the interference from other user's traffic being simultaneously transmitted on different spreading codes. Of course, it is beneficial for overall traffic capacity to minimize the transmitted power for any given user in order to minimize the interference to other users. Hence, overall traffic capacity is maximized by achieving, for each packet, no greater SINR than is necessary to meet the QoS objective. If we use power control only, then in a CDMA system we will be transmitting to a particular user at less than 100% duty cycle whenever the bit rate required by that user is less than the peak rate enabled by the chip rate and processing gain. Power control has some important advantages:

- It is easy to achieve variable reliability over a wide range of bit error rates by changing the power level. The power level can be dynamically varied to track
time-varying channel conditions and maintain relatively constant reliability for any substream.

- Power control is transparent to the receiver, requiring no special processing.
- For CDMA, using a fixed-rate code to provide maximum coding gain without bandwidth expansion is easy, since the spectrum has already spread; i.e., channel coding and spreading can be combined to provide redundancy without bandwidth expansion [63].

The first question is whether, ignoring implementation issues, it is most advantageous to use UEP or power control. To address this issue, let us examine UEP and power control from an information theory perspective, using the following elementary calculation. In a CDMA system, focus on a single user’s data and approximate the total interference as white Gaussian noise with power spectrum $N_0$, and let the bandwidth be $B$. Let $P$ be the average transmitted power for this particular user, and transmit with a duty cycle $\gamma \leq 1$. Then, the transmitted power during that duty cycle is $P/\gamma$. The channel capacity using this duty cycle is $\gamma$ times the capacity if we transmitted at this same power level at 100% duty cycle, where the latter is $B \cdot \log \left(1 + \frac{P/\gamma}{2N_0B}\right)$. Thus, the overall channel capacity is

$$C = \gamma B \cdot \log \left(1 + \frac{P}{2N_0\gamma B}\right), \quad (6.3)$$

which is precisely the same as the capacity of a channel with bandwidth $\gamma B$ with 100% duty cycle transmission and transmitted power $P$. Since this capacity is maximum for $\gamma = 1$, we conclude that it is advantageous to transmit with 100% duty cycle in order to minimize the average transmitted power $P$ for a fixed bit rate $C$. To minimize $P$, and hence the interference to other users, we should always transmit at 100% duty cycle by adding channel coding redundancy as necessary. Intuitively, it is advantageous to use coding to increase the duty cycle of transmission to 100%, regardless of the required bit rate, and take advantage of the coding gain to reduce the average transmit power.

Thus, information theory teaches us that in an interference-dominated wireless channel such as CDMA, it is best to use coordinated UEP and power control. Either UEP or power control in isolation is suboptimum at the fundamental limits. If the bit rate for a given CDMA spreading code is low, coding redundancy should be added and the transmitted power simultaneously reduced. The bitway coding and power control layers in a wireless cellular bitway design are illustrated in Figure 6.14. A set of substreams is applied to a coding layer that is cognizant of the propagation characteristics of the channel and that configures itself to provide the
negotiated QoS contract. Based on the coding selected, each substream is associated with a required SINR. The power control layer then associates a transmitted power with each substream, taking into account a maximum power requirement, the SINR requirement for each substream, and which substreams currently have packets awaiting transmission.

Finally, let us quantify the capacity gain in a wireless CDMA system attainable by using joint power and error control for variable QoS. The traffic capacity is the amount of traffic that the system can support, subject to the (possibly distinct) QoS demands of the traffic. Let \( M \) be the number of users and \( K_m \) be the number of substreams of user \( m \). Specify a user (CDMA spreading code) by subscript \( m \) and a substream by subscript \( k \). The SINR experienced by the \( k \)th substream of user \( m \) on the uplink is

\[
\text{SINR}_{\text{experienced}} = \frac{G_m x_{k,m}}{\sigma^2 + I_{m}^{\text{intra}}},
\]

where \( G_m \) is the path loss from user \( m \) to the base station; \( x_{k,m} \) is the transmit power assigned to substream \( k \); \( I_{m}^{\text{intra}} \) is the intracell interference experienced by user \( m \), and \( \sigma^2 \) is the lump sum of background noise and intercell interference experienced at the base station.

\[\footnote{For example, in the design of the trellis code, using a metric different from the Euclidean metric typically used for the additive Gaussian noise channel is advantageous for Rayleigh fading channels [64].} \]

\[\footnote{In practice, we would also like to schedule the most opportune time for a packet transmission to take advantage of allowed delay jitter. This problem is considered elsewhere [60].} \]
The intracell interference experienced by a substream of user $m$ is

$$f_{m}^{\text{intra}} = \sum_{n=1}^{M} f_{m,n} G_n \sum_{k=1}^{K_n} \beta_{k,n} x_{k,n}$$

(6.5)

where $\beta_{k,m}$ is an indicator function, equaling one if substream $k$ of user $m$ is currently active, zero otherwise. Each user’s traffic can be decomposed into multiple substreams, but the substreams are statistically multiplexed together onto one user stream so that only one of the user’s substreams is active at any time. In (6.5), $f_{m,n}$ is the partial correlation coefficient (or degree of nonorthogonality) between channels of users $m$ and $n$: because signals from different mobile users travel through different multipath channels to reach the base station, perfect orthogonality between user codes may be lost and $f_{m,n}$ may be non-zero. Uplink transmission is inherently asynchronous, so $f_{m,n}$ is well modeled by $f$, the correlation between random signature sequences, with $E[f] = 2/3$ [65].

The indicator function of a substream as it evolves over time $\beta_{k,m}(t), t \in [0, \infty)$ is a random process. Let $\bar{\beta}_{k,m}$ denote the long-term time average of $\beta_{k,m}(t)$; e.g., $\bar{\beta}_{k,m} = 1/4$ if the average bit rate of substream $k$ is 500 kbps and it belongs to a 2-Mbps user stream. We assume ergodicity in the mean, so that $E[\beta_{k,m}(t)] = \bar{\beta}_{k,m}$. This assumption is based on the intuition that at any given time slot, the probability that you receive a packet from substream $k$ equals the average rate of that substream, divided by the aggregate rate of the user stream to which it belongs. The expected value of the total power is then

$$E[P] = E \left[ \sum_{m=1}^{M} \sum_{k=1}^{K_m} \beta_{k,m}(t) x_{k,m} \right] = \sum_{m=1}^{M} \sum_{k=1}^{K_m} \bar{\beta}_{k,m} x_{k,m}$$

(6.6)

Our objective is to minimize the average overall power $E[P]$ while promising each substream that the expected value of the SINR it experiences will meet or exceed the desired SINR:

minimize $E[P] = \sum_{m=1}^{M} \sum_{k=1}^{K_n} \bar{\beta}_{k,m} x_{k,m}$ such that

\[ \forall k, m, \quad \frac{G_n x_{k,m}}{\sigma^2 + E[f] \sum_{n=1}^{M} G_n \sum_{j=1, j \neq m}^{K_n} \beta_{j,n} x_{j,n}} \geq \text{SINR}_{k,m} \]

(6.8)

$x_{k,m} \geq 0$, SINR$_{k,m} > 0$
In (6.8), $\text{SINR}_{k,m}$ is the SINR requested by substream $k$ of user $m$, and the inequality in 6.8 implies that the expected value of the SINR achieved at the receiver must equal or exceed the desired SINR.

It can be shown [66] that the feasible capacity region of a CDMA system is given by

$$\gamma < 1,$$  \hspace{1cm} (6.9)

where

$$\gamma = \sum_{m=1}^{M} \gamma_{m}$$  \hspace{1cm} (6.10)

$$\gamma_{m} = E[f] \sum_{k=1}^{K_{m}} \alpha_{k,m}, \text{ and}$$  \hspace{1cm} (6.11)

$$\alpha_{k,m} = \frac{\text{SINR}_{k,m} \bar{\beta}_{k,m}}{1 + E[f] \sum_{j=1}^{K_{m}} \bar{\beta}_{j,m} \text{SINR}_{j,m}}$$ \hspace{1cm} (6.12)

$k = 1, \ldots, K_{m}, \quad m = 1, \ldots, M.$

The left-hand side of (6.9), $\gamma$, represents the load of the system. The closer $\gamma$ is to unity, the closer the system is to violating the QoS requirements for all users and substreams. If the QoS requirements are too stringent, then the interference will be too great and no solution exists, regardless of the transmit power. Examining (6.12) we see that for a cellular wireless system whose capacity is interference-limited, the “cost” of transmitting an information substream is the product of its reliability requirement (specified by an SINR) and bandwidth requirement (specified by its average rate $\bar{\beta}$).

As noted earlier, information theory suggests that the application of VR coding to adapt a variable-rate information source to a bandlimited channel is necessary in order to maximize capacity. In a CDMA system, a user is associated with a code, and the bandwidth afforded by the code is shared by the user’s substreams. Each substream is allocated a time-averaged fraction of the bandwidth, $\bar{\beta}$. To ensure optimal capacity in the information theoretic system, the bandwidth associated with the code should be used at 100% duty cycle; i.e., in (6.9)–(6.12), $\{\bar{\beta}_{j,n}\}$ should satisfy

$$\sum_{j=1}^{K_{n}} \bar{\beta}_{j,n} = 1 \text{ for all users } j,$$ \hspace{1cm} (6.13)

with the SINR requirements of all substreams adjusted for the resulting coding gain. We note that the interference-limited capacity result as stated in (6.9)–(6.12) is
quite general and can also be applied toward suboptimal systems that do not achieve 100% utilization of the channel bandwidth.

We can apply these results to find the capacity gain of power control for variable QoS with substreams over power control without substreams. In the absence of substreams, each stream’s reliability requirement would be equal to the reliability need of its worst-case (most error-sensitive) information component. A fine-grained substream architecture therefore achieves a capacity gain of

$$\text{capacity gain} = \frac{\sum \sum \sum \alpha_{k,m}^{(\text{max})} \beta_{i,k,m}}{\sum \sum \sum \alpha_{i,k,m} \beta_{i,k,m}}$$

where $\alpha_{k,m}^{(\text{max})}$ corresponds to the maximum SINR requirement among the substreams of stream $k$.

### 6.4.2 MPEG-2 Compression

The International Organization for Standardization’s Moving Pictures Experts’ Group (ISO/MPEG) has developed several well-known audiovisual compression standards:

- MPEG-1 is designed for VCR-quality audio and video compression and for delivery via reliable media such as CD-ROMs [16].
- MPEG-2 is designed for high-quality broadcast applications, including entertainment, remote learning, electronic publishing, and more [17].
- MPEG-3 was intended to address high-definition television, but this effort was folded into MPEG-2.
- MPEG-4, just beginning development, is intended to address wireless interactive multimedia coding and transmission [67]. MPEG-4 currently borrows some technology from the ITU-T Study Group 15, for example, SG15’s wireless multiplexing protocol, H.245.

MPEG-2 is not designed for wireless multiaccess but rather for wireless and wired broadcast; as such, the decisions made in the design of MPEG-2 are quite different from those that would be made for a wireless multiaccess system. Broadcast channels differ from wireless multiaccess channels in that they are noise rather than interference-limited and thus can be provisioned to deliver lower bit-error rates and very much lower burst-error rates. Thus, error resiliency tools for broadcast applications should be designed and optimized differently than those for
wireless multiaccess systems. Nevertheless, it is instructive to review the error resiliency features included in MPEG-2. In Section 6.4.3, we illustrate a much different approach.

MPEG-2 is a service layer standard and today is used with a range of bitways: direct broadcast satellite, digital switched line, cable television, ATM, and more. As a suite of service layer standards, MPEG-2 does not provide bitway QoS-enhancing functionality such as data interleaving, selective packet discard, or FEC. Still, MPEG-2 does provide a range of functionality to help resynchronize and recover quickly from bitway errors and to configure to trade off efficiently between bandwidth, delay, loss, and service quality.

6.4.2.1 Features

MPEG-2 contains three subparts that define bitstream formats: The audio specification defines a compressed representation of a multichannel audio signal. The video specification defines a compressed representation of a moving picture sequence. The systems specification defines, among other things, how to multiplex multiple audio, video, and data streams into a single packetized bit stream.

The audio and video compression methods defined by MPEG-2 contain many predictive coding steps. For example, the video specification includes interframe motion-compensated DPCM, predictive coding of motion vectors within a frame, predictive coding of DCT brightness coefficients within a frame. Predictive coding, which represents a signal's difference from a predicted value rather than representing the signal value directly, removes signal redundancy very effectively but suffers from error propagation. Errors cause predictive decoders to incorrectly render data which is used in future predictions. These subsequent predictions with errors lead to more incorrectly decoded data; this propagates errors throughout spatio-temporally nearby audio or video. Fortunately, MPEG-2 allows an encoder to define "resynchronization points" almost as often or infrequently as the designer desires, to trade between bandwidth efficiency and rapid error recovery. At a resynchronization point, signal values are coded directly rather than via a predictor.

Both the audio and video compression algorithms use Huffman coding, which uses short bitstrings to represent frequently occurring values and long bitstrings to represent infrequent values. A property of Huffman codes is that they cannot be decoded without some context—knowledge of the bit position of the start of some bitstring. Huffman codes also suffer error propagation: one bit error destroys the decoder's context, and the decoder may incorrectly decode a long sequence of values. To alleviate this problem, the audio and video specifications both define "startcodes," which are patterns in the bit streams that decoders can find easily and at which Huffman codes are known to be at the start of a bitstring. These startcodes do consume a small but non-zero percentage of audio and video
stream bandwidth, but they enable decoder recovery after a transmission error. The video specification allows bitstream encoders to insert "slice" startcodes at either a default minimum rate or at a higher rate to enable faster-than-default error recovery.

The system specification defines a bitstream format ("transport streams") that consists of short (compared to TCP/IP) fixed-length packets. Short packets ensure that if a receiver identifies a packet as corrupted, comparatively little data is suspect. Fixed-length packets facilitate rapid identification of packet delineators after errors.

MPEG-2 transport streams can contain "duplicate packets." A duplicate packet is a copy of the previous packet with the same source identifier. Duplication is a simple (but not efficient) flavor of FEC. Combined with bitway interleaving, duplication greatly reduces the occurrence of uncorrectable burst errors, however. A sensible strategy is to duplicate all packets that contain the highest-level startcodes.

Example

Suppose we transmit an MPEG-2 transport stream with 30 packets per second that contain critical video headers (one such packet per picture). Suppose the bitway layer uses interleaving to ensure that the probability of bit error is uniform and identically distributed and uses FEC to ensure a probability of bit error of $10^{-8}$. The probability that any packet contains an error is $1 - (1 - 10^{-8})^{188 \times 8} = 1.5 \times 10^{-5}$ since there are 188 bytes in an MPEG-2 transport stream packet. This means that without duplicate packets, we can expect a packet with a critical header to be corrupted about once every 2,216 seconds or every 37 minutes. If we send duplicate packets for each packet with a critical header, the probability that both an original critical packet and its duplicate are corrupted is $(1 - (1 - 10^{-8})^{188 \times 8})^2 = 2.3 \times 10^{-10}$. Both an original critical packet and its duplicate are lost about once every $1.5 \times 10^8$ seconds, or once per 4.7 years.

The MPEG-2 system specification defines a timing recovery and synchronization method that utilizes two types of timestamps. "Program clock references" (PCRs) allow receivers to implement accurate phase-locked loop clock recovery. Stringent limitations on the encoder clock frequency accuracy and drift rate allow decoders to identify and discard corrupted PCRs. Video frames and audio segments are identified by decode/presentation timestamps (DTS/PTS), which tell the decoder the proper time to decode and present the associated video and audio data. Since each frame has a fixed (and known to the decoder) duration, there is a lot of redundancy in the DTS/PTS values. However, the small amount of bandwidth spent on PCRs and DTS/PTSs helps decoders properly prefill their input buffers and properly synchronize their video and audio outputs after errors.

The MPEG-2 systems, video, and audio subparts specify only bitstream formats. Another part of MPEG-2, the RTI, defines constraints on real-time delivery of
systems bit streams to actual decoders. The RTI defines a method for measuring the delay jitter present when a bit stream is delivered to a decoder; this approach aids in ensuring interoperability between bitstream providers and decoders. The RTI does not mandate a specific delay jitter value; the designer chooses a value suitable for the system, e.g., 50 μs for a low-jitter connection between a digital VCR and a decoder, or more than 1 ms for an international ATM connection. The RTI defines decoder memory requirements and bitstream delivery constraints based on the chosen jitter value. The specification of decoder memory requirements as a function of delay jitter is very important for many MPEG-2 applications, where decoder cost, largely driven by memory, is the biggest determinant of commercial viability.

6.4.2.2 Scalability Tools

Hierarchical or layered coders are good candidates for use with QoS-impaired bitways. As shown in Figure 6.15, the base layer of a hierarchical coder represents the input signal at some coarse fidelity. Higher layers code the residual between the base layer decoded output and the original input; the output of the base layer combined with higher-layer decoded output is more accurate than the base layer output alone. For a given service quality level, the aggregate bit rate of a good hierarchical coder is close to that achievable by the best nonhierarchical coders.

A hierarchical coder’s outputs have different QoS requirements; often it is acceptable for only the base layer to be decoded for short periods of time. Thus, only the base substream requires high QoS in order to achieve acceptable application quality; if data from other substreams is lost, the decoded signal is corrupted, but not catastrophically.

A simple example of a hierarchical video coder simply transmits the most significant bits of a picture’s pixels on one substream and the least significant bits on another. If some of the least significant bits are lost, affected picture regions appear coarsely quantized but certainly recognizable.

Figure 6.15 Structure of a hierarchical encoder.
The MPEG-2 video specification defines several much more sophisticated ways by which an encoder can produce a two- or three-layer hierarchically encoded bit stream. MPEG-2 video allows an encoder to generate hierarchical bit streams that decompose a signal into low frame-rate vs. high frame-rate components, small picture size vs. large picture size, or low image fidelity vs. high image fidelity (called “SNR scalability”) components. Several of these decompositions can be used in tandem as well.

MPEG-2 video defines another scalability tool, called data partitioning. Data partitioning defines how to split a non-hierarchically-encoded bit stream into two substreams. The high-priority substream contains video headers and other important syntax elements such as motion vectors. The low-priority bit stream contains lower-priority syntax elements such as high-frequency discrete cosine transform (DCT) coefficients. The encoder can choose its definition of high-priority and low-priority syntax elements to achieve its best trade-off between high-priority bandwidth, high-priority QoS, low-priority bandwidth, and low-priority QoS.

### 6.4.3 JSCC for Delay: Delay-Cognizant Video Compression

Having described MPEG, an established standard, let us now illustrate a dramatically different approach motivated by the need for efficient use of wireless channels. The design of today’s CM services are a holdover from the circuit switched era, when bitways did not introduce significant delay jitter. Existing compression standards for both audio and video thus assume a fixed-delay transport model, imposing on the bitway the need to emulate a fixed-delay circuit. This emulation requires the artificial delay of packets arriving early. In the context of delay-critical interactive services, it seems intuitively unattractive to artificially add delay, and one wonders if it is not possible to take advantage of these early-arriving packets.

Since substreams have different delay characteristics, it is inherent that they are asynchronous at the receiving terminal. They can be resynchronized by an appropriate medley transport protocol, but not resynchronizing them allows the delay characteristics of the bitway to be exploited. To this end, the medley gateway abstractions offer several key benefits:

- The service is allowed to specify different delay characteristics for different substreams. This is a direct way for the service to control which packets arrive earlier and which arrive later, which in turn makes the differential delays more useful. The model also encourages the medley bitway to deliver certain packets earlier, whereas conventional approaches do not.
- For a fixed traffic capacity, bitways generally trade higher reliability for increased delay (through techniques like FEC, interleaving, retransmission,
etc.). The medley gateway model allows the service to explicitly control as well as exploit this trade-off and to force this trade-off to be quantitatively different for different packets.

- Since the overall delay is no longer determined by the *worst-case* delay, the bitway worst-case delay can be relaxed, which can in turn be traded for increased traffic capacity through traffic smoothing.

What we have just described is JSCC in the delay dimension. By making the source coding delay-cognizant, that is, segmenting its information into delay classes, we hope to achieve a more desirable combination of perceptual delay and traffic capacity.

An early example of delay-cognizant video coding is *asynchronous video* [68], a coding technique that exploits variations in the temporal dimension of video to segment information into distinct delay classes. We leave the details to other references [68] but illustrate the basic idea in Figure 6.16. The frame is block-segmented into different delay and reliability classes in accordance with motion estimation (three classes are shown). These different classes are allowed to be offset at the receiver by one or more frames in the reconstruction process. The hope is that low-motion blocks are less susceptible to multiple-frame delay jitter at the receiver than are high-motion blocks and that the user perception of delay will be dominated by the high-motion blocks. If this is the case, low-motion blocks can be assigned to a medley bitway substream with a relaxed delay objective, and the bitway can

![Figure 6.16](image)

*Figure 6.16* Asynchronous video as an example of delay-cognizant video coding. Blocks of video are reconstructed in different frames at the sink, based on motion segmentation.
exploit this relaxed delay jitter objective to achieve higher traffic capacity. In addition, high-motion blocks are assigned to substreams with a relaxed reliability objective since the motion tends to subjectively mask losses or corruption. Fortuitously, the bitway naturally provides precisely the needed exchange of higher reliability for higher delay.

6.4.4 Multiple-Delivery Transport Protocol

The importance of the transport protocol as a way to change the characteristics of the bitway to the benefit of the application was discussed in Section 6.2.5. An example of a transport protocol tailored to the needs of CM services is a multiple delivery service [70, 71]. Interference-limited wireless access links typically have two undesirable characteristics: restricted bandwidth and low reliability. Error control techniques to compensate for the latter increase the rate (for redundancy or retransmissions), and this rate increase trades unfavorably against delay because of the restricted bandwidth. Thus, reliable delivery mechanisms increase delay substantially. This will be problematic for interactive applications, for example refreshing a graphics window in a WWW browser. If graphics are treated as a pixel map (as in the InfoPad™ system [72]), it is advantageous to display corrupted information early, but it is also important that corruption artifacts do not stay on the screen indefinitely (asymptotic reliability). This corruption control can be accomplished without a traffic capacity penalty by exploiting the redundancy needed anyway to deliver two or more copies of a packet to the receiver, each with increasing reliability, as illustrated in Figure 6.17. The application delivers a single copy of each packet to the transport protocol. The transport delivers, in general, more than

![Image of a multiple-delivery transport protocol]

Figure 6.17 A multiple-delivery transport protocol.
one copy of the packet to the receiver, where it is agreed that each copy has statistically greater fidelity (fewer bit errors) than the previous copy. Internally, the transport protocol can utilize packet combining techniques, where it transmits the packet as many times as required and caches all received renditions of the packet, delivering to the application its best estimate of the packet based on all the cached information. Acknowledgments built into the protocol allow the number of transmissions to be adjusted dynamically to channel conditions. This protocol has proven useful for video [32].

Protocols such as the multiple-delivery transport protocol should also have a mechanism to purge stale packets; that is, packets that will not be used by the receiver if they are delivered.

6.5 Concluding Remarks

The most important point of this chapter is that in an integrated-services multimedia network, it is advantageous to take an overall systems perspective, rather than designing wireless access networks in isolation. We have seen how, by coordinating the design of the backbone network, terminals, and servers with the wireless access network, greater traffic capacity can be achieved subject to subjective quality objectives. At the same time, it is important to adhere to good principles of complexity management and ensure that the different parts of the multimedia network are made modular and as independent as possible, with appropriate levels of scalability and configurability. Achieving modularity requires a carefully crafted architecture for the network. We have proposed the medley gateway model based on substreams or flows (supported by existing or emerging protocols in both IP and ATM networks) as a basic unifying principle of the architecture. Once an architectural approach is chosen, many opportunities for research in the various modules open up. We have illustrated the design of a video source coder, a variable QoS wireless CDMA media access layer, and a transport protocol within the context of this architecture.

The considerations covered in this chapter suggest many opportunities for research, which include:

- The design of medley services that take advantage of the medley bitway (such as delay/loss trade-offs and segmentation) and that have the needed level of scalability and configurability.
- The design of medley bitways that maintain the structural integrity of the substreams, which have the ability to configure to different impairment profiles for different substreams and which exploit the substream structure to achieve higher traffic capacity.
• An understanding of JSCC, as constrained by the structure of the medley gateway model. Similarly, an understanding of the design of hierarchical compression algorithms for multicast heterogeneous terminals, as constrained by the same substream structure.

• An understanding of issues inherent in the aggregation of concatenated bitway links for CM services.

• Development of negotiation strategies for resolving the trade-off between subjective quality vs. bitway QoS and cost.

• The upgrade of signaling systems to provide the needed capabilities in support of the edge architecture, including aggregation of bitway links and negotiation between the endpoint terminals and the aggregated links.

REFERENCES


[16] ISO/IEC Standard 11172, "Coding of Moving Pictures and Associated Audio at up to about 1.5 Mbits/s." (MPEG-1).


ACKNOWLEDGMENTS

The authors appreciate the contributions of their colleagues Jonathan Reason, Richard Han, and Yuan-Chi Chang to the insights reported in this chapter. This research is supported by Bell Communications Research, Pacific Bell, Tektronix, MICRO, and the Defense Advanced Research Projects Agency.