

Principle and Technique for Encapsulation of User Control and Data Information in Separate Frames

By: Chooi-Tian (Alex) Lee and Jack W. Harris, © 1996.

UNI-Net, Inc.

438 Colony Woods Drive
St. Louis, Missouri 63122.

Tel: (314)-821-1198

E-mail: jwharris@mo.net

Abstract

Present and future communication services require the integration of data and media applications to provide real-time interactive multimedia services that are reliable and secure. The popularity of the World Wide Web provides a glimpse into the potential of distributed multimedia application services. However, the Internet protocol, TCP/IP, does not have the provisions to adequately handle future real-time interactive multimedia services. Present network protocols are programmed to view and treat all user information packets alike, since application (user) control and data information are bundled within the same packet. ATM protocol provides more flexible multimedia transmission services for dynamic and static types of information. This paper proposes techniques for enhancing ATM protocol to enable the support of application communication services as well as network communication services. The paper advocates strict enforcement of the encapsulating of User-Control and User-Data information in separate and discrete frames at the network-level so that unique application communication services can be specified for the disparate application information traffics. The paper proposes the ability to identify user control and data messages at the network-level. The objective of this paper is to provide an understanding of a new protocol, which provides the capability to optimize services to each of the disparate type of traffics and facilitates the integration of media and data applications. The proposed FAL protocol enables the various network protocols to be aware of the required application class-of-services and provide the appropriate network services which facilitates traffic shaping and policing.

1 Introduction

Present and future communication services require the integration of existing data and media applications to provide

real-time interactive multimedia services. Digital technology and ATM communication provide a means to achieve this goal. Future multimedia services also require that communication services be secure and reliable. Secure and reliable communication services require a network protocol that is able to provide application-based access filtering at the network-level as well as the capability to allocate differing communication services for the disparate application information traffics. This paper advocates strict enforcement of the encapsulation of User-Control and User-Data information in separate and discrete frames at the network-level. *The paper proposes that the identification and awareness of user control and data messages be extended to the network-level so that unique application communication services can be specified for the disparate information traffics.*

2 Problems and Solutions

The popularity of the Internet and the World Wide Web provides a glimpse into the potential of distributed multimedia application services. However, there are still some technological problems yet to be overcome before real-time interactive multimedia applications and services can be implemented over a more secure virtual dedicated-media network environment. Present Internet protocol, TCP/IP, does not have the provisions to adequately handle future real-time multimedia services as TCP/IP was designed solely for handling non real-time burst-mode data. The following discussion highlights major technical problems and recommends solutions to facilitate the integration of real-time media and burst-mode data services.

2.1 Management of User-Control during Traffic Congestion

Application-related commands or user control information, such as Remote Procedure Call (RPC), remote object-oriented

Table 1: Attributes of Multimedia Information Traffic

Message Type	Loss Sensitivity	Traffic Type	Traffic Priority	Traffic Size
Network Control	Cannot tolerate loss	Streaming	Top Priority	Very small
User Control	Cannot tolerate loss	Streaming	Top Priority	Very small
User bursty data	Cannot tolerate loss	Store-and-forward	Low Priority	Small
User audio data	Can tolerate loss	Streaming	High Priority	Large
User video data	Can tolerate loss	Streaming	High Priority	Very large

invocations, OS instruction codes, application daemon commands, and an application's User Interface commands, are processed and utilized at the application-level. *Although the ability to separately transfer and identify user control and data messages has been implemented at the application-level, such ability to differentiate user control and data messages has not been extended to the network-level.* It should also be noted that the ability to separately transfer user control and data messages is not uniformly implemented throughout all applications or all network protocols. For example, the vendors of network protocols and applications have the tendency to bundle user control and data messages within a frame or packet in order to increase the efficiency of the transmission.

Existing network protocols, including advanced ATM protocol, presently only perform separate encapsulation and identification of network switch control messages from application messages at the network-level. ATM protocol does not separately encapsulate and identify an application's user control messages from user data messages at the network-level. Hence, ATM protocol is not able to define the optimal communication services for the differing user control and user data information.

This can be problematic since ATM protocol routes information cells among network switches having limited buffer resources. During traffic congestion, the ATM network switch tends to drop cells when the buffer becomes saturated. However, to prevent dropping of vital Network-Control Plane service information, ATM separately encapsulates and identifies network control information as network control frames or cells. The network control cells are also transmitted over a separate virtual channel connection. Currently, the ATM Header Layer of network control cells are encoded with Cell Loss Priority (CLP) value set to 0 to indicate high priority traffic. This is done so that ATM protocol is aware of the need to guarantee reliable delivery of those cells during traffic congestion.

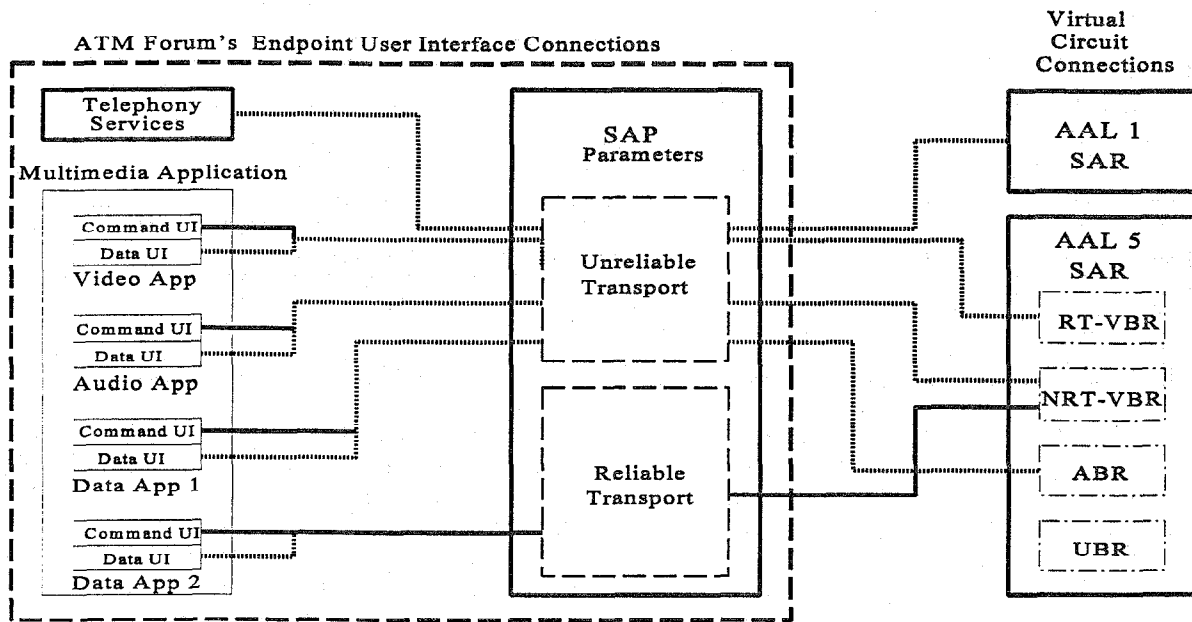
The inability of network protocols to differentiate application or user control messages from data message at the network-level means that application datagrams are assigned

the same traffic services. For example in ATM protocol, the protocol data unit CLP value for user datagrams is usually set to 1 to indicate low priority traffic. Cells with CLP=1 are more likely to be discarded during traffic congestion. Consequently, user control information transported in cells having CLP=1 has a high probability of being corrupted during traffic congestion. This has a detrimental effect on the operation of the application if time-out schemes for command messages are not implemented.

The proposed solution to the above problems is 1) to enable and to strictly enforce, at the application- and network-level, the encapsulation of user control and user data information within separate frames, and 2) extend the identification and recognition of user control frames to the network-level. This is accomplished by using the proposed unique Frame Adaptation Layer (FAL) protocol and API, which allows applications to issue communication-based control and data send primitives.[1] A discussion on the differences between FAL and existing network protocol send primitives is covered herein in section 3. FAL protocol facilitates a user control cell's PDU to be assigned with CLP=0 for indicating high priority traffic, while at the same time, assigning CLP=0 to the user data cell's PDU. This paper also proposes to use an ATM Header Layer PTI value of 111 as an indicator for SDUs containing only User-Control Information in the payload segment of the cell. This PTI value enables ATM network switches to recognize and differentiate User-Control SDU cells from Network-Control SDU cells. The key benefit is the ability of FAL to assign reliable transport services and allocate high cell loss priority to user-control SDU traffic. This reduces the probability of cell loss during traffic congestion without overwhelming the ATM network.

2.2 Handling Different Priority and Loss Sensitivity

The attributes and service parameters of the differing types of information messages in a multimedia transmission should be considered individually in order to achieve an optimum level of traffic services. To achieve reliable multimedia



Source UNI-Net, Inc.

Figure 1: Combined-Encapsulation of User Control and User Data Information

communication, network protocol must be able to identify, recognize, and provide the required services to the differing traffic types. The types and characteristics of information in multimedia traffic can generally be grouped into five main categories. The categories and the attributes of multimedia information traffic are illustrated in Table 1. An information frame may use either reliable or unreliable transport protocol. In unreliable transport protocol, the transmission of a frame is by the "best effort delivery" concept and is used by User-Data SDU types that can tolerate some loss of data, such as real-time user media datagrams containing audio and video information. User control information requires reliable delivery and requires a reliable transport protocol. The optimization of network services for the differing multimedia information traffic requires the capability to specify the differing traffic service parameter best-suited for each type of application message. For example, an interactive real-time media application requires different types of service parameters in order to support both user media data and user command traffic. There are currently two possible methods for transmitting information messages generated by a media applet. One information is the commonly used *Combined-encapsulation method* that encapsulates *both* the user media data information and the user command information within the same frame at the network-level. As shown in Table 1, the

user control message attributes and service requirements differ from user data messages of burst-mode, audio, and video information. However, the combined-encapsulation of user control and data messages within the same SDU frame compromises the allocation of network services for that information.

Figure 1 is illustrative of the application Command-User Interface and Data-User Interface traffic flows and the communication services provided when using the currently used *Combined-encapsulation method*. When the user control and data information originates from a communication link established for a real-time video (media) application, it must be decided either to establish a reliable network transport service or an unreliable but temporal sensitive network transport service. As shown, the media data information and the application daemon command information are encapsulated in the same SDU frame and then transmitted through use of unreliable transport services established for media transmission. The dotted line indicates that unreliable transport services are provided to the communication link. Transmitting application daemon command information through media-based communication services exposes the user command information to probable loss of data.

An alternative solution currently used is to assign reliable transport services to the real-time media application.

However, some features of reliable transport service protocols, such as Transmission Control Protocol (TCP), make those protocols unsuitable for sending streaming media user data. For example, the requirements of TCP to resend packets of data that have not been correctly transmitted and to automatically reduce the packet window size cannot be used with temporal sensitive media user data. Although TCP is inefficient in sending real-time audio and video information, it is useful in providing reliable transport for user control information. Consequently, when user control and video data information are encapsulated within the same SDU frame, the resulting dilemma is in deciding which transport protocol is to be specified for the frame. This phenomenon is known as transport protocol mismatch for the user control and data information.

An existing solution to transport protocol mismatch is to assign two virtual circuits to each application, one for the user interface control channel and the other for the user interface data channel. However, there is a need to synchronize information originating from the separate virtual circuits. Synchronization of information originating from two virtual circuits can be problematic. In addition, using separate virtual circuits for each application daemon control and data user interface channel utilizes precious virtual circuit resources.

Our proposed solution is to implement a separate-encapsulation method for user control and user data information during transmission at the network-level. This proposal facilitates implementation of specialized communication services for the separately encapsulated User-Control SDU and User-Data SDU frames. Reliable transport for the *vital and critical* user control information can be specified without providing similar transport service to other large loss insensitive user data information, such as audio and video. When application control and application data are encapsulated separately, the ability to specify differing communication services, such as capabilities to check the frame integrity, to prioritize information traffic, to define access authorization, and for traffic-flow control management, are feasible for the differing information traffic. For example, loss correction services can be specifically designed for streaming media information which minimizes disruptions arising from frame loss or data corruption.

2.3 Extending Connection-based Communication to Burst-mode Applications

Connectionless-based network protocols are well-suited for burst-mode communication as client/server computing is designed to terminate the communication after each message transfer session. In connectionless-based communication, call establishment is not required. However, connection-based communication requires call establishment. The way in which

connection-based communication is implemented to client/server computing can create inefficiency.

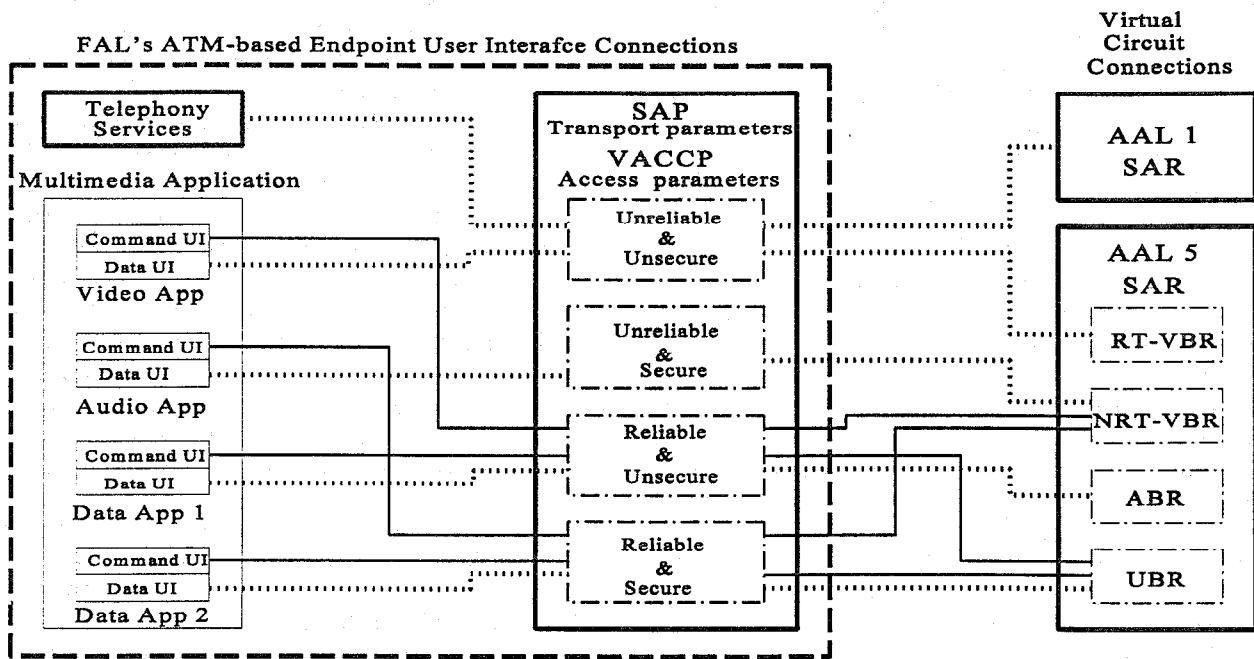
ATM Forum's ATM API semantics[2] proposes a means to extend ATM communication capability directly to application user interfaces. However, it is inappropriate to extend the ATM API scheme of connection-based call establishment and termination procedures to client/server computing applications. Client/server computing communications involve several small request and response messaging. This results in spending a long period of time in establishing a call connection which is used for a very short period of time to send the data. After the data is sent ATM protocol terminates the call immediately. The repetitive small request and respond messaging required in the operation of client/server applications results in inefficiency.

The proposed solution the above problem is to utilize ATM protocol to establish private lines of communication between communicating host computers Network Interface Cards (NICs) and then use FAL protocol to communication with the application. This avoids the requirement to terminate the ATM switched virtual channel connection immediately after each client/server communication. The connection-oriented and packet-switching FAL network protocol forms communication links with ATM protocol and the application user-interfaces. The use of FAL protocol as an intermediary communication link between application user-interfaces and ATM switches eliminates the need to take down the ATM section of the communication link at the end of each application send session. The FAL protocol terminates the ATM section of the communication when the application terminates the user-interface connection to the FAL protocol.

When used in this fashion, call termination of a prior connection can be executed either when the user terminates the application or when the application requires a connection to a new host computer site. For example, in a Web application, when the Web Universal Resource Locator (URL) indicates that the information is located on a different Web host server, HyperText Transfer Protocol sends a message to the FAL protocol to establish a new connection. The FAL protocol then prompts the user whether or not to terminate the existing line of communication. If the user chooses not to terminate the existing connection, FAL protocol proceeds to execute only call connection procedures and the application has point-to-multipoint communications with the other Web site. When the user terminates from a Web application session, FAL protocol automatically terminates all communication links associated with the Web application.

3 Implementation of Network Communication

In digital communication, a network protocol collects messages from the sending end-system and formats the



Source UNI-Net, Inc.

Figure 2: Proposed Separate Encapsulation of Command-UI and Data-UI Information

message for transmission, and also specifies the required communication services. The network protocol also needs to inform the receiving end-system where to deposit the user data information in the application's memory at the destination endpoint. Currently, this is achieved by either sending the user control information prior to the user message data or by incorporating user control information within the message data. The information is used by the destination endpoint to determine exactly where the message data should be deposited. There are also instances when the user needs to interrupt the normal operation of the media data communication to invoke a remote object request or to execute a new control processing request. An application invokes network-based send primitives to enable sending user control and data messages through a network. The above operations generate user control and user data information which are deposited in the appropriate buffers. The deposition of the user control and the user data messages into control and data memory spaces is handled directly within the hardware, kernel, or application. Methods for memory address mapping and protection issues associated with direct delivery of messages to an application have been developed by C. Thekkath and vonEicken.[4][5]

The following discussion covers the separate-encapsulation methods used by FAL protocol and techniques in which FAL protocol strictly enforces separate encapsulation of user control and data information at the network level. The below

discussion describes how the awareness of the disparate user information is extended to the network-level. It also covers the benefits gained when FAL protocol is used, and discusses the implementation of FAL in multiplexing disparate information traffic within a single virtual channel connection.

3.1 Separate-encapsulation of User Control Information at the Network-level

The way in which the separate-encapsulation method influences the transmission of user information differs from the combined-encapsulation method. Figure 2 is illustrative of the Command-User Interface (UI) and Data-User Interface traffic flows and the communication services provided when using the proposed FAL *separate-encapsulation* method. The method uses different endpoint identifiers to create separate communication links having the appropriate quality of service and transport protocol for the differing communication links.[2] As shown in Figure 2, information originating from the application Command-UI and Data-UI is strictly enforced to be encapsulated within separate and discrete frames through the use of FAL's Control and Data send primitives, respectively. The figure also illustrates how differing transport protocol and security protocol can be specified for each communication link by assigning the appropriate SAP and VACCP parameters. The required communication services information or information elements are encoded

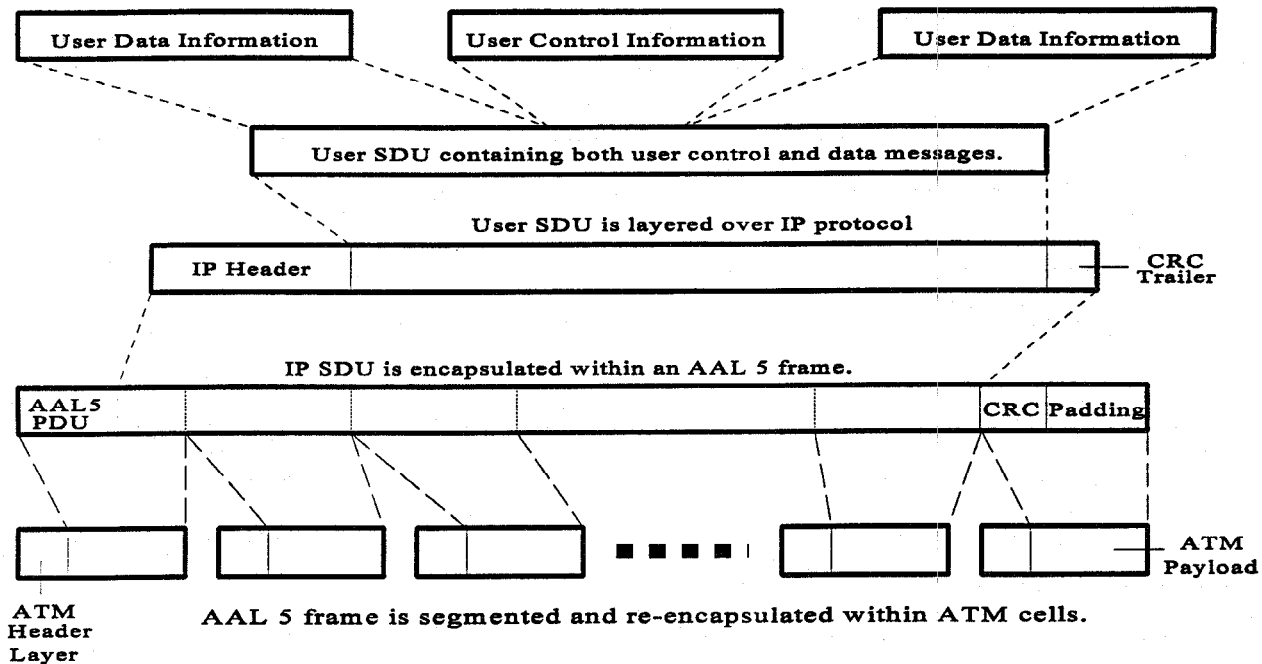


Figure 3: Current Combined Encapsulation Processing of Existing Network Protocols

within FAL's PDU. The Service Access Point (SAP) parameter is used to allocate the appropriate Quality-of-Service (QoS) parameter, and reliable or unreliable transport services, by forming a link to the appropriate transport protocol processing module. Our proposed Virtual Access Control Configuration Protocol (VACCP) parameter is used to define user access rights to the control functions of the application daemon command user interface. The remote access and remote control capabilities to a host computer is negotiated and approved during connection setup [3].

Present ftp, http, RPC, and other computer application daemon programs issue both user command and data messages. The user control information may at times be the only message encapsulated within a IP packet. However, using existing IP-based send primitives, user control information can be occasionally encapsulated together with user data information within the same information packet. Such conditions exist during an application operation when servers send control and data messaging in successive operations or when application switches to a telnet operation.

Figure 3 illustrates existing network protocols use of combined-encapsulation processing of information packets at the network-level. Currently, during combined-encapsulation processing, the user control and data information is encapsulated within an IP packet. The IP packet is subsequently encapsulated with an AAL 5 frame and encoded with appropriate AAL PDU information along with other protocol information, such as CRC information. Padding is

added to the AAL 5 frame so that the bytes in the frame, are divisible into 48-bytes blocks. The AAL frame is then segmented and encapsulated into several ATM cell payload segments for transmission which includes an AAL SAR-PDU byte at the beginning of each cell payload. In summary, user control and data information are encapsulated together within the payload segment of the frame.

Since the user control and user data information are encapsulated within the same frame, the existing protocol stack at the sending end-system incorporates identification to differentiate the user control field from the user data field within the same frame. Currently, one solution is to set the 8th-bit of the byte to 1 for all user control information in order to differentiate control messages from data messages. The protocol stack at the receiving end-system incorporates the ability to recognize and sort out user control messages from user data messages within the same frame. The receiving protocol stack then removes the special identification mechanisms used to differentiate between control and data messages before passing the user control message to the application program.

This paper advocates that each User-Control SDU and User-Data SDU be separately encapsulated within individual FAL frames. Figure 4 illustrates the FAL network protocol separate-encapsulation processing of information packets at the network level. As shown in Figure 4, only application-layer http command messages are individually encapsulated within a FAL frame. When FAL PDU information elements

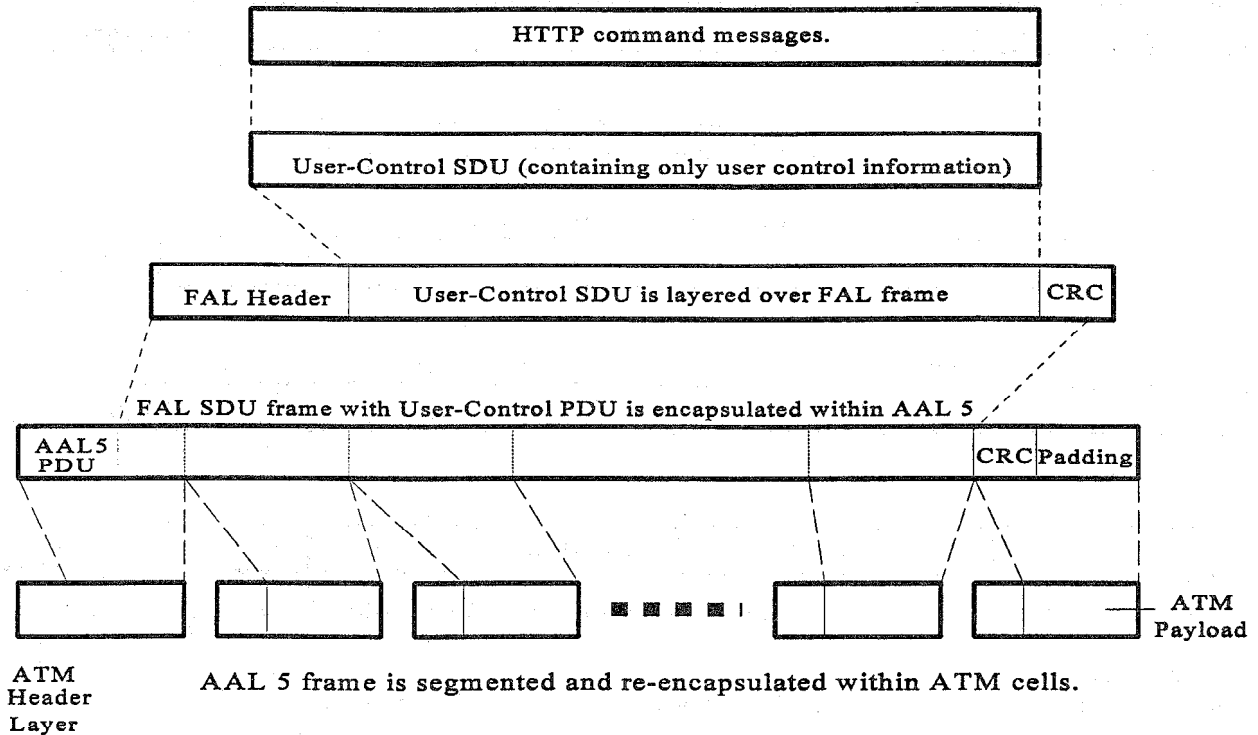


Figure 4: Separate Encapsulation Processing for ATM traffic

are encoded, each FAL-based User-Control SDU or User-Data SDU frame is then individually re-encapsulated within an AAL frame. The resulting AAL frame is then segmented and encapsulated into several ATM cell payload segments. The ATM cells are then transmitted over the same virtual channel connection using a First-In-First-Out (FIFO) scheme. The use of a FIFO scheme assures the synchronization of delivery of user control and user data information. Although the FAL frame is shown to be re-encapsulated within the AAL frame, FAL frames can also be re-encapsulated within IP, IPX, Frame-Relay and other carrier-based network protocols.

3.2 Technique to Separately Encapsulate User-Control and User-Data Information

The way in which an application's network-based communication Application Programming Interfaces (APIs) are designed can greatly affect the interaction of the application with the network protocol and the resultant mode of transmission. The following compares the differences between IP-based send primitives API and FAL-based send primitives APIs. The below semantic is issued by an application using IP-based send primitive API.

For IP User SDU:

```
IPsend(
    IN port_number
    IN ctrl&data_source
    OUT sending_result
)
```

Note that the IP-based send primitive does not indicate whether the message is a control or data message and allows programmers to bundle control and data messages. This is indicated by the input parameter "ctrl&data_source" which allows control and data messages to be downloaded simultaneously from the application memory and encapsulated together within the same packet.

FAL protocol differs in that FAL strictly enforces separate-encapsulation of user control and data information and extends the awareness of the differing application frames to the network-level through the use of unique FAL-based communication APIs. When application command information is sent by an application daemon control user interface channel, the application issues a FAL-based send control primitive API. The application's FAL-based API then issues a send data primitive when information is sent from the

application daemon data user interface channel. The following semantics detail the FAL-based send primitive APIs for sending separate user control and data to be encapsulated within separate User-Control SDU and User-Data SDU frames.

For User-Control SDU:

```
FALsendCtrl(  
  IN endpoint_identifier  
  IN ctrl_source  
  IN frame_length  
  OUT sending_result  
)
```

For FAL User-Data SDU:

```
FALsendData(  
  IN endpoint_identifier  
  IN data_source  
  IN frame_length  
  OUT sending_result  
)
```

This illustrates that, unlike existing network-based communication APIs, FAL-based communication APIs use distinct control and data send primitives. One important difference to note is that the input parameter "ctrl_source" constrains the input of messages to only control messages. Hence, the strict enforcement of the separate-encapsulation scheme is implemented with FAL-based send primitives. When FALsendCtrl primitives are used by an application control user interface, the FAL network protocol is also being notified that the message is user control information. FAL protocol then encodes the corresponding protocol information elements so that the frame can be identified as a user control frame. FAL protocol also instructs ATM protocol to encode the cells containing user control frames with CLP=0 and PTI with 111 coding. Applications using FAL-based communication APIs are also endowed with the ability to extend the awareness of user control and user data messages to the network-level.

3.3 Benefits of the FAL Protocol

Frame Adaptation Layer (FAL) protocol optimizes application-to-ATM communication services. *FAL protocol is designed with an extensible architecture.* With an extensible architecture, program libraries or handlers can be "plugged-in" or linked to FAL protocol to provide additional application-oriented network services. This would include emulation programs for interoperability among several existing legacy protocols. FAL is a connection-oriented packet-switching protocol designed to layer over ATM virtual

circuit-switching protocol. The main function of FAL protocol is to facilitate the integration of media and data communication. FAL has several unique advantages and the following discussion covers some of those major benefits and advantages.

Reliable transport services can be assigned to the burst-mode user control frames while media user data frames can be assigned unreliable transport services. This allows specifying user control information with reliable transport services and high priority indicators for transmission without overwhelming the ATM network. The unique ability to assign high priority traffic services to user-control SDUs reduces the potential of cell loss for the vital user control information during traffic congestion.

Other benefit which capitalizes on FAL's separate-encapsulation method is the ability to send different binary codes for user control and user data information. For example, the user control frames may contain machine-independent byte-codes while the user data frames may contain MPEG-2 encoded information. The separate user control frame simplifies the execution of interpreter programs for machine-independent byte-code commands since the requirement to sort out control and data messages is not required.

FAL protocol also enables the multiplexing and demultiplexing of User-Control SDU and User-Data SDU cells within the same Virtual Channel Connection (VCC) for delivery to a destination application which reduces the usage of precious virtual circuit resources. The interleaving of User-Control and User-Data SDUs within a single VCC also solves the problems associated with synchronization. Moreover, FAL-based User-Control and User-Data SDU frames can be re-encapsulated within either circuit-switching or packet-switching networks.

For secure communication within the Internet, software developers are using Point-to-Point Tunneling Protocol (PPTP). PPTP software allows end users to establish private communication using proprietary encrypted information transmission within the party line communication mode of the Internet. However, the use of PPTP software fragments the seamless communication feature of the Internet. The PPTP software utilizes encryption algorithms and each PPTP software vendor tends to implement their own proprietary algorithms. The plurality of proprietary PPTP software represents a handicap to the seamless communication feature of the Internet.

The use of virtual private lines of communication established by ATM and FAL protocols enables users to have private communications without using encryption schemes. FAL is designed with the flexibility and versatility to support new communication-oriented services, such as our proposed VACCP protocol for providing security to a network. The

FAL-enabled VACCP security mechanism performs network-based application access control filtering at the network-level. The network-based application access control security measures are implemented for remote access and remote control capabilities to a host computer so that unauthorized access can be terminated at the perimeter of the computer resources before access is granted to the computer's application programs or operating system. The use of network-based application access control security measures denies a malicious user, who has gain a connection to a targeted computer, the opportunity to snoop or to utilize the targeted computer resources to defeat the applications security mechanism. *More critically, when incorporated into FAL protocol services, the add-on security protocol can be standardized cross a platform of networks and computer systems for seamless communication.* The FAL-enabled VACCP security protocol can be implemented for computing communication in both client/server and peer-to-peer architectures.

FAL protocol can design the communication topology so that each application communication establishes a separate line of communication. This is especially useful when a user may have several applications communicating with several other applications on the called host computer. FAL protocol can also be used to establish several virtual circuit connections for a single application communication. This feature allows an application to utilize a Switched Virtual Circuit (SVC) channel hopping algorithm which is used to scramble the delivery paths of encrypted messages to another application over several virtual channel connections in order to defeat eavesdropping. *FAL protocol provides applications the ability to establish point-to-point and point-to-multipoint packet-switching communications to circuit-switching network communications.*

However, FAL protocol encounters the same problems as any other new protocol in that the requirement to create the respective FAL protocol program and Frame Header indexing must be developed. Configurable object-oriented higher layer protocol libraries would also need to be developed. The libraries would be used to dynamically program the network and communication services during call connection establishment based on the requirement of the application. Also proposed is to develop a software-based FAL protocol and signaling program which would be incorporated into computer operating systems as a socket program, like Winsock. Existing application-layer protocol services could also be modified to recognize and incorporate FAL-based API. The associated paper[1] covers the concept of FAL protocol in further detail.

3.4 Interleaving of User Control and Data Frames Within a VCC

Typically, Virtual Channel Connections (VCCs) are established over a virtual circuit-switching network between two endpoint host computer NICs. The VCCs serve as virtual communication links between the source and destination host computers and must be completely defined before any data transfer can occur and remain in place until the connections are terminated. ATM cells use the ATM Header Layer VPI/VCI values for data-link routing functions over the ATM network. A host computer may establish one or more pairs of upstream and downstream VCCs. The host computer may also multitask several applications and each application can generate differing types of application User Control and Data information at various intervals. Such information is individually encapsulated within the AAL5 frame. AAL 5 frames are widely used for transporting packetized data, voice, and video information over VCCs due to the simplicity of AAL 5 codings.

A comparison with IP elements illustrates how the interleaving of user data and control over the same VCC is accomplished. The VPI/VCI values are similar in usage to the IP destination-source addresses in that they are used as switching information for routing over a network topology. FAL's Endpoint Identifier serves a similar function as the Port Number of TCP. The FAL's Endpoint Identifier and the TCP's Port Number are associated or connected with an application user interface port or channel. An application may have two user interface channels: a control channel and a data channel. In this proposal, application control and data channels are each assigned a different FAL Endpoint Identifier value. Information generated by application control and data channels are encapsulated in separate frames and encoded with the corresponding FAL PDU information elements. Each FAL frame, with the appropriate Endpoint Identifier, is then layered over AAL5, which in turn is segmented and encoded within the payload segment of each ATM cell.

Information generated by an application control channel and a data channel can be multiplexed and routed using the same VPI/VCI value, since the application control and data channels are associated with a different FAL Endpoint Identifier value. This enables the interleaving of User Control and Data over the same VCC at the source end-system. At the receiving end-system, the FAL Endpoint Identifier value is used to demultiplex the information packets and to route the packets to their appropriate application control and data channels. This eliminates the need to establish a separate VCC for each data and control channel for each application. In perspective, the VPI/VCI value is used to route cells over a circuit-switching network to the host computer. At the host

computer, the FAL Endpoint Identifier values are used to route information frames to their destination application channels. It should be noted that end-to-end connection-based communications are established and maintained not only to the host computer, but also to the application user interfaces through the combined use of circuit-switching and packet-switching. Such end-to-end connection-based communication using FAL and ATM protocols can be used as a foundation to create a secure virtual dedicated-media internet system.[6]

4 Conclusion

Present network protocol is inadequate to provide optimum traffic services for reliable and secure real-time interactive multimedia application usages. This paper proposes a new Frame Adaptation Layer (FAL) protocol having unique frame encapsulation procedures. By strictly enforcing separate encapsulation of user control and user data information, different traffic services can be specified even when they originate from the same application. FAL protocol is designed to be used in combination with ATM technology in order to optimize traffic services to the differing types of information traffic that will exist in future multimedia applications. FAL protocol enables not only strict enforcement of separate-encapsulation of user control and user data information but also extends the awareness of disparate user information frames to the network-level. FAL protocol is a connection-oriented and packet-switching protocol which interfaces with application user interfaces. Using FAL over ATM, connection-oriented packet-switching communication can be implemented over a connection-based, circuit-switching network traffic topology. These end-to-end connection-based switching services, achieved through the use of the new FAL protocol and ATM protocol technology, can be used as a foundation for creating a virtual Dedicated-Media Internet system.

Terminology

Protocol Control - Information	Network control information exchanged between corresponding protocol entities to coordinate their joint operation.
Network Control - Information	Control information exchanged between corresponding network interface control-plane (NC-plane) entities to coordinate their joint operation.
User Control - Information	Control message or information exchange between corresponding user-control plane (UC-Plane) entities, host computer and application-defined control, application daemon command, user-interface (UI) command, RPC, ORB, or

User Data - Information	applet, to coordinate their joint operation. Non-control messages or information (burst-mode, audio, and video) exchange between corresponding user-data plane (UD-Plane) entities.
User-Control - SDU	A unit containing only user control information whose identity is preserved from one end of the layer connection to the other.
User-Data SDU -	A unit containing only user data (non-control) information whose identity is preserved from one end of the layer connection to the other.
User-Control - PDU	A unit of data specified in a layer protocol and consisting of protocol control information.
User-Data PDU -	A unit of data specified in a layer protocol and consisting of protocol control information.
Frame Adaptation-Layer	A connection-oriented packet-switching protocol for establishing protocol services for the user (application-defined) control and data information.

References

- [1] "A Versatile Frame Adaptation Layer (FAL) Architecture for ATM User Plane.", Chooi-Tian Lee and J.W. Harris, UNI-Net, Inc., April 1996.
- [2] "Native ATM Services: Semantic Description", Robert Callaghan, SAA API Ad-hoc Work Group, ATM Forum Technical Committee.
- [3] "Designing a Virtual Access Control Configuration Protocol for Implementation over ISDN and Shared-Media Networks.", Chooi-Tian Lee and J. W. Harris, IEEE 21st Conference on Local Computer Networks, Oct 1996.
- [4] "Separating Data and Control Transfer in Distributed Operating Systems", C. Thekkath, H. Levy, and E. Larowska, Sixth Int'l Conference in Architectural Support for Programing Languages and Operating systems, October 1994.
- [5] "Active Message: A Mechanism for Integrated Communication and Computation", T. Von Eicken et al. Int'l Symposium on Computer Architecture, May 1992.
- [6] "Proposal for Developing a Secure Electronic Commerce Environment Using VACCP and Virtual Dedicated-Media Internet", Chooi-Tian Lee and Jack W. Harris, UNI-Net Inc., April 1996.

