Chapter 3

TCP over ATM

Currently, in the computer communication area, the TCP/IP (transmission control protocol/internet protocol) is one of the most widely used protocol suites which interconnects a considerable number of workstations, terminals, hosts, and even PCs. TCP/IP is placed on top of ATM and ATM supports this protocol by way of so called "best-effort service". This service is supposed to replace the internet with the higher transmission speed in an ATM-based environment. Thus, ATM-based networks must also be capable of carrying and supporting the TCP/IP protocol.

The basic architecture of TCP/IP over ATM is shown in Figure 3.1. IP is a connectionless

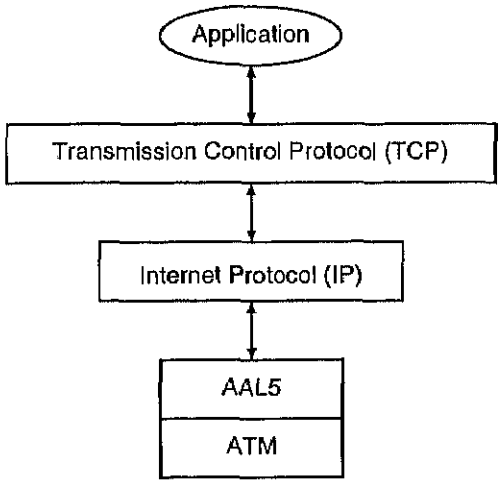

Figure 3.1: TCP/IP over ATM architecture

protocol technique designed for data applications, while TCP is a connection-oriented reliable transport protocol. IP can be run over various transmission systems such as Ethernet, token ring, and leased lines. The IP packets are transferred in a best-effort manner, which means no error correction capabilities and no information about successful delivery are provided. They are transported in an unassured manner. In other words, there is no guarantee that they will arrive in the same order as they went out. Instead, the higher layer protocols, such as TCP,

have to control data integrity as well as network congestion. With respect to congestion control algorithms, IP is a transparent layer for TCP over ATM. Hence, only the congestion control mechanisms at the TCP land ATM layer are described in this dissertation.

The author has already introduced some features of TCP and ATM with regard to congestion control or flow control methods. Here, this author will describe slow start and congestion avoidance, fast retransmit and recovery of TCP, and the ABR and UBR services of ATM.

3.1 TCP congestion mechanisms

For supporting TCP over ATM, one of the most important parts is the congestion control mechanism. The TCP protocol suite contains various flow control algorithms. It provides reliable data transfer using a window-based end-to-end flow control and error control algorithm. This includes slow start, congestion avoidance, fast retransmit, and fast recovery [STEV97][JACO92].

3.1.1 Slow start and congestion avoidance

TCP uses a window based end-to-end protocol for flow control to limit the number of packets in the network. For this purpose, two windows are implemented. The receiver maintains a receiver's window (RCVWND) as a measure of the receiver's buffering capacity. At the other end, the sender maintains a variable congestion window (CWND) as a measure of the capacity of the network at a specific time. The sender cannot transmit more data than the minimum of RCVWND and CWND into the network at a time if the transmitted packets are not acknowledged.

Slow start and *congestion avoidance* is the basic TCP congestion control scheme. When a new connection is set up or an already-established connection is re-set, CWND is initialized to one segment and is increased by one segment whenever a new acknowledgment (ACK) from the receiver arrives at the sender. In this way, the CWND doubles every round trip time (RTT) until it reaches a maximum value (typically 65535 octets). This is called *slow start*, although it is actually not slow start, because the CWND increases exponentially during this phase.

However, due to the difference of the capacity of pipes in the network (e.g., a big pipe for a fast LAN and a smaller pipe for a slower WAN), congestion might occur. Because of this congestion, a data segment can be lost in the network. If a segment is known to be lost, the receiver sends duplicate ACKs on the receipt of the subsequent segments. In addition, the sender maintains a retransmission timer for the last unacknowledged packet[1]. If the timer expires, it is assumed that the network is suffering congestion. In this case, the TCP sender must slow down its transmission rate of segments into the network in order to regulate the load. When a sender has been notified

[1]Two terms, "packet" and "segment", are used interchangeably here.

that a packet is lost, the sender saves half of the current CWND value in a variable slow start threshold (SSTHRESH). In addition, the CWND is newly set to one segment and the sender retransmits the segments from the first unacknowledged one. Then the slow start begins again and the CWND is increased by one on the receipt of each new ACK until it reaches SSTHRESH. On reaching the SSTHRESH, the CWND is increased by $(1/current\_CWND)$ on every arrival of ACK. Thus, CWND is increased one segment approximately every round trip time, which results in a linear increase of the CWND. This is called the *congestion avoidance* phase. The slow-start phase and the congestion avoidance phase are distinguished by the SSTHRESH, which is initialized to 64K octets.

Figure 3.2 shows the slow start and congestion avoidance stages for a typical TCP connection. In the first SS (slow start phase), the CWND is increased exponentially. While an ACK is not

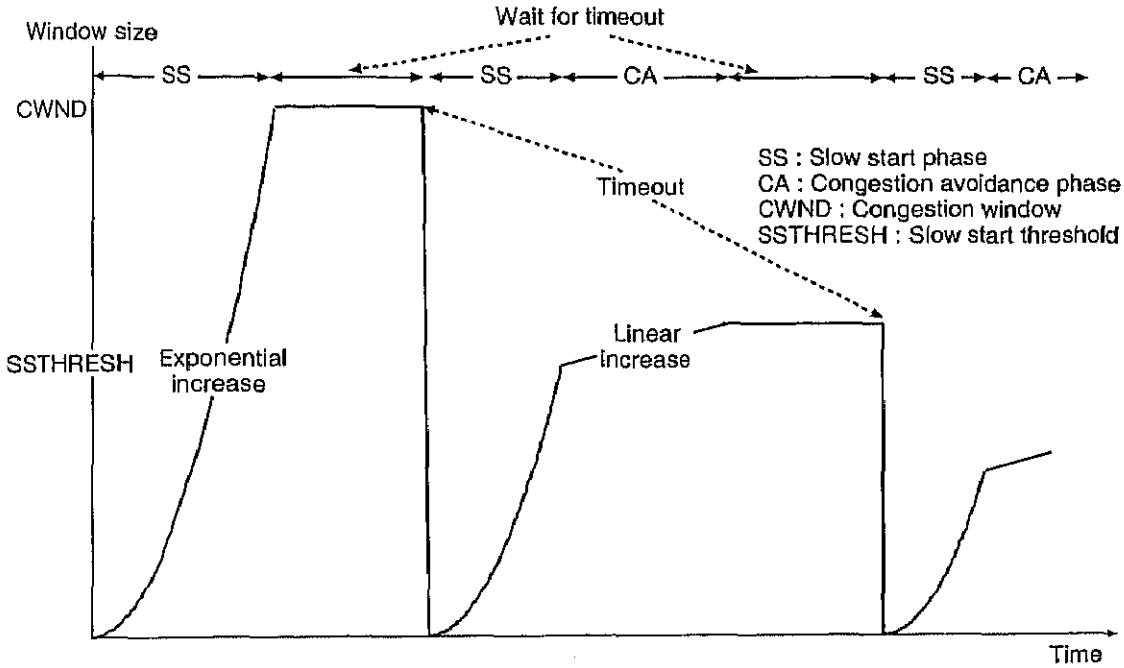

Figure 3.2: TCP window behavior of slow start and congestion avoidance

returned, the TCP sender waits for a timeout. When a timeout occurs, half of the current CWND value is set into SSTHRESH. Then, the CWND is newly set to one segment and a new SS begins. After the CWND reaches the SSTHRESH, the CA (congestion avoidance phase) is started, increasing CWND linearly. This procedure repeats while the connection is established.

3.1.2 Fast retransmit and fast recovery

The retransmission timer calculation is based on the average and mean deviation of the RTT, which is estimated by measuring the time between the transmission of a segment and the receipt of an ACK for the corresponding segment. On the other hand, most TCP implementations

basically use a coarse timer granularity (typically 500 ms) for triggering the retransmission timeout. Hence, the TCP connection may waste considerable time waiting for the timeout when it experiences a segment loss. During this waiting time, the TCP neither sends new packets nor retransmits the lost packets. Moreover, once the timeout occurs, the TCP connection will enter the slow start stage, setting the CWND to one segment. This means that the connection will take a long time to fully utilize the network links again. As a result, the links remain idle for a long time and experience low throughput.

Fast retransmit and *fast recovery* are proposed to alleviate the penalty of a packet loss. The receiver TCP may generate a duplicate ACK immediately when an out-of-order segment is received. This duplicate ACK should not be delayed to notify the sender that a segment was received out of order, which might be a packet loss. If three or more duplicate ACKs are received by the sender, it assumes that the corresponding segment is lost. The TCP then retransmits the lost segment immediately without waiting for a retransmission timer to expire. This is called *fast retransmit*.

The receipt of the duplicate ACKs means that not only has a packet been lost, it also means that there is still data flowing between the sender and the receiver. This is because the receiver can generate the duplicate ACK only when it receives another segment, which comes to be stored in the receiver's buffer. Therefore, the TCP does not need to regulate the flow abruptly by shifting to the slow start phase. Hence, after fast retransmit sends the lost segment, the TCP enters the congestion avoidance phase, not the slow start phase. This is the fast recovery algorithm.

The fast retransmit and fast recovery algorithms are usually implemented together[2] as shown in Figure 3.3. When the third duplicate ACK is received, SSTHRESH is set to half the current CWND ($SSTHRESH \leftarrow CWND_i/2$) but no less than two segments and the missing segment is retransmitted immediately (① in Figure 3.3). The new CWND is set to SSTHRESH plus three segments size, which corresponds to the increment of the three duplicate ACKs, because they are generated based on three other segments which are correctly received and cached in the receiver's buffer. Then, for each subsequent duplicate ACK of other segments, CWND is increased by one segment size and tries to transmit a new packet if it is allowed to by the new CWND value. This is for the additional segment that has already left the network. The effect of these actions is that the sender keeps the transmission rate at least half of its original rate when it performs fast retransmit. When the next ACK for a new data segment arrives, the CWND is set to equal the value of SSTHRESH directly instead of setting the CWND to one segment and doing the slow start again until CWND reaches SSTHRESH (② in Figure 3.3). After approximately one round trip time (RTT) following the retransmission of the lost packet, the ACK should be the acknowledgment of the retransmission of the lost packet. Then congestion avoidance is performed. This is called *fast recovery*. By using fast recovery, network

[2]Thus they are often called the FRR algorithm.

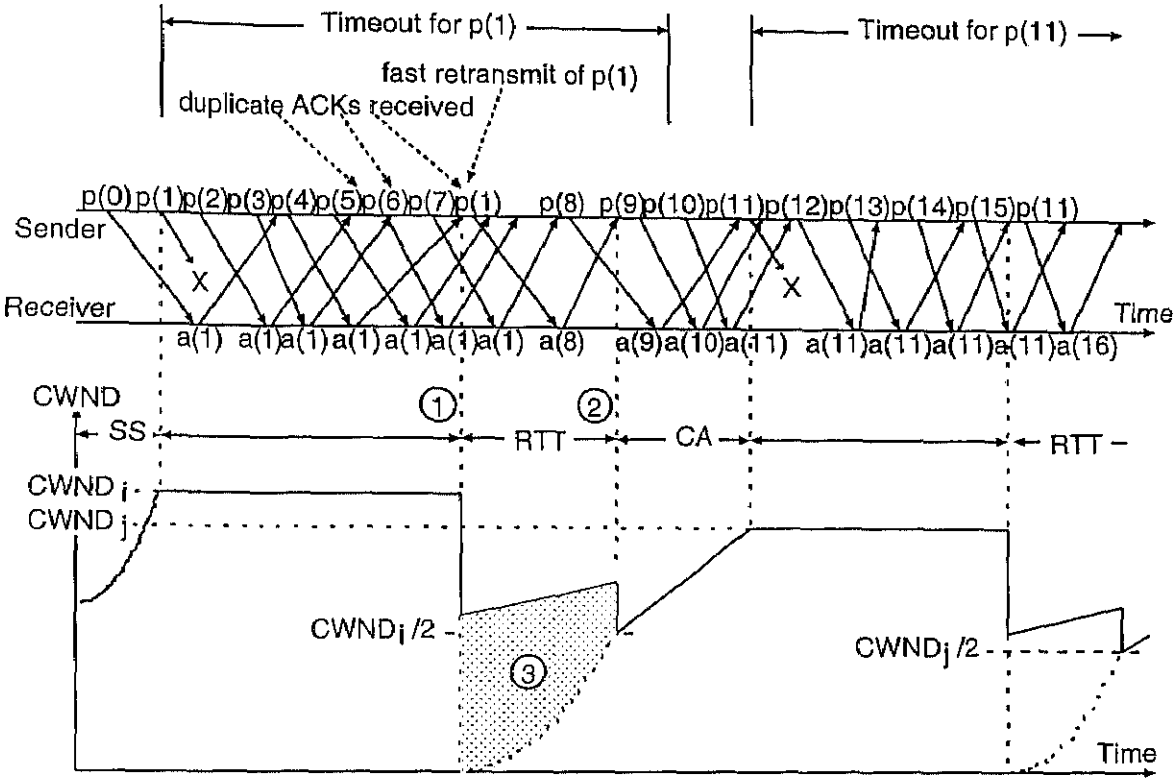

Figure 3.3: Behavior of fast retransmission and recovery

transmission rate is improved by an amount of ③ in Figure 3.3.

The slow start, congestion avoidance and fast retransmit algorithms first appeared in the 4.3BSD Tahoe release. The fast recovery algorithm is appended to it first in the 4.3BSD Reno release.

So far, the main features of TCP congestion control techniques have been introduced. Various congestion control schemes of ATM to support TCP are described in the following section.

3.2 Congestion control of ATM

ATM supports various quality of service (QoS), which include constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), and unspecified bit rate (UBR). Among these service categories, best-effort service using ABR or UBR are deemed to be suitable to support TCP over ATM.

3.2.1 Congestion control of ABR

Having been proposed by the ATM Forum, ABR traffic gives the network the opportunity to offer guaranteed performance to high priority traffic, and divide the remaining network resources among ABR connections. In order to support ABR services, the ABR specifically requires a

feedback mechanism to regulate each source. A number of schemes concerning the feedback mechanism have been proposed. Two main schemes for congestion control with ABR are the rate-based and the credit-based algorithms.

Credit-based scheme

The credit-based scheme is a link-by-link (or hop-by-hop) per-VC flow control [KUNG95] which is an implementation of the *isarithmic method* described in [BERT92]. The credit-based scheme works as follows (Figure 3.4). Before forwarding any cell over a link, the sender needs to possess

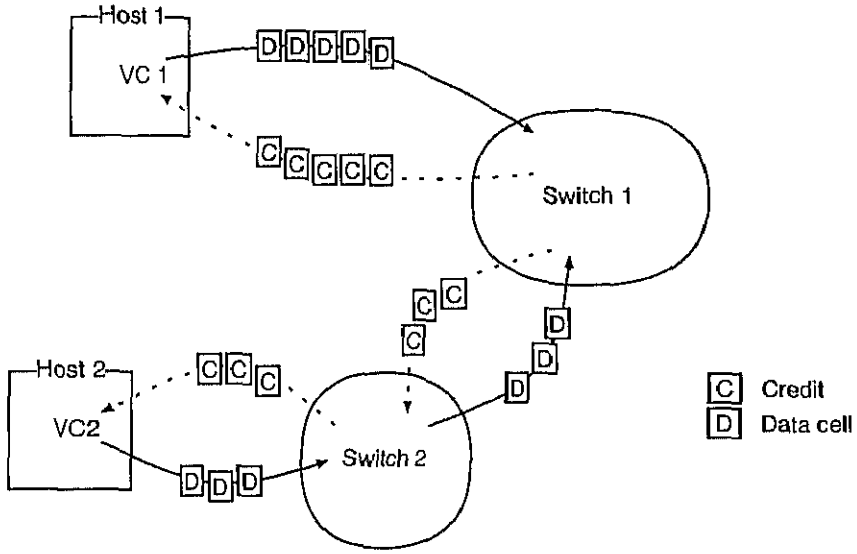

Figure 3.4: Credit-based flow control

credits for the VC from the receiver. These credits are sent from the receiver to the sender at various times indicating the availability of buffer space for receiving data cells from the VC. The sender is only eligible to forward data cells amount of received credits to the receiver when it has received enough credits. Each time the sender forwards a data cell of a VC, it decrements its current credit balance for the VC by one. Similarly, when a node (or switch) forwards one cell to another node of its downstream link, it increments the credit balance for that VC by one. Thus, for a given virtual channel, each link maintains its own control loop independent of the other links on the virtual channel connection. This may be viewed as a particular form of window flow control where there is a single global window for the entire network. The rationale of the scheme is to regulate the total number of cells in the network by using credits circulating in the subnet to control congestion. Owing to this mechanism, transient congestion can be relieved effectively. Ideally, in addition, no cell loss occurs. However, as each switch should maintain complicated queue management for every connection, the credit-based control scheme is thought to be quite complex.

Rate-based scheme

On the other hand, a rate-based algorithm, which was finally voted for in the ATM Forum in late 1994, controls the cell emission rate of each connection. This type of control scheme is very intelligent. However, the more intelligent the scheme has evolved to be, the more complex it has become. Here, merely the basics of the rate-based scheme are described. After that, specific rules on the ABR rate-based scheme by the ATM Forum will be briefly described.

In order to control the source transmission rate, the ATM switch is required to constantly monitor the current load to calculate the allowable rates for the sources, which are fed back to the sender to dynamically adjust the input into the network. The switch takes advantage of **resource management** (RM) cells for this feedback information. The structure of an RM cell will be shown later. The RM cells are generated by the source and inserted into the network periodically or under some exceptional condition. They are forwarded from the source to the destination (**forward RM cells** (FRM cells)) periodically and returned from the receiver to the source (**backward RM cells** (BRM cells)) allowing the switches along the route to modify the contents of the RM cells depending on the congestion state of the network. Based on the feedback information, the sender should take appropriate action to regulate its transmission rate. Figure 3.5 shows a simple introduction of the closed-loop feedback control scheme for unidirectional flow.

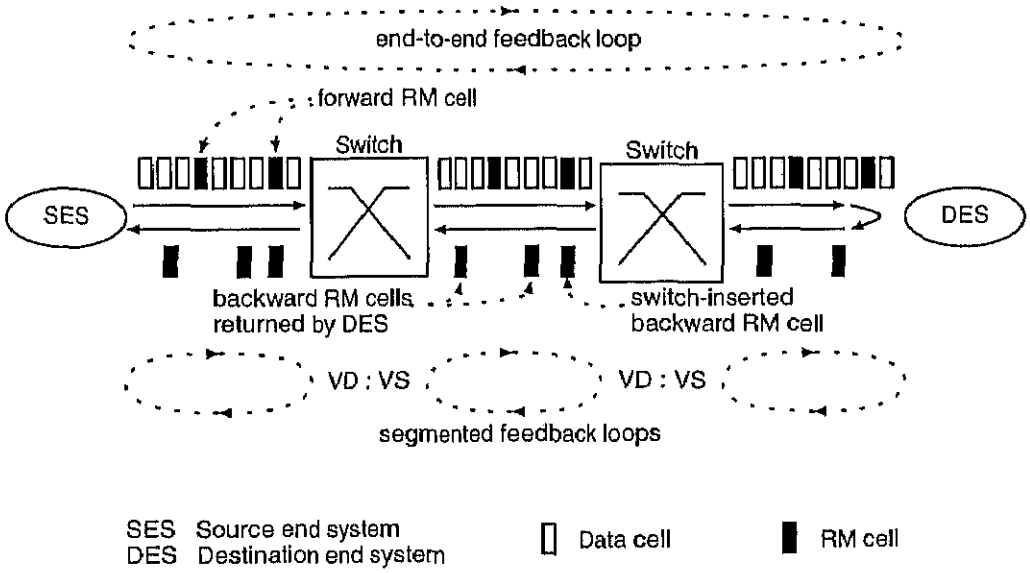

Figure 3.5: Simple introduction of a close-loop control of ABR

Provided the control loop is long, it takes a long time for a feedback to be returned in these types of schemes. Thus, it is not always an up-to-date information and it should result in poor performance. Therefore, in order to reduce the length of the control loop, leading to performance improvements, and to create separate control domains with respect to administrative reasons, the

virtual source/virtual destination (VS/VD) feature is optionally implemented to segment the ABR control loop into smaller loops (Figure 3.5). In a VS/VD network, the intermediate switching elements additionally behave both as a source end system and as a destination end system. As a destination end system, it turns back the RM cells to the sources from one segment. As a source end system, it generates RM cells for the next segment. This feature can allow feedback from nearby switches to reach sources faster, and allow hop-by-hop control as well.

Generally, there are three types of feedback methods for switches:

- EFCI bit: Each data cell header contains a single-bit indicator called an **EFCI** (explicit forward congestion indication) bit in the payload type indicator (PTI) field. The switch indicates the network congestion state to the source by using this bit.
- Relative rate marking: RM cells have two bits in their payload. One is called the "congestion indication" (**CI**) bit and the other is called the "no increase" (**NI**) bit. They can be set by congested switches to be informed to the source.
- Explicit rate marking: The RM cells also have another field in their payload called the "explicit rate" (**ER**)" that can be reduced by congested switches to any desired value. It is applied to the transmission rate of the source.

Since the appearance of congestion control schemes, the schemes have been greatly evolved. The most essential part of the scheme is described in next section.

EFCI

The first generation ATM switches were implemented prior to the specification of RM cells. Thus, they do not deal with RM cell based algorithms. To indicate congestion, these switches use a congestion bit called the "EFCI bit" in the payload type indicator (PTI) field of the ATM cell header, which is set before the cell arrives at the destination. In this case, a switch uses a well-defined threshold in its buffer occupancy to determine when to activate the EFCI bit. When the queue length in its buffer exceeds the threshold, the node activates the EFCI to indicate that the cell on this ATM connection has encountered congested resources. In addition, the set EFCI bit notifies the user that congestion avoidance procedures should be initiated for traffic in the same direction as the received cell. The destination node monitors the EFCI bit of the arrived cell. If the bit is set, the destination application protocol is invoked to communicate to the appropriate counterpart to adaptively lower the cell rate of the connection (Figure 3.6). The originating end node then may slow down its transmission rate for that connection, selectively discard cells, or a combination of both. In either case, the source causing the congestion is throttled to recover from the congestion state. This method can work when the congestion interval is substantially longer than the round-trip delay. Otherwise the congestion will be

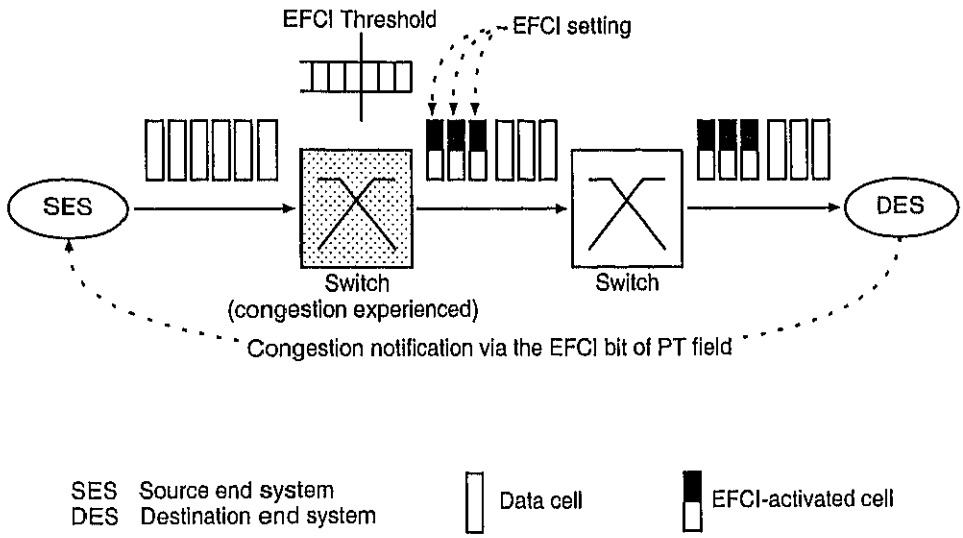

Figure 3.6: Enforcement of the EFCI

relieved by the time the feedback control functions. The worst-case scenario for such a feedback scheme would be that the input is generated periodically, with the period approximately equal to the round-trip time, because all reactions taken based on the feedback information might not be up-to-date and thus, not appropriate to solve the network congestion. However, for major network trunk and/or nodal failures, this method is expected to be an effective technique to control the input rate of different sources by fair manner.

Relative rate marking

After that, the ATM Forum has proposed traffic management models for ABR which use RM cells to handle the transmission rate. As a forerunner of the rate marking algorithm, Yin and Hluchyj's scheme is simply introduced here [YIN94]. This control mechanism also uses the EFCI bit. The destination node of the connection checks the EFCI bit of the data cell. If it is not set, the destination node feeds a control cell[3] back to the source containing permission in order to increase the transmission rate of the source by a fixed increment. If the source does not receive any permission to increase its rate over the same interval, it decreases its allowed rate by an amount proportional to its current rate. Hence, the transmission rate is increased linearly and decreased exponentially. The allowed rate of a connection is adapted between a minimum and maximum value. The interpretation and reaction of the EFCI bit are different from scheme to scheme. The scheme proposed by Yin and Hluchyj interprets a set EFCI in a single data cell as sufficient grounds for denying positive feedback, while another scheme proposed in [RAMA90] performs the increase and decrease of the rates depending on an average of the bits. In addition,

[3]In their paper[YIN94], Yin and Hluchyj did not mention an RM cell. However, the RM cell is the very implementation they meant by "control cell".

the Yin and Hluchyj scheme interprets all explicit feedback messages from the network in the backward direction as positive feedback (which means an increment of the rate) and interprets the absence of explicit feedback messages as negative feedback (which means a decrement of the rate). This makes the scheme robust to lost or delayed feedback, which unfortunately results in consumption of bandwidth for feedback cells when the network is not congested and there is no need to regulate the traffic [BONO95b].

Let us now examine a new form of relative rate marking scheme based on the precursors using the RM cell. Before describing the RM cell-based control scheme, note the complete format of the RM cell in Figure 3.7. The payload type indicator (PTI) in the ATM cell header is binary

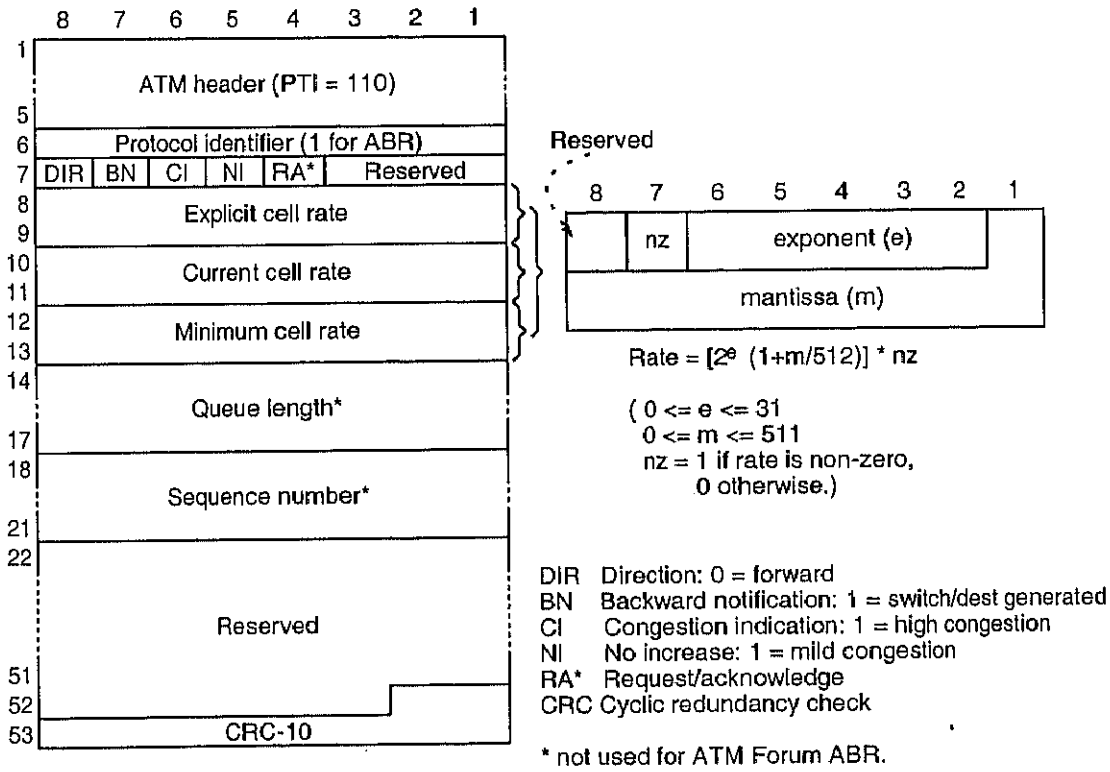

Figure 3.7: Resource management cell fields

110 to indicate that the cell is an RM cell. Moreover, the CLP bit in the cell header is 0 or 1 depending on the characteristics of the RM cell[4]. One octet of the protocol identifier field is set to one for ABR connections. The Direction (DIR) bit distinguishes the direction of the RM cell. For forward RM cells, it is 0. The backward notification (BN) field is set when the RM cell is generated by a switch. This is to distinguish the ordinary RM cell generated by a source and a switch-generated RM cell for backward explicit congestion notification (BECN). CI, NI, and ER (explicit cell rate) fields are used by the network to give feedback to the sources. The current cell rate (CCR) is used by the source to indicate its current rate to

[4]This is determined relative to whether it is "in-rate" or "out-of-rate", which will be described later.

the network. All rates (ER, CCR and MCR) in the RM cell are presented using a special 16-bit floating point format using 9 bits of mantissa and 5 bits of exponent. The maximum value is 4,290,772,992 cells per second (1.8 terabits per second)[5]. During the connection setup, however, the rate parameters are negotiated using an 24-bit integer format, which limits their maximum value to 16,777,215 cells per second or 7.1 Gbit/s[6][JAIN96]. 10-bit of CRC is computed over the content of the RM cell payload excluding the CRC field (374 bits) by the generator polynomial $1+x+x^4+x^5+x^9+x^{10}$. RA (request/acknowledgment), QL (queue length), and SN (sequence number) fields are not used for ATM Forum ABR.

The algorithm named "relative rate marking" uses the CI and NI fields of the RM cell in cooperation with the EFCI indicator. The CI (congestion indication) field is used to force the source to decrease its allowed cell rate (ACR) by a predefined amount or manner. The source initially sets the CI bit to $CI = 0$ when it inserts the RM cell. The destination sets $CI = 1$ in the BRM cell to indicate that a data cell of $EFCI = 1$ was previously received. For each RM cell returned to the source, the source adjusts its new ACR based on the information of the RM cell. If the congestion indication bit is set ($CI = 1$), the source must decrease its ACR by a proportion of its current value, called the "rate decrease factor" (RDF), but the ACR must never be less than the MCR. On the other hand, if the CI bit is not set ($CI = 0$), the source can increase its ACR by a fixed increment, $RIF * PCR$, where RIF is the "rate increase factor". This time, the new ACR can never exceed the PCR. The NI (no increase) field is used to notify the source to not increase its ACR. Unlike the CI bit, $NI = 1$ does not require any decrease in the ACR. It is typically used when a switch senses impending or mild congestion. Then, the source is informed to observe the CI and ER fields in the RM cell and to not increase its ACR above its current value.

Explicit rate marking

In contrast to the relative rate marking algorithms, the switches can explicitly indicate the rates that they can support by setting the ER field of the RM cell to any desired value. These switches are called "explicit rate marking switches".

As can be seen from Figure 3.7, the RM cell contains the current cell rate (CCR) field and the explicit cell rate (ER) field. The CCR is the cell rate at which a source is transmitting. The ER is the rate at which a switch on the path will allow the source to transmit. Any switch that receives an FRM cell may set the EFCI bit but not the ER yet. When the FRM cell reaches the destination, the CCR and the ER fields of the cell are examined to determine whether the destination can support the source rate. If the destination cannot support the rate that the source wants, it modifies the ER field to a particular value it can support. Then, it modifies

[5] As $0 \leq e \leq 31$, and $0 \leq m \leq 511$, the maximum rate value is $2^{31} \times (1 + 511/512) \times 1 = 4290772992\ cell/s = 1819287748608\ bit/s$.

[6] $2^{24} - 1 = 16777215\ cell/s = 7113539160\ bit/s$.

the direction (DIR) bit of the RM cell to 1 and the RM cell is turned around to become a BRM (backward RM) cell. As the cell passes in the backward direction to the source, each switch examines the ER field and determines if it can support the rate. If it cannot, it reduces the value of the ER field in the RM cell to whatever value it can support. Otherwise, it sends the cell to the next switch upstream keeping that field unchanged. Notice that as the RM cell travels toward the source, if a smaller ER has already been inserted by a downstream switch, the current switch does nothing to the BRM cell, because the higher rate of the switch may be beyond the capacity of a downstream switch. When the BRM cell arrives at the source, the source is expected to set its new CCR to the received ER, provided that the ER is not less than the minimum cell rate (MCR).

The most essential part of rate-based algorithms have been described by now. The next section explains the specific rules of the ABR rate-based scheme suggested by the ATM Forum [TM4.1].

ABR flow control: the ATM Forum traffic management specification 4.1

The ATM Forum has proposed the Traffic Management Specification 4.1 (TM 4.1) for ABR flow control. The rules concerning the ABR flow control are mentioned in this section as a summary of rate-based ABR control schemes.

When a connection is set up, the parameters defining the characteristics of the connection are negotiated between a user and the network. The parameters for ABR connections are defined in Table 3.1 [TM4.1]. CRM and ICR are computed or updated when the call setup is completed and the FRTT and the other parameters are known. CRM is computed as $CRM = \lceil \frac{TBE}{Nrm} \rceil$, where $\lceil x \rceil$ is the smallest integer greater than or equal to x. ICR is updated after call setup is complete to insure TBE compliance as $ICR = max(MCR, min(ICR, \frac{TBE}{FRTT}))$.

Based on those parameters, the ABR flow is controlled. A number of rules are specified by the TM 4.1 concerning the behavior of the source, the destination, the switch, and so on. Here, those of the source, the destination, and the switch are described. The source rules are as follows:

1. A source should always transmit at a rate equal to or below its ACR. In addition, the value of ACR should not exceed the PCR, nor should it ever be less than the MCR. This means, $MCR \leq ACR \leq PCR$ and $Sourcce\ Rate \leq ACR$.

2. Before a source sends the first cell after the connection setup, it sets the ACR to, at most, the ICR. Thus, a source starts transmission at the ICR. The first cell is always an in-rate forward RM cell, which enables the source to receive network feedback as soon as possible.

3. Then, the source has three types of cells to send: data cells, forward RM cells, and backward RM cells for reverse flow.

Table 3.1: Parameter descriptions

Label	Description	Units and range	Default
PCR	Peak cell rate: the cell rate that the source may never exceed.	Cell/sec *	**
MCR	Minimum cell rate: the rate at which the source is always allowed to send.	Cell/sec *	0
ICR	Initial cell rate: the rate at which a source should send initially and after an idle period.	Cell/sec *	PCR
RIF	Rate increase factor: controls the amount by which the cell transmission rate may increase upon receipt of an RM-cell	Power of two, ranging from 1/32768 to 1.	1
Nrm	The maximum number of cells a source may send for each forward RM-cell.	Power of two, ranging from 2 to 256.	32
Mrm	Controls the allocation of bandwidth between FRM-cells, BRM-cells, and data cells	A constant fixed at 2	
RDF	Rate decrease factor: controls the decrease in the cell transmission rate.	Power of 2, ranging from 1/32768 to 1	1/32768
ACR	Allowed cell rate: the current rate at which a source is allowed to send.	Cell/sec	
CRM	Missing RM-cell count: limits the number of FRM cells that may be sent in the absence of received BRM cells.	Integer, implementation-specific	
ADTF	ACR decrease time factor: the time permitted between sending RM cells before the rate is decreased to ICR.	Second, ranging from .01 to 10.23 sec with granularity of 10 ms.	0.5
Trm	Provides an upper bound on the time between FRM for an active source.	Millisecond, 100 times a power of two, from 100×2^{-7} to 100×2^{0}	100
FRTT	Fixed round-trip time: the sum of the fixed and propagation delays from the source to a destination and back.	Microsecond, ranging from 0 to 16.7 seconds	***
TBE	transient buffer exposure: the negotiated number of cells that the network would like to limit the source to sending during startup periods, before the first RM cell returns.	Cell, ranging from 0 to 16,777,215	16777215
CDF	Cutoff decrease factor: controls the decrease in ACR associated with CRM.	Zero, or a power of two ranging from 1/64 to 1.	1/16
TCR	Tagged cell rate: limits the rate at which a source may send out-of-rate FRM cells.	A constant fixed at 10 cell/sec.	

*: Rates are signaled as 24 bit integers that have a minimum value of zero, and a maximum value of 16,777,215. However, RM cells use a 16 bit floating point format that has a maximum value of 4,290,772,992.
**: It should be determined mandatorily at call setup time.
***: FRTT should be set by the source to the fixed delay. It is then accumulated during the call setup. FRTT is used to determine other parameters. It should be the sum of all the RM cell fixed delays and propagation delays in the round trip path.

(a) The source sends a forward RM cell if and only if the last in-rate forward RM cell was sent and either (i) or (ii).

i. At least Mrm in-rate cells have been sent and at least Trm time has elapsed.

ii. $Nrm - 1$ in-rate cells have been sent.

The source is required to send an FRM after every 31 (default of $Nrm - 1$) cells. When the source rate is low it takes a long time to send another RM cell. Thus, a source can send an FRM cell if more than 100 ms (default of Trm) has elapsed since the transmission of the last FRM. However, if the source rate is very low and there are not many cells to be sent, it is needless to send an FRM every period of 100 ms, which would use up the available bandwidth. Hence, there must be at least Mrm cells between FRMs.

(b) The backward RM cell should not be unnecessarily delayed. Therefore, the next in-rate cell is a BRM if the above condition (a) is not met, if a backward RM cell is waiting for transmission, and if either (i) or (ii).

- No in-rate BRM cell has been sent since the last in-rate FRM cell.
- There are no data cells to send.

(c) The next in-rate cell can be a data cell if neither the above condition (a) nor condition (b) is met.

4. All RM cells sent in accordance with rules 1, 2, and 3 are in-rate RM cells, so they have $CLP = 0$. On the other hand, a source can send a limited number of out-of-rate RM cells if necessary, which have $CLP = 1$.

5. The duration for which an allowed source rate is valid is approximately 500 ms (the default value of $ADTF$). If a source does not send any RM cells for this duration, an ACR higher than the ICR is not allowed. Thus, based on the *use-it-or-lose-it* policy, the ACR is reduced to the ICR to allocate a new cell rate.

6. If a network link is broken or is highly congested, the transfer of an RM cell cannot be guaranteed. In such cases, the sources are required to reduce their rates if the network feedback is not received for a certain period of time. Therefore, if at least the CRM in-rate forward RM cells have been sent since the last backward RM cell with $BN = 0$ was received, then the ACR is reduced by at least $ACR * CDF$ as long as the new ACR is no less than the MCR. Here, the CDF is negotiated at the connection setup time, and the CRM is computed as described above. This rule can be disabled by setting the CDF to 0.

7. After the above behaviors 5 and 6, the sources should place their ACR in the CCR field of the outgoing forward RM cell.

8. When the CI bit of a backward RM cell (in-rate or out-of-rate) is set, it means congestion. Thus, the ACR is reduced by at least $ACR * RDF$, unless that reduction would result in a rate below the MCR, in which case the ACR is set to the MCR. If both $CI = 0$ and $NI = 0$, then the ACR may be increased by no more than $RIF * PCR$, but cannot exceed the PCR. In mild congestion, $NI = 1$, the ACR should not be increased.

9. After the ACR is adjusted according to rule 8, the ACR is again set to the minimum of the ACR from rule 8 and the ER field, but no lower than the MCR. These two rules (7 and 8) are summarized as follows:

NI	CI	Action
0	0	$ACR \leftarrow Min(ER, ACR + RIF \times PCR, PCR)$
0	1	$ACR \leftarrow Min(ER, ACR - ACR \times RDF)$
1	0	$ACR \leftarrow Min(ER, ACR)$
1	1	$ACR \leftarrow Min(ER, ACR - ACR \times RDF)$

$$ACR \leftarrow Max(ACR, MCR)$$

10. Each field of a forward RM cell is initialized as follows. Please refer to Figure 3.7. For virtual path connections (VPCs), the virtual channel identifier (VCI) is set to 6. For virtual channel connections (VCCs), the VCI of the connection is used. In either case, the payload type indicator (PTI) is 6 (110). The protocol identifier of the RM cell is 1. The direction bit is unset for forward RM cells. The backward notification (BN) bit is unset because the RM cell is generated by a source. The explicit rate field is set to the maximum rate that the source can support, which cannot exceed the PCR, though. The current cell rate is set to the current ACR. The minimum cell rate is set to the negotiated value. All reserved octets are set to 6A (hex) or 01101010 (binary). Other reserved bits are set to 0. The sources can also set the ER and NI fields to indicate their own congestion.

11. Forward RM cells may be sent out-of-rate (i.e., not conforming to the current ACR). The out-of-rate FRM cells should not be sent at a rate greater than a PCR whose default value is 10 cells per second.

12. A source should reset EFCI on every data cell sent.

13. The source may implement a *use-it-or-lose-it* policy to reduce its ACR to a value that approximates the actual cell transmission rate.

In the calculation of the cell rate, in-rate forward and backward RM cells are included in the source rate of a connection. In addition, the source has to deal with its local congestion in a fair manner.

The destination follows the rules described below:

1. When a data cell is received, the destination monitors and saves the EFCI bit as the EFCI state of the connection.

2. On receiving forward RM cells, destinations should return the FRM cells to their sources. The DIR bit is changed to 1 to indicate "backward". The BN bit is set to 0 to indicate that the cell was not a switch-generated RM cell. The CI bit in the next BRM is set to 1 if the saved EFCI state is set, and then the stored EFCI state is cleared. Provided the destination has internal congestion, it may reduce ER to the rate it can support and/or set the CI or NI bits. This rule is used in the VS/VD configuration where the virtual destination is bottlenecked in the next segment. In any case, the ER should never be increased. The octets and bits defined as reserved may be set to 6A (hexadecimal) and zero, respectively, or left unchanged. All other values (PTI, protocol identifier, CCR, MCR) remain unchanged.

3. The destination is required to return RM cells as quickly as possible. Due to the low reverse ACR, however, if a forward RM cell is received while another turned-around RM cell on the same connection is scheduled for in-rate transmission, it indicates either:

   - The contents of the old cell may not be up-to-date. Thus, it is recommended that the contents of the old cell are overwritten by the contents of the new cell.
   - There is not much need to send the old cell even though it has been over-written. Thus, it is recommended that the cell may be sent out-of-rate setting the CLP bit to 1, which will allow it to be selectively dropped by the switch if congestion is experienced. Alternatively, the old cell may be discarded or remain scheduled for in-rate transmission.
   - The new cell is required to be scheduled for in-rate transmission.

4. In any case in Rule 3 above, the contents of an older cell must not be transmitted after the transmission of the contents of a newer cell.

5. Provided a destination is severely congested, it may generate a backward RM cell without having received a forward RM cell to reduce the transmission rate of a source. The BN and DIR is set to 1 to indicate "backward notification" and "backward direction", respectively. To inform the source of its congestion, it may set either CI=1 or NI=1. The other fields of the RM cell are assigned appropriately. The rate of these BRM cells (including both in-rate and out-of-rate) is limited to 10 cells/second, per connection.

6. A forward RM-cell with CLP=1 may be turned around either in-rate (with CLP=0) or out-of-rate (with CLP=1).

In the above description, "turn around" means a destination originated process of transmitting a backward RM cell in response to the reception of the corresponding forward RM cell. The RM cells are required to be returned as quickly as possible. Therefore, in order to minimize the turn-around delay, it is recommended that as many RM cells (in-rate and/or out-of-rate) as possible be inserted from the destination to the source.

The switch behavior is as follows:

1. A switch may include at least one of the following congestion control functions:
   (a) EFCI marking: The switch may set the EFCI bit (in PTI field) in the data cell headers.
   (b) Relative Rate Marking: The switch may set CI=1 or NI=1 in FRM and/or BRM cells.
   (c) Explicit Rate Marking: The switch may reduce the ER field in FRM and/or BRM cells.
   (d) VS/VD Control: The switch may segment the ABR control loop to make a virtual source and destination.
2. A switch may generate a backward RM cell if necessary. Both the BN and DIR are set to 1. To inform the source of its congestion, it may either set CI=1 or NI=1. The other fields of the RM cell are assigned appropriately. The rate of these BRM cells (including both in-rate and out-of-rate) is limited to 10 cells/second, per connection.
3. RM cells may be transmitted out of order with respect to data cells. However, the sequence between any RM cells must be maintained.
4. The values of the various fields before CRC-10 field in an RM cell should not be changed by a switch except:
   (a) The CI, the NI, and the ER may be modified (rule 1).
   (b) The fields unused by the ATM Forum (RA, QL, and SN) may be modified by another standard recommendation.
   (c) The MCR may be corrected if the incoming MCR value is incorrect.
5. The switch may also implement a *use-it-or-lose-it* policy to reduce its ACR to a value that approximates the actual cell transmission rate from the source.

A number of rate-based congestion control schemes in ABR have been proposed. Most of them are based on the same rationale described above. These closed-loop rate-based control mechanisms are very intelligent and cautious. However, if the round-trip-time is long, it takes

quite a while to start flow control in a closed-loop based scheme, which is why it is sometimes called a reactive method. ABR makes use of a technique called virtual source/virtual destination. Nevertheless, it makes the scheme more and more complex and sophisticated. Therefore, simple and cost effective schemes for non-real-time applications have been studied. The applications are serviced over a UBR category. The next section describes the congestion control of UBR services.

3.2.2 Congestion control of UBR

The type of applications ABR supports are similar to those UBR supports. The basic rationale of TCP over UBR is that the traffic control algorithms should be applied at a higher layer than the ATM layer. Provided that the network is congested and network resources such as buffer space are insufficient, any cells of UBR services can be discarded. Thus, UBR guarantees no quality level in the ATM layer.

ABR has attracted much attention for a long time and is expected to deliver better quality of service than UBR. However, congestion control of UBR is much simpler than that of ABR and the implementation complexity and cost of ABR is significantly higher than that of UBR. In addition, in spite of the ABR's intelligent and sophisticated control scheme, because of its reactive property it may not be a timely control at times. Focusing on the support of TCP over ATM, a UBR congestion control scheme should provide a simple and low-cost alternative to ABR service.

The Partial packet discard (PPD) scheme

A packet handed down from the TCP layer must be segmented into fixed sized cells through the AAL layer to be transmitted over ATM networks. However, while TCP supports reliable services, ATM does not guarantee any quality level for UBR services. In other words, provided a packet is known to be lost or damaged, retransmission of the packet is triggered in the TCP layer, not in the ATM layer. Thus, once a cell is dropped in an ATM network, the TCP layer will have to transmit the entire TCP packet again afterward (Figure 3.8). If cells are dropped whenever a buffer overflows, the cell dropping action is dispersed over many packets. Hence, a large quantity of damaged packets are transferred through the ATM network to be finally thrown away at the receiver TCP. Figure 3.8 illustrates this fragmentation and retransmission of a damaged packet very simply. Actually, however, TCP has much more sophisticated mechanism to acknowledge its correct packet receipt, and in order to know that a packet is damaged or lost, a sender TCP generally has to wait for a timeout, which is multiple of a round-trip-time (RTT). Even worse, those still active useless cells are propagated to their destination TCP node, thereby wasting network resources and sometimes contributing to network congestion itself. To relieve this packet fragmentation problem and to control the congestion of TCP over ATM, some simple

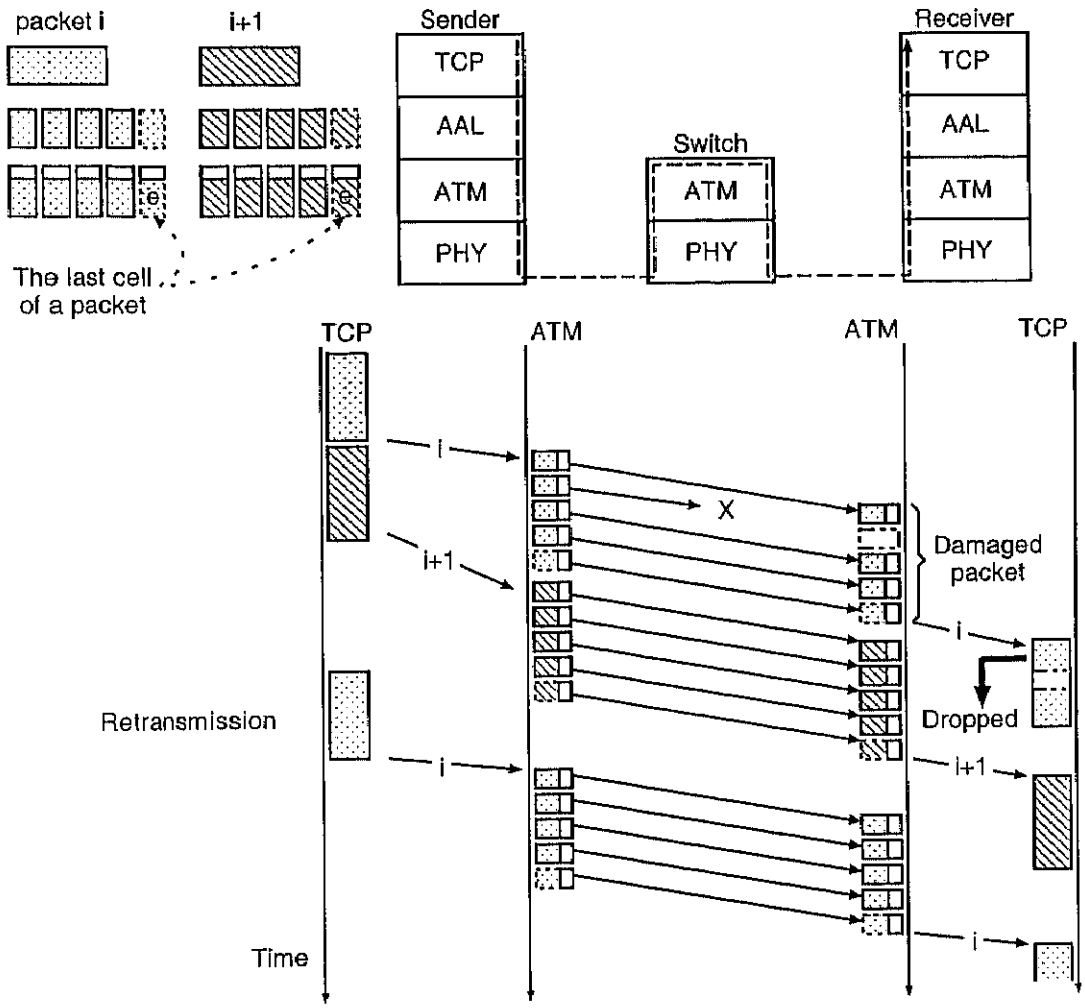

Figure 3.8: TCP packet fragmentation over ATM

strategies to sacrifice a small number of packets to save many have been studied.

The **Partial packet discard** (PPD) scheme is one of these strategies[ARMI93]. In this strategy, when a buffer in a switch is full and a cell is to be dropped, an ATM switch marks the VCI of the connection over which the packet is transmitted (D flag in Figure 3.9), and discards all the subsequent cells from the same packet except the last one cell (i.e., the tail part of the packet). This is why PPD is also called "Packet Tail Discard" in [TURN96]. A new packet

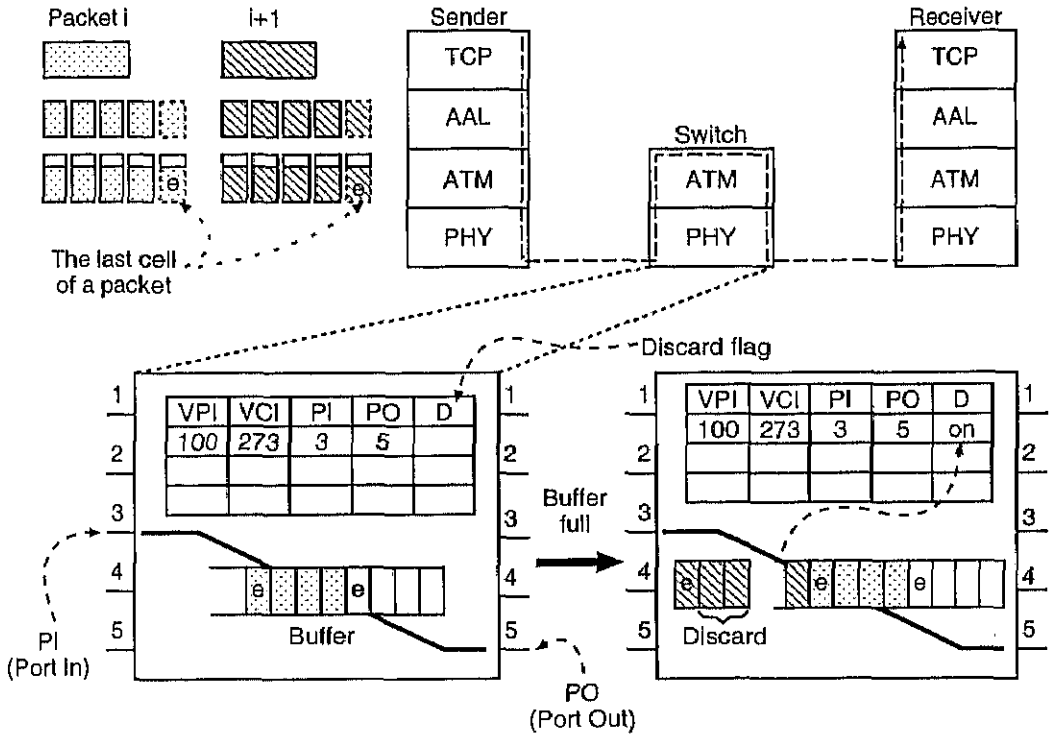

Figure 3.9: Partial packet discard scheme

can be identified by the EOP flag in an AAL 5 frame cell header. In this scheme, the buffer resources are not used for those useless cells, thus producing more available space in the buffer. This results in congestion control by means of a preventive mechanism.

With PPD, it is possible to partially reduce the forwarding of those needless cells. However, since part of the packet is already queued waiting to be transmitted, the switches still try to relay the cells belonging to those damaged packets. The "early packet discard" (EPD) scheme, which will be described next, solves this problem.

Early packet discard (EPD) scheme

In the early packet discard (EPD) scheme[ROMA95], a certain buffer occupancy level is set as a threshold value which is used by an ATM switch to estimate the congestion level of a network. A switch monitors the queue length and if it exceeds the threshold value, a discard flag

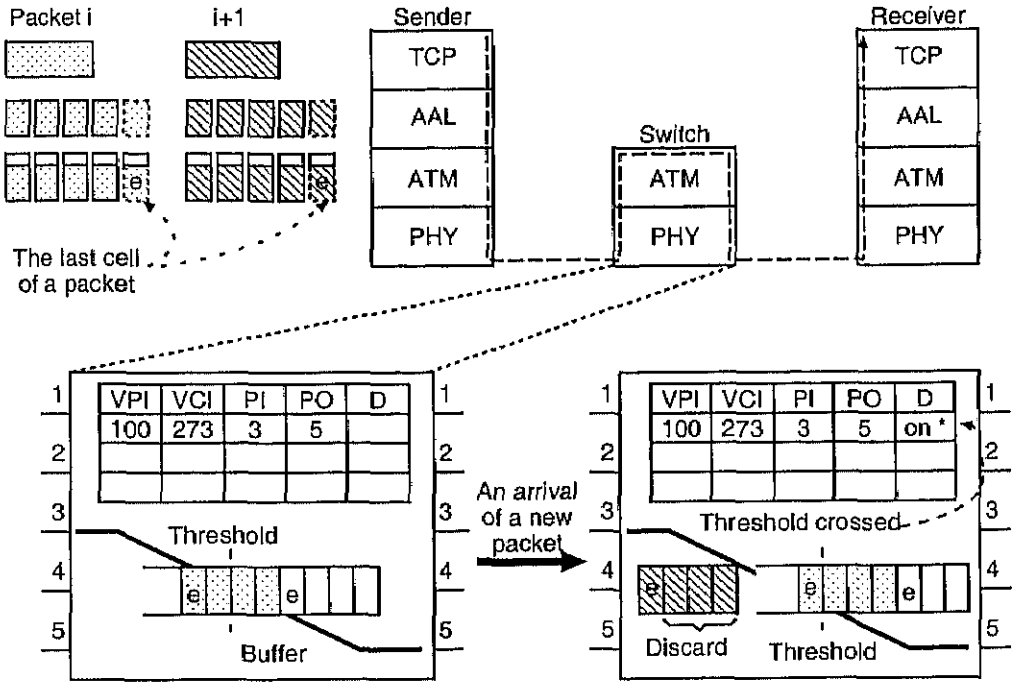

Figure 3.10: Early packet discard scheme

is set (Figure 3.10). If a new packet arrives when the flag is set, the packet is not allowed to be buffered. Thus, with this EPD scheme, the entire packet except the last cell is discarded while only part of the packet is discarded in the PPD scheme. In other words, it is also possible to prevent the transmission of partially damaged packets by using the EPD algorithm. Therefore, EPD prevents useless transmission more aggressively than PPD does.

Similar to the PPD scheme, a new packet is identified by an end of packet (EOP) flag in a cell header in the EPD scheme. Basically, in both the PPD and EPD schemes, the last cells should not be discarded. This is because if the last cell is dropped, the TCP cannot discriminate the packet boundary between an actually damaged packet and the following flawless one. Hence, it assumes that two packets are damaged. Thus, the last cell should be buffered in almost all cases.

Numerous studies of the EPD scheme have been conducted. Lapid, Rom, and Side examined the performance of EPD, PPD, and plain TCP[7] by an analytical model with different loads and different arrival processes[LAPI97]. They showed that the EPD policy outperforms PPD and plain TCP in cases when the load is not light. On the other hand, one of the most important parameters of EPD is the threshold value (in Figure 3.11). If the threshold is too large, or the restricted part (RP)[8] is too small, the EPD scheme cannot take advantage of the early

[7] A TCP that no congestion policy is applied in the ATM level.

[8] The portion over the threshold is called the "restricted part" here, because it restricts the entrance of the new packets.

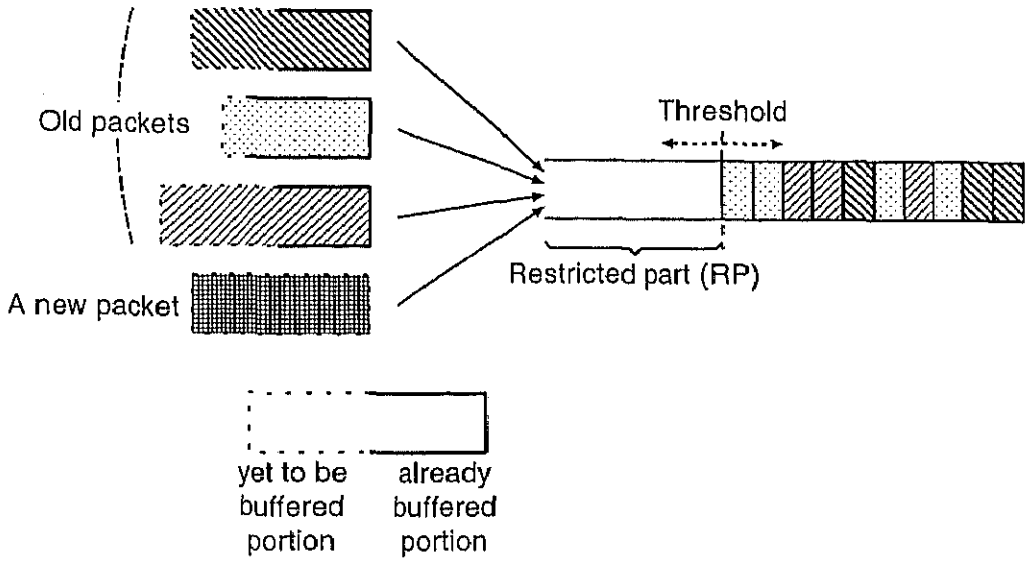

Figure 3.11: Buffer threshold

discard of packets. This is because there is not much space left for both new packets and the rest of the already received ones and the queue length reaches its end too quickly to hold both the new packets and the rest of the already buffered ones. On the contrary, when the threshold is so small that a number of new packets are prohibited from entering, the buffer utilization as well as the propagation rate is reduced. As we can see, the determination of the threshold value has been an important factor of EPD. In [JAUS98], Jaussi, Lorang, and Nelissen evaluated EPD with a simple model. They showed that the spare space of one packet size to two/three packet sizes (depending on the packet size) is enough to achieve moderate performance. Basic TCP, however, is not said to fully utilize network resources in high-speed networks due to the limitation of its congestion window size. In addition, since TCP regulates its own streams by means of acknowledgments, it cannot send any more packets while it is waiting for a corresponding acknowledgment. Thus, if cells from TCP packets are controlled by an EPD algorithm in the ATM layer, the buffer in the ATM switch (specifically the RP) is not fully utilized, as Okuda and Ishihara pointed out[OKUD96]. Therefore, if this RP of buffer space is exploited more efficiently, it should improve TCP performance.

In this dissertation, the author shows the possibility of using buffer resources more efficiently by an optimistic and simple packet discard mechanism through generalization of ordinary packet discard schemes.