In summary, PPP is a data link layer protocol by which two communicating link-level peers, one on each end of a point-to-point link, exchange PPP frames containing network-layer datagrams. The principal components of PPP are:

- *Framing.* A method for encapsulating data in a PPP frame, identifying the beginning and end of the frame, and detecting errors in the frame.
- *Link-control protocol.* A protocol for initializing, maintaining, and taking down the PPP link.
- *Network-control protocols.* A family of protocols, one for each upper-layer network protocol, that allows the network-layer modules to configure themselves before network-level datagrams begin flowing across the PPP link.

5.9 ♦ Asynchronous Transfer Mode (ATM)

The standards for ATM were first developed in the mid-1980s. For those too young to remember, at this time there were predominately two types of networks: telephone networks, which were (and still are) primarily used to carry real-time voice, and data networks, which were primarily used to transfer text files, support remote login, and provide e-mail. There were also dedicated private networks available for video conferencing. The Internet existed at this time, but few people were thinking about using it to transport phone calls, and the World Wide Web was as yet unheard of. It was therefore natural to design a networking technology that would be appropriate for transporting real-time audio and video as well as text, e-mail, and image files. Asynchronous transfer mode (ATM) achieved this goal. Two standards committees, the ATM Forum [ATM 2002] and the International Telecommunications Union [ITU 2002] developed standards for broadband digital services networks.

The ATM standards call for packet switching with virtual circuits (called virtual channels in ATM jargon). The standards define how applications directly interface with ATM, so that ATM provides a complete networking solution for distributed applications. Paralleling the development of the ATM standards, major companies throughout the world made significant investments in ATM research and development. These investments have led to a myriad of high-performing ATM technologies, including ATM switches that can switch terabits per second. In recent years, ATM technology has been deployed very aggressively within both telephone networks and the Internet backbones.

Although ATM has been deployed within networks, it has been less successful in extending itself all the way to desktop PCs and workstations. And it is now questionable whether ATM will ever have a significant presence at the desktop. Indeed, while ATM was brewing in the standards committees and research labs in the late 1980s and early 1990s, the Internet and its TCP/IP protocols were already operational and making significant headway:

- The TCP/IP protocol suite was integrated into all of the most popular operating systems.
- Companies began to transact commerce (e-commerce) over the Internet.
- Residential Internet access became inexpensive.
- Many wonderful desktop applications were developed for TCP/IP networks, including the World Wide Web, Internet phone, and interactive streaming video. Thousands of companies are currently developing new applications and services for the Internet.

Today, we live in a world in which most networking applications interface only with TCP/IP. Nevertheless, ATM switches can forward data at very high rates and, consequently, they have been deployed in Internet backbone networks, where the need to transport traffic at high rates is most acute. When ATM is in the Internet backbone, TCP/IP runs on top of ATM and views an entire ATM network, which might span a continent, as one large link-layer network. In other words, although ATM has not caught hold as a process-to-process solution (or even a desktop-to-desktop solution), it has found a niche at the link level within parts of the Internet backbone; this is referred to as IP-over-ATM—a topic we'll cover in Section 5.9.5. For these reasons, we have included our discussion of ATM in this chapter on the link layer, rather than in the previous chapter on the network layer.

5.9.1 Principal Characteristics of ATM

The principal characteristics of ATM are as follows:

- The ATM standard defines a full suite of communication protocols, from an application-level API all the way down through the physical layer.
- The ATM service models include constant bit rate (CBR) service, variable bit rate (VBR) service, available bit rate (ABR) service, and unspecified bit rate (UBR) service. We've already considered some of the service models in detail in Section 4.1.
- ATM uses packet switching with fixed-length packets of 53 bytes. In ATM jargon, these packets are called **cells.** Each cell has five bytes of header and 48 bytes of "payload." The fixed-length cells and simple headers have facilitated high-speed switching.
- ATM uses virtual circuits. In ATM jargon, virtual circuits are called **virtual channels.** The ATM header includes a field for the virtual channel number, which is called the **virtual channel identifier (VCI)** in ATM jargon. As discussed in Section 1.3, packet switches use the VCI to forward cells toward their destinations.

- ATM provides no retransmissions on a link-by-link basis. If a switch detects an error in an ATM cell header, it attempts to correct the error using error-correcting codes. If it cannot correct the error, it drops the cell rather than request a retransmission from the preceding switch.
- ATM provides congestion control only within the ATM ABR service class (see Table 4.1). We covered ATM ABR congestion control in Section 3.6, where we saw that it belongs to the general class of network-assisted congestion-control approaches. ATM switches themselves do provide feedback to a sending end system to help it regulate its transmission rate in times of network congestion.
- ATM can run over just about any physical layer. It often runs over fiber optics using the SONET standard at speeds of 155.52 Mbps, 622 Mbps, and higher.

As shown in Figure 5.44, the ATM protocol stack consists of three layers: the ATM physical layer, the ATM layer, and the ATM adaptation layer (AAL):

- The **ATM physical layer** deals with voltages, bit timings, and framing on the physical medium.
- The **ATM layer** is the core of the ATM standard. It defines the structure of the ATM cell.
- The **ATM adaptation layer** (AAL) is roughly analogous to the transport layer in the Internet protocol stack. ATM includes several different types of AALs to support different types of services.

Currently, ATM is most commonly used as a link-layer technology within localized regions of the Internet. A special AAL type, AAL5, has been developed to allow TCP/IP to interface with ATM. At the IP-to-ATM interface, AAL5 prepares IP datagrams for ATM transport; at the ATM-to-IP interface, AAL5 reassembles ATM cells into IP datagrams. Figure 5.45 shows the protocol stack for the regions of the Internet that use ATM. Note that in this configuration, the three ATM layers have been squeezed into the lower two layers of the Internet protocol stack. In particular, the Internet's network layer views ATM as a link-layer protocol. A nice tutorial on ATM, reflecting its original goals, is given in [LeBoudec 1992].

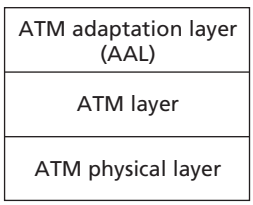

Figure 5.44 ♦ The three ATM layers

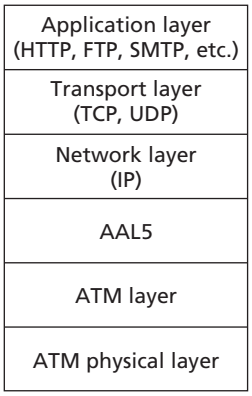

Figure 5.45 ♦ Internet-over-ATM protocol stack

5.9.2 ATM Physical Layer

The physical layer is concerned with sending an ATM cell over a single physical link. As shown in Table 5.2, the physical layer has two sublayers: the physical medium dependent (PMD) sublayer and the transmission convergence (TC) sublayer.

The Physical Medium Dependent (PMD) Sublayer

The PMD sublayer is at the very bottom of the ATM protocol stack. As the name implies, the PMD sublayer depends on the physical medium of the link; in particular, the sublayer is specified differently for different physical media (fiber, copper, and so on). It is also responsible for generating and delineating bits. There are two classes of PMD sublayers: PMD sublayers that have a transmission frame structure (for example, T1, T3, SONET, or SDH) and PMD sublayers that do not. If the PMD has a frame structure, then it is responsible for generating and delineating frames. (The terminology "frames" in this section is not to be confused with link-layer

Sublayer	Responsibilities
Transmission convergence (TC) sublayer	Idle cell insertion Cell delineation Transmission frame adaptation
Physical medium dependent (PMD) sublayer	Physical medium Bit voltages and timings Frame structure

Table 5.2 ♦ The Two Sublayers of the Physical Layer and Their Responsibilities

frames used in the earlier sections of this chapter. The transmission frame is a physical-layer TDM-like mechanism for organizing the bits sent on a link.) The PMD sublayer does not recognize cells. Some possible PMD sublayers include:

- SONET/SDH (synchronous optical network/synchronous digital hierarchy) over single-mode fiber. Like T1 and T3, SONET and SDH have frame structures that establish bit synchronization between the transmitter and receiver at the two ends of the link. There are several standardized rates, including:

  OC-1: 51.84 Mbps
  OC-3: 155.52 Mbps
  OC-12: 622.08 Mbps

- T1/T3 frames over fiber, microwave, and copper.
- Cell-based with no frames. In this case, the clock at receiver is derived from a transmitted signal.

Transmission Convergence (TC) Sublayer

The ATM layer is specified independently of the physical layer; it has no concept of SONET, T1, or physical media. A sublayer is therefore needed (1) at the sending side of the link to accept ATM cells from the ATM layer and prepare them for transmission on the physical medium, and (2) at the receiving side of the link to group bits arriving from the physical medium into cells and pass the cells to the ATM layer. These are the jobs of the TC sublayer, which sits on top of the PMD sublayer and just below the ATM layer. We note that the TC sublayer is also physical medium–-dependent—if we change the physical medium or the underlying frame structure, then we must also change the TC sublayer.

On the transmit side, the TC sublayer places ATM cells into the bit and transmission frame structure of the PMD sublayer. On the receive side, it extracts ATM cells from the bit and transmission frame structure of the PMD sublayer. It also performs header error control (HEC). More specifically, the TC sublayer has the following tasks:

- At the transmit side, the TC sublayer generates the HEC byte for each ATM cell that is to be transmitted. At the receive side, the TC sublayer uses the HEC byte to correct all one-bit errors in the header and some multiple-bit errors in the header, reducing the possibility of incorrect routing of cells. The HEC is computed over the first 32 bits in the cell header, using an eight-bit polynomial-coding technique, as described in Section 5.2.3.
- At the receive side, the TC sublayer delineates cells. If the PMD sublayer is cell-based with no frames, then this delineation is typically done by running the HEC

on all contiguous sets of 40 bits (that is, five bytes). When a match occurs, a cell is delineated. Upon matching four consecutive cells, cell synchronization is declared and subsequent cells are passed to the ATM layer.

- If the PMD sublayer is cell-based with no frames, the sublayer sends an idle cell when the ATM layer has not provided a cell, thereby generating a continuous stream of cells. The receiving TC sublayer does not pass idle cells to the ATM layer. Idle cells are marked in the PT field in the ATM header.

5.9.3 ATM Layer

When IP runs over ATM, the ATM cell plays the role of the link-layer frame. The ATM layer defines the structure of the ATM cell and the meaning of the fields within this structure. The first five bytes of the cell constitute the ATM header; the remaining 48 bytes constitute the ATM payload. Figure 5.46 shows the structure of the ATM header.

The fields in the ATM cell are as follows:

- *Virtual channel identifier (VCI).* Indicates the VC to which the cell belongs. As with most network technologies that use virtual circuits, a cell's VCI is translated from link to link (see Section 1.3).
- *Payload type (PT).* Indicates the type of payload the cell contains. There are several data payload types, several maintenance payload types, and an idle cell payload type. (Recall that idle cells are sometimes needed by the physical layer for synchronization.)
- *Cell-loss priority (CLP) bit.* Can be set by the source to differentiate between high-priority traffic and low-priority traffic. If congestion occurs and an ATM switch must discard cells, the switch can use this bit to first discard low-priority traffic.
- *Header error control (HEC) byte.* Error-detection and -correction bits that protect the cell header, as described above.

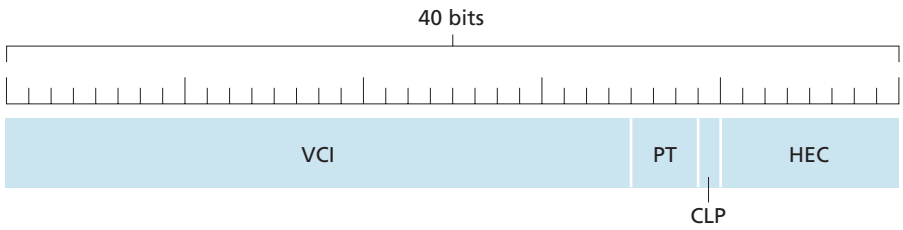

Figure 5.46 ♦ The format of the ATM cell header

Virtual Channels

Before a source can begin to send cells to a destination, the ATM network must first establish a virtual channel (VC) from source to destination. A virtual channel is nothing more than a virtual circuit, as described in Section 1.3. Each VC is a path consisting of a sequence of links between source and destination. On each of the links the VC has a virtual circuit identifier (VCI). Whenever a VC is established or torn down, VC translation tables must be updated (see Section 1.3). As noted earlier, ATM backbones in the Internet often use permanent VCs; they obviate the need for dynamic VC establishment and teardown.

5.9.4 ATM Adaptation Layer

The purpose of the AAL is to allow existing protocols (for example, IP) and applications (for example, constant bit rate video) to run on top of ATM. As shown in Figure 5.47, AAL is implemented only at the endpoints of an ATM network. Such an endpoint could be a host system (if ATM provides end host–to–end host data transfer) or an IP router (if ATM is being used to connect two IP routers). In this respect, the AAL layer is analogous to the transport layer in the Internet protocol stack.

The AAL sublayer has its own header fields. As shown in Figure 5.48 these fields occupy a small portion of the payload in the ATM cell.

The ITU and the ATM Forum have standardized several AALs. Some of the most important AALs and the ATM service classes (see Section 4.1.3) they typically support include:

AAL 1: For constant bit rate (CBR) services and circuit emulation.
AAL 2: For variable bit rate (VBR) services.
AAL 5: For data (for example, IP datagrams)

AAL Structure

AAL has two sublayers: the segmentation and reassembly (SAR) sublayer and the convergence sublayer (CS). As shown in Figure 5.49, the SAR sits just above the ATM layer; the CS sublayer sits between the user application and the SAR sublayer. Higher-layer data (for example, an IP datagram) are first encapsulated in a common

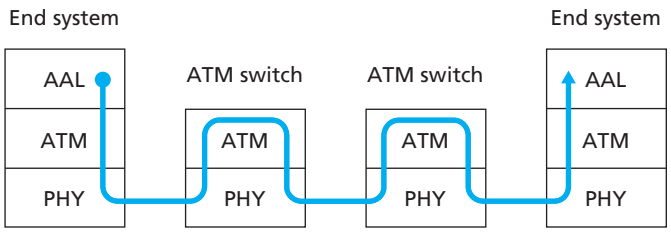

Figure 5.47 ♦ The AAL layer is present only at the edges of the ATM network.

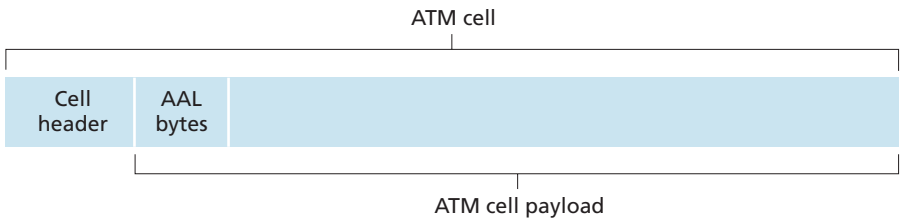

Figure 5.48 ♦ The AAL fields within the ATM payload.

part convergence sublayer (CPCS) PDU in the convergence sublayer. This PDU can have a CPCS header and CPCS trailer. Typically, the CPCS-PDU is much too large to fit into the payload of an ATM cell; thus the CPCS-PDU has to be segmented at the ATM source and reassembled at the ATM destination. The SAR sublayer segments the CPCS-PDU and adds AAL header and trailer bits to form the payloads of the ATM cells. Depending on the AAL types, the AAL and CPCS header and trailers could be empty.

AAL5 (Simple and Efficient Adaptation Layer—SEAL)

AAL5 is a low-overhead AAL that is used to transport IP datagrams over ATM networks. With AAL5, the AAL header and trailer are empty; thus, all 48 bytes of the ATM payload are used to carry segments of the CPCS-PDU. An IP datagram occupies the CPCS-PDU payload, which can be from one to 65,535 bytes. The AAL5 CPCS-PDU is shown in Figure 5.50.

The PAD ensures that the CPCS-PDU is an integer multiple of 48 bytes. The length field identifies the size of the CPCS-PDU payload, so that the PAD can be

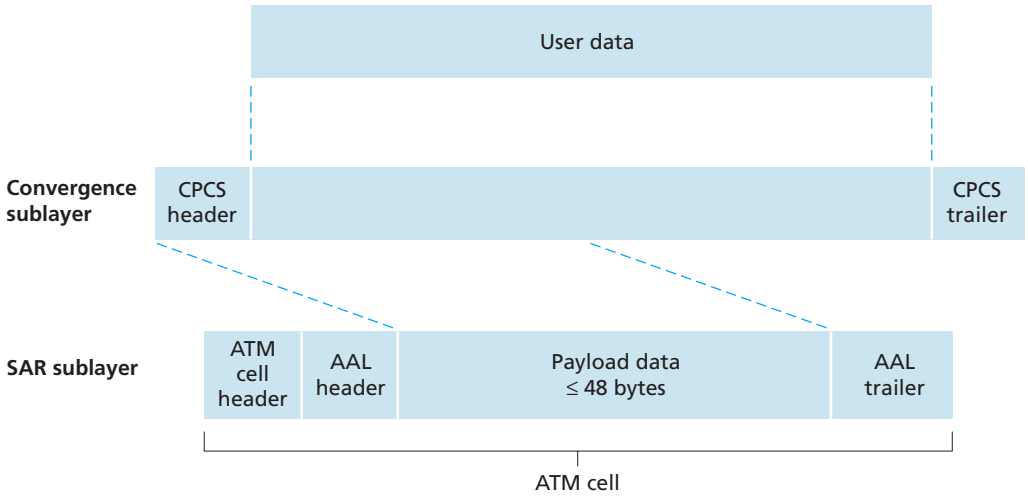

Figure 5.49 ♦ The sublayers of the AAL.

Figure 5.50 ♦ CPCS-PDU for AAL5.

removed at the receiver. The CRC is the same one that is used by Ethernet, token ring, and FDDI. At the ATM source, the AAL5 SAR chops the CPCS-PDU into 48-byte segments. As shown in Figure 5.51, a bit in the PT field of the ATM cell header, which is normally 0, is set to 1 for the last cell of the CPCS-PDU. At the ATM destination, the ATM layer directs cells with a specific VCI to an SAR-sublayer buffer. The ATM cell headers are removed, and the `AAL_indicate` bit is used to delineate the CPCS-PDUs. Once the CPCS-PDU is delineated, it is passed to the AAL convergence sublayer. At the convergence sublayer, the length field is used to extract the CPCS-PDU payload (for example, an IP datagram), which is passed to the higher layer.

5.9.5 IP over ATM

Figure 5.52 shows an ATM backbone with four entry/exit points for Internet IP traffic. Note that each entry/exit point is a router. An ATM backbone can span an entire continent and may have tens or even hundreds of ATM switches. Most ATM backbones have a permanent virtual channel (VC) between each pair of entry/exit points. By using permanent VCs, ATM cells are routed from entry point to exit point without having to establish and tear down VCs dynamically. Permanent VCs, however, are feasible only when the number of entry/exit points is relatively small. For n entry points, $n(n-1)$ permanent VCs are needed to directly connect n entry/exit points.

Each router interface that connects to the ATM network will have two addresses. The router interface will have an IP address, as usual, and the router will have an ATM address, which is essentially a LAN address (see Section 5.4).

Consider now an IP datagram that is to be moved across the ATM backbone in Figure 5.52. Note that to the four IP routers, the backbone appears as a single logical link—ATM interconnects these four routers just as Ethernet can be used to

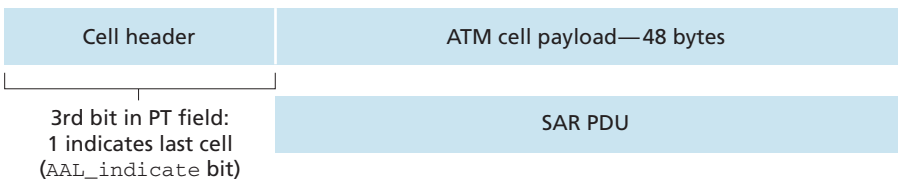

Figure 5.51 ♦ The `AAL_indicate` bit is used to reassemble IP datagrams from ATM cells.

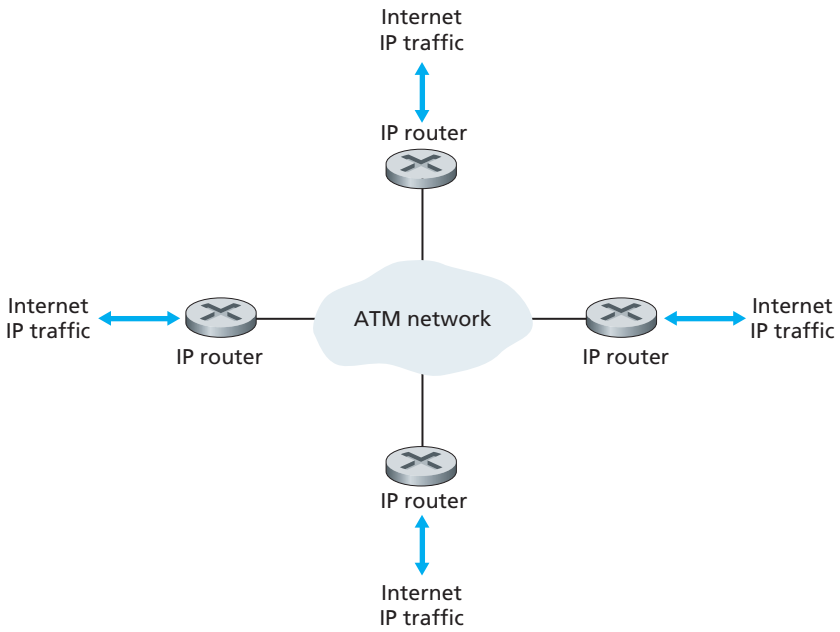

Figure 5.52 ♦ ATM network in the core of an Internet backbone

connect four routers. Let us refer to the router at which the datagram enters the ATM network as the "entry router" and the router at which the datagram leaves the network as the "exit router." The entry router does the following:

1. Examines the destination address of the datagram.
2. Indexes its routing table and determines the IP address of the exit router (that is, the next router in the datagram's route).
3. To get the datagram to the exit router, the entry router views ATM as just another link-layer protocol. To move the datagram to the next router, the physical address of the next-hop router must be determined. Recall from our discussion in Section 5.4, that this is done using ARP. In particular, the entry router indexes an ATM ARP table with the IP address of the exit router and determines the ATM address of the exit router.
4. IP in the entry router then passes down to the link layer (that is, ATM) the datagram along with the ATM address of the exit router.

After these four steps have been completed, the job of moving the datagram to the exit router is out of the hands of IP and in the hands of ATM. ATM must now move the datagram to the ATM destination address obtained in Step 3 above. This task has two subtasks:

- Determine the VCI for the VC that leads to the ATM destination address.
- Segment the datagram into cells at the sending side of the VC (that is, at the entry router), and reassemble the cells into the original datagram at the receiving side of the VC (that is, at the exit router).

The first subtask listed above is straightforward. The interface at the sending side maintains a table that maps ATM addresses to VCIs. Because we are assuming that the VCs are permanent, this table is up to date and static. (If the VCs were not permanent, then an ATM signaling protocol would be needed to establish and tear down the VCs dynamically.) The second task merits careful consideration. One approach is to use IP fragmentation, as discussed in Section 4.4.4. With IP fragmentation, the sending router would first break the original datagram into fragments, with each fragment being no more than 48 bytes, so that the fragment could fit into the payload of the ATM cell. But this fragmentation approach has a big problem—each IP fragment typically has 20 bytes of header, so that an ATM cell carrying a fragment would have 25 bytes of "overhead" and only 28 bytes of useful information. ATM uses AAL5 to provide a more efficient way to segment and reassemble a datagram.

Recall that IP in the entry router passes the datagram down to ATM along with the ATM address of the exit router. ATM in the entry router indexes an ATM table to determine the VCI for the VC that leads to the ATM destination address. AAL5 then creates ATM cells out of the IP datagram:

- The datagram is encapsulated in a CPCS-PDU using the format in Figure 5.51.
- The CPCS-PDU is chopped up into 48-byte chunks. Each chunk is placed in the payload field of an ATM cell.
- All of the cells except for the last cell have the third bit of the PT field set to 0. The last cell has the bit set to 1.

AAL5 then passes the cells to the ATM layer. ATM sets the VCI and CLP fields and passes each cell to the TC sublayer. For each cell, the TC sublayer calculates the HEC and inserts it in the HEC field. The TC sublayer then inserts the bits of the cells into the PMD sublayer.

The ATM network then moves each cell across the network to the ATM destination address. At each ATM switch between the ATM source and the ATM destination, the ATM cell is processed by the ATM physical and ATM layers, but not by the AAL layer. At each switch the VCI is typically translated (see Section 1.3) and the HEC is recalculated. When the cells arrive at the ATM destination address, they are directed to an AAL buffer that has been put aside for the particular VC. The CPCS-PDU is reconstructed using the `AAL_indicate` bit to determine which cell is the last cell of the CPCS-PDU. Finally, the IP datagram is extracted out of the CPCS-PDU and is passed up the protocol stack to the IP layer.