Faculty of Computing and Information Technology
Department of Digital Systems

Technical Report 96-5

Caroline Flood Routing Technique

- Forward Difficulty Metric Method -

Jai Hyung Cho , James Breen

January, 1996

Enquiries:-

Technical Report Coordinator
Digital Systems
Monash University
Clayton VIC 3168
AUSTRALIA

tr.coord@dgs.monash.edu.au +61 3 9905 3402

Contents

Abstract

Reverse Metric Flooding is a routing method designed for the Caroline Network [4]. This method uses a flooding technique for setting up a virtual circuit along an optimal route. It is a novel approach in that the virtual circuit is established without using a pre-established routing table. However the complexity of the routing mechanism and the uncertainty of the time out periods requires a further improvement of the algorithm.

A revised technique; The Forward Difficulty Metric Flooding, is proposed in this document. This method is expected to be simple and more efficient than the previous method. In particular, it reduces many timer problems which are sensitive to network size and transit delay, and it also lends itself more readily to extension to support multicast routing applications.

Keywords

Network, Communication, Routing, Protocol, Caroline, Forward Difficulty Metric Flooding, Flood Routing, ATM, Asynchronous Transfer Mode, LAN, Local Area Network, Virtual Circuit Connection

1. Introduction

1.1 Scope

This document describes the routing mechanism and network layer protocols for the Caroline Network. An introduction on flood routing, protocol description and discussion of major issues in developing the protocol will be presented. ISO standard style FSM tables and equivalent state transition diagrams are provided to give a formal description of the protocol actions. Physical implementation of routing algorithms, the hardware structure of the Caroline Network, and the inter-relations with higher layers are out of the scope of this paper.

1.2 Terminology

Packet : A Packet is a basic message unit of information transfer on communication medium.

Cell : A cell is a fixed length packet which is treated as a basic unit of asynchronous transfer mode.

Network Node : A Network Node is a nodal point in describing a communication network. The meaning generally encompasses both end user equipment and switching device.

Terminal Node: A Terminal Node is an end user equipment or a group of equipment attached to the switching node.

Transit Node: A Transit Node is a switching device which links Terminal Nodes and other Transit Nodes.

2. General Features of Flooding as a Routing Method

Flooding is a routing technique which sends duplicates of a packet to all output links of a communication node. Once a source node in network initiates a data transmission, packets are copied and transmitted to every link on each communication node, and flooded to the entire network. As a result, all network users receive at least one copy of the original packet [6]. Without a strict controlling mechanism, flooding packets can saturate the network and for this reason, flooding has often been regarded as an inefficient method [5]. However, because of its inherent simplicity and robustness, flooding has been applied in a few areas [7][8][9][10].

In this section, the general advantages of the flood routing methods are introduced.

a) Simplicity : Pure flood routing does not rely on the pre-established routing knowledge. This helps to ease many routing problems. In adaptive routing methods based on routing tables, it is a difficult issue to reflect the changes of network configuration into the routing table in each node and maintains the integrity of entire network image. Therefore most such methods implement sophisticated routing table maintenance algorithms. This is not a problem in a flood routing network because the network nodes do not use a routing table and are not aware of the location of the destination, the network configuration. This makes the routing and control algorithm simple.

b) Robustness : A flooding network is inherently robust because the network produces many copies of the original packet and it guarantees at least one copy of the packet reaches the destination provided a path exists. In particular, if a network is configured in a way that more than one path exists between every two nodes, the network connection becomes very reliable because an alternative route may be taken without any recovery mechanism.

c) Fast Routing : Generally speaking, the routing speed of the flooding network is faster than the table based routing. It is because most conventional routing algorithms requires significant routing table referencing time. Pure flooding does not need the table searching time and it can make the switching speed fast. This is useful in some high speed, local area networks which require rapid route set up.

d) Configuration Flexibility : The flooding network is relatively easy to set up and change the network topology. It can be configured to any shape of network, such as mesh, tree, ring or star, as the flooding network does not have a routing table to maintain strict configuration information.

3. Overview of the Caroline Network

3.1 Introduction

The Caroline Network is an experimental cell switching local area network which is being developed by Department of Digital Systems (formerly Robotics and Digital Technology), Monash University. The network employs a mesh structure to provide a reliable and scaleable communication medium and it is similar in concept to the ITU ATM. Within a node, cell switching activity over virtual circuit connections is achieved by the hardware without direct control of the CPU. The initial switching fabric is a bus structure and uses output buffer for queuing cells and consists of 7 input/output lines which are each capable of transmitting data cells at a maximum rate of 2 Mbps [3]. The network management and routing software uses a flooding technique to find an optimal route in a fully distributed and dynamic fashion. More details of the Caroline Network are written in departmental technical reports [1][2][3].

3.2 Protocol Architecture

The Caroline protocol architecture is depicted in Figure 3.1. There are two types of nodes defined in the Caroline architecture. One is a Terminal Node of which the major roles are providing the user interface, and initiating and closing the connections. Typically the Terminal Node will be implemented in workstations or user devices. The other type of node is a Transit Node which is the Caroline switching fabric and it carries out the flooding and cell switching operations.

Three layers are defined in the Caroline protocol architecture - CAL (Caroline Adaptation Layer), CCL (Caroline Cell Layer) and CPL (Caroline Physical Layer). The CCL layer provides an interface to the upper layer through which independence from the network routing and management details are hidden. This layer is responsible for setting up, maintaining and terminating the virtual circuit connections. The CPL is concerned with the fixed size cell switching activity and the transmission of bit streams over a physical medium.

Once a virtual circuit route is established, the normal data cell switching activity is carried out by the CPL entity. The higher layer adaptation functions, including segmentation and reassemble, are provided by the CAL. The Transit Node only encompasses the CPL and CCL, however the Terminal Node will include the CAL.

This document describes the CCL protocol and associated algorithms. The CCL service descriptions and primitives are defined in [1] and will not be presented again in this document. Since the CCL action represent the Terminal Node in a network, the term 'Terminal Node' will also mean 'the CCL entity' in this document.

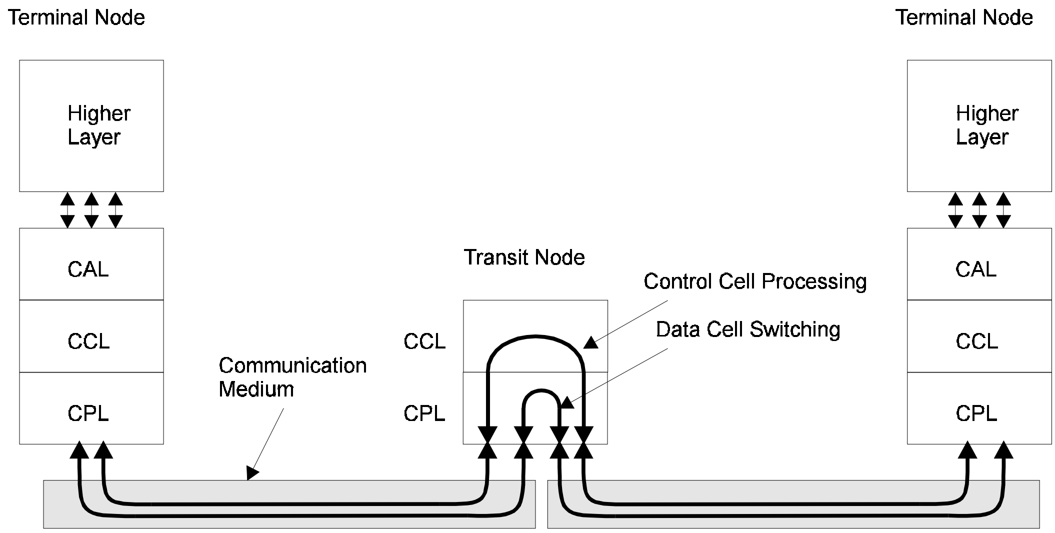


Figure 3.1 Caroline Protocol Architecture

4. Forward Flood Routing Description and Protocol

4.1 Design Goal

The Caroline protocol design aims to satisfy several criteria. This section introduces these design goals.

4.1.1 Light Weight Protocol

The underlying assumption in designing the Caroline protocol is that the network is fast and reliable enough that only the minimum set of protocol functions need to be implemented in the CCL and CPL protocol. For example, link by link functions of error detection, error recovery, and flow control do not exist in the CCL nor CPL protocols. These functions would typically be provided by the end to end protocol of the Terminal Nodes. This is the same assumption as the ATM concept that the data switching speed is increased by eliminating the redundant link by link protocol functions.

4.1.2 Robust Protocol

The Caroline Network may have occasional cell loss or node/link failures. The protocol is required to work in a stable condition even in worst cases. Usually it is the control cells which are critically affected by the cell losses, as loss of data cell can be detected and handled by higher layer protocols. Although the underlying layer of the CCL, which is the CPL, does not check or report the loss of control cells, the CCL should be prepared for any protocol errors caused from the cell losses. However this should not make the protocol complex. In summary, the protocol should be sophisticated and reliable, with keeping the simplicity.

4.1.3 Efficient Use of Network Resource

The network resource control become an important issue as the capacity of the network link is increasing. Although the current Caroline Network is a small-scale, experimental network, the protocol itself needs to maximize the utilization of the resources. This can be achieved through various ways such as an optimal choose of the route, reducing the control complexity and call blocking rate, increasing the protocol processing speed, etc.

4.2 Routing and Control Cell Structure

This section introduces the cell formats defined in this version of the protocol. Currently, 5 control cells and 1 data cell are defined. Among them, 4 control cells are used for connection management, 1 control cell (LSTA) is for link management and 1 data cell is for normal data transmission. All cell types are fixed length of 16 byte size and have the first octet for virtual circuit number field. The control cells have the virtual circuit number zero. Data cells have a virtual circuit number ranging from 2 to 31.

Field Name	Size
VC Number	8bit
Cell Type	8bit
CDM	8bit
Source Address	16bit
Source Conn #	8bit
Dest. Addr. Type	2bit
Dest. Address	~78bit

Table 4.1 CREQ Cell Format

It is the CREQ (Connection REQuest) cell that is duplicated for flooding upon a connection request of a Caroline service user. Once a Terminal Node sends a CREQ cell to the adjacent Transit Node, the cell is duplicated and disseminated to the neighbor nodes according to the Caroline Flooding rule.

The 8 bit 'CDM (Connection Difficulty Metric)' field contains a value representing a measure of the route difficulty. This field is used for controlling the number of CREQ cells and optimal choice of the route. Whenever a CREQ cell goes through a link, the link difficulty value, e.g. link cost, is added to the CDM value. As a consequence, the CDM value shows the route difficulty which is the aggregated link difficulty value.

The 'Source Address' field contains the 16 bit Caroline Address. 'Source Conn #' is an unique random number chosen by the source Terminal Node. This number is necessary to distinguish the CREQ cells of the different connections between same source and destination nodes. See section 5.4 for more details.

The CREQ cell is also designed to carry different types of destination addresses. The 2 bit 'Dest. Addr. Type' field indicates what type of address is contained in the 'Destination Address' field. According to this address type field, the 'Destination Address' field can carry an address of up to 78 bits long. Currently determined address types are 16 bit for native Caroline address and 32 bit Ipv4 address.

Field Name	Size
VC Number	8bit
Cell Type	8bit
Reference VC #	8bit
Source Address	16bit
Source Conn #	8bit
Dest. Address	16bit
Padding	8byte

Table 4.2 CACC Cell Format

The CACC (Connection ACCeptance) cell is created by a destination Terminal Node upon the call acceptance of the Caroline service user. A virtual circuit is allocated to establish a new connection and the selected circuit number is written into the 'Reference VC#' field. More details about the allocation of the virtual circuit number will be provided in section 5.6.

The 'Source Address' field and the 'Source Conn #' field contains the same number as the CREQ cell. However the 'Dest. Address' field of the CACC cell will have the Caroline Address of the destination Terminal Node.

Field Name	Size
VC Number	8bit
Cell Type	8bit
Source Address	16bit
Source Conn #	8bit
Dest. Address	16bit
Reason	8bit
Padding	8byte

Table 4.3 CCLO Cell Format

The CCLO (Connection CLOse) cell is primarily used for clearing the connections by both the Terminal Nodes and the Transit Nodes. The Terminal Node can send the CCLO cell to disconnect a connection to refuse the call request, or to reply a short message without making a connection The Transit Node also can close the connection. In any case, the 'Reason' field carries a message such as the disconnect reason. All the rest of the fields have the same meaning as the CACC cell format.

Field Name	Size
VC Number	8bit
Cell Type	8bit
Reference VC #	8bit
Source Address	16bit
Source Conn #	8bit
Dest. Address	16bit
Rcvd Cell #	32bit
Proposed Rate	16bit
Padding	2byte

Table 4.4 VCST Cell Format

The VCST (Virtual Circuit STatus) cell is a virtual circuit management cell which is regularly transmitted by the source and the destination Terminal Nodes during the life of a virtual circuit. The 'Reference VC #', 'Source Address', 'Source Conn #' and 'Dest. Address' fields have the same meaning as the CACC cell format. As for the details of the VCST mechanism, see sections 4.4.5 and 5.2.

The VCST cell also provides a mechanism to report status information between the Terminal Nodes and the Transit Nodes. The 'Rcvd Cell #' and 'Proposed Rate' fields are reserved for the traffic monitoring and control.

Field Name	Size
VC Number	8bit
Cell Type	8bit
Sender Address	16bit
Padding	12byte

Table 4.5 LSTA Cell Format

The LSTA (Link STAtus) cell is a management cell used for checking the connectivity between the adjacent nodes. While the Caroline network nodes are active, all the nodes send the LSTA cell to the neighboring nodes and check the link and node availability regularly. The details of the LSTA cell will be explained in sections 5.2 and 5.6.

Field Name	Size
Payload Indicator	1bit
VC Number	7bit
Payload	15byte

Table 4.6 DATA Cell Format

The DATA cell is used for normal data transmission. The 'Payload Indicator' bit is used for the segmentation and reassembly process in the CAL. Virtual circuit number 1 is reserved for network wide broadcasting. Currently, the available virtual circuit number space for normal connection is between 2 ~ 31.

4.3 Virtual Circuit Set Up and Release procedure

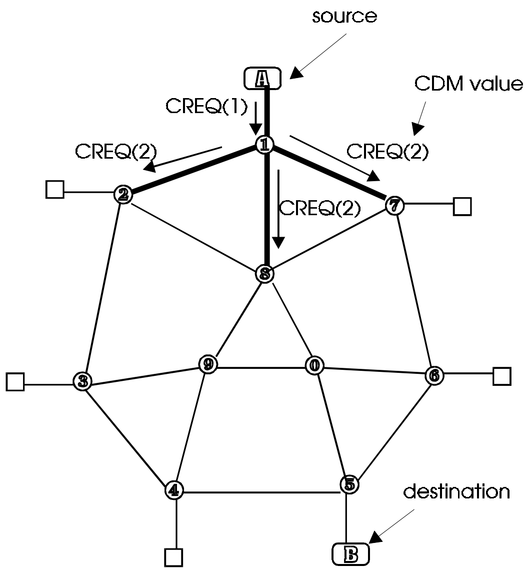


Figure 4.1

In this section, an example of the Forward Flooding on the model network is described. In Figure 4.1, the flooding starts from the Terminal node A with sending a CREQ cell to the Transit Node 1.

The CREQ cells have the CDM value fields which were shown between the parentheses in the diagram. When the Transit Node 1 receives the CREQ cell, it first increases the CDM value. The amount of CDM increase depends on the route difficulty metric set by the network manager. Second, the node checks the internal flooding record to see if a CREQ cell of the same flood has been received before. If this is the first received CREQ cell for this flood, the node records the details of the flood, such as the incoming link number, the source and destination addresses, etc., and sends the copies of the CREQ cell to all output links except the received one. This 'copy and spread' action is the basic flooding function. In Figure 4.1, the CREQ cells flood out to the adjacent nodes 2, 7, 8. The CDM values of the output cells have been increased to 2.

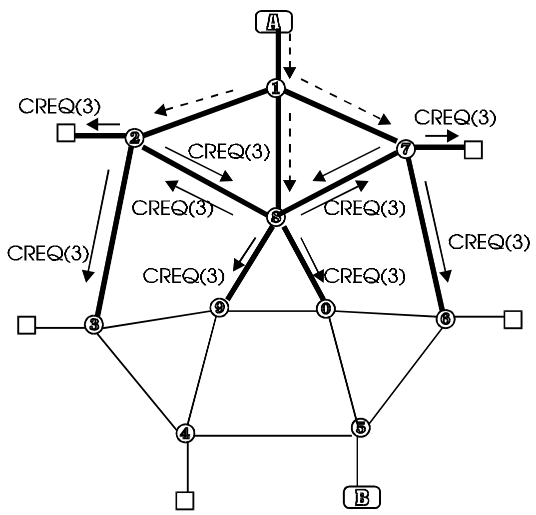


Figure 4.2

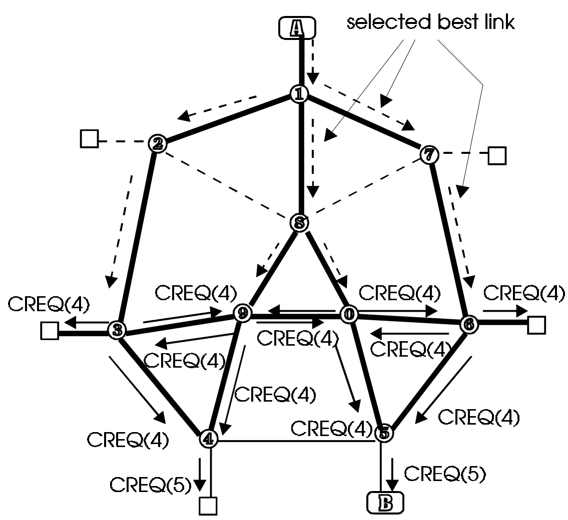


Figure 4.3

In Figure 4.2, the Transit Nodes 2, 7, 8, received the CREQ cells from the transit node 1. Since these are the first arriving CREQ cells to the nodes, they repeat the same flooding action as above in each of the nodes. As a result, some Terminal Nodes (shown as a small square box in the diagram) receive the CREQ cell, but ignore it because the destination address is not their addresses. Some Transit Nodes such as nodes 2, 7, 8, receive the CREQ cells again which were duplicated by their neighbors. Since these are the repeats of the CREQ cells, the nodes check their internal record and compare the CDM values. If the CDM value is not better than the previous record, they discard the received CREQ cell, otherwise they accept the new CREQ cell, update the internal record, and repeat the flooding. Through this CDM comparison and flooding update process, the Transit Nodes can control the number of flooding cells and keep the optimal CDM record.

In Figure 4.3, the flood of CREQ cells has swept the entire network and finally some CREQ cells have reached the destination Terminal Node B.

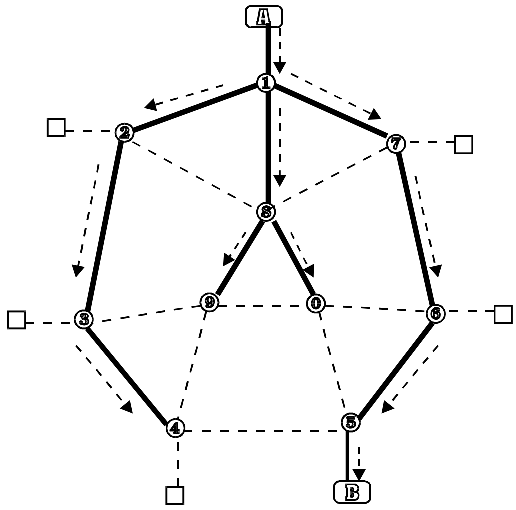


Figure 4.4

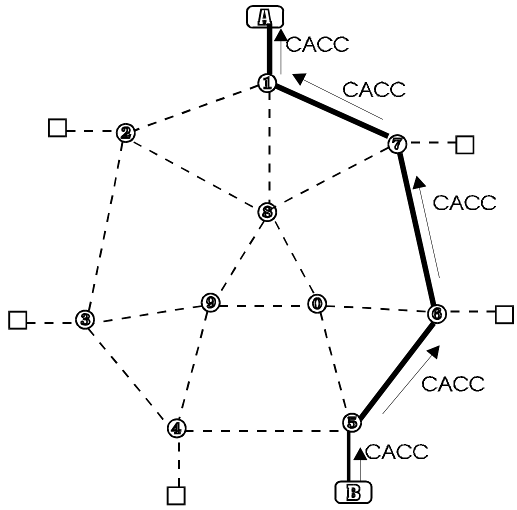


Figure 4.5

When the flood of the CREQ cells subside, all the Transit Nodes are keeping a record including the best CDM value and relevant incoming link number. Note that once these best incoming links are put together, they consist a shortest path tree(or lowest CDM tree), e.g. Figure 4.4.

Finally the destination Terminal Node B accepts the call request and sends the CACC cell to the adjacent Transit Node 5. At the time of sending the CACC cell, the destination node allocates a virtual circuit. The selected virtual circuit number is carried by the CACC cell.

When a Transit Node receives the CACC cell, the node sets up the virtual circuit route and passes the CACC cell to the next node which had sent the CREQ cell of the best CDM value. Consequently, the CACC cell is passed through one branch of the shortest path tree in Figure 4.5. This shows that the route set by the Forward Flooding procedure is the optimal route.

Finally, the connection procedure is completed by the source node sending the first VCST cell or DATA cell to the destination after the route is established.

4.4 Timer Definition

In this section, definitions of timers and their roles are explained. Currently, 5 timers are defined. Among them, T1,T2,T6,T7 are used in the Terminal Node and only one T5 timer is used in the Transit Node for the virtual circuit management. T3, T4 timers are reserved for link management. The time quantum relation between timers are :

$$T6 << T1 < T7 < T5 < T2$$

Symbol	Name	Using Node
T1	Connection Timer	Source Terminal
T2	Destination Timer	Destination Terminal
T3		(unused)
T4		(unused)
T5	Transit Timer	Transit Node
T6	Route Improve Timer	Destination Terminal
T7	VCST Timer	Source/Destination Both

Table 4.7

4.4.1 T1, Connection Timer

T1 sets the maximum waiting time of the source Terminal Node for connection establishment.
This timer starts with the transmission of the CREQ cell and stops on reception of destination node's response cells - CACC or CCLO. The expiration of this timer indicates a connection failure, and the source node announces total connection failure after three failed connection attempts.

For any given connection request, the time quantum of T1 starts from a few millisecond and increases exponentially at each connection re-trial by the following formula :

$$T1_i = a * b \hat{}\, i \ , \ (i=0,\ 1,\ 2 \ldots.)$$

where, a = initial time value, b = exponential growth base

4.4.2 T2, Destination Timer

T2 sets the maximum waiting time of the destination Terminal Node for the connection activation. The destination node cannot send a data until the connection is activated. This timer starts with the transmission of the CACC cell, and stops either when the destination user refuses the connection, or receives the first VCST cell or DATA cell .

The source node sends the VCST cell or DATA cell in response to the CACC reception and the connection is activated. If the destination node does not receive any of these responses before T2 expires, the node cancels the connection. The time quantum of the T2 should be sufficiently long enough to cover the sum of the T1 times and the network delay.

The CACC cell can be lost or corrupted during the reply and the T1 connection timer of the source node may expire. In this case, the source node will try the new flood with a different connection number. This new flood will reach to the destination as a new connection request and the destination node reply the CACC cell with setting another T2 timer. The previous T2 timer will soon expire and the failed connection will be removed.

4.4.3 T5, Transit Timer

T5 is the periodic timer which tells the Transit Node of the maintenance time for checking the inconsistent connection information and corrects them. This timer starts at the initialization time of the Transit Node and is continually re-armed again after the expiration.

The VCST cell is used for maintaining the active virtual circuit information. Normally, the source and the destination nodes periodically send VCST cells via the virtual circuit route and the Transit Nodes receiving this maintenance cell mark the virtual circuit active. The time period for sending this VCST cell is smaller than the T5 time, therefore the Transit Nodes on the virtual circuit route should receive this cell at least once before the T5 expiration. At every T5 time out, the Transit Node checks the virtual circuit table and marks any inactive virtual circuit since the last T5 time out. If a virtual circuit is still inactive in next T5 time out, the Transit Node determines this virtual circuit connection is broken and deletes the connection information.

4.4.4 T6, Route Improvement Timer

Since the flooding cells are racing together, the destination node may receive a series of the CREQ cells and the lately arriving CREQ cell may suggest more improved path. It is worth while to wait a short time when the destination first receive the CREQ cell. The T6 timer sets a waiting period for absorbing these consecutively arriving flooding cells and waits a better chance to receive a CREQ of improved route information.

When the destination Terminal Node receives a CREQ cell, the node sets the T6 timer. The destination node resets the T6 timer when another CREQ cell arrives before the T6 time out. This T6 is a very short time only enough to absorb the consecutively arriving CREQ cells with small time variance. When the T6 expires and the connection is accepted by the destination, the CACC cell is replied. Once the destination node releases the CACC cell, the node ignores any further arriving CREQ cells.

4.4.5 T7, VCST Timer

T7 is the virtual circuit management timer used by both the source and the destination Terminal Nodes to send the VCST cell via the virtual circuit periodically, hence the Transit Nodes updates the active virtual circuit information. If either source or destination node fails to send the VCST cell in (3 * T5) time, the connection will be deleted by the Transit Nodes.

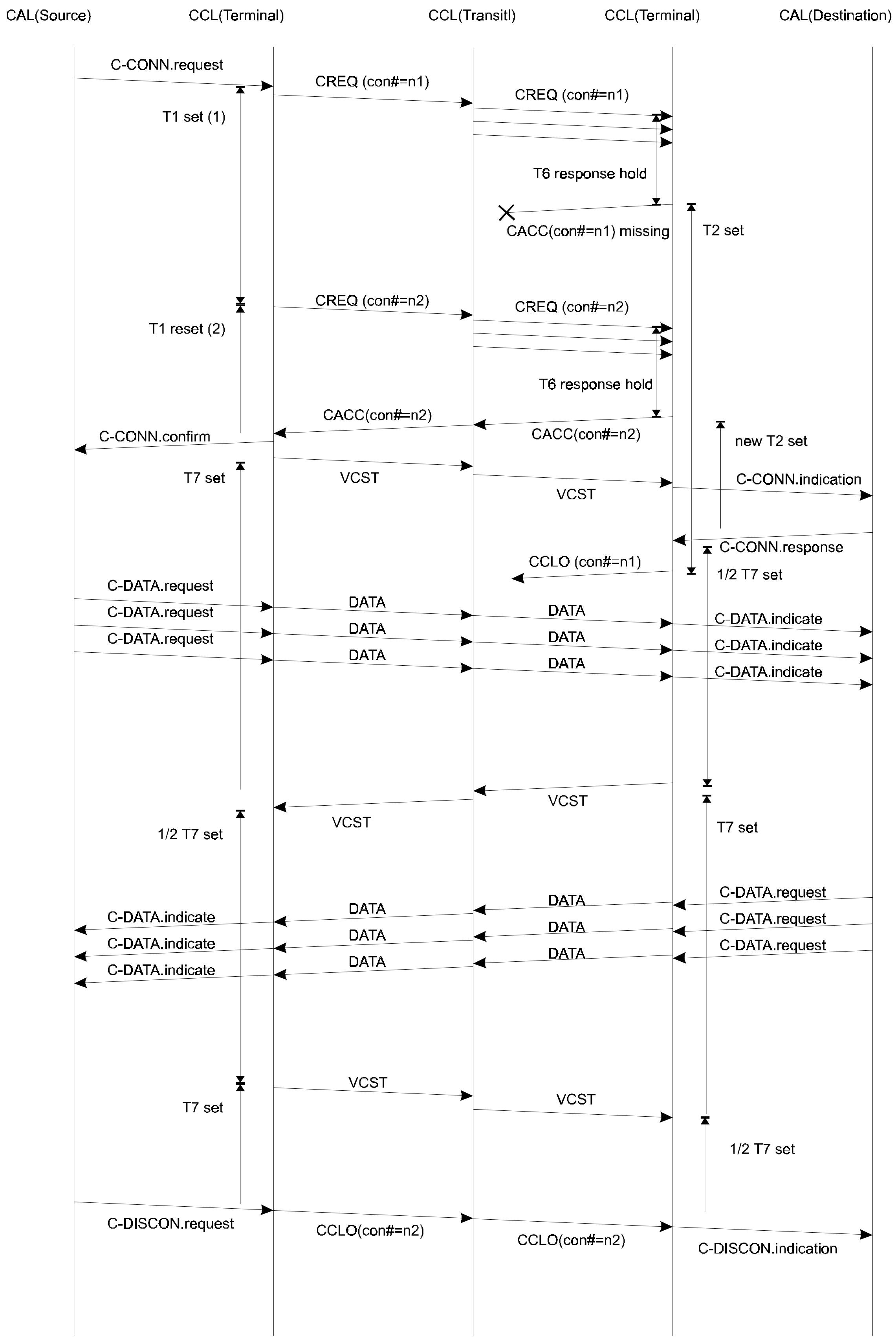
After the successful connection set up, the source node initiates the periodic sending of the VCST cell to the destination node. The reception of the first VCST cell in the destination node activates the connection. Only one T7 timer is used for reciprocal process of the VCST exchange at the Terminal Nodes.

When a Terminal Node (source or destination) has sent the VCST cell, the node sets T7 time to wait the VCST return from the other end node. If the Terminal Node fails to receive the

VCST return 3 times, disconnect the connection. If the node receive the VCST cell in T7 time, the node set 1/2 T7 timer to transmit next VCST cell. As a result, the source and destination node send and receive the VCST cell with 1/2 T7 time interval. This reduces the number of timers to manage for one connection.

4.4.6 Timing Diagram

In this section, the timing relation and the interaction of the primitives and PDUs are illustrated. This diagram shows an example of one Terminal Node making a connection with the other Terminal Node through a Transit Node and closing the connection after sending three DATA cells. The two vertical bars of the left side depicts two CAL and CCL layers of the source node and the center bar is the CCL of a Transit Node and two bars of the right are the CCL and the CAL of a destination Terminal Node. The connection starts with the CCL user requesting the C-CONNECT primitive to the CCL and finishes with the same user requesting the C-DISCONNECT primitive.


CAL(Source)
CCL(Terminal)
CCL(Transitl)
CCL(Terminal)
CAL(Destination)
C-CONN.request
CREQ (con#=n1)
CREQ (con#=n1)
T1 set (1)
T6 response hold
CACC(con#=n1) missing
T2 set
CREQ (con#=n2)
CREQ (con#=n2)
T1 reset (2)
T6 response hold
CACC(con#=n2)
C-CONN.confirm
CACC(con#=n2)
new T2 set
T7 set
VCST
VCST
C-CONN.indication
C-CONN.response
CCLO (con#=n1)
1/2 T7 set
C-DATA.request
C-DATA.request
C-DATA.request
DATA
DATA
DATA
DATA
DATA
DATA
C-DATA.indicate
C-DATA.indicate
C-DATA.indicate
VCST
VCST
1/2 T7 set
T7 set
C-DATA.request
C-DATA.request
C-DATA.request
DATA
DATA
DATA
DATA
DATA
DATA
C-DATA.indicate
C-DATA.indicate
C-DATA.indicate
VCST
VCST
T7 set
1/2 T7 set
C-DISCON.request
CCLO(con#=n2)
CCLO(con#=n2)
C-DISCON.indication

4.6 Terminal Node Specification

4.6.1 Overview of the Terminal Node Action

The Terminal Node will usually be integrated with a user device and a terminal I/O card. The control software can be installed on work station as a part of the communication software or it can be an embedded form in terminal hardware. In this section, the overview of the Terminal Node action is described. More formal definition of the state transition table and the diagram are followed.

The network level connection procedure starts with the CCL (Caroline Cell Layer) of the source Terminal Node receiving the connection request from the CCL user (Usually, Caroline Adaptation Layer). In this version of the protocol, the Terminal Node (or CCL entity) is responsible for selecting a unique 'Source Connection Number' for distinguishing the connections. A random number generation is recommended. The (source address, connection number, destination address) triple conforms network wide unique number. The initial CDM value can start from 0.

The source node sets the T1 timer with sending the CREQ cell to the first hop Transit Node. The initial T1 value will start from a few milli-seconds. If the destination node does not respond the CACC cell in T1 time, the source node 1) select the new ‘Source Connection Number 2) retransmit the CREQ cell 3) reset the T1 timer with increasing the waiting time by the exponential rate (see 4.4.1).

The destination Terminal Node sets the T6 timer when it receives the first CREQ cell. The T6 is the short time window to absorb the subsequently arriving CREQ cells of the same flood. If the destination node receive the CREQ cell before the T6 expiration, reset the T6 timer and wait the T6 time again. When the T6 expires and the CCL entity accepts the connection request, the CACC cell is released to establish the virtual circuit (see 4.4.4). The destination node sets T2 timer to wait the completion of the virtual circuit establishment and the source node respond either by the DATA cell or the VCST cell. Any reception of these cells activates the connection and the CCL entity informs this to the CCL user (CAL) by the connection indication primitive.

4.6.2 State Transition Diagram

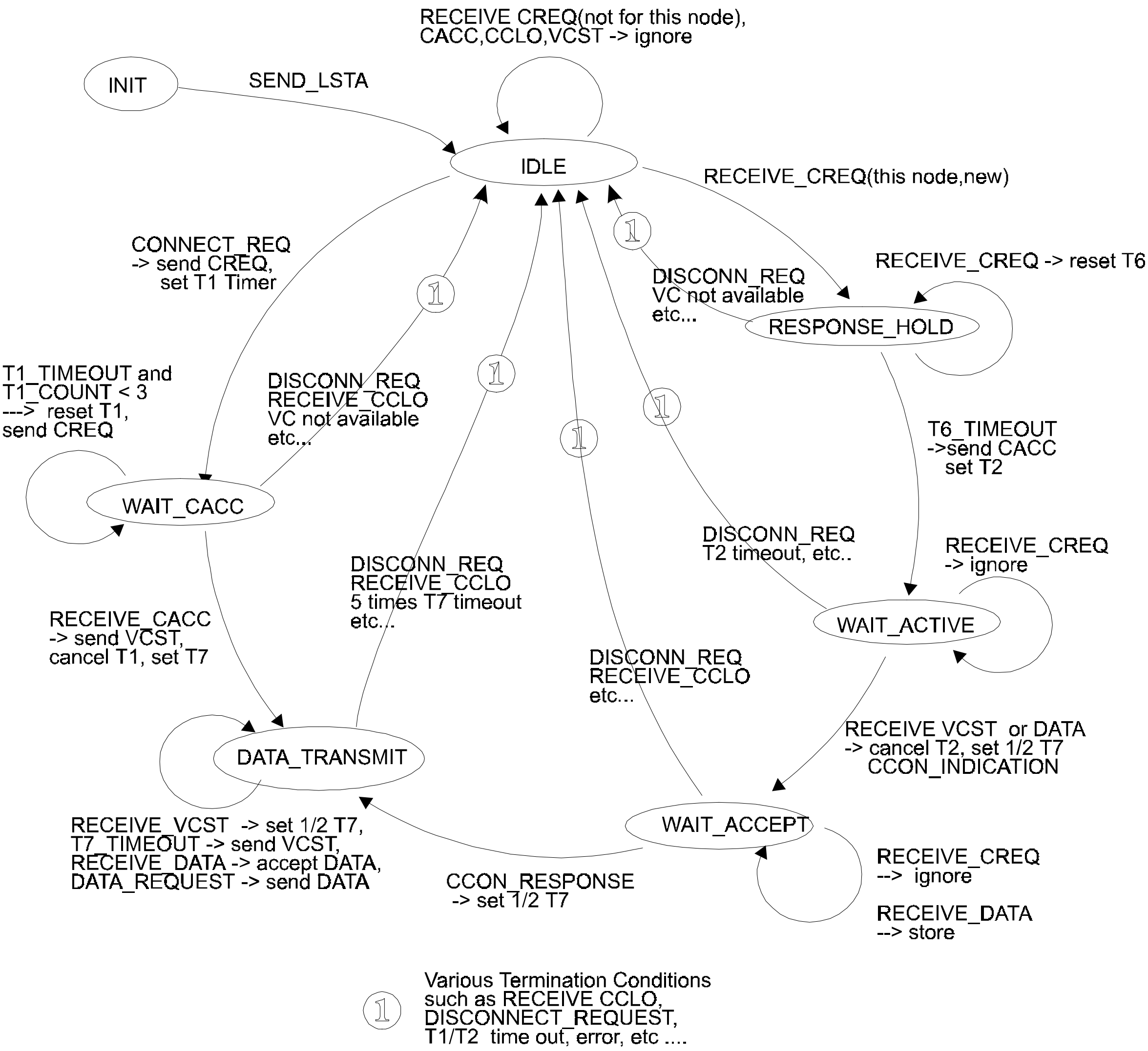


Figure 4.7 The Terminal Node State Transition Diagram

4.6.3 State Transition Table

The process is instanced by each virtual circuit.

State	Event	Action	Next State
INIT		Send LSTA	IDLE
All state	RECEIVE_LSTA	ignore	Prev State
IDLE	RECEIVE_CREQ	set T6 (route improve) timer	RESPONSE HOLD
	CCON_REQUEST and VC available	select connection ID, send CREQ set T1_count = 0, set T1 timer	WAIT CACC
	CCON_REQUEST but VC not available	pass DISCONNECT_IND (to upper layer)	IDLE
	RECEIVE _CACC/ CCLO/ VCST/ DATA	ignore (% no error condition)	IDLE
RESPONSE HOLD	RECEIVE_CREQ	reset T6 (route improve) timer	RESPONSE HOLD
	T6_TIMEOUT and VC available	allocate VC, send CACC, set T2 timer	WAIT ACTIVE
	T6_TIMEOUT but VC not available	send CCLO (reason=VC not available)	IDLE
	DISCONNCET_REQ (by own reason)	send CCLO (reason=unable to connect)	IDLE
	RECEIVE _CACC/ VCST, DATA, CCLO	error	IDLE
WAIT ACTIVE	RECEIVE_CREQ	ignore	WAIT ACTIVE
	RECEIVE_VCST or RECEIVE_DATA	cancel T2 timer, pass CCON_INDICATION set 1/2 T7 (vcst) timer, set T7_count = 0	WAIT ACCEPT
	T2_TIMEOUT	send CCLO (reason=dest give up)	IDLE
	DISCONNCET_REQ (from upper layer)	send CCLO (reason=call refused)	IDLE
	RECEIVE_CACC	error	IDLE
WAIT ACCEPT	RECEIVE_CREQ	ignore	WAIT ACCEPT
	RECEIVE_DATA	store data	WAIT ACCEPT
	RECEIVE_CCLO	pass DISCONNECT_IND	IDLE
	CCON_RESPONSE	set 1/2 T7 timer, pass DATA_IND (if received)	DATA TRANSMIT
	DISCONNECT_REQ (from upper layer)	send CCLO (reason=user refused)	IDLE
	RECEIVE_CACC	error	IDLE
WAIT CACC	RECEIVE_CACC	pass CCON_CONFIRM, cancel T1 timer, send VCST, set T7 (vcst) timer, set T7_count = 0	DATA TRANSMIT
	RECEIVE_CCLO	pass DISCONNECT_IND	IDLE
	T1_TIMEOUT and T1_count < 3	select new connection ID, send CREQ, reset T1 timer, increase T1_count	WAIT CACC
	T1_TIMEOUT and T1_count >= 3	pass DISCONNECT_IND (to upper layer)	IDLE
	DISCONNCET_REQ (from upper layer)	cancel all	IDLE
	RECEIVE_CREQ/ VCST/ DATA	error	IDLE
DATA TRANSMIT	RECEIVE_DATA	pass DATA_INDICATION	DATA TRANSMIT
	RECEIVE_VCST	reset 1/2 T7 timer, set T7_count = 0	DATA TRANSMIT
	RECEIVE_CCLO	pass DISCONNECT_IND	IDLE
	T7_TIMEOUT and T7_count < 5	increase T7_count, reset T7 timer, send VCST	DATA TRANSMIT
	T7_TIMEOUT and T7_count >= 5	pass DISCONNECT_IND, send CCLO (reason=conn failure)	IDLE
	DATA_REQUEST	send DATA	DATA TRANSMIT
	DISCONNECT_REQ	send CCLO (reason=disconnect)	IDLE
	RECEIVE_CREQ/ CACC	ignore	DATA TRANSMIT

4.7 Transit Node Specification

4.7.1 Overview of the Transit Node Action

The main roles of the Transit Node are to carry out the flooding process and to provide the virtual circuit path. The Transit Node maintains three types of tables for these purposes. One is the Flooding Queue used for recording the information of the CREQ cell flood and the other is the Virtual Circuit Table (VC table) for keeping the virtual circuit information, and the third is the ARU (Address Resolution Unit) memory for routing data cells by hardware control.

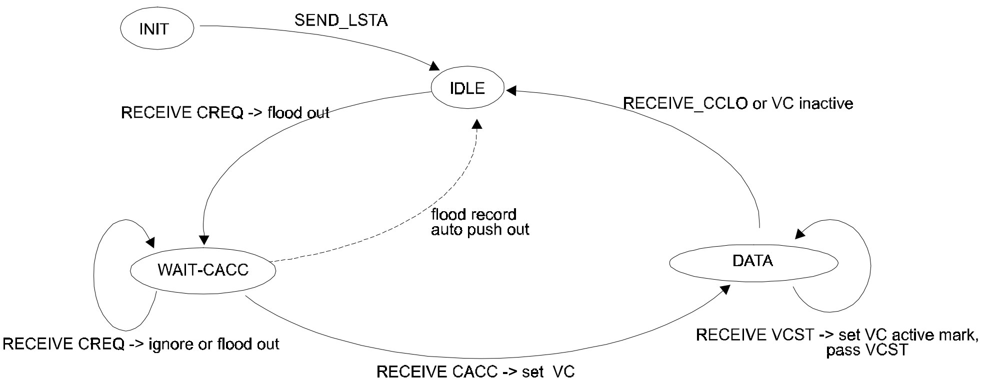
When a Transit Node receives a CREQ cell, the node checks the Flooding Queue to see if the same flooding has been received before. If this is the first CREQ cell or the CDM value is better than the previously received one, it records the details of the CREQ cell on the Flooding Queue and send the copies of the CREQ cell to all output links. The Flooding Queue is a fixed size, first-come-first-out style queue. As a new information is pushed into the front of the Flooding Queue, the oldest information is deleted. The flooding information remain in the Flooding Queue for a time depends on the size of the queue and the network call generation rate. More details of the Flooding Queue is discussed in section 5.5.

When the Transit Node receives a CACC cell, the node allocates the virtual circuit route based on the best route information in the Flooding Queue. The virtual circuit information is recorded in the VC table and the ARU (Address Resolution Unit) routing memory. The VC table is referenced by the CPU for processing the connection control cells. The ARU memory is referenced by the control hardware for routing data cells directly.

The CACC cell is then passed to the neighboring node which has given the CREQ cell of the best CDM. When the CACC cell finally arrives to the source node and the virtual circuit path is established, the source node sends the VCST cell and starts the data transmission. The DATA cells are switched by the hardware directly but the VCST cell is passed hop by hop to be processed by the CPU. The Transit Node marks the virtual circuit when it receives a

VCST cell, and at every T5 time interval, all the unmarked virtual circuits are check and removed.

4.7.2 State Transition Diagram



Fugure 4.8 The Transit Node State Transition Diagram

4.7.3 State Transition Table

The process is instanced on a Flood.

State	Event	Action	Next Event
INIT		Send LSTA	IDLE
IDLE	RECEIVE_CREQ	push the flood record into the Flooding Queue; increase the CDM; send the copies of the CREQ to all links;	WAIT CACC
	RECEIVE_CACC/ CCLO/ VCST	ignore	IDLE
WAIT CACC	RECEIVE_CREQ	if the CDM is not better than the previous, ignore it; else push the flood record into the Flooding Queue; increase the CDM; send the copies of the CREQ to all links;	WAIT CACC
	RECEIVE_CACC	allocate the VCs, send the CACC to the neighbor who gave the CREQ	DATA
	RECEIVE_CCLO	send the CCLO to the neighbor who gave the CREQ	IDLE
	RECEIVE_VCST	error	IDLE
DATA	RECEIVE_CREQ	ignore	DATA
	RECEIVE_CACC	ignore (% for multicast extension)	DATA
	RECEIVE_VCST	mark the VC to 'active'; send VCST to the next hop node;	DATA
	RECEIVE_CCLO	clear the VC	IDLE
All state	RECEIVE_LSTA	(% not yet defined)	Prev State
All state	T5_TIMEOUT	check the VC table; delete unmarked VCs for 3 times T5 time out. reset T5 timer;	Prev State

5. Discussions on the Major Issues

5.1 Connection Time Out and Control Mechanism

The T1 and T2 timers are defined as a source and a destination Terminal Node's connection timer. These timers set the maximum waiting time to receive a response. The decision of the optimal time period of these timers was long argued issue. There are several reasons that made the decision difficult.

The connection delay is mainly related to the round trip time from the source to the destination node. Unfortunately , it is difficult to know the distance information in a Flood Routing network because the network does not keep the distance table. This becomes more complex problem when several different types of network are interconnected. The distance measure (or the delay measure) may be different in a different type of network.

Another connection delay problem is involved in the target host computer's average response time. The host delay varies by the performance and the work load and it makes the optimal connection time more difficult to predict. In a network, the overall connection delay may range from a few milli-seconds to several minutes or forever if the destination host is crashed or the response cell is lost. The source node does not know how much it should wait and when it should retry the connection.

In the case when connection response cell was lost or corrupted, it is wasteful for the source node wait a long time to determine the connection failure. The connection needs to be retried again as soon as possible because the connection may be successful in next trial. A proposed time to retry in this case is a few milli-seconds. However if the response is delayed because the destination host is busy or the network is congested, etc., it would be better to wait a long time. Otherwise the signalling traffic will overload the situation more. The proposed time to wait in this case is up to 3 ~ 5 minutes before finally giving up the connection. Therefore the compromise between these two different time out requirements is a variable timer, starting from a short period and increases the interval exponentially up to several minutes on each time out.

The initial T1 timer starts from a few milli-seconds. The most local area network's turnaround time is less than few milli-seconds in normal traffic conditions. If the connection request cell is lost on the way, another request cell follows in a few milli-seconds. However, if the connection failure is repeated, it may be because the network needs a long time to transfer the response. In this case, the Terminal Node gradually increases the T1 length by this exponential formula:

$T1_i = a * b \wedge i$ where, a = initial T1 time, b = exponential growth rate

The longer T1 will be used in later connection request.

The destination node also needs a connection timer; T2. However this timer does not need to be so sensitive. T2 only needs to be long enough to cover the longest T1 time because no source node will wait longer than T1 time. Then the destination node is less responsible for the connection procedure while the source node actively to recover the connection failures.

5.2 Lost CCLO and VC Reuse Problem

This rare but harmful phenomenon occurs when the virtual circuit is heavily reused. The virtual circuit number tends to be reused especially when the available number space is small. In particular, if a recently returned virtual circuit is quickly reallocated for a new connection while the closing process is progressing in network, the data cells coming from the opposite side can wrongly be rerouted to the new connection. Figure 5.1 to 5.3 illustrates this procedure in detail.

In Figure 5.1, a new virtual circuit is established from node 2 to 3 while the other existing connection is closed from the Transit Node 1 -> 3-> 4. Unfortunately, the CCLO cell is lost in the link (3, 4) due to the cell discarding or corruption. Note that node 4 is still keeping the old connection information unchanged.

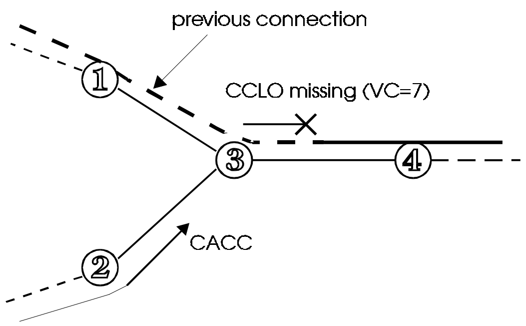


Figure 5.1

A short period later, the Transit Node 3 receives a virtual circuit allocation request and reuses the previously returned virtual circuit number and passes the CACC cell to node 4. Unfortunately again, this CACC cell is also lost in the link (3, 4) as the same reason before. In this case, Transit Node 3 thinks of the virtual circuit as a newly allocated one while node 4 thinks of the virtual circuit as an old connection, thereby the circuit is abnormally connected.

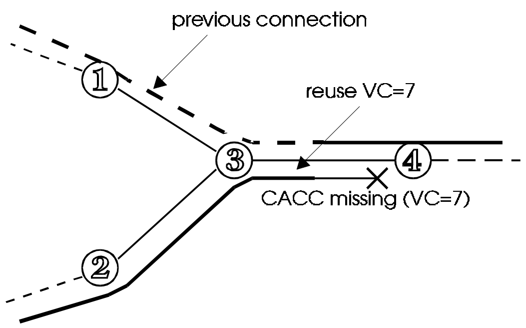


Figure 5.2

The Terminal Node of the old connection does not know this fact and sends data cells. Between node 3 and 4, the data cells are wrongly routed to the Terminal Node of the new connection. This causes a protocol discrepancy in the Terminal Node.

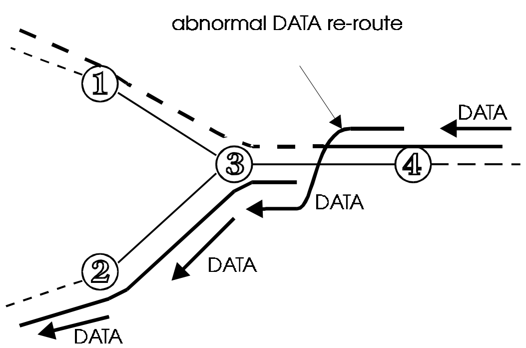


Figure 5.3

A temporary solution to this problem is postponing the recently returned virtual circuits from direct re-use, until the VCST mechanism can clean incomplete VCs. However this method is wasteful when there is a high demand for VCs. The more fundamental solution is in consideration.

5.3 Source Connection Number

In a normal network, the (source address, destination address) pair is used for packet identification. However this pair of addresses is not enough to support various application needs in the Caroline Network. For example, the CREQ cell carries different types of destination addresses. This multi-type destination address scheme is useful especially when the Caroline protocol is used in conjunction with other protocols such as the ARP mechanism.

When the destination node replies with a CACC or CCLO cell, it is convenient for the destination node reports its Caroline Address in the 'Destination Address' field. Otherwise the cells should be capable of carrying the multi-type address as well as the Caroline Address both. However, if the destination node replies with only the Caroline Address, the Transit Node cannot recognize the CACC cell because the address form is different from the CREQ cell.

The other difficulty is, when the source node wants to set up more than two connections with the same destination, the two different CREQ cells will generate their own floods. If the cells of these two different floods have only (source address, destination address) pairs, they can not be distinguished each other.

For these reasons, the control cells use the (Source Address, Source Connection #, Destination Address) triple for their identification. The 'Source Connection #' is selected by the source Terminal Node randomly. As long as the source node keeps the 'Source Connection #' unique in its node, the (Source Address, Source Connection #) pair can be an unique identifier in the entire network. Therefore, the Transit Node only references the (Source Address, Source Connection #) fields to distinguish the control cells of the different calls. The destination node is now free to use the Caroline Address to reply the control cells no matter what type of address was given by the CREQ cell. It is also possible to distinguish the different floods as long as the 'Source Connection #' is different.

In addition, it is desirable to generate the 'Source Connection #' randomly in order to guarantee a different number each time the Terminal Node is powered off and on. Finally, the number is also can be used as a group identification number in the multicast application.

5.4 Flooding Queue

Flooding is a networkwide event. This means that all the network nodes participate in the flooding action when the connection request cells propagate. Among them, only a small proportion of the nodes are chosen to provide the route between the source and destination nodes. The excluded nodes from the connection will return to the original state. There are several different methods used in various flooding algorithms. Some algorithms [7] use a timer to give up waiting, others send an explicit cancelling signal [12]. The previous specification of the 'Reverse Metric Flooding' method with the Caroline Network also used an explicit connection rejection cell to cancel the waiting state [1][2][4]. However the original algorithm had the potential to make three adjacent nodes waiting a rejection cell from each other, and to fall into a deadlock state [1].

The 'Forward Metric Flooding' method solves the problem using the novel 'Flooding Queue' method. The Flooding Queue is a fixed size push out queue which contains the flooding information. Whenever a CREQ cell enters into the Transit Node, the information of the cell is checked with the flood records kept in the Flooding Queue. If this is the first CREQ cell regarding the flooding, the new information is pushed into the queue. If the new CREQ cell is better than the information of the previous CREQ cell, the previous record is updated. The queue is a fixed size. If there is no space left for new information, the oldest record is removed. Thus the older records in the queue are gradually moved to the exit and are finally removed without using any timer.

The Transit Node accesses the flooding record from the Flooding Queue when the node receives the CACC cell and sets up the route. The flooding record should remain sufficient time in the queue until the Transit Node receives the CACC cell. The time from the queue insertion to the deletion is called the 'Queuing Delay' and it depends on the queue size and the new call input rate. The general formula of the average queuing delay in the Flooding Queue is:

$T = L / \lambda$ where, L = Flooding Queue size, λ = new call input rate (calls/seconds).

For example, if the Flooding Queue size of the Transit node is 3000, and the maximum call generation rate in the network is 10 calls/seconds, then the flooding records remains in the queue about 5 minutes (= 3000/10 seconds). This is enough time to cover a connection delay in most of the networks. The average queuing delay of the Flooding Queue should be greater than the largest T1 time.

Care should be taken regarding the use of the CDM. It is the CDM value of the CREQ cell which is compared with the flood record of the queue. If the CDM value of the new CREQ cell is equal to or greater than the CDM value in the queue, the cell is discarded. Therefore the Transit Node accepts only the CREQ cell of the lowest CDM. The source Terminal Node send the CREQ cell again when the T1 timer expires. If the CREQ cell starts with the same CDM value as before, the cell will arrive to the Transit Node with the same CDM value or greater than the previous record. The cell will eventually be discarded before it reaches to the destination. For this reason, the source Terminal Node starts the first CREQ cell with certain value of the CDM and reduces it in next trial. This ensures the CDM of the second CREQ cell lower than the flood record in the queue and reaches to the destination node.

6. Conclusion and Future Study

A new flood routing mechanism based on the Forward Difficulty Metric Flooding is proposed. The Forward Flooding will simplify the Caroline network software implementation and it has potential to solve many other network routing problems. The major improvements of this revised flooding are: 1) it reduced many timer concerns 2) it simplified the complex backward flooding process and the failed virtual circuit cancellation mechanism 3) it removed the problem of CREQ cells racing and possible deadlock problem.

There are several un-solved problems requesting further study. The VC reuse and the lost CCLO problem (see 5.3) is still not perfectly solved. An intensive simulation study of the characteristic of the flooding has not been carried out. The extension to the Multicast routing is the generalization process of the point-to-point flood routing. This is the research to provide the point to multi-point or multi-to-multi point connections with the optimal use of network resources. More effective way of reducing the number of flooding cell needs to be researched. We call this the 'Constrained Flooding' problem. In conjunction with this, the research to develop the flood routing for a large scale network by introducing a network hierarchy or clustering is important. Considering the use of a simple terminal device on Caroline Network such as the voice telephone, the protocol actions in the Terminal Node is complex to implement on one chip solution. The protocol overhead needs to be shifted from the Terminal Node to the Transit Node in order to simplify the terminal adaptor design and reduce the manufacturing cost. The current flooding protocol does not detect non-responding or non-existing destination node. Whether to use the negative acknowledgment function to get a quick response is the issue to discuss more.

References

[1] J. W. Breen "The Caroline Project: Service and Protocol Description", Technical Report 93-15, Monash University, Depart of Robotics and Digital Technology, 1993

[2] C. Blackwood, R. Harris, A. T. McGregor and J. W. Breen, "The Caroline Project: An Experimental Local-Area Cell-Switching Network", ATNAC-94,1994

[3] C. Blackwood, R. Harris and A. T. McGregor, "Caroline Node Interface Specification and Implementations Guide", Technical Report 94-5, Monash University, Department of Robotics and Digital Technology, 1994

[4] R. Harris, "", Master Thesis, Monash University, Department of Robotics and Digital Technology, 1995

[5] G. Ludwig, and R. Roy, "Saturation Routing Network Limits", Proceedings of IEEE, Vol. 65, No.9, September 1977

[6] M. Schwartz and T.E. Stern, "Routing Techniques Used in Computer Communication Networks", IEEE Transactions on Communications, Vol. COM-28, No.4, April 1980

[7] C. Petitpierre,"Meshed Local Computer Networks",IEEE Communications Magazine,Vol.22, No.8, August 1984

[8] V. O. K. Li and R. Chang, "Proposed Routing Algorithms for the U.S. Army Mobile Subscriber Equipment (MSE) Network", Proceedings - IEEE Military Communications Conference, Monterey, CA, paper 39.4, 1986

[9] M. Kavehrad and I.M.I. Habbab, "A Simple High speed Optical Local Area Network Based on Flooding", IEEE Journal on Selected Areas in Communications, Vol.6, No.6, July 1988.

[10] S. Corson and A. Ephremides, "Distributed Routing Algorithm for Mobile Radio Networks", Proceedings - IEEE Military Communications Conference, Piscataway, NJ, pp210-213,1989

[11] O. Lesser, "Routing by Controlled Flooding in Communications Networks", Proceedings of the Conference on Computer Communications, IEEE Infocom, pp910-917,June 1990

[12] H. K. Pung, "Simulation Study of a flood routing technique for mesh-like LANs", Computer Communications, Vol. 13, No. 6, July/August 1990

[13] Douglas E. Comer, "Internetworking with TCP/IP", Vol I, Prentice Hall