

Native ATM 서비스 상의 웹 시스템

성 종 진[†]

요 약

본 논문에서는, native ATM 서비스를 이용한 WWW 시스템을 제안한다. ATM API를 통한 native ATM 서비스의 이용은 IP over ATM이나 LAN Emulation 또는 Multiprotocol over ATM 등의 방법들에 비해 우수한 성능과 기능을 제공하는 것으로 알려져 있다. 우리가 개발한 WWW 브라우저와 서버는 이러한 native ATM 서비스 이용의 장점을 살려 성능의 우수성은 물론 QoS 보장이 가능한 진보된 WWW 서비스를 제공할 수 있도록 한다. 본 논문에서는 먼저 native ATM 서비스 방법과 ATM 인터넷 서비스 방법의 장단점을 비교한다. 그리고 native ATM 서비스의 제공을 위해서 ATM 포럼에서 표준화 하고 있는 ATM API에 대해서 그 특징 및 개발 추세를 설명한 후, 이를 이용한 WWW 시스템의 구조와 동작에 대한 내용을 기술한다. 시스템의 구조는 연결의 QoS를 보장할 수 있도록 하는 HTTP over ATM API를 근간으로 하며, HTML의 하이퍼링크 요소 내에서의 속성을 확장 정의하여 UNI 3.1과 Q.2931에서 정의하고 있는 ATM의 QoS 및 트래픽 특성 변수를 나타낼 수 있도록 하고 있다. 개발된 시스템에서의 ATM API는 WinSock 2 API를 사용한다.

Web System over Native ATM Service

Jongjin Sung[†]

ABSTRACT

In this paper, we present WWW system over native ATM services. The use of native ATM services through ATM API can provide better performance and functionality than that of IP over ATM, LAN Emulation or Multiprotocol over ATM. Our WWW browser and server provide advanced WWW services based on enhanced performance and guaranteed QoS support by using native ATM service benefits. This paper describes and compares advantages and disadvantages of Native ATM Services and ATM Internet Services, and addresses ATM API standardization and development trend that are made by the ATM Forum for the support of native ATM services, and then describes the architecture and operation of our WWW browser and server using ATM API. The system architecture is based on HTTP over ATM API capable of supporting guaranteed QoS over its connections. The system defines and uses new HTML attributes within hyperlinking HTML elements for the description of ATM QoS and traffic characteristics that are derived from UNI Signaling 3.1 connection characteristics information elements. Our system uses WinSock 2 API as its ATM API.

1. 서 론

ATM(Asynchronous Transfer Mode) 네트워크 상에

서 WWW(World Wide Web) 서비스를 이용하고자 하는 움직임이 늘고 있다. ATM의 우수한 네트워크 성능을 바탕으로 WWW의 편리한 하이퍼링크(hyperlink) 정보 탐색 방법과 HTML(Hypertext Markup Language) [1]을 사용한 비교적 간단하면서도 풍부한 정보표현

[†] 정 회 원: 한국전자통신연구원

논문접수: 1997년 8월 22일, 심사완료: 1997년 10월 29일

방법 등의 장점을 사용자에게 제공할 수 있기 때문이다.

ATM상의 WWW 서비스 구축을 위한 방법은 ATM 상의 응용 구축 기술에 따라 크게 두가지 유형으로 구분될 수 있다. 첫번째 방법은 인터넷 응용을 그대로 수용하기 위하여 인터넷의 IP를 ATM상에 올리는 ATM 인터넷 서비스 방법으로서, IPOA(IP over ATM) [2], LANE(LAN Emulation) [3], 그리고 MPOA (Multiprotocol over ATM) [4]등의 방법이 있으며, 현재 대부분의 ATM 네트워크 환경에서 이용되고 있다. 두번째 방법은 IP를 사용하지 않고 native ATM 서비스만을 이용하는 방법으로서 ATM상에 직접 WWW 서비스를 구축하는 방법이다. 이 방법은 ATM이 제공할 수 있는 기능과 성능을 그대로 응용이 이용할 수 있는 장점이 있다. 그리고 현재 이 기술에 대한 표준화와 그것을 따르는 제품의 상용화가 본격적으로 이루어지고 있어서 앞으로 그 사용의 활성화가 기대된다.

본 논문에서는 이 두번째 방법인 ATM상에서 직접 WWW 서비스를 구축하는 방법을 통하여 개발된 WWW 시스템에 대해 기술하고자 한다. 우리가 개발한 WWW 시스템에서는 자료전송을 위한 통신 프로토콜인 HTTP(Hypertext Transfer Protocol)[5]의 하부에 TCP/IP가 아닌 native ATM 서비스[6]를 이용함으로써, 자료 송수신을 위한 연결의 생성시 사용자가 원하는 QoS(Quality of Service) 및 트래픽 특성을 보장할 수 있도록 하여, 만족할 만한 품질의 정보 검색 서비스를 제공할 수 있도록 한다.

이 WWW 시스템에서 연결에 대한 트래픽(traffic) 특성을 입력하는 방법으로는, HTML을 확장하여 연결의 트래픽 특성을 기술할 수 있도록 하는 방법과 WWW 브라우저 응용 프로그램이 사용자에게 대화창을 제공하여 연결에 대한 트래픽 특성값을 입력할 수 있도록 하는 방법을 제공한다.

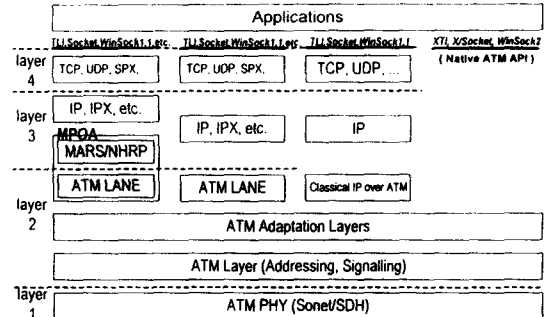
2. Native ATM 서비스와 ATM 인터넷 서비스

ATM의 통신 서비스를 이용하기 위한 방법으로는 ATM의 서비스를 직접 응용이 이용하도록 하는 방법과 ATM 서비스 상에서 기존의 네트워크 프로토콜들을 올려서 응용이 이를 통하여 이용하도록 하는 방법이 있다. 전자의 방법이 native ATM API를 통한 native

ATM 서비스 이용 방법이며, 후자의 방법은 IPOA, LANE, MPOA 등과 같은 방법들로서 ATM을 기반으로 인터넷 서비스를 제공할 수 있는 ATM 인터넷 서비스 방법들이다. 현재 상황에서는 기존의 인터넷 응용들이 인기를 끌고 있고 보편화되어 있는 까닭에 ATM의 통신 서비스를 쉽게 사용자에게 제공하기 위하여 ATM 인터넷 서비스 방법들이 많이 이용되고 있는 실정이다.

〈표 1〉 Native ATM 서비스와 ATM 인터넷 서비스와의 비교
 〈Table 1〉 Comparison between Native ATM Services and ATM Internet Services

	Native ATM Service	ATM Internet Service
APIs	Native ATM API (XTI, X/Socket, WinSock2, DLPI)	TCP(UDP)/IP API (XTI, X/Socket, WinSock2 /1.1)
Access to ATM Services	Direct access to AAL and/or ATM layer	Using LANE, MPOA, IPOA, ISI, etc.
Advanced Functionality	Full ATM characteristics (QoS, ATM multicasting, etc.)	Future support of QoS and multicasting by using RSVP, MARS, NHRP, PIM, etc.
Applications	New applications	Legacy and new applications
Internetworking with legacy LANs	No (but, possible with CIF method)	Yes



(그림 1) MPOA, LANE, IPOA, Native ATM 기술의 프로토콜 계층
 (Fig. 1) Protocol Stacks for MPOA, LANE, IPOA, and Native ATM

이 두가지 방법에 대한 비교를 〈표 1〉에 나타내었다. 두 방법의 장단점을 살펴보면 다음과 같다. ATM 인터넷 서비스를 이용할 경우는 ATM 서비스 상에서 IPOA, LANE, MPOA 등의 기술을 적용하여 인터넷의 TCP/IP 서비스를 제공하게 된다. 따라서 기존의 TCP/IP 응용들을 수정없이 그대로 이용할 수 있다는 장점을 갖는 반면 ATM의 고유한 특성 중 QoS 제어와 ATM의 멀티캐스팅 기능등 TCP/IP가 제공하지 못하는 중요한 장점부분을 잃어버리게 된다. 이를 극

복하기 위해서 MPOA나 차세대 인터넷 기술들에서는 RSVP(Resource Reservation Protocol), MARS (Multicast Address Resolution Service), NHRP(Next Hop Resolution Protocol) 등의 새로운 기술들을 모색하고 있으나 프로토콜 구조가 복잡해져 처리속도의 지연을 유발하는 등 또 다른 단점들이 나타나게 된다.

Native ATM 서비스를 이용할 경우는 응용이 ATM 계층이나 AAL(ATM Adaptation Layer)의 서비스를 직접 이용하므로 ATM 고유의 장점을 충분히 살릴 수 있는 반면에, 기존의 인터넷에서 사용되던 응용들을 그대로 사용할 수 없고 이들과 연동(interworking)이 되지 않는 단점이 있다. 그러나 이러한 단점을 논하기에 앞서 기존의 응용들은 ATM의 장점을 이용하지 않고 있는 것들이므로 ATM의 장점을 이용하고자 하는 진보된 응용의 개발을 고려한다면 native ATM 서비스를 이용하는 새로운 응용의 제작이 바람직하다는 점을 염두에 두어야 할 것이다.

우리가 개발한 WWW 시스템은 native ATM 서비

스를 이용한 QoS를 제어하고 보장할 수 있는 기능을 갖추고 있다. 다음 절에서는 native ATM 서비스의 이용을 위한 native ATM API 기술에 대해 설명한다.

3. Native ATM API

Native ATM 서비스를 이용하기 위해서 응용이 접근하는 인터페이스를 native ATM API(이하, 줄여서 ATM API라 칭함)라고 하며 이것을 통하여 응용이 ATM 계층이나 AAL에 직접 접근할 수 있게 된다.

ATM API 기술에 대한 표준화 작업은 ATM 기술 전반에 대한 표준화를 주도하고 있는 ATM 포럼에서 활발하게 이루어지고 있다. 현재 ATM 포럼에서는 ATM API 자체의 표준을 정의하는 대신에 native ATM 서비스에 대한 의미 명세인 "Native ATM Services: Semantic Description[6]"을 표준으로 제정하여 이를 따르는 ATM API의 개발을 권고하고 있으며, 1996년 2월에 UNI(User Network Interface)3.0과 3.1의 서비스[7]를 바탕으로 한 버전 1.0 표준을 발표하

<표 2> ATM 포럼의 native ATM 서비스와 XTI, WinSock 2 API의 프리미티브 매핑

<Table 2> Mapping of Primitives between ATM Fourm's native ATM service, XTI and WinSock 2 API

	XTI	WinSock 2
ATM_abort_connection(req)	t_close	WSASendDisconnect + closesocket
ATM_accept_incoming_call(res)	t_accept	WSAAccept returning CF_ACCEPT
ATM_add_party(req)	t_addleaf(new)	WSAJoinLeaf
ATM_add_party_reject(con)	t_rcvleafchange(new)	FD_CONNECT with error codes
ATM_add_party_success(con)	t_rcvleafchange(new)	FD_CONNECT without error codes
ATM_arrival_of_incoming_call(ind)	t_listen	WSAAccept
ATM_associate_endpoint(req)	t_open	socket, WSASocket
ATM_call_release(req)	t_snddis	WSASendDisconnect
ATM_call_release(ind)	t_rcvdis	FD_CLOSE network event
ATM_connect_outgoing_call(req)	t_connect	WSAConnect, WSAJoinLeaf
ATM_drop_party(req)	t_removeleaf(new)	WSASendDisconnect + closesocket()
ATM_drop_party(ind)	t_rcvleafchange(new)	FD_CLOSE network event
ATM_get_local_port_info(req)		
ATM_P2MP_call_active(con)	t_rcvconnect	FD_CONNECT(connecting side), none(listening side)
ATM_P2P_call_active(con)	t_rcvconnect	FD_CONNECT(connecting side), none(listening side)
ATM_prepare_incoming_call(req)	t_bind	bind + listen
ATM_prepare_outgoing_call(req)	t_bind	bind
ATM_query_connection_attributes(req)	t_optmgmt	WSAIocli, getsockopt
ATM_reject_incoming_call(res)	t_snddis	WSAAccept returning CF_REJECT
ATM_set_connection_attributes(req)	t_optmgmt	WSAIocli, setsockopt
ATM_wait_on_incoming_call(req)	t_bind	WSAAccept, WSAAsyncSelect with FD_ACCEPT
ATM_send_data(req)	t_snd	send, WSAend
ATM_rcv_data(req)	t_rdv	recv, WSARcv

였다.

ATM 포럼의 표준 의미 명세에 따라 ATM 서비스를 이용할 수 있도록 개발된 실제적인 ATM API의 대표적인 두가지 예를 들면 UNIX 계열 플랫폼에서 사용될 수 있는 XTI[9] 및 X/Socket[10]과 마이크로소프트사의 윈도우즈 계열 플랫폼에서 사용될 수 있는 WinSock 2 API[11]를 꼽을 수 있다. 이들 두가지 ATM API들은 최근에 ATM 포럼에 의해 표준 의미 명세를 따르는 사양을 정의한 것으로 인정 받았으며, 앞으로 각각 UNIX와 마이크로소프트 윈도우즈 환경의 통신 응용 제작에 영향력을 미칠 것으로 전망된다.

ATM API 개발의 초기인 1996년도 전반 이전에는 ATM 어댑터 카드 개발 회사들에 의해 여러 ATM API들[12-14]이 개발되고 있었으나 최근의 추세는 통신 응용 프로그래밍에서 가장 많이 사용되어 업계의 표준으로 인식되고 있는 WinSock 2 API와 XTI, X/Socket API 등을 지원하는 방향으로 나아가고 있어 대부분의 ATM 장비 개발 업체들은 자사의 장비 위에 WinSock 2나 XTI, X/Socket을 지원하기 위한 서비스 제공자 드라이버를 개발하여 제공하고 있거나 할 계획이다.

본 논문에서 나타내고 있는 native ATM 서비스 상의 WWW 시스템은 WinSock 2 API를 사용하였으며, ATM API로서의 WinSock 2 API 이용을 위해서 특히 WinSock 2 API 규격 중에서 Protocol-Specific An-

nex의 ATM 부분[15]에 기술되어 있는 ATM-Specific QoS Extension 부분을 중점적으로 활용하였다. 이 부분은 ITU-T의 Q.2931[16]에 정의되어 있는 IE(Information Element)의 내용을 그대로 도입한 것이라 할 수 있다.

4. QoS 보장이 가능한 WWW 브라우저와 서버

WWW 브라우저와 서버는 자료 전송을 위하여 HTTP를 이용하며, 그 하부에는 TCP/IP를 이용한다. TCP/IP는 전송 서비스의 품질, 소위 QoS를 고려하지 않는 통신 프로토콜로써 트래픽에 대한 특성을 보장할 수 없다.

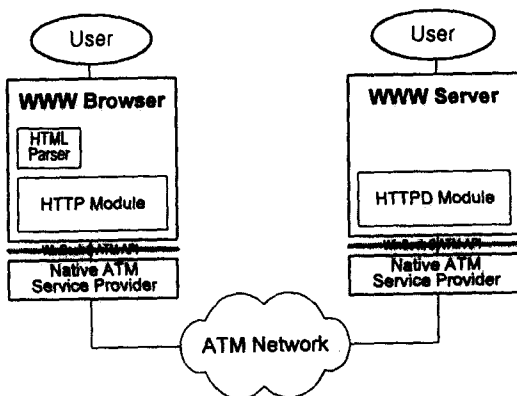
한편, WWW을 통해서 전달되는 자료는 멀티미디어 자료로써 그 자체의 미디어 구성 형태와 자료의 용도 및 성격에 따라 전송시에 요구되는 트래픽 특성과 통신 서비스의 품질은 다양해진다. 즉, 전송의 긴급성과 연속성, 그리고 전송량의 균일성 등의 특성과 서비스 품질 면에서 자료들 마다 서로 다양한 값을 요구하게 된다.

TCP/IP 상의 기존의 WWW 서비스는 이러한 다양한 전송 특성에 대한 요구를 수용하지 못하고 모든 종류의 자료에 대해 동일한 성질의 무원칙적인 통신 서비스 연결만을 제공하여 왔다. 따라서 동시에 다수의 사용자가 WWW 서비스를 이용하게 될 경우에는 만족할 만한 자료의 전송이 이루어질 수 없었다.

HTTP 모듈이 native ATM 서비스의 장점을 활용함으로써 이를 바탕으로 한 WWW 시스템은 사용자가 설정하는 QoS 및 트래픽 특성을 보장할 수 있게 된다. 본 절에서는 native ATM 서비스 상에서 HTTP를 사용하는 WWW 브라우저와 서버에 대해서 기술한다. 먼저 시스템의 구조와 환경을 나타내고, ATM API를 이용한 HTTP의 동작 절차와 특징들에 대해서 알아본다. 그리고 WWW 브라우저에서 제공하는 QoS 및 트래픽 특성의 입력 방법을 구체적으로 설명한다.

4.1. 시스템의 구조 및 환경

Native ATM 서비스를 이용하여 QoS 및 트래픽 특성을 보장할 수 있는 WWW 브라우저와 서버의 통신 환경은 순수한 ATM 네트워크를 바탕으로 한다. Native

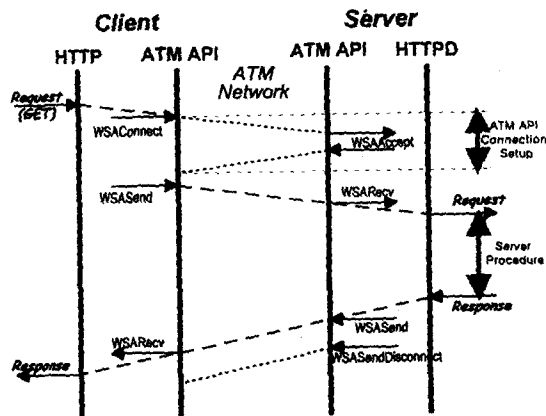


(그림 2) Native ATM 서비스 상의 WWW 브라우저와 서버의 구조

(Fig. 2) Architecture of the WWW Browser and Server using native ATM Services

ATM 서비스를 이용하게 되는 WWW 시스템은 IP를 사용하고 있는 인터넷에서의 기존의 WWW 시스템들과는 연동이 되지 않는다. 본 논문에서 나타내고 있는 WWW 시스템에서 이용하는 ATM API는 앞 절에서 설명한 WinSock 2 API이며 시스템의 환경은 PC 플랫폼 상의 마이크로소프트 윈도우즈 95 환경이다.

본 WWW 시스템의 구조는 (그림 2)와 같다. 그림에서 보는 바와 같이 시스템의 구성은 클라이언트 응용 프로그램인 WWW 브라우저와 서버 응용 프로그램인 WWW 서버로 이루어져 있으며, 클라이언트 측에서는 HTTP 모듈이 존재하고 서버 측에서는 HTTPD (HTTP Demon) 모듈이 존재한다. QoS 보장이 가능한 WWW 시스템에서의 HTTP와 HTTPD는 ATM API를 통해 제공되는 하부의 native ATM 서비스를 이용하여 연결을 생성하고 자료 송수신을 수행하게 된다.



(그림 3) HTTP over ATM API에서의 GET 프리미티브의 동작 (Fig. 3) Primitive Operation Procedure of HTTP over ATM API

4.2. HTTP over ATM API

(그림 3)에서 HTTP의 GET 메소드(method)를 예로 들어 HTTP가 ATM API(WinSock 2 API)의 프리미티브 함수를 이용하는 절차를 도식적으로 표현하였다.

이 그림의 ATM API 연결 설정(connection setup) 부분인 WSAConnect와 WSAAccept 함수의 이용 절차에서 바로 연결에 대한 QoS 및 트래픽 특성을,

WinSock 2 API 규격의 ATM-Specific QoS Extension 자료 구조를 이용하여 입력시키게 된다. 일단 이 ATM API 연결 설정이 성공적으로 완료되어 원하는 특성으로 연결이 생성되고 나면, 그 연결을 통해서 HTTP의 GET Request 메시지가 서버 측의 HTTPD로 전달되게 된다. 서버 측에서는 그 GET Request 메시지에 실려온 질의 내용에 따라 클라이언트가 요구하는 자료를 찾아내어 클라이언트 측으로 전송할 Response 메시지에 담아서 보내게 된다. Request와 Response 메시지의 전달시에는 WSASend와 WSARecv 함수가 이용된다. GET 이외의 OPTIONS, HEAD, POST, PUT, DELETE, TRACE의 기타 메소드들에 대해서도 GET과 동일한 절차와 방법으로 ATM API를 이용하여 HTTP가 동작하게 된다.

본 시스템에서 사용하는 HTTP는 기존의 HTTP가 사용하는 TCP/IP 인터페이스가 아닌 ATM API를 사용하지만, 두 인터페이스의 경우 모두 마이크로소프트 윈도우즈 환경에서는 WinSock 2 API를 이용할 수 있다. 다만 이 WinSock 2 API를 통하여 접근하게 될 하부의 통신 서비스 제공자가 Native ATM용과 TCP/IP 용으로 달라지게 되고 사용하는 프리미티브 함수와 매개변수들이 일부분 달라지는 것이다. ATM API로서의 WinSock 2 API를 위한 ATM 서비스 제공자는 ATM 어댑터카드 제조 회사에 의해 공급된다.

ATM API를 이용한 데이터의 전송에 있어서 신뢰성을 보장하기 위한 방법으로 ATM 포럼의 표준에서는 데이터 전송 단계에서 SSCOP(Service Specific Connection Oriented Protocol)을 AALS 상에서 사용하도록 하고 있다. 기존의 SSCOP은 제어 단계의 연결설정시 시그널링을 위한 신뢰성 보장을 위하여 사용되지만, ATM API의 이용에서는 데이터 전송 단계에서도 신뢰성있는 전송을 위하여 SSCOP을 사용하도록 하였다. 실제 WinSock 2를 이용한 응용 제작에서는 AALS_PARAMETERS 자료구조의 SSCSType 멤버 변수를 "SSCOP_RELIABLE_SSCS"로 세팅하면 된다.

4.3. WWW 서비스 절차

아래에서는 HTTP over ATM API를 바탕으로 한 WWW 브라우저와 서버에서의 실제 WWW 서비스 절차를 알아본다.

- ① 사용자의 입력행위에 의해 WWW 브라우저내에

서 특정 원격지에 존재하는 자료를 읽어 올 요구가 발생한다.

- ② 요구된 자료로의 연결 생성시 고려해야 할 QoS 및 트래픽 특성이 WWW 브라우저내의 HTML 파서(parser)에 의해 파악되어 인식된다.
- ③ 인식된 연결에 대한 특성 요구 사항을 HTTP 모듈로 전달한다.
- ④ 전달된 연결에 대한 특성 요구 사항에 따라, HTTP 모듈은 적절한 ATM API 프리미티브와 매개변수를 이용하여 HTTPD와의 연결을 생성한다.
- ⑤ 생성된 연결을 통하여 브라우저의 HTTP 모듈은 자료 요청을 Request 메시지를 서버측으로 보낸다.
- ⑥ HTTPD를 통하여 브라우저 측으로부터의 Request 메시지를 받은 WWW 서버는 그 메시지에 지시된 브라우저 측의 요청에 따라 자료를 찾아내는 처리를 한다.
- ⑦ 서버의 처리 결과를 Response 메시지에 담아 브라우저에게로 보낸다.
- ⑧ 브라우저가 HTTP를 통해 전달받은 결과를 적절히 사용자에게 출력한다.

이러한 WWW 서비스 처리 절차에서 QoS 및 트래픽 특성은 HTML 파서에 의해 파악되어 HTTP 모듈로 전달된다. 이것은 QoS 및 트래픽 특성을 표기하는 방법에 대한 설명은 다음 절에서 자세하게 나타낸다.

4.4. QoS 특성 입력 방법

본 WWW 브라우저와 서버에서는 QoS 및 트래픽 특성을 입력하는 방법으로 다음의 두가지를 사용한다.

- 연결의 QoS 및 트래픽 특성의 설정을 위한 HTML 속성(attribute)을 새로 정의하여 포함시키는 방법
- WWW 브라우저에서 대화창을 제공하여 사용자로 하여금 원하는 연결의 특성값을 기입하게 하는 방법

첫번째 방법은 HTML을 이용하여 웹 자료를 작성하는 정보 제공자가 자신의 웹 페이지에서 다른 웹 정보를 연결하여 가져올 수 있도록 하고자 할때 그 연결에 대해 적절한 특성값을 HTML 문서 내에 입력하기 위해 사용할 수 있는 방법이다. 그리고 두번째 방법은 첫번째 방법에 의한 연결 특성의 입력이 되어 있지 않은 HTML 자료들을 이용하게 될 경우 사용할 수 있는 기본적인(default) 연결의 특성을 사용자의

의도에 의해 입력하여 적용되도록 하는 방법이다.

4.4.1. 새로운 HTML 속성 정의

본 WWW 브라우저에서의 주된 연결 특성 입력 방법으로, 새로운 HTML 속성을 정의하여 연결에 대한 특성값을 기술하는 방법이다. 아래의 예에서와 같이, HTML 문서 내에서 원격지 자료로의 연결 생성이 요구되는 HTML 요소(element)들을 사용할 때, 그것의 시작태그(start tag)내에서 속성(attribute)의 형태로 나타낼 수 있도록 연결에 대한 QoS 및 트래픽 특성을 표현하기 위한 새로운 HTML 속성을 정의한다.

[예 1]

```
<A HREF="http://pec.etri.re.kr/welcome.html"
FWD_PCR_CLP0=10
BAK_PCR_CLP0=10
.....
FWD_QOS_CLASS=2)
TEXT
</A>
```

[예 2]

```
<IMG SRC="../icons/abc.gif"
FWD_PCR_CLP0=20
BAK_PCR_CLP0=20
.....
FWD_QOS_CLASS=5)>
```

예 1에서는 <A> 요소의 시작태그 내에 "pec.etri.re.kr/welcome.html"로의 연결에 적용될 QoS 및 트래픽 특성을 위해서 FWD_PCR_CLP0, BAK_PCR_CLP0, FWD_QOS_CLASS 등의 새로 정의된 속성들이 사용되었다. 예 2에서는 요소의 태그 내에서 이미지 소스인 "../icons/abc.gif"로의 연결에 적용될 QoS 및 트래픽 특성을 위해서 역시 FWD_PCR_CLP0, BAK_PCR_CLP0, FWD_QOS_CLASS 등의 속성과 그에 대한 특성 값의 할당이 기술되어 있다.

HTML에서 원격지로의 연결생성이 요구되는 부분은 하이퍼링크를 포함하는 다음과 같은 HTML 요소들이다.

- <A>-HREF 속성을 포함한 경우

- <LINK>
-
- <INPUT>--SRC 속성을 포함한 경우
- <ISINDEX>
- <FORM>--"METHOD=GET" 포함한 경우

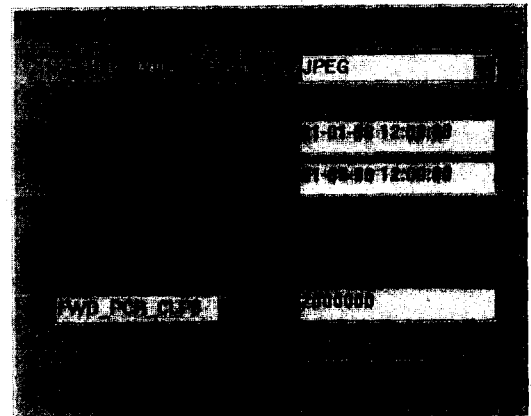
이 요소들에 의해 나타내어지는 하이퍼링크 연결에 적용하기 위한 새로 정의된 연결특성의 속성들은 <표 3>과 같다. HTML 문서의 작성시에 이들의 일부를 필요에 의해 선택적으로 사용하게 된다. <표 3>의 HTML 연결 속성들은 ATM API에 대한 표준인 "Native ATM Services: Semantic Description, Version 1.0"의 Annex A 부분에 정의되어 있는 연결 특성(connection attributes)과 동일한 내용이다. 이것들은 ATM UNI 시그널링 3.1에서 정의하고 있는 QoS 및 트래픽 기술 변수들, 즉 ITU-T Q.2931에 정의되어 있는 Traffic Descriptor 관련 IE들에 해당하는 것들이다. 이 내용에 따라 WinSock 2 API의 ATM-Specific QoS structure 자료구조가 정의되어 있으므로, <표 3>의 HTML 연결 속성을 이용하여 기술한 연결에 대한 QoS 및 트래픽 특성은 그대로 WinSock 2 API의 매개변수로 매핑될 수 있게 된다.

BLU_SELECTOR	
LAYER 2_ID	
LAYER 2_MODE	
LAYER 2_WINDOW_SIZE	
LAYER 2_USER_ID	
LAYER 3_ID	Broadband Low Layer Information
LAYER 3_MODE	
LAYER 3_PACKET_SIZE	
LAYER 3_WINDOW_SIZE	
LAYER 3_USER_ID	
LAYER 3_PPI_ID	
LAYER 3_OH_ID	
LAYER 3_PID_ID	
CALLED_ADDR_FORMAT	Called Party Number
CALLED_ADDR	
CALLER_ADDR_TYPE	Called Party Subaddress
CALLER_ADDR	
CALLING_ADDR_FORMAT	Calling Party Number
CALLING_ADDR	
PROBATION_ADDR_IND	
SCREENING_IND	
CALLING_SUBADDR_TYPE	Calling Party Subaddress
CALLING_SUBADDR	
CAUSE_CODE_IND	Cause
CAUSE_LOCATION	
CAUSE_VALUE	
CAUSE_DIAGNOSTICS	
FWD_QOS_CLASS	Quality of Service Parameter
BAK_QOS_CLASS	
NETWORK_ID	Transit Network Selection

이렇게 정의된 새로운 HTML 속성들은 WWW 브라우저 내의 HTML 파서에 의해 인식되도록 하고 그렇게 인식된 연결의 특성 내용들을 그대로 HTTP 모듈이 이용하도록 하여 그 특성 내용에 맞는 연결을 생성하도록 한다.

<표 3> 새로 정의된 HTML 연결 특성
<Table 3> Newly Defined HTML Connection Attributes

Connection Attribute	Category
AAL_TYPE	ATM Adaptation Layer Parameters
AAL1_SUBTYPE	
AAL1_CBR_RATE	
AAL1_MULTIPLIER	
AAL1_CLOCK_RECOVERY_TYPE	
AAL1_ERROR_CORRECTION	
AAL1_STRUCTURED_DATA_TRANSFER	
AAL1_PARTIALLY_FILLED_CELLS	
AALS_FWD_MAX_SDU	
AALS_BAK_MAX_SDU	
AALS_SSCS_TYPE	
USER_DEFINED_AAL_INFO	
FWD_PCR_CLP0	
FWD_PCR_CLP1	
BAK_PCR_CLP0	
BAK_PCR_CLP1	
FWD_SCR_CLP0	
FWD_SCR_CLP1	
BAK_SCR_CLP0	
BAK_SCR_CLP1	
FWD_MBS_CLP0	
FWD_MBS_CLP1	
BAK_MBS_CLP0	Broadband Bearer Capability
BAK_MBS_CLP1	
BEST_EFFORT	
FWD_TAGGING	
BAK_TAGGING	Broadband High Layer Information
BEARER_CLASS	
TRAFFIC_TYPE	
TIME_REQ	
CLIPPING_IND	
CONNECT_CONFIG	
APPL_ID_TYPE	
APPL_ID	



(그림 4) 연결 특성 입력을 위한 대화창
(Fig. 4) Dialog Box for Accepting Connection Attributes

4.4.2. 대화창을 통한 입력

WWW 브라우저에 의해 사용자에게 (그림 4)와 같은 대화창을 제공하여 원하는 연결의 특성을 입력할

수 있도록 한다. 이 방법에서 대화창을 통해서 사용자에게 입력할 수 있도록 제시하는 연결특성 항목으로는 역시 HTML의 확장된 속성 내용과 동일한 것들로서, <표 3>에 나열한 연결속성에 해당하는 항목들이다.

이 방법에서는 사용자로 하여금 특정 시간대와 특정 종류의 자료에 대해 원하는 연결의 특성을 기입할 수 있도록 하여 다양한 연결의 특성 설정이 가능하도록 한다.

대화창을 통한 연결의 특성 입력은 앞 절에서의 HTML 문서 내에 기입하는 방법을 사용하지 않을 경우 기본적으로 적용될 수 있는 방법이 된다. 특히 가져올 자료의 미디어 유형별로 적절한 특성 값을 미리 정해둘 수 있으므로 확장된 HTML을 사용하지 않는 대부분의 다른 WWW 자료를 읽어올 때, 이미지, 텍스트, 비디오, 오디오 등 큰 차이를 보이는 미디어별 전송 특성을 고려한 효과적인 WWW 서비스를 실현할 수 있다. 미디어 유형에 대한 기본값은 HTML이다.

4.5. 개발된 WWW 시스템의 사용 시험

이상의 기능을 갖춘 WWW 브라우저는 QoS 및 트래픽의 설정을 이용하여 사용자가 원하는 품질의 서비스를 보장할 수 있다. 주로 트래픽에 관련된 PCR (Peak Cell Rate), SCR(Sustainable Cell Rate), MBS (Max Burst Size), Traffic Type 등과 관련된 변수들을 사용하여 만족할 만한 성능의 WWW 검색을 할 수 있다. 개발된 WWW 브라우저에서의 테스트에서는 대부분의 경우 2Mbps의 BAK_PCR_CLP0(서버→브라우저 방향 PCR)을 할당하여 연결을 설정한 후 검색하면 웬만한 HTML 데이터는 순식간에 가져와서 화면에 보여준다.

개발된 WWW 브라우저에서 5메가 바이트 크기의 MPEG 동영상 데이터를 가져오도록 할 때, MPEG 타입에 대해 BAK_PCR_CLP0을 2Mbps로 설정했을 때와 200Kbps로 설정했을 때의 차이가 뚜렷이 비교된다. BAK_PCR_CLP0가 2Mbps일 경우는 클릭을 한 후 약 5초 이내에 화면에 출력이 시작된다. 그러나 BAK_PCR_CLP0가 200Kbps일 경우는 클릭을 한 후 약 27초 이후에나 출력이 시작된다. 트래픽 형태는 CBR(Constant Bit Rate)로 설정했었으며 MPEG 출

력을 위한 프로그램은 윈도우 95용 MPEG 재생기를 WWW 브라우저에서 연결시켜 불러내도록 하였다. 이것은 서버 측으로부터 브라우저 측으로 데이터를 전송 받을 때 걸리는 시간외에도, 브라우저 측에서 서버측으로의 메시지 전달 시간과 서버측에서의 데이터 액세스 처리 시간 그리고 브라우저로 전달된 데이터를 출력하기 위한 MPEG 재생기를 불러내어 동작시키는데 걸리는 시간들을 모두 포함한 시간이다.

이러한 연결 특성 설정 기능의 제공과 함께 설정된 연결 특성을 보장해 줄 수 있는 서비스이므로 일단 연결이 생성된 이후에는 주위에 다수의 다른 연결의 생성이 시도되더라도 이 연결의 특성을 저하시키지는 못한다.

개발된 웹 브라우저와 서버를 이용하게 되면, 멀티미디어를 사용한 웹 정보의 구축과 사용에 있어서 동영상과 오디오, 이미지 등의 데이터를 적절한 연결 특성으로 설정하여 사용자가 원하는 만족할 만한 품질의 웹 서비스를 제공할 수 있을 것이다.

5. 결 론

ATM 네트워크 상에서 WWW 시스템의 구축을 위한 기존의 방법들은 IPOA, LANE, MPOA 등과 같은 ATM 인터넷 서비스 기술을 바탕으로 인터넷의 IP를 덧씌운 후 HTTP over TCP/IP의 계층을 이용하고 있어서 TCP/IP가 제공하지 못하는 ATM 고유의 장점들을 이용하지는 못하고 있다.

본 논문에서는 보다 나은 품질의 WWW 서비스를 native ATM 서비스 상에서 실현한 방법과 시스템을 나타내었다. Native ATM 서비스를 제공할 수 있는 ATM API를 이용하여 QoS의 보장이 가능한 WWW 서비스를 제공하도록 하였다. 시스템의 구성은 연결의 QoS를 보장할 수 있도록 하는 HTTP over ATM API를 근간으로 하며 HTML의 하이퍼링크 요소 내 속성을 확장 정의하여 UNI 3.1과 Q.2931에서 정의하고 있는 ATM의 QoS 및 트래픽 특성 변수를 나타낼 수 있도록 하고 있다. 개발된 시스템에서의 ATM API는 WinSock 2 API를 사용하고 있다.

참 고 문 헌

- [1] Berners-Lee, T. and Connolly, D., "Hypertext Markup Language-2.0," IETF RFC 1866, Nov., 1995.
- [2] Lauback, M., "Classical IP and ARP over ATM," IETF RFC 1577, Jan., 1994.
- [3] ATM Forum, '*LAN Emulation over ATM, Version 1.0*,' af-lane-0021.000, Jan., 1995.
- [4] ATM Forum, '*Multiprotocol over ATM, Version 1.0*,' Just Passed in the Final Ballot, Jul., 1997.
- [5] Fielding, R., et al, "Hypertext Transfer Protocol-HTTP/1.1," IETF RFC 2068 Proposed Standard, Jan., 1997.
- [6] ATM Forum, '*Native ATM Services: Semantic Description, Version 1.0*,' af-saa-0048.000, Feb., 1996.
- [7] ATM Forum, '*ATM User-Network Interface Specification, Version 3.1*,' af-uni-0010.002.
- [8] ATM Forum, '*Traffic Management 4.0*,' af-tm-0056.001.
- [9] ATM Forum, '*ATM Protocol-Specific Appendix of X/Open's XTI API*,' ATM_Forum/95-1375R2, Dec., 1995.
- [10] ATM Forum, '*XNET ATM API Specifications, Appendix Y*,' ATM_Forum/96-1169, Oct., 1996.
- [11] WinSock Group, '*Windows Sockets 2 Application Programming Interface, Revision 2.1.0*,' Jan., 1996.
- [12] Almesberger, W., '*Linux ATM API Draft, Version 0.3*,' Mar., 1996.
- [13] Fore system, '*ATM Sbus Adapter User's Manual*'.
- [14] Gupta, U., '*ATM Sbus Adapter Solaris Driver Design Specification*,' Interphase Corp.'s Doc. No. ES00050, Dec., 1993.
- [15] ATM Forum, '*WinSock 2 ATM Annex*,' ATM_Forum/96-0190, Feb., 1996.
- [16] ITU-T Recommendation Q.2931, '*UNI Layer 3 Specification for Basic Call/Connection Control*'.
- [17] Ross, T., "ATN APIs: The Missing Links," Data Communications, pp. 119-128, Sep., 1995.



성 증 진

1990년 경북대학교 전자공학과
졸업(학사)
1992년 경북대학교 전자공학과
대학원 졸업(석사)
1992년~현재 한국전자통신연
구원 근무
관심분야: ATM 통신망 및 서비
스, 인터넷 서비스