


    Overview Description.


    34(1), January 1986.

Figure 17: Extension of the Rerouting Protocol to Handle VC’s Which Cannot be Accommodated by the Backup VP

Setup to Passive and sends a CNCL-F message to J. The takedown protocol of VC(\(L\rightarrow J\)) is completed at A when a CNCL-F is received from L.

6 Conclusions

The paper has shown that the virtual path concept of ATM, where two levels of connections are defined, may significantly enhance survivability of VC connections. The paper has presented a two-way handshake rerouting protocol, to be invoked upon the failure of a VP connection, whereby all the VC connections established over the failed VP are rerouted to a backup VP. Due to the simplicity of the protocol, it can be completed within a short time, thus fulfilling an essential requirement of any rerouting scheme.

The paper has shown that the rerouting protocol enables the end points of the failed VP to recover also from loss of signaling messages sent over the signaling VC associated with the failed VP. Since the REROUTE messages exchanged by the two end points of the failed VP contain the state of every VC established over the failed VP, these nodes can deduce whether and which signaling messages were lost, and to act accordingly. The protocol also enables the source node of the failed VP to determine which VC’s can be rerouted and which of them should be taken down due to insufficient bandwidth in the backup VP. To this end, the state of the VC’s that cannot be rerouted is changed at the source, before the REROUTE message is sent, from Active to Takedown.

References


The last two cases will become possible when we shall treat in Sec. 5.4 the case of VC's that cannot be rerouted because of lack of bandwidth.

Note that the state of each handled VC should not change between the time a REROUTE is sent and the time the REROUTE is received from the other end. This can be achieved for VC(L→J) if node A queues signaling messages received from L regarding VC(L→J) during the time it waits for the REROUTE message from D. Similarly for node D. In addition, both A and D should not participate in any other rerouting instance before the current one ends (see Fig. 14).

5.4 Handling Active and Pending-Active VC’s That Cannot be Rerouted

So far it was assumed that there is sufficient bandwidth in the backup VP to allow all active and pending-active VC’s to be rerouted. If the bandwidth in the backup VP is limited, some of the VC’s will have to be taken down. Node A should decide which of the VC’s will be rerouted, as described before, and which of them will be taken down. This can be stated as an optimization decision in the following way.

Let VC₁, VC₂, ..., VCₙ be the collection of VC’s which are either in Active state or in Setup state at node A when node A invokes the rerouting algorithm (i.e. before it sends a REROUTE message to D). For every VCᵢ, 1 ≤ i ≤ N, let β(VCᵢ) be the bandwidth needed by VCᵢ, π(VCᵢ) be the rerouting priority of VCᵢ, and ρ(VCᵢ) be 1 if VCᵢ is rerouted and 0 otherwise. In addition, let B be the bandwidth available in the backup VP for the rerouting. We want to find a vector ρ₁, ρ₂, ..., ρₙ such that \( \sum_{i=1}^{N} (\rho_i \cdot \pi_i) \) is maximized subject to \( \sum_{i=1}^{N} (\rho_i \cdot \beta_i) \leq B \).

If there is more than one rerouting priority, the problem is equivalent to the "knapsack problem", which is known to be NP-complete. If the rerouting priority is the same for all the VC’s, i.e. for every 1 ≤ i ≤ N \( \pi_i = 1 \), the problem can be easily solved by a simple greedy algorithm. The solution dictates that the VC’s should be selected for rerouting based on their bandwidth requirement from low to high. The consequence is that the number of VC’s that cannot be rerouted due to lack of bandwidth is minimized.

After node A determines which VC’s cannot be rerouted, it initiates a takedown protocol for each. This can be easily done by changing the state of each VC from Active or Setup to Takedown before sending the REROUTE message to D, and by sending a CNCL-B message to the upstream VC node (e.g. to node L for VC(L→J)). Consequently, cases 12 and 13 in Fig. 17, become possible. When D receives the REROUTE of A, it changes the state of the new selected entry, VC-Table(Dₑ)[dₗ⁻], from used or frozen to free, it changes the state of VC(L→J) from Active or
this message had been received, node $D$ would have been in Setup or Active or Takedown state for this VC). Unlike all the previous cases, when $D$ receives the reroute it cannot act as if the lost message was received. The reason is that the SETUP-F message contains some parameters regarding the to-be-established VC. However, node $A$ can deduce that the SETUP-F message was lost. Thus, it resends the message over S-VC($A\rightarrow D$), the S-VC associated with the backup VP.

11. The VC is in Takedown state at $A$ and in Passive state at $D$. This implies that no SETUP or CNCL signaling message regarding this VC has been lost. VC($I\rightarrow J$) is in the process of being taken down, and $A$ and $D$ do not expect CNCL messages from each other. Thus, when they receive the REROUTE message of each other, they do nothing regarding VC($I\rightarrow J$). At some later time, node $A$ will receive CNCL-B from node $L$.

Since the VC finite state machine has 4 states, the total number of state combinations for a given VC at two nodes is 16. The list presented above is missing the following 5 cases:

- Both $A$ and $D$ are in Passive state. In such a case the VC is not included in the REROUTE messages, and no actions are taken regarding the rerouting of such a VC.

- The VC is in Active state at $A$, and in Setup state at $D$. This case is not possible because $A$ enters Active only upon receiving the SETUP-B from $D$. However, when $D$ sends SETUP-B it leaves the Setup state and cannot re-enter this state.

- The VC is in Active state at $A$, and in Passive state at $D$. This case is not possible because $A$ enters Active only upon receiving the SETUP-B from $D$. When $D$ sends SETUP-B, it leaves the Setup state and enters Active state. It may move from Active to Passive only after having received CNCL-F from $A$. However, if $A$ sends such a message it must leave the Active state.

- The VC is in Takedown state at $A$, and in Active state at $D$. With the algorithm presented so far, this case is not possible. The reason is that node $A$ enters Takedown only upon receiving CNCL-B from $D$ (e.g., due to destination-driven takedown) or upon taking down the VC following a failure of VP($A\rightarrow D$). The first case is not possible because after $D$ sends CNCL-B it cannot be in Active state. The second case is not applicable since we consider the case where $A$ tries to REROUTE the VC rather than to take it down.

- The VC is in Takedown state at $A$, and in Setup state at $D$. This case is impossible since it is equivalent to the previous one.
4. The VC is in Setup state at both A and D. In this case no SETUP or CNCL signaling message regarding this VC has been lost. Recall that before sending the REROUTE-B, node D allocated an entry $d_1^r$ in VC-Table$(D\rightarrow A)$, and puts this entry in *frozen* state. Later, when node D receives a SETUP-B message from J, it will send SETUP-B$(d_1^r)$ to A.

5. The VC is in Passive state at A and in Setup state at D. This implies that A has sent to D a CNCL-F message which was lost due to the failure of S-VC$(A\rightarrow D)$. When D receives the REROUTE-F of A, it acts as if the CNCL-F was received: it changes the state of the new selected entry VC-Table$(D\rightarrow C)[d_1^r]$ from *frozen* to *free*; it changes the state of the VC from Setup to Passive; and it sends a CNCL-F message to node J — the next VC node of VC$(L\rightarrow J)$.

6. The VC is in Active state at A, and in Takedown state at D. This implies that D has sent to A a CNCL-B message which was lost due to the failure of S-VC$(A\rightarrow D)$. When A receives the REROUTE-B from D, it acts as if the CNCL-B was received: it changes the state of VC-Table$(A_K)[a_1]$ from *used* to *free*; it changes the state of the VC from Active to Passive; and it sends a CNCL-F to D (over S-VC$(A\rightarrow D)$) and CNCL-B message to L — the upstream VC node of VC$(L\rightarrow J)$.

7. The VC is in Setup state at A and in Takedown at D. This case is similar to the previous one. Both A and D perform the same actions as in the previous case, and the VC is taken down due to an unsuccessful set-up.

8. The VC is in Takedown state at A and D. This implies that A has sent to D a CNCL-F message which was lost due to the failure of S-VC$(A\rightarrow D)$ (because if this message had been received, node D would have been in Passive state for this VC). When D receives the REROUTE-F from A, it acts as if the CNCL-F was received: it changes the state of VC-Table$(D_C)[d_1]$ from *frozen* to *free*, and the state of VC$(L\rightarrow J)$ from Active to Passive. Node A has nothing to do, it keeps waiting for CNCL-F from L in order to move the VC into Passive state.

9. The VC is in Passive at A and in Takedown at D. This case is similar to the previous one. The only difference is that when the rerouting protocol takes place, node A is in Passive rather than Setup state. Node D performs the same actions as in the previous case, and the VC is taken down.

10. The VC is in SETUP state at A and in Passive state at D. This implies that node A has sent to D a SETUP-F message which was lost due to the failure of S-VC$(A\rightarrow D)$ (because if
summarized in Figure 16 and explained in the following. In the following we explain the actions taken by A and D for every possible combination.

1. This is the case addressed in Sec. 5.2, where the VC is active while the failure takes place, and thus is rerouted to the backup VP.

2. The VC is in Setup state at A, and in Active state at D. This implies that D sent to A a SETUP-B message which was lost due to the failure of S-VC(A→D). The lost message contained the VCI to be used by A. However, due to the failure this VCI is no more relevant. On the other hand, the REROUTE-B sent by D contains for every VC in Active state at D a new VCI to be used by A on the backup VP. Thus, this REROUTE can be used instead of the lost SETUP-B. Upon receiving the REROUTE-B, node A writes in the empty entry VC-Table(AK)[a1] the triplet (b1, d1*, A_B), changes the state of this entry from frozen to used, and changes the state of the VC from Setup to Active. Node D makes no changes upon receiving the REROUTE-F from A. Following the reroute execution, the state of the VC changes from pending-active to active.

3. The VC is in Takedown state at A, and in Active state at D. This implies that A has sent to D a CNCL-F message which was lost due to the failure of S-VC(A→D). When D receives the REROUTE-F and realizes that this is the case, it acts as if the CNCL-F was received: it changes the state of the new selected entry VC-Table(DC)[d1*] from used to free; it changes the state of the VC from Active to Passive; and sends a CNCL-F message to node J — the next VC node of VC(L→J).

Figure 16: The Rerouting Protocol
5.3 Handling Pending VC Connections That can be Rerouted

In the following we shall address the cases when a VC is not active at the time when one of its building block VP’s fails. We shall show how a pending-passive VC is taken down, and how the setup of a pending-active VC that can be transfered to the backup VP is successfully completed. The main difficulty is that signaling messages belonging to pending VC’s may be lost because of the failure of the signaling VC associated with the failed VP. Several ways to overcome this difficulty are:

(a) Pending VC’s are taken down when there is a failure of the VP $(A \rightarrow D)$.

(b) Reroute signaling VC to the backup VP before the non-signaling VC’s. Then, after the SSCOP protocol executed over the signaling VC recovers from the failure\(^8\), the non-signaling VC’s can be rerouted.

(c) The end points of the failed VP use the state information in the REROUTE messages to deduce which signaling messages were lost and act as if those messages were received.

We select the third approach because of the following deficiencies of the first two. Approach (a) decreases the reliability of signaling VC connections and makes signaling over S-VC’s less robust than signaling over existing Common Channel Signaling networks. Approach (b) leads to a significant increase of the period of time during which the VC’s are not alive compared to approach (c) and still requires a special mechanism to detect and handle pending VC’s.

In the sequel, we present approach (c). As already shown for the rerouting of VC’s in Active state, both $A$ and $D$ attach to the REROUTE message the global VC-id and the state of every VC for which VC $(A \rightarrow D)$ serves as a building block. In addition, node $D$ selects a new entry for every VC in Active state, where it copies the contents of the old entry. For every VC in Setup state, node $D$ selects a new entry, but copies nothing into it because the old entry is still empty (the contents are written upon receiving a SETUP-B, in which case the state of the VC changes from Setup to Active).

After $A$ and $D$ receive the REROUTE of each other, they know the state of every VC at the other side of the failed VP (if the VC does not exist in the received REROUTE, it is in Passive state). The actions they perform depend on the state of the VC in each side. These actions are

\(^8\)The SSCOP uses acknowledgment mechanism and retransmission mechanism to ensure reliable transmission of signaling messages as long as the S-VC over which it runs is alive. Thus, if the S-VC is rerouted, the protocol can detect and retransmit lost messages.
selects an empty entry in VC-Table($D_{1^{-}}$), entry $d^*_1$, say, and copies the contents of VC-Table($D_{C}$)[$d_1$] to VC-Table($D_{1^{-}}$)[$d^*_1$]. This entry is put in used mode. Then, $D$ appends to the REROUTE-B message it will send to $A$ the global VC-id of VC($L^{-}J$), the state of this VC at $D$ (Active), and the new VCI ($d^*_1$).

- Before sending REROUTE-F, node $A$ puts entry VC-Table($A_{K}$)[$a_1$] in frozen mode. Then, $A$ appends to the REROUTE-F message it will send to $A$ the global VC-id and the state (Active) of VC($L^{-}J$).

- Upon receiving the REROUTE-B from $D$, node $A$ changes the contents of entry VC-Table($A_{K}$)[$a_1$] from ($b_1, d_1, A_B$) to ($\bar{b}_1, d^*_1, A_{\bar{B}}$), and changes the state of this entry back to used. Label $\bar{b}_1$ is the VPI of all the cells sent by $A$ over VP($A^{-}D$), whereas $d^*_1$ is the new VCI selected by $D$.

- Upon receiving the REROUTE-F from $A$, node $D$ checks the state of VC($L^{-}J$) at $A$. Since the state is Active, node $D$ has nothing more to do regarding the rerouting of this VC.

The purpose of the REROUTE-F message sent by $A$ to $D$ is explained in Sec. 5.3.

Note that cells belonging to VC($L^{-}J$) will be received by $D$ over the backup VP only after such cells stop being received over the original VP. This “VC level FIFO” must be retained according to the ATM standard.

over the failed VP. Thus, entry VC-Table($D_{C}$)[$d_1$] can be safely put in free state.
there is sufficient bandwidth in the backup VP and pending-passive VC’s. Finally, we shall present the handling of active and pending-active VC’s that cannot be rerouted because of lack of bandwidth. As an example for the VC’s needed to be addressed, i.e. VC’s for which VP(A→D) is used as a building block, we consider VC(L→J). As shown in Fig. 13, this VC has three building blocks: VP(L→A), VP(A→D) and VP(D→J).

5.2 Handling Active VC Connections That can be Rerouted

In this section we consider VC’s that are in Active state at A and D when VP(A→D) fails and can be rerouted from VP(A→D) to VP(A→D). Let VC(L→J) be such a VC. Suppose that cells of VC(L→J) are transmitted by node K to node A with VCI=a1. Recall that this value is written into these cells by L and then does not change by the intermediate nodes of VP(L→A). Assuming that before rerouting VC-Table(AK)[a1] = (b1, d1, A,B), as Fig. 15 shows, these cells continue over VP(A→D) with VPI=b1 and VCI=d1. Label b1 is the VPI of all cells sent by A over VP(A→D), regardless of the VC connections to which they belong. This label was determined during the setup of VP(A→D). Label d1 is the VCI of the cells belonging to VC(L→J), as determined when VC(L→J) was set up.

In order to reroute VC(L→J) from VP(A→D) to VP(A→D), changes should be performed in the routing tables of both A and D. This is done in the following way (Fig. 15).

- Before sending REROUTE-B, node D puts entry VC-Table(DC)[d1] in free state7. Then, it se-

\footnote{As said in Sec. 4.2, we assume that after node D detects the failure of VP(A→D), it may receive no more cells...
protocol is invoked each of the VC connections established on the failed VP can be in one of the following three situations:

**active:** The set-up of the VC at $A$ and $D$ has been completed. Thus, the VC is in Active state at $A$ and $D$.

**pending-active:** The set-up of the VC at $A$ and $D$ has been started but not yet completed. Thus, the VC is in Setup state at $A$ and in Passive, Setup or Active state at $D$.

**pending-passive:** A take-down protocol for this VC has been started and not yet completed. Thus, one of the following holds: (i) $A$ is in Active or Setup states and $D$ is in Takedown state; (ii) $A$ is in Takedown state and $D$ is in any state. We explain in Sec. 5.3 why other combinations of states are not possible.

Our rerouting protocol ensures that the failure of the VP will not affect the VC. This implies that if the VC is active when the failure of VP($A\rightarrow D$) takes place, it will remain active on the backup VP. If the VC is pending-active when the failure takes place, the set-up will be completed over the backup VP. If the VC is pending-passive when the failure takes place, the take-down protocol will be completed. In the latter two cases, the rerouting protocol can overcome any disagreement between the FSM's of $A$ and $D$, which might happen due to loss of signaling messages sent over S-VC($A\rightarrow D$), the S-VC associated with the failed VP. For instance, if during the set-up of a VC, node $D$ sends to $A$ a SETUP-B message and the latter is lost due to the failure of the S-VC, the set-up of the VC will be completed over the backup VP despite of the loss of the SETUP signaling message.

When VP($A\rightarrow D$) fails, the associated S-VC($A\rightarrow D$) fails as well. Asynchronously, nodes $A$ and $D$ get notification that the VP has failed. Then, node $A$ – the source of the failed VP – sends to $D$ a REROUTE-F message, whereas node $D$ sends $A$ a REROUTE-B message. The REROUTE messages are signaling messages, sent by $A$ and $D$ over S-VC($A\rightarrow D$), which, by Assumption 2, is alive.

Before sending the REROUTE and after receiving the REROUTE from the other side, $A$ and $D$ need to do some local actions for every VC served by VP($A\rightarrow D$). However, the REROUTE message each of them sends is common for all the VC’s. The rerouting protocol is completed at each of $A$ and $D$ when the node receives and processes the REROUTE of the other (see Fig. 14).

In the following we shall first present the rerouting of active VC’s for which there is sufficient bandwidth in the backup VP. Then, we shall explain how to handle pending-active VC’s for which
or of a node along the route of a VP connection would trigger a large number of VC cancelation protocols and succeeding VC setup protocols, which may impose an unreasonable burden on the network.

The approach we suggest is the following. For every VP connection, or at least for those that are likely to carry a large number of VC connections, there exists a predefined backup VP connection between the same two end point nodes. A VP connection and its backup will be referred to as parallel VP connections. It is essential that parallel VP connections will have a minimum of common links and nodes, so as to minimize the possibility that a single failure will destroy both.

Fig. 13 shows a pair of parallel VP connections, between nodes A and D. Let VP(A→D) be the active one, and VP(A→D) the backup. Suppose that VP(A→D) fails, due to a failure of an intermediate link or node, while VP(A→D) is alive. We want to reroute to VP(A→D) the VC connections established over VP(A→D), such that the end nodes of the rerouted VC connections (e.g. L and J for VC(L→J)) will not be involved. The bandwidth available in the backup VP might be sufficient to accommodate only part of these VC’s. This is the task of A, the source node of the failed VP, to determine how much bandwidth is available on the backup VP and which VC connections should be rerouted. Those VC connections which cannot be rerouted due to lack of bandwidth in the backup VP will be taken down by means of the failure-driven takedown protocol.

As far as A and D, the two end points of the failed VP, are concerned, when the rerouting...
the destination VC node because the latter does not have a downstream neighbor from which a SETUP-B message can be received. Thus, upon receiving a SETUP-F message, a destination VC node enters Active state directly.

**Active:** SETUP-F and SETUP-B messages have been received for this VC (only SETUP-F if the node is the destination VC node for the particular VC). The allocated entry in the VC tables is in *used* state.

**Takedown:** The VC is in the process of being taken down. A CNCL-B message or cncl-B indication was received in Active state. As far as the node concerns, the takedown process will be completed upon the receipt of a CNCL-F message. The entry allocated to this VC is in *frozen* state. Such a state does not exist for the source VC node because the latter does not have an upstream neighbor from which a CNCL-F message should be received. Thus, upon receiving a CNCL-B message, a source VC node enters Passive state directly.

Recall that a failure indication serves as a CNCL message. Thus, an intermediate VC node in Active state that receives a failure-F indication moves to Passive, exactly as if a CNCL-F message had been received. Upon receiving a failure-B indication, such a node moves to Takedown state, as if a CNCL-B message had been received. However, in the latter case only CNCL-B message is sent. A CNCL-F message, which is sent when a CNCL-B is received, cannot be sent due to the failure. Recall that both the source VC node and the destination VC node can initiate a normal execution of the takedown protocol. At the source, this happens following the receipt of a cncl-F indication from the upper layer. At the destination, this happens following the receipt of a cncl-B indication from the upper layer. In addition, every VC node (source, destination or intermediate) can initiate an abnormal execution of the takedown protocol following the receipt of a failure-F or failure-B indication.

5 The VC Rerouting Protocol

5.1 The Backup VP

One of the important properties of the virtual path concept is that it can be used to significantly improve the survivability of regular (non-signaling) VC connections and the reliability of signaling VC connections. As shown in this section, a simple protocol executed by the two end points of a failed VP connection may save all VC connections that use that VP. Since a large number of VC connections can be bundled in each VP, without such a rerouting mechanism, the failure of a link
Figure 12: The VC Finite State Machine
only if the VC cannot be rerouted.

4.3 The VC Finite-State-Machine

Fig. 12 presents the finite-state-machine (FSM) of a VC. Such an FSM is kept for every VC at every end point of a VP over which the VC is set up. Fig. 12(a) presents the FSM kept by the intermediate VC nodes (nodes D and H for VC(A→H)), Fig. 12(b) presents the FSM kept by the source VC node (node A for VC(A→H)) and Fig. 12(c) presents the FSM kept by the destination VC node (node H for VC(A→H)). Each transition is associated with one or more [event/action] pairs. A transition is made when an event of any pair takes place, in which case the action associated with this event is performed.

Each of the four states has an interpretation as follows:

Passive: The VC does not exist in this node, because it does not go through a VP for which this node serves as an end point. In particular, no entry is allocated to this VC in the node routing tables.

Setup: A SETUP-F message has been received and an entry in the routing table is allocated to the VC. This entry is in a frozen state. The node waits for a SETUP-B in order to move the VC entry to used state and the VC itself to the Active state. Such a state does not exist for
submitted by the AAL, and sends a CNCL-F message to $D$ over S-VC$(A\rightarrow D)$. Upon receiving this
message, node $D$ knows that it will receive no more cells belonging to VC$(A\rightarrow H)$. Thus, it safely
changes the status of entry VC-Table$(D_C)[d_4]$ to free and sends a CNCL-F message forward to $F$.
Upon receiving the message from $D$, node $F$ changes the status of entry VC-Table$(F_E)[f_4]$ to free
and sends a CNCL-F message to $H$. Node $H$ changes the status of entry VC-Table$(H_A)[h_4]$ to free.
This protocol is described in Fig. 11(a).

Next, consider a destination-driven taken down of VC$(A\rightarrow H)$ (Fig. 11(b)), initiated by $H$ following
the receipt of a cncl-B indication from the higher layer. Unlike in the source-driven taken down,
ode $H$ cannot change the status of entry VC-Table$(D_C)[d_4]$ to free before it makes sure that no
more cells belonging to VC$(A\rightarrow H)$ (carrying VCI=$h_4$) will be received from $A$. The reason is that
due to race conditions, if such an entry is allocated to another VC and cells belonging to VC$(A\rightarrow H)$
continue to arrive, they will be mis-routed. Thus, node $H$ first puts entry VC-Table$(D_C)[d_4]$ in
frozen state and sends CNCL-B to $F$. When $F$ receives the CNCL-B message from $H$, it changes
the status of entry VC-Table$(F_E)[f_4]$ from used to frozen and sends a CNCL-F message to $H$ and
a CNCL-B message to $D$. Upon receiving the CNCL-F message, $H$ can safely change the state of
entry VC-Table$(H_A)[h_4]$ from frozen to free. The protocol continues in the same way until node $A$
receives a CNCL-B from $D$. Since $A$ is the source, it can inform the AAL associated with this VC
to stop sending cells. Then, it can change to status of VC-Table$(A_A)[a_4]$ directly from used to free.
Finally, node $A$ sends CNCL-F to $D$. Upon receiving this message, node $D$ changes the status of
VC-Table$(D_C)[d_4]$ from frozen to free, and the protocol is completed.

An abnormal taken down is demonstrated in Fig. 11(c) for the case where VP$(D\rightarrow F)$ fails. The
failure-F indication node $F$ receives from the lower layer serves as a CNCL-F, because $F$ knows it
will receive no more cells belonging to the VC over the failed VP. Similarly, the failure-B indication
node $D$ receives from the lower layer serves as a CNCL-B. Consequently, the protocol is executed
in the right-hand side of the failure as a source-driven taken down, and in the left-hand side of the
failure as a destination-driven taken down.

In Sec. 5 it will be shown that a failure of a VP over which a VC has been established does
not necessarily lead to the execution of the taken down protocol. Rather, the VC can be rerouted to
an alternative, backup, VP connection. The taken down protocol presented above should be invoked

5From assumption 2 follows that if node $A$ sends to $D$ at time $t$ a signaling message, this message is received and
processed by $D$ after $D$ has received all cells transmitted by $A$ to $D$ before $t$.

6We assume that no more cells are accepted over the failed link after failure indication. This can be achieved by
putting the relevant entry in the VP-Table in a frozen state.
Figure 10: Unsuccessful Executions of the Setup Protocol
can be changed from to \textit{free}. If, however, the VC is taken down before the setup is completed, as in our case, the state of the entry at the downstream neighbor ($F$) can be safely changed to \textit{free} regardless of the CNCL-F sent by $F$. The only reason we dictate the transmission of a CNCL-F message in this case is to enable the two end points of a failed S-VC to deduce from the state of each other which signaling messages have been lost. More details are given in Sec. 5. Upon receiving the CNCL-B from $D$, node $A$ releases the resources it has allocated to the VC and sends a CNCL-F message back to $D$. This completes an unsuccessful execution of the setup protocol.

A different case of unsuccessful setup is shown in Fig. 10(b). This time an S-VC (the one between $D$ and $F$) fails after the SETUP-F is sent, but before the SETUP-B is received. Node $D$ is notified of the failure by means of a failure-B notification from the SSCOP layer\(^3\). Upon receiving failure-B, $D$ behaves as before and the setup protocol terminates unsuccessfally. At the other side of the failure, node $F$ gets a failure-F indication from the SSCOP layer, either before (as in Fig. 10(b)) or after receiving SETUP-B from $H$, and sends a CNCL-F to $H$. Upon receiving this message, node $H$ releases the resources allocated to the VC.

In Sec. 5 it will be shown that a failure of a VP over which a VC has to be established, or an S-VC over which SETUP-F/SETUP-B need to be exchanged, does not necessarily lead to the execution of the takedown protocol. Rather, the VC can be established over an alternative, backup, VP connection, and signaling messages can be exchanged over the S-VC associated with the backup VP.

### 4.2 Virtual Channel Takedown

Consider a VC which has been successfully set up. Such a VC can be taken down in one of the following ways: (a) a \textit{normal} takedown, initiated by the originator of the VC (e.g. node $A$ for VC($A\rightarrow H$)), referred to as source-driven takedown, or by the destination (node $H$ for VC($A\rightarrow H$)), referred to as destination-driven takedown, when the VC is no more needed; (b) an \textit{abnormal} takedown, referred to as failure-driven takedown, initiated due to a failure of a VP over which the VC has been established.

Consider first a source-driven takedown of VC($A\rightarrow H$) (Fig. 11(a)). The protocol is initiated when node $A$ receives a cncl-F indication\(^4\) from a higher layer. Node $A$ changes the status of entry VC-Table($A_4$)[$a_4$] to \textit{free}, after making sure that no more cells belonging to this VC will be

\(^{3}\)A dashed arrow in the figures represents an \textit{indication} given from another layer. This is as opposed to a solid line, which represents a message.

\(^{4}\)Messages of the protocol are denoted by capital letters, e.g. CNCL-F, while interaction with other layers is denoted in lower case, e.g. cncl-F and failure-F indications.
Figure 9: The Relevant VC-Tables of Nodes A, D, and F following the Setup of VC(A→F).

- Set VC-Table(D_C)[d_4] ← (e_2, f_4, D_E) and put this entry in used state. The value f_4 is known from the SETUP-B message received from node F. The value e_2 is the VPI used by D for all the cells transmitted over VP(D→F). This value was selected during the setup of this VP.
- Send SETUP-B(d_4) message to A on S-VC(D→A).

Upon receiving this message, the Network layer at A performs as follows:
- Set VC-Table(A_A)[a_4] ← (b_2, d_4, A_B) and put this entry in used state. The value d_4 is known from the SETUP-B message received from node D. The value b_2 is the VPI used by A for all the cells transmitted over VP(A→D).

The relevant VC-Tables of Nodes A, D, F and H following the setup protocol are shown in Fig. 9. Upon the completion of the setup protocol, every cell submitted by the AAL of A to the switch of A with VPI= a_3 and VCI= a_4, where [a_3, a_4] is the multiplexing label of the created VC at A, will be received by the AAL of H with the demultiplexing label [h_1, h_2] (assuming that VP-Table(A_A)[a_3],VPI was initialized to nil).

The setup protocol works as described above provided that the S-VC’s needed for the exchange of SETUP messages are alive and the resources, like bandwidth, entries in routing tables etc., are available. If an S-VC is down at the time when a SETUP-F must be sent, or if resources for this VC cannot be allocated (e.g., due to a failure of a VP over which the VC is to be established), the propagation of the SETUP-F messages is stopped, and CNCL-B messages are transmitted in the reverse direction. The purpose of these messages is to release the resources that have been already allocated to the VC.

For instance, Fig. 10(a) shows the case where F receives from D the SETUP-F message and S-VC(F→H) is down or resources for this VC cannot be allocated by H. Thus, F sends a CNCL-B to D over S-VC(F→D). Upon the receipt of the message, node D releases the resources allocated to the VC and the State of VC-Table(D_C)[d_4] changes from frozen to free. Then, D sends a CNCL-B message to A to continue the takedown protocol, and a CNCL-F message to F as an acknowledgment. As shown later, a CNCL-F message is always needed if the VC is taken down after the setup is completed, in order to let the downstream VC node that the state of the entry
in tandem from the originating node $A$ to node $D$ (over $S$-$VC(A \rightarrow D)$), from $D$ to $F$ (over $S$-$VC(D \rightarrow F)$) and from $F$ to $H$ (over $S$-$VC(F \rightarrow H)$). During this phase the resources needed by the to-be-established VC, namely bandwidth and routing table entries, are reserved. In the second phase, SETUP-B messages are sent in the reverse direction: from $H$ to $F$ (over $S$-$VC(H \rightarrow F)$), from $F$ to $D$ (over $S$-$VC(F \rightarrow D)$) and from $D$ to $A$ (over $S$-$VC(D \rightarrow A)$). During this phase the setup of the routing tables is completed. The following is a detailed description of the protocol.

The Network layer of $A$ initiates the setup protocol by performing the following steps:

- Select a free entry, $a_4$ say, in VC-Table($A_A$), and put it in frozen state.
- Allocate the required bandwidth in VP($A \rightarrow D$).
- Send a SETUP-F message to node $D$ on S-$VC(A \rightarrow D)$. This message contains the VC-id of the to-be-established VC, as well a description of the route over which the VC should be established.

Upon receiving the SETUP-F message, node $D$ performs as follows:

- Select a free entry, $d_4$ say, in VC-Table($D_C$), and put it in frozen state.
- Allocate the required bandwidth in VP($D \rightarrow F$).
- Send a SETUP-F message to node $F$ on S-$VC(D \rightarrow F)$.

Upon receiving the SETUP-F message, node $F$ performs as follows:

- Select a free entry, $f_4$ say, in VC-Table($F_E$), and put it in frozen state.
- Allocate the required bandwidth in VP($F \rightarrow H$).
- Send a SETUP-F message to node $H$ on S-$VC(F \rightarrow H)$.

Upon receiving this SETUP-F message, the Network layer at $H$ performs as follows:

- Select a free entry, $h_4$ say, in VC-Table($H_G$), and put it in frozen state.
- Select an available demultiplexing label $[h_1, h_2]$ for the created VC.

Now, the second phase of the protocol is started by node $H$ which performs as follows:

- Set VC-Table($H_C$)[$h_4$] $\leftarrow (h_1, h_2, H_H)$ and put this entry in used state. Recall that $h_1$ and $h_2$ are the two parts of the demultiplexing label selected by $H$.
- Send SETUP-B($h_4$) message to $F$ on S-$VC(H \rightarrow F)$.

Upon receiving this message, the Network layer at $F$ performs as follows:

- Set VC-Table($F_E$)[$f_4$] $\leftarrow (g_2, h_4, F_G)$ and put this entry in used state. The value $h_4$ is known from the SETUP-B message received from node $H$. The value $g_2$ is the VPI used by $F$ for all the cells transmitted over VP($F \rightarrow H$). This value was selected during the setup of this VP.
- Send SETUP-B($f_4$) message to $D$ on S-$VC(F \rightarrow D)$.

Upon receiving this message, the Network layer at $D$ performs as follows:
In addition to the [VPI,VCI] pair that identifies a VC uniquely at each switch for cell routing purposes, we assume that a VC is also identified by a global identifier, referred to as VC-id, which has a global significance across the ATM network. The latter may be created, for example, by concatenating the originating node identity to the VC call-reference assigned by the originating node ([14]). All VC related signaling messages contain the global VC-id to indicate the VC they belong to.

The VC setup protocol will be presented by means of an example. Consider Fig. 8, where three existing VP connections: VP(A→D), VP(D→F) and VP(F→H), are supposed to serve as the building blocks of a VC from A to H. The associated pairs of signaling VC’s: S-VC(A→D), S-VC(D→F) and S-VC(F→H), exist as well, and will be used for the exchange of the signaling messages between A and D, between D and F and between F and H respectively.

As said earlier, in order to avoid details related to the User-Network-Interface (UNI), it is assumed in the following description that A and H are not only the network end nodes of VC(A→H), but also the actual source and destination of this VC. If this were not the case, a UNI protocol would be needed between node A and its host, and between node H and its host, in addition to the NNI protocol between A, D, F and H.

As Fig. 8 shows, the protocol has two phases. In the first phase SETUP-F messages are sent
4 VC Setup and Cancelation

4.1 Virtual Channel Setup

The VC setup protocol establishes a unidirectional VC connection over the VP network. The protocol suggested here is performed by a two-way handshake of SETUP-F and SETUP-B signaling messages and is similar to the protocol proposed in [13] for non-ATM virtual circuit networks. These messages are exchanged by the Network layers of the VP endpoints over the corresponding S-VC’s.
The cell is received by PP($B_A$), that uses the received VPI=4 as a pointer to VP-Table($B_A$). Entry 4 of that table indicates that the cell should be transmitted on link $BC$ with VPI=8. The VCI does not change in this case. Similarly, PP($C_B$) finds in entry 8 of its VP-Table (not shown in the figure) that the cell should be transmitted on link $CD$ with VPI=17. PP($D_C$) receives the cell, refers to entry 17 in VP-Table($D_C$) and finds$^2$ VP-Table($D_C$)[17].VPI =$nil$. This indicates that PP($D_C$) should refer to its VC-Table using the received VCI=6 as a pointer. In VC-Table($D_C$)[6] it finds the new VPI=12, the new VCI=18 and the link $DE$ over which the cell should be transmitted. Note that VCI=6 is used over VP($A\rightarrow D$) and VCI=18 is used over VP($D\rightarrow F$).

The cell now travels along VP($D\rightarrow F$), while the VPI is swapped, until it reaches switch $F$ with VPI=35 and VCI=18. It is then forwarded to PP($F_F$), and from there to the AAL, with VPI=8 and VCI=5. The couple (8,5) is referred to as the demultiplexing label and is used by the higher layer to determine the AAL to which the cell is to be forwarded.

### 3.3 The State of an Entry and the PP Algorithm

Every entry in a VC-Table has an additional field, called State, that not shown in Fig. 5. This field indicates whether the entry is (a) used, (b) free, or (c) frozen. A used entry is an entry employed by an active VC. Such an entry cannot be allocated to a new VC connection and cells arriving with the VCI associated with this entry are routed according to the contents of the entry. A free entry is an entry not used by any VC connection. Such an entry can be allocated to a new VC connection and cells received with the VCI associated with such an entry are discarded. Although normally no cells with the VCI of a free entry are received, this might happen due to some failure. Frozen is a state used sometimes during the transition from used to free, in order to indicate that on one hand such an entry is used by a VC and, therefore, cannot be allocated to a new VC, but on the other hand cells received with the VCI associated with such an entry should be discarded.

Transitions of an entry, as shown in Fig. 6, occur during VC setup, takedown and reroute and are controlled by the Network layer. During the setup protocol, the State of an entry changes from free to frozen and then to used. During the takedown protocol, the State of an entry changes from used to free either directly or through the frozen state, depending on the side of the VC from which the takedown is initiated. During the rerouting protocol, the status changes from used to frozen upon the failure of the VP. If the VC can be rerouted, the status changes back to used. Otherwise, it changes to free.

$^2$We employ Pascal-like notations; e.g., the VPI field of entry 6 of VC-Table($D_C$) is denoted VC-Table($D_C$)[6].VPI.
This will eliminate the necessity to consider in this paper the interaction between the Network layer and the Control Processor.

### 3.2 Cell Routing

In the following we illustrate routing of ATM cells over a virtual channel connection, \( VC(A\rightarrow F) \), which has already been set up over two VP connections: \( VP(A\rightarrow D) \) and \( VP(D\rightarrow F) \) (Fig. 5). In this example, as well as in those presented later, it is assumed that the end points of the VC connections are network nodes, rather than user hosts, that need to exchange information (e.g. routing information). This is because we want to concentrate on NNI issues, while avoiding unnecessary details regarding the UNI.

Cells belonging to \( VC(A\rightarrow F) \) in Fig. 5 are submitted to \( PP(A_A) \) with VPI=3 and VCI=10. The [VPI,VCI] pair used by the upper layer for a connection will be referred to as the *multiplexing label*. \( PP(A_A) \) uses the received VPI part of the multiplexing label as a pointer to its VP-Table. It finds in the VPI field of entry 3 the value nil, which indicates that this is a start or an end point of a VP connection. This implies that \( PP(A_A) \) should refer to its VC-Table using the received VCI=10 as a pointer. The values found in entry 10 of VC-Table\( (A_A) \) are \( (4, 6, AB) \) and therefore \( PP(A_A) \) changes the VPI and VCI fields of the cell to 4 and 6 respectively, and transmits the cell, via \( PP(A_B) \), on the output link \( AB \).
the routing tables in a VC node. For instance, since one VC-Table can be associated with every VP-Table entry, multiple VC-Tables can be employed by every PP. A discussion regarding the advantages and disadvantages of every possible architecture is beyond the scope of the present paper.

When a PP receives a cell on its link, it uses the VPI/VCI labels of the cell to determine the VP-Table/VC-Table entries to be accessed. The contents of these entries indicate the new values of the VPI/VCI labels and the port to which the cell is to be transferred. The cell is locally routed to this port by the internal interconnection network and then transmitted over the corresponding link.

Every node contains a Control Processor that updates the various tables maintained by the Port Processors. Since the Control Processor is informed about the required changes by the Network layer, it will be convenient to refer to the Network layer as the entity that updates these tables.
The original network

setup of VP connections

the resulting VP network

setup of VC connections

Figure 3: Network Example

associated pair of switched S-VC’s from A to D and vice versa, denoted S-VC(A→D), over which A and D can exchange connection signaling messages for setup, tear down and rerouting of VC connections.

Assumption 2 The switched S-VC’s between every two end points of a VP are in associated mode. This implies that if a VP is available, the associated pair of S-VC’s are available as well.

Note that unidirectional connections are denoted by → (for example VP(A→D)) while pairs of such connections are denoted by ← (for instance S-VC(A←D)).

3 ATM Switch Model and Routing Algorithm

3.1 Conceptual Structure of an ATM Switch

The ATM layer functions are performed by high-speed hardware called the ATM switch. A conceptual structure of an ATM switch for a VC node is shown in Fig. 4. The switch we consider consists of several independent Port Processors (PP’s). There is one PP associated with every adjacent link. The PP at node A associated with the link to and from B is denoted by PP(A_B). One PP is associated with the upper layer, and is responsible for the traffic generated by the node or destined for the node. This processor at node A is denoted PP(A_A).

A PP of a VP node maintains for routing purposes a VP-Table, containing up to $2^{12}$ entries. A PP of a VC node maintains two routing tables: a VP-Table (containing up to $2^{12}$ entries) and a VC-Table (containing up to $2^{16}$ entries). The VP- and VC-Tables at PP(A_B) will be denoted by VP-Table(A_B) and VC-Table(A_B) respectively. One may consider other ways for maintaining
their routing tables, and therefore their implementation is less expensive. Due to this distinction, VP nodes can be used only as building blocks of VP connections, whereas VC nodes can be used as building blocks of both VP and VC connections.

As said before, we assume that between every two neighboring nodes there exists a permanent S-VC in each direction. These pairs of S-VCs are used in order to set up VP connections and switched S-VCs between pairs of VC or VP nodes. There also exists a pair of switched S-VC’s between every two VC nodes which serve as the endpoints a VP connection. Such a pair of switched S-VC’s may or may not be established on the same route as the VP. In the first case it is said to be in associated mode, whereas in the second it is in quasi-associated mode. For simplicity, we shall assume that every (unidirectional) VP has a pair (one on each direction) of switched S-VC’s in associated mode.

The VP connections induce a VP network, which is embedded in the original network. Fig. 3 demonstrates this concept. Fig. 3(a) shows the original network, consisting of both VP and VC nodes. The various VP connections, established in the network, are shown in Fig. 3(b). Each VP connection has an associated switched S-VC (not shown in the figure). The resulting embedded VP network is shown in Fig. 3(c), and three VC connections set up over the VP network are shown in Fig. 3(d). As shown in Sec. 4 and 5, connection signaling messages used by the protocols for setup, takedown and rerouting of VC connections are exchanged over the switched S-VC connections associated with the VP’s used by the considered VC’s.

The following is a summary of the assumptions made during this section regarding the signaling network:

**Assumption 1** If there exists a VP from A to D, denoted VP(A → D), then there exists an
Sec. 5 shows how VC connections that cannot be accommodated by the backup VP, and therefore cannot be rerouted, are taken down. Sec. 6 concludes the paper.

All presented protocols apply to the Network-Network-Interface (NNI), since we view the rerouting mechanism as an NNI tool.

## 2 Signaling in ATM

### 2.1 Signaling Virtual Channels and the SAAL

An ATM network consists of ATM nodes connected to each other by physical links. Two ATM nodes are said to have a signaling relation if they are connected to each other by a Signaling VC (S-VC)\(^1\), over which the two end nodes can send signaling messages to each other. Throughout the paper it is assumed that signaling relations are bidirectional, while VP and VC connections are unidirectional.

Fig. 2 depicts the signaling ATM Adaptation Layer (SAAL) according to draft ITU-T recommendation Q.SAAL.0[10]. The SAAL makes use of the service provided by the Common Part Convergence Sublayer (CPCS) and Segmentation and Reassembly (SAR) Sublayer, which form the common part of AAL type 5. The Service Specific Connection Oriented Protocol (SSCOP) is used to transfer variable length Protocol Data units (PDUs) between users of SSCOP. It provides for the recovery of lost and corrupted units. The Service Specific Coordination Function (SSCF) has several types. One of them is used in order to map particular requirements of the NNI Network layer protocol (MTP level 3 of B-ISDN) to the requirements of the SSCOP. Thus, the primitives required to support the SAAL user (MTP-3) are the same as provided by MTP level 2 to MTP level 3 in B-ISDN [1, 3, 4, 7, 8, 12].

To simplify the discussion in this paper, we shall use the term “Network layer” in order to refer to both level 3 (i.e. MTP level 3) and level 4 of the B-ISDN functional levels. Using the services provided by SAAL, two nodes that have an S-VC can send to each other signaling Network layer messages with no loss, misordering or duplication as long as the S-VC is alive.

### 2.2 The Embedded VP Network

In an ATM network, there is a distinction between VC nodes, that are capable of interpreting both the VCI and the VPI parts of the cell header, and VP nodes, that are capable of interpreting only the VPI part. The reason for such a distinction is that VP nodes need less high-speed memory for

\(^1\)Note that S-VC stands for “Signaling-VC” rather than for “Switched-VC”
VP(\(A \rightarrow D\)) fails. This would result in a failure of VC(\(A \rightarrow F\)) as well as of all other VC connections that use VP(\(A \rightarrow D\)) as a building block. In large networks there may be a large number of such connections and if each is independently rerouted (i.e. taken down and then set up over an alternative route) by its end nodes, the burden on the network in terms of communication and processing might be excessive. The present paper suggests an alternative approach, where the affected VC connections are rerouted by a common protocol to an alternative predefined VP connection.

Although the VC survivability provided by VP connections in ATM networks has been discussed in the past in various papers [5, 11], the present paper is the first to give a detailed rerouting procedure, to address the various difficulties, and to show that all the VC connections established over a VP connection can be rerouted together using a single protocol. Due to the simplicity of the protocol, it can be completed within a short time, thus fulfilling an essential requirement of any rerouting scheme.

Another important contribution of the present paper is improving the reliability of signaling VC connections in ATM networks. In ATM, signaling VC connections can be employed instead of a dedicated common channel signaling (CCS) network [2]. In such a case, however, it is important to meet at least the reliability of existing CCS networks. CCS networks are based on SS7 protocols, which uses changeover and retrieval mechanisms in order to recover from loss of signaling messages due to link failures [3, 4, 7, 8]. A Signaling VC (S-VC) connection associated with a failed VP connection cannot be rerouted to an alternative VP, because such an action would delay the rerouting of non-signaling VC connections. Nevertheless, the paper shows that the two end points of the failed VP can recover from loss of connection signaling messages sent over the failed signaling VC associated with the VP, such that the loss will not affect connection signaling protocols. To this end, the paper presents connection signaling protocols for VC setup and takedown, and shows how loss of signaling messages sent by these protocol can be handled.

The organization of the rest of the paper is as follows. Sec. 2 describes some signaling aspects in ATM networks. Sec. 3 presents a model of an ATM switch: it shows the structure of the switch routing tables and gives the switch algorithm. Sec. 4 presents protocols for VC setup and takedown. As already said, these protocols are described in order to show in Sec. 5 how they use the rerouting protocol in order to recover from loss of connection signaling messages. Sec. 5 presents the rerouting mechanism. It first shows how the rerouting protocol reroutes active VC connections to a backup VP. Then, this section shows how the rerouting protocol enables other signaling protocols (i.e. the VC setup and takedown protocols) to recover from loss of connection signaling messages. Finally,
1 Introduction

In ATM networks, information is carried in fixed size packets, called cells. ATM cell transport requires a connection. Two levels of ATM connections are defined: virtual path (VP) connections and virtual channel (VC) connections. Routing of cells is performed by label swapping at the intermediate nodes. In the NNI (Network Network Interface), the label consists of an 8-bit VPI (Virtual Path Identifier) and a 12-bit VCI (virtual channel identifier). A virtual path is similar to a virtual circuit in traditional networks. A virtual channel is also similar, except that its building blocks are virtual paths rather than physical links.

The approach of two connection levels, known as the virtual path concept, has several advantages. In order to illustrate these advantages, consider Fig. 1. The figure shows two unidirectional virtual path connections: VP\((A \rightarrow D)\), from \(A\) to \(D\), and VP\((D \rightarrow F)\), from \(D\) to \(F\). These virtual path connections are used as building blocks by a virtual channel connection, VC\((A \rightarrow F)\), from \(A\) to \(F\).

One advantage of the virtual path concept is that only a small number of nodes along the route of a VC are involved in the VC setup and takedown protocols. These protocols need to be executed only by the end nodes of the VP connections used by the VC. For example, the virtual channel between \(A\) and \(F\) in Fig. 1 is established by a VC setup protocol executed by nodes \(A\), \(C\) and \(F\) only. This reduces the processing load at network nodes and expedites the VC setup and takedown protocols.

Other advantages of the virtual path concept, as discussed in [6, 9, 11], are as follows. The concept reduces the processing of cells and the size of the high-speed routing tables in the network nodes. It simplifies network architecture and management, and it facilitates dynamic control of bandwidth allocation.

As shown in the present paper, the VP concept has the additional advantage of increase survivability of VC connections. Consider Fig. 1, and suppose that an intermediate link or node of

\[\text{Figure 1: The Virtual Path Concept}\]
Connection Management and Rerouting in ATM Networks

Reuven Cohen and Adrian Segall* 
Dept. of Computer Science 
Technion, Haifa 32000, Israel 

December 18, 1994

Abstract

In ATM networks, two levels of connections are defined: virtual path connections and virtual channel connections. Virtual paths are the building blocks of the virtual channels. This hierarchy, known as “the virtual path concept”, provides considerable advantages in the design of high-speed networks. In this paper we show that this concept has the advantage of providing a means of improving the survivability of virtual channels. To this end, we present a NNI rerouting protocol, which can be invoked upon the failure of an intermediate link or node of a virtual path. This protocol reroutes all the affected virtual channels to an alternative virtual path. In addition, the protocol enables the two end points of the failed VP to recover from a possible loss of connection signaling messages. Due to the simplicity of the protocol, it can be completed rapidly, thus fulfilling an essential requirement of any rerouting scheme.

*This research was supported by the Fund for the Promotion of research at the Technion