UBR 위에서 동작하는 TCP/IP 성능 평가
(Performance Evaluation for TCP/IP over UBR)

안 성 수 † 유 흥 식 † 황 선 호 †† 이 준 원 ††† 김 성 운 ††††
(Sungsoo Ahn) (Hyungsik Yu) (Sunho Whang) (Junwon Lee) (Sungun Kim)

요 약 멀티미디어 서비스 통합에 대한 핵심적 기술로 ATM기술이 각광을 받게됨에 따라 이에 대한 검증 및 사용자에게 일정 수준의 신뢰성을 보장하기 위해서 다양한 성능 인자를 측정하고 이를 분석함으로써 수행가능한 망의 최대 수율, 즉 망의 성능을 평가하는 성능 시험에 대한 관심과 중요성이 점점 높아지고 있는 추세이다. ATM 망의 성능 시험 및 평가는 망의 관점에서 셀 레벨의 QoS를 이용하는 방법과 사용자 관점에서 프레임 레벨의 메트릭을 이용하는 방법이 있으며, 현재 이에 대한 표준화가 진행중이다. 본 논문에서는 ATM UBR 서비스를 통해서 TCP/IP를 수용하기 위해 TCP 측면에서의 성능 요구사항을 추출하고, 이를 ATM 망 및 패킷망에 적용시켜 시뮬레이션을 수행하였다. 시뮬레이션을 통해 성능을 분석 및 평가한 결과, TCP혼잡제어 기법과 ATM 셀 폐기 정책간의 상호 작용 문제로 인해 전체적인 성능을 낮게 측정되었으며, 이에 대한 개선 방안으로 기존 TCP 혼잡제어 기법 연구에서 제안된 TCP Vegas를 발전시킨 Accelerated Vegas를 제안한다.

Abstract ATM is a key technology of integration of multimedia service. Recently, Many study have been concentrated on performance testing for evaluation network performance are stronger everyday. The performance testing is on evaluation of maximal throughput of network by measuring and analyzing of various performance parameters. There are two ways to test ATM network performance; one is using QoS in cell level on the point of network's view, and the other is using metric in frame level in the point of user's view. And, the standardization process is also under way. In this paper, we derive a performance requirement of TCP in TCP/IP data transmission over ATM UBR service. By applying the derived requirements to ATM and packet networks, we evaluate the performance of TCP over UBR based on the result of our simulations. Therefore, we evaluate the result of simulation and find degradation of network throughput by interaction between TCP congestion control and ATM cell drop policy. So we suggest the accelerated Vegas that modify traditional TCP Vegas in congestion control mechanism for batter network throughput.

1. 서 론

멀티미디어 서비스에 대한 요청이 급증함에 따라 대역폭, 트래픽 수용 능력 및 확장성 등 여러 측면에서 멀티미디어 서비스 제공에 가장 적합한 기술로 평가받는 ATM 망에 대한 연구가 활발히 진행중이다. 이에 따라 사용자에게 일정 수준의 신뢰성을 보장하기 위해 ITU-T, ATM Forum 등 여러 표준화 기구들은 ATM 프로토콜에 대해 적합성 시험 및 상호운용성 시험에 주력하여 왔으나, 최근에는 QoS 및 망 성능 매개변수를 측정함으로써 수행되는 성능시험에 대한 관심과 중요성이 높아지고 있는 추세이다.

이러한 표준화 기구들은 ATM 성능 시험의 효율성을 위해, 유사한 특성을 지닌 응용 서비스들을 분류해서 정의해 놓고 있는데, 대표적으로 ATM 포럼의 TM 4.0에서는 이를 5가지 서비스 카테고리로 규정하고 있다. 즉, 음성이나 비디오 등 실시간 서비스에 이용되는

† 비 회 원 : 부경대학교 정보통신공학과
ahnss@woongbi.pknu.ac.kr
yhc007@pel.pknu.ac.kr
†† 비 회 원 : 한국전파기지국관리(주) 기술연구소 연구소장
hsh4@ppp.kornet21.net
††† 정 회 원 : 안동대학교 정보통신공학과 교수
lccjw@anu.andong.ac.kr
†††† 종신회원 : 부경대학교 정보통신공학과 교수
kimsu@dolphin.pknu.ac.kr
논문접수 : 1999년 1월 15일
심사완료 : 1999년 11월 11일

CBR(Constant Bit Rate) 및 rt-VBR(real time-Variable Bit Rate), 비실시간 서비스에 응용되는 nrt-VBR(non real time-Variable Bit Rate), UBR(Unspecified Bit Rate), ABR(Available Bit Rate)로 분류하고, 각각 망에 대한 QoS 요구사항을 정의한다[1].

ATM 망의 성능 시험은 망의 관점에서 셀 레벨의 QoS를 이용하는 방법과 사용자 관점에서 프레임 레벨의 메트릭을 이용하는 방법이 있으며, 현재 이에 대한 표준화가 진행중이다. 먼저, 셀 레벨의 성능 시험은 OAM(Operation and Maintenance) 셀 또는 시험 셀을 이용하여 ATM 계층의 QoS 매개변수를 측정함으로써 수행되는데, 이는 다시 시험을 위한 특정 경로의 사용 여부에 따라 in-service 및 out-of-service 방법으로 나누어 수행될 수 있다[2][3]. 반면, 프레임 레벨의 성능 시험은 ATM 계층 이상, 즉 AAL 및 응용 레벨에서 수행되는데, 성능 측정의 결과가 사용자에게 더 잘 인식될 수 있도록 프레임 레벨의 메트릭을 측정함으로써 수행된다[4][5].

현재, 다양한 멀티미디어 응용이 인터넷 기반으로 등장하고 있으나 망 관점에서 인터넷 망은 데이타그램 개념으로 라우터를 거쳐 데이타를 전송하므로 궁극적으로 멀티미디어 사용자가 요구하는 성능 요구사항을 만족시킬 수 없다. 그러나, 통신망을 구성하는 요소들간의 모든 경로가 ATM 경로가 될 때까지는 B-ISDN 망의 성패 여부는 기존에 사용중인 데이타 통신망, 특히 전세계적으로 가장 많은 사용자를 확보하고 있는 인터넷 망을 어떻게 ATM 기술에 적용할 것인가에 따라 좌우될 것으로 예상된다. 즉, 대부분의 기존 응용은 적절한 데이타 포맷 문제 해결과 함께 서비스 에뮬레이션 또는 상호 연동을 통해 여전히 ATM을 통해 동작할 수 있음을 의미한다.

이에 따라, 본 논문에서는 프레임 레벨의 성능 시험을 위한 대상으로 최근 그 중요성이 부각되고 있는 TCP/IP over UBR을 채택하고, ATM 망 및 패킷망으로 나누어 시뮬레이션을 수행한다. 이를 위해 본 논문의 2장에서는 TCP 혼잡제어 기법 및 ATM과의 문제점을 분석한다. 3장에서는 시뮬레이션을 위한 네트워크 구성 및 특성과 시뮬레이터인 x-kernel에 대해 기술한다. 이를 바탕으로, 4장에서는 TCP 성능 인자 변화에 따른 시뮬레이션 결과 및 성능 인자간의 상호동작을 분석한다. 그리고, 5장에서는 TCP/IP over UBR의 성능 개선 방안을 제안하며, 끝으로 6장에서 결론 및 향후 연구사항을 기술한다.

2. TCP 혼잡제어 기법 및 ATM간 상호 연관성

ATM을 통한 TCP/IP 성능은 각 프로토콜간 상호 연관성과 밀접한 관계가 있다. 특히, TCP/IP 및 ATM이 수행하는 혼잡제어와 관련된 메커니즘들의 상호 행위는 실제 TCP/IP over ATM의 전체적인 성능에 큰 영향을 미치게 된다. 본 장에서는 이러한 관계를 분석하기 위해 TCP 혼잡제어 기법 및 ATM간의 상호 연관성에 대해 기술한다.

2.1 TCP 혼잡제어 기법

TCP는 RFC 793에 근거하여 (그림 1)과 같이 발전되었다. 최초로 널리 사용된 4.2BSD에서 4.4BSD-Lite까지 발전된 TCP 구현의 목적은 전송률 감소를 최소화시키고 자원의 낭비를 줄이는 것이다[8][9][10].

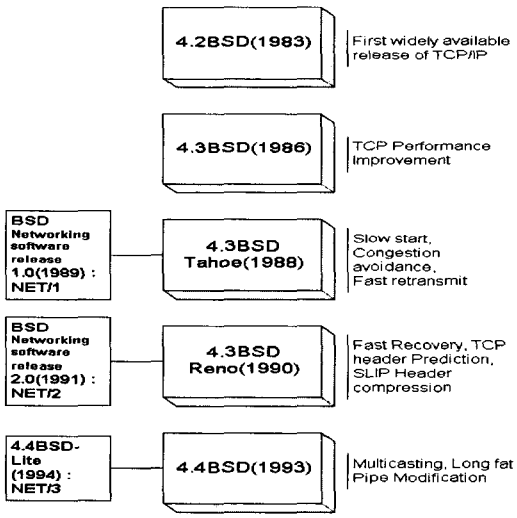


그림 1 TCP 구현의 발전단계

실험을 위해서 사용된 TCP Vegas 의 경우 TCP Reno 를 수정한 것이다. Reno 의 경우는 (그림 1)과 같이 Tahoe 에서 Fast Recovery 알고리즘이 추가 되었다.

TCP(Transmission Control Protocol)는 종단간 신뢰성을 보장하기 위해 설계된 트랜스포트 프로토콜로, 이를 수행하기 위해 다음과 같은 혼잡제어 기법들이 수행되어야 한다.

(1) Additive Increase / Multiplicative Decrease

TCP 제어를 위한 흐름은 다음과 같이 이루어 지고 있다. 사용 가능한 네트워크 자원을 조절하기 위해서 혼잡 윈도우(Congestion windows)와 같이 네트워크 연결

때 마다 가변적인 값을 갖는 변수를 정한다. 이것은 전송량의 조절을 위해서 사용되고, 다음과 같은 방법으로 최대 윈도우(Max Windows) 크기를 설정한다.

Max Win = MIN(Congestion Window , Advertised Windows)

즉, 최대 윈도우(Max Win) 크기는 혼잡 윈도우(Congestion Window)와 수신자쪽에서 보내온 윈도우(Advertised Windows) 중 작은 것을 최대 윈도우 크기로 선택한다.

혼잡 윈도우는 매 RTT(Round Trip Time) 때마다 1 세그먼트씩 증가하고 타임 아웃이 발생 되면 1/2로 급격히 감소한다. 이와 같은 증가와 감소로 인해서 전송량은 톱날 형태(Sawtooth behavior)의 모습으로 전송이 이루어 진다.

(2) Slow - start and congestion avoidance

앞서 소개된 Additive Increase / Multiplicative Decrease 방법은 연결 초기과정에서 많은 지연을 초래하게 된다. Slow - Start 및 congestion avoidance 혼잡 제어는 슬로우 스타트 및 혼잡 기피 기간으로 구성되며, 슬로우 스타트 한계치를 나타내는 변수 SSTHRESH(slow start threshold)는 그 기간을 구별하기 위해 소스에서 유지된다.

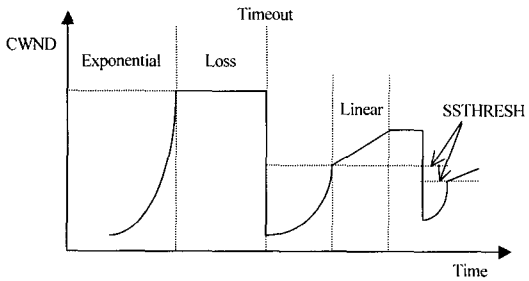


그림 2 Slow - start and congestion avoidance behavior

Slow-start and congestion avoidance 메커니즘의 행위는 지수적으로 CWND가 증가하는 슬로우 스타트 기간에서는 매 RTT마다 CWND가 2배씩 증가하며, 이는 CWND가 SSTHRESH에 도달할 때까지 계속된다. CWND가 선형적으로 증가하는 혼잡 기피 기간에서 소스는 송신된 패킷에 대한 Ack(acknowledgement) 수령시마다 CWND를 1/CWND 만큼 증가시킨다.

한편, 혼잡으로 인한 패킷 손실 발생시 TCP는 재전송 타임 아웃이 발생될 때까지 기다리게 되므로, 그 만큼의 시간이 낭비 된다. 재전송 타임 아웃이 일어나면, 소스는 CWND의 반으로 SSTHRESH 값을 세팅한 후 슬로우 스타트 기간으로 들어간다.

결과적으로, 링크는 낭비된 시간만큼 휴지 상태가 될 것이고, 소스는 손실된 패킷 이후 모든 패킷을 재전송하게 되는데, 이는 TCP/IP over ATM의 성능 저하에 가장 핵심적인 요인이 된다.

(3) Fast Retransmit and Fast Recovery

TCP는 timeout으로 인해서 전송에서 상실된 패킷을 기다리기 위해서 쓸데 없는 idle 시간을 보내게 된다. 이것을 피하기 위해서 사용되는 것이 Fast Retransmit 알고리즘이다. 예를 들어 (그림 3)에서 세번째 Packet이 전송에 실패하고 수신자 측에서는 네번째 Packet 을 받게 된다. 이런 경우 수신자 측에서 두번째 Packet 에 대한 ACK 값을 보내게 되고 이것을 'duplicated ACK' 라고 한다. 이 duplicated ACK를 n개(일반적으로 3) 이상 받게 되면 RTO에 이르지 않더라도 재전송을 하게 된다.

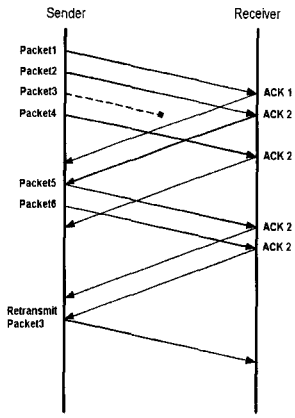


그림 3 Retransmit Packet

Fast Recovery 알고리즘은 Fast retransmit 알고리즘에서 slow - start를 제외시킨 것이다. Slow-start를 네트워크 연결 초기 과정에서만 사용하는 것으로 timeout 발생시, Congestion Windows 크기를 1로 만들지 않고 1/2로 감소된 크기의 Congestion Window 크기로부터 1씩 증가 시키게 된다.

본 논문에서 시뮬레이션을 위해 사용된 TCP 메커니즘은 Larry Peterson에 의해 제안된 Reno 버전을 향상시킨 Vegas 버전으로 이는 상기 TCP 메커니즘 중, slow-start and congestion avoidance 및 fast retransmit and fast recovery와 관계된다. 후자는 optional한 부분으로 실제 시뮬레이션에서는 전자를 중

심으로 성능 시험을 수행하였다.

2.2 TCP/IP over ATM 문제점 분석

ATM과 패킷 네트워크 사이의 중요한 차이점은 ATM 스위치의 폐기 단위가 셀인 반면, 종단 호스트의 재전송 단위는 세그먼트 단위라는 것이다. TCP/IP가 ATM 네트워크 상에서 동작할 때 TCP 세그먼트는 AAL5 계층을 통해 53byte의 ATM 셀로 분해된다. 만약 ATM 망에서 혼잡상태가 발생하면 셀을 버리게 된다. 결과적으로, TCP 세그먼트를 구성하는 셀 중 하나의 셀이 버려져도 전체 세그먼트가 다시 전송되어져야 한다. 반면, 패킷 네트워크에서는 재전송 단위와 폐기 단위가 동일하다.

특히, TCP/IP over UBR 구현시 UBR 자체는 flow control을 제공하지 않고 TCP의 혼잡제어 메커니즘에 의존하는데, 이러한 TCP 메커니즘은 셀 단위가 아닌 세그먼트에 기반해서 재전송이 이루어지므로 이들간의 불일치는 TCP/IP over UBR의 전체적인 성능에 큰 영향을 미치게 된다. 또한, 버려진 세그먼트의 일부분으로부터 남겨진 셀은 단지 목적지에 도착해서만 폐기되어지기 때문에 전체 네트워크 자원 및 연결된 전송률을 낭비시킨다. 이러한 문제점에서 오는 성능 저하는 본 논문의 시뮬레이션 결과에서도 확인이 되었으며, TCP 레벨에서 성능을 개선하기 위한 방안으로 과다한 세그먼트 손실을 보다 효율적으로 방지하는 향상된 TCP 메커니즘을 제안한다.

3. 시뮬레이션 환경

본 장에서는 TCP/IP over UBR에 대한 성능을 평가하기 위해 사용된 시뮬레이션 환경에 대해 기술한다. 시뮬레이션은 실제 네트워크 상황을 반영하면서 ATM 및 packet 망에 대해 수행되었는데 이를 위한 시험구성과 사용된 시뮬레이터인 x-kernel 및 네트워크 특성에 대해 서술한다.

3.1 Network Configuration

네트워크 구성은 두 ATM 스위치가 중심으로 스타형으로 구성되어져 있고, 이중 구조로 되어져 있어 어느 한쪽이 다운되어도 네트워크는 계속 운영될 수 있도록 설계되어져 있다. 각 ATM 스위칭 장비간은 155 Mbps(OC-3)로 연결되어져 있고 LAN 스위치들은 Ethernet 100Mbps로 연결되어져 있다.

이를 기반으로 해서 구성된 시험 구성 형태는 (그림 4-1) 및 (그림 5-1)과 같다. (그림 4-1)은 ATM 망에서의 TCP/IP 데이타 전송을 위해 사용한 것으로 주로, 멀티미디어 서비스를 목적으로 구성되는 형태이다. (그림 4-2)는 시험 구성 1에 대한 프로토콜 스택을 나타낸 것으로 TCP 에서 세그먼트 단위로부터 ATM 계층의 Cell 까지 전송경로를 표현한 것이다.

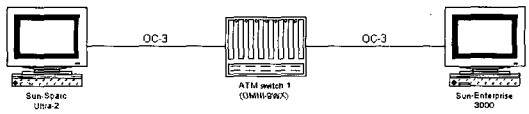


그림 4-1 시험 구성 1

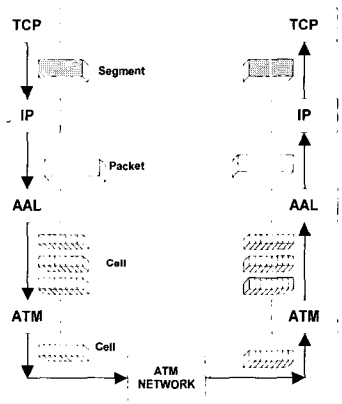


그림 4-2 시험 구성 1에 대한 프로토콜 스택

(그림 5-1)은 Packet 망에서의 TCP/IP 데이타 전송을 위한 구성으로, 허브(Hub)를 통해서 호스트가 연결되어 있다. (그림 5-2) 은 시험 구성 2에 대한 프로토콜 스택을 나타낸 것이다.

상기 시험 구성에서, 종단 호스트의 세부적인 하드웨어 사양 및 ATM 스위치 사양은 각각 <표 1>, <표 2>과 같다. 사용 시스템은 SunOS 5.5.1을 운영체제 기반의 워크스테이션으로 서버와 클라이언트로 구분하여 시험하였다. 그리고 ATM과 Packet 네트웍에서 각기

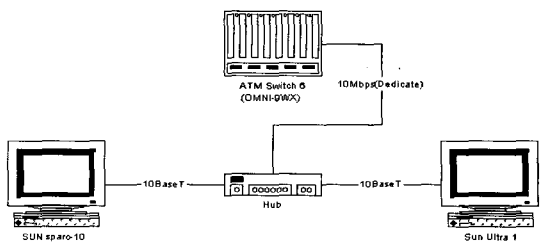


그림 5-1 시험 구성 2

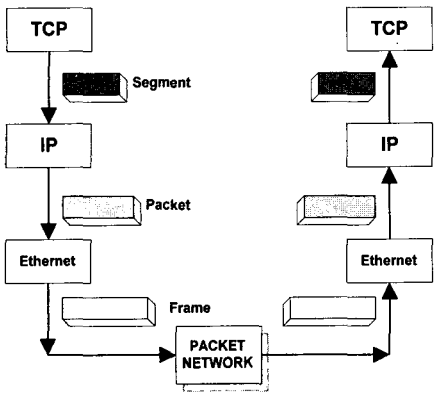


그림 5-2 시험 구성 2에 대한 프로토콜 스택

다른 시스템을 사용하였다. 또한 ATM 네트웍 시험에 사용된 스위칭 장비는 Xylan사의 OMNI-9WX로 ATM forum UNI 3.0/3.1과 ISO Q.2931, TM 4.0, PNNI 1.0의 표준을 모두 만족하고 있다.

표 1 End-Host Specification

Section	Testing Configuration 1 (ATM Network)		Testing Configuration 2 (Packet Network)	
	Server	Client	Server	Client
Model	SUN Ultra 2	SUN Entprise 3000	SUN sparc-10	SUN Ultra 1
Platform	Ultra-2	Ultra Enterprise	SPARC station-10	Ultra-1
Processor Type	Sparc	Sparc	Sparc	Sparc
CPU	2	2	1	1
Main Memory	256MB	512MB	256MB	128MB
OS	SunOS 5.5.1			

3.2 Simulator : x-kernel

시뮬레이션에 사용된 x-kernel은 네트워크 프로토콜 구현을 위해 Arizona Natl. Univ.의 Norm Hutchinson과 Larry Peterson에 의해 개발된 객체 지향 구조의 툴이다. 수년간 컴퓨터 네트워크에서 End-to-End에 관한 연구가 이를 통해서 이루어 지고 있다. X-kernel 은 크게 두 부분으로 실제 네트웍 기반에서 프로토콜의 성능을 분석할 수 있는 x-kernel 모드와 사용자가 정의한 네트웍 환경에서 성능을 분석할 수 있는 x-sim 모드로 나눌 수 있는데, 이번 시험에서는 실제 네트웍 환경에서 각종 프로토콜을 분석할 수 있는 x-kernel 모드에서 시험하였다. 이 모드는 기존 운영체제에서 사용하는 커널 부분의 네트웍 모듈을 이용하지 않고, 객체 기반의 x-kernel 구조 자체에서 지원하는 프로토콜을 이용한다 [6][7][8].

표 2 ATM Switch(OMNI-9WX)

Section		Description
Support		ETHERNET, TOKEN RING, FAST ETHERNET
Fabric	Rate	13.2Gbps
	Type	Distributed matrix
Buffer Management	Type	DIBOC(Distributed input buffering with output control)
	Max cell buffers	2,097,152 / switch
Cell	Discard	RED(Random Early Detect)
Traffic Management		Congestion-based GCRAs
CSM-155 Module	Public Standards	ATM forum UNI 3.0/3.1; ISO Q.2931; TM4.0; PNNI 1.0
	Data Rate	155 Mbps
	# Of Virtual circuit	4K per port
	Cell buffer size	8192 cells per port

X-kernel이 지원하는 객체 클래스는 크게 프로토콜과 세션으로 구분되는데, 프로토콜 객체는 TCP, IP와 같은 프로토콜을 표현하고 세션 객체는 메시지 해석과 채널에 관련된 모든 상태를 관리하며 채널 open/close 시 동적으로 생성되고 삭제된다. 즉, 프로토콜 객체는 채널을 열기 위한 동작을 제공하고, 그 결과로 세션 객체들이 생성되며 세션 객체는 메시지를 송수신하기 위한 동작을 제공한다. 다음 (그림 6)은 X-kernel 구성의 예를 나타낸 것이다. 그림에서 사각형은 프로토콜 객체를, 원형은 세션 객체를, 굵은 실선은 메시지를 나타낸다[9][10][11].

3.3 네트워크 특성

Testbed로 사용된 ATM 망의 특성을 살펴보면 다음과 같다. ATM 망의 연결 대역폭은 155.52Mbps이고 Peak Cell Rate는 초당 353208 cell이다. 그리고 서비스

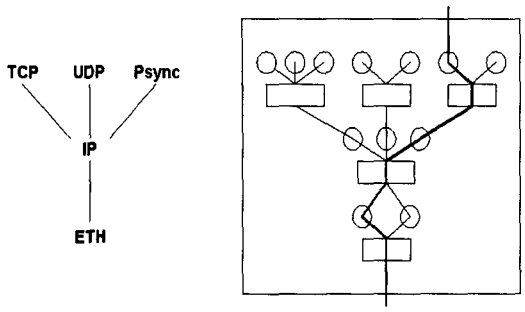


그림 6 Example (x-kernel configuration)

카테고리는 UBR을 사용하고 스위치에 구현된 버퍼 관리 정책은 RED(Random Early Detection)이며 라운드 로빈 형태로 PPD(Partial Packet Discard)/EPD(Early Packet Discard)를 사용한다. PPD의 경우 스위치 버퍼로부터 셀 손실이 발생될 경우 이후의 상위 계층의 프로토콜 데이타를 모두 버리게 된다. 그리고 EPD의 경우 스위치 버퍼 큐가 일정상태에 도달하게 되면 상위 계층의 데이타 모두를 버리게 된다. 일반적으로, TCP/IP over ATM의 성능에 영향을 미치는 요인으로 다음 <표 3>과 같다[12][13].

표 3 TCP/IP over ATM에 영향을 주는 매개변수

구분	매개변수
General	소스수 UBR 트래픽 형태(Infinite vs Bursty, Uni vs Bidirectional) Background Traffic 상위계층 형태(TCP vs non-TCP) 버퍼 형태(Single vs Multiple FIFO) 링크 지연, 링크용량, 버퍼크기
TCP/IP	TCP MSS(Maximum Segment Size) TCP Timer granularity Delay Ack timer(Enable vs Disable) Processing Delay, Delay Variation(zero vs non-zero) IP MTU Size
UBR	UBR 관련 트래픽 매개변수(PCR) UBR 스위치에서 구현된 트래픽 정책(Tail drop, FBA, EPD 등) 선택된 트래픽 정책의 매개변수

4. 시뮬레이션 결과 및 성능 평가

UBR의 경우 ABR이나 CBR과는 달리 셀 상실이나 지연에 대한 관리정책이 없고 단지 그 한계 용량을 초과하면 셀을 폐기한다. 그렇기 때문에 PCR이나 트래픽 정책이 성능에 큰 영향을 미치게 된다. 이를 바탕으로 시뮬레이션에 적용된 핵심 매개변수는 <표 4>과 같다. 시뮬레이션은 주로 TCP/IP 매개변수의 Default 값을 중심으로 변화시켜가며 수행하였고, 또한 각 매개변수에 대해서 Sending Size를 변화시켜 가며 각각 100번씩 수행하였다.

표 4 시뮬레이션에 적용된 핵심 매개변수

구분	매개변수명	Default 값
General	UBR 트래픽 형태	Bursty, Unidirectional
	상위계층 형태	TCP
	버퍼 형태	Input buffering
TCP/IP	TCP MSS	512 (byte)
	IP MTU	1500 (byte)
	TCP Buffer Size	4096 (byte)
	Timer granularity	500 (ms)
UBR	PCR(Peak Cell Rate)	353,208 cell/s
	UBR 스위치 policy	RED(Random Early Detection)

시뮬레이션은 주로 TCP 측면에서 TCP 버퍼 크기, MSS, Timer Granularity를 핵심 매개변수로 적용하여 수행하였고 사용된 TCP 버전은 Vegas이다. 하지만 Timer Granularity 부분에서는 Vegas에서 사용하는 동적인 윈도우 크기 할당은 시행하지 않았다. 그 이유는 정적인 Timer가 TCP 성능에 미치는 영향을 분석하기 위해서이다. 본장에서는 시뮬레이션 결과 및 이에 대한 결과를 분석함으로써 성능을 평가한다.

4.1 시뮬레이션 결과

(1) TCP 버퍼 크기가 미치는 영향

실제 시험을 수행하기에 앞서 시험의 체계성 및 효율성을 높이기 위해 각 시험 항목에 대한 시험 항목(TC : Test Case)이 먼저 작성되어야 한다. 일반적으로, 시험 항목의 구성은 TC Name 시험 목적 rationale 등 시험의 개괄적 사항을 포함한 부분, 시험 장치 설정 절차 및 시험 결과를 포함한 시험 기술 부분, 주로 시험에 관한 주석 사항을 기술한 Note 부분으로 구성되어 있

다. 예를 들어, (그림 7)는 TC Name이 X0001로 TCP 버퍼 크기에 따라 TCP/IP over UBR에서의 전체적인 성능에 얼마나 영향을 미치는가를 분석하기 위한 시험 항목이며, 이에 대한 이론적인 근거를 제공하는 부분이 Rationale 부분이다.

TC Name	X0001										
TC Id	TCP IP/UBR/TC X0001										
Reference	x-kernel 관련 문서										
Test Purpose	TCP/IP over UBR의 전체적인 성능에 TCP 버퍼 크기가 미치는 영향을 분석하기 위해 시험한다.										
Rationale	TCP 버퍼 크기는 TCP 연결에서 데이터 양을 결정하는 파라미터로 ATM 망에서 필요한 저장 능력과 관련되는 핵심적인 변수이다.										
Test Description											
Test Procedure	P1	TCP 버퍼 사이즈를 변화시키기 위해 다음을 수행한다. ① 다음 위치로 이동 : cd /usr/x-kernel/protocol/test ②tcptest.c 에서 다음을 수정 : line 27 [#define BUFFER_SIZE (1*1024)] ③ 다음 위치로 이동 : cd /usr/x-kernel/protocol/tcp ④tcp.h 에서 다음을 수정 : line 23 [#define TCP_BUFFER_SPACE (1*1024)]									
	P2	라이브러리 재생성을 위해 다음을 수행한다. ① 다음 위치로 이동 : cd /usr/x-kernel/user-level/build/solaris ② 다음 명령어 수행 : make system									
	P3	시험을 위해 x-kernel 을 실행한다. ① 다음 위치로 이동 : cd /usr/x-kernel/user-level/build/solaris ② 다음 명령어 순차적으로 수행 : make compnec → make depend → make									
	P4	각 버퍼크기에 따라 P1~P3 의 과정을 반복한다.									
Test Result	다음과 같은 형태로 시험 결과를 기록한다.										
Result Output	TCP Buffer Size	Sending data	Test Number / measured time(msec)									
			1	2	3	4	5	6	7	8	9	10
	1K	1k										
		16k										
		100k										
		1M										
		10M										
	.	.										
	.	.										
	.	.										
	32K	1k										
		16k										
		100k										
		1M										
		10M										
Notes	TCP 버퍼 크기(default : 4096byte), 시험 횟수, sending data 크기는 사용자 목적에 맞게 임의대로 설정 가능하다.										

그림 7 시험 항목(TCP 버퍼 크기)

시험 기술 부분에서, 해당 시험을 수행하기 위한 절차를 나타내는 Test Procedure에서는 먼저, TCP 버퍼 사이즈를 변화시키기 위해 x-kernel이 제공하는 소스를 분석하여 관련 매개변수, 즉 검은색 부분을 다양하게 변화시킨 후 라이브러리를 재생성하고 x-kernel을 실행시킨다. 시험 결과는 각 버퍼 크기 및 sending size에 대해, 시험 횟수를 100번씩하여 각각에 대한 시간을 측정한 후 결과를 Result Output에 나타난 표에 기록함으로써 버퍼 크기에 대한 시험이 완료된다.

(그림 8)은 상기 시험 항목에 따라 ATM 및 packet 망에서 버퍼의 크기를 1K에서 32K까지 변화시켜가며 sending size 16Kbyte에 대한 전송시간을 측정한 결과이다. 그림에서 보는 바와 같이 ATM 망의 경우 4K 이상, Packet 망의 경우 8K 이후로는 차이가 거의 나타나지 않았고 다른 sending size에 대해서도 유사한 결과가 나타났다.

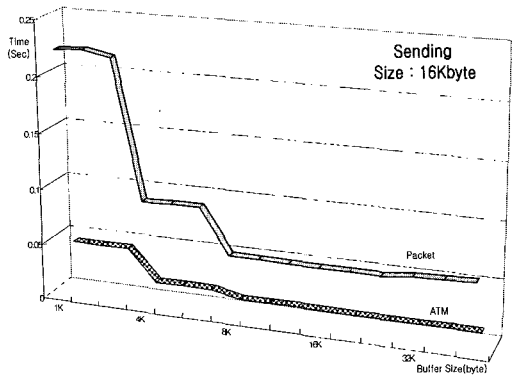


그림 8 Effect of TCP buffer Size

(2) TCP의 MSS가 미치는 영향

시험 항목에 따라 ATM 및 패킷망에 대해 TCP MSS의 크기를 각각 4K, 8K, 16K로 변화시켜가며 각 sending size에 대한 전송 시간을 측정한 결과, MSS의 변화에 따라 성능에 있어 큰 차이는 없는 것으로 나타났다.

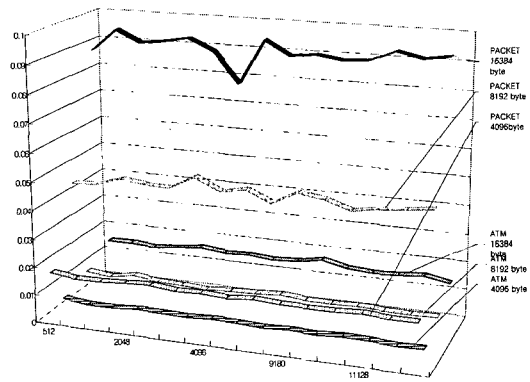


그림 9 Effect of TCP MSS(Maximum Segment Size)

(3) TCP Timer Granularity 가 미치는 영향

TCP timer granularity 값은 패킷 loss 결정과 관련되는 핵심적인 매개변수로, 이 값이 작으면 재전송이 많이 일어나므로 network load의 증가를 유도하고, 크면 상대적으로 긴 timeout 기간으로 인한 network 대역폭

및 자원을 낭비하게 되므로 network 상황에 따라 적절한 TCP timer granularity 값의 설정이 중요하다. 시험 항목에 따라 ATM망에 대해 TCP timer granularity 값을 100ms에서 1000ms까지 변화 시켜가며, sending size 1Mbyte에 대한 전송 시간을 측정한 결과, (그림 10)와 같이 timer granularity 값을 낮게 설정 했을 때, 성능이 우수하게 나타났다.

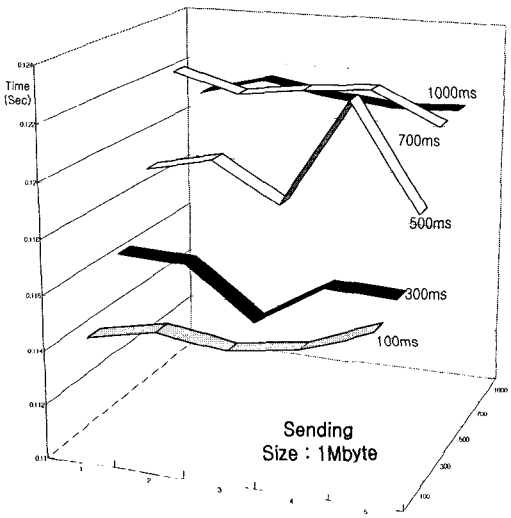


그림 10 Effect of TCP timer granularity

4.2 성능 평가

상기 시뮬레이션은 TCP의 각 매개변수 값의 변화에 따라 TCP over UBR의 전체적인 성능에 어떠한 영향을 미치는가를 살펴보기 위해 수행되었다.

이 중, TCP 버퍼 크기는 TCP 연결에서 데이타 양을 결정하는 매개변수로 ATM 망에서 필요한 저장 능력과 관련되는데, ATM 및 패킷망에 대해 각각 4K 및 8K이상에서 높은 성능을 보였다.

또한, TCP의 최대 세그먼트 크기가 크면 패킷 loss 시 재전손량이 많아지고 작으면 오버헤드를 증가시켜 전송 효율이 낮아지므로 적절한 MSS 값의 설정이 중요하다. 그러나, 본 논문의 시험 구성에서는 MSS의 변화에 따라 큰 성능 차이를 나타나지 않았다. 이와 같은 결과는 Mogul[1993]의 연구 결과에서 크기가 큰 패킷을 전송하는 것이 상대적으로 많은 수의 작은 크기의 패킷을 전송하는 것에 비해서 패킷의 오버헤드라든지, 라우팅 시간, 호스트에서의 프로토콜 처리 측면에서 효율적이라고 했다. 하지만 현재의 실험결과는 이와는 맞지 않는 측면이 있는데 그 이유는 Bellovin[1993]의 연구결과와 같이 ATM 스위치나 라우터와 같이 store-and-forward 기법을 사용하는 장치들이 중간에 처리 비용이 세그먼트 크기를 크게 했을 때와 비교해서 상대적으로 큰 부분을 차지하고 있으면 그 효율성이 더 떨어진다는 이유에서 (그림 9)와 같은 결과를 나타내고 있다고 볼 수 있다. 더욱이 ATM망에서는 TCP MSS가 패킷망에 비해서 데이타 전송에 큰 작용은 하지 못하는 것으로 나타나고 있다. 그 이유는 ATM Cell로 변환되는 과정에 있어서 비효율적인 요인이 있기 때문이다.

예를 들어 512Byte를 전송할 경우, ATM 계층에서는 받는 총 데이타 크기는 다음과 같다.

568byte = 512byte (data)
+ 20 byte(TCP header) + 20 byte(IP header)
+ 8byte(LLC header) + 8byte(AAL5 trailer)

568Byte의 경우 총 12 ATM Cells이 필요하고 ATM 계층에서는 636Byte가 전송되게 된다. 그러므로 155Mpbs의 경우 실제 사용 용량은 125Mbps로 80% 정도의 효율성을 나타내고 있다. 그러므로 TCP MSS 크다고 해서 효율성의 증대를 의미하지 않는다.

특히, TCP timer granularity와 관련하여 본 논문에서는 이 값을 정적으로 설정해서 시험을 수행한 결과 본 논문과 같이 안정적인 시험 구성에서는 이 값을 낮게 설정했을 때 성능이 우수하게 나타났었다. 그러나 혼잡 상태가 보다 복잡하게 발생하는 네트워크의 경우, 이에 대응하기 위해선 혼잡 상태의 사전 예측을 반영한 동적인 TCP 메카니즘이 필요하다. 뿐만 아니라 Linear한 전송 데이타 크기의 증가로는 순간적인 변화가 심한 네트웍 상황에서 큰 효율성을 나타낼 수 없음을 알수 있었다. 그러므로 본 연구에서는 증감 구간을 좀 더 세분화 시켜 Linear한 형태와 Exponential 한 형태 모두를 지원할 수 있는 개선된 알고리즘을 다음과 같이 제안한다.

5. TCP/IP over UBR의 성능 개선 방안

UBR 서비스 클래스는 흐름 제어를 수행하지 않으므로, TCP가 이를 수행한다. TCP 패킷은 ATM 계층에서 셀들로 구성되므로 셀의 손실이 발생하면 결과적으로 패킷의 손실이 발생하고 TCP는 재전송 메커니즘을 사용하여 이를 복구한다. TCP over UBR에선 제한된 버퍼 용량에서 낮은 성능을 증진시키는 것이 가장 큰 이슈인데, (그림 11)과 같이 UBR 스위치 정책 및 종단 시스템 정책에 의해 개선될 수 있다[14][15].

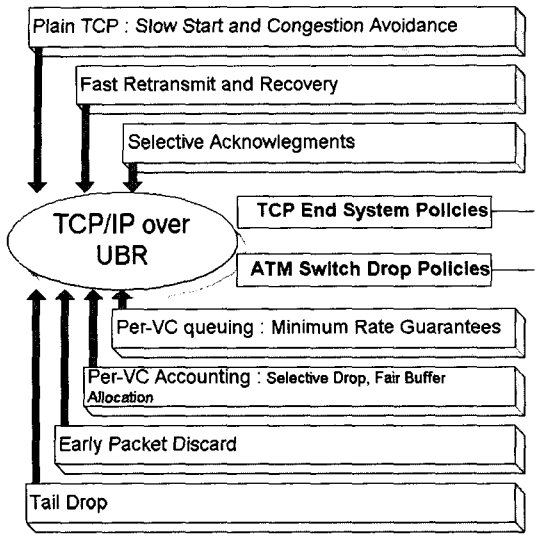


그림 11 TCP/IP over UBR Policies

5.1 UBR 스위치 정책

종단 시스템에 대한 정책은 2장에서 언급한 바와 같고, 여기서는 UBR 스위치에 대해 기술한다.

먼저, 폐기 정책은 셀들이 폐기되는 시기와 관련된 기능으로 EPD(Early Packet Discard), PPD(Partial Packet Discard), LPD(Lazy Packet Discard) 기법이 대표적인 예이다. 그러나 폐기 정책으로 전송률이 증가되지만 높은 불공정성이 나타난다. 특히 LPD의 경우 기존의 EPD나 PPD와는 달리 혼잡 상태 발생시 셀을 폐기하지 않고 TCP Packet 의 Sequence 를 비교해서 재전송이 이루어져 중복된 셀 전송을 피할 수 있도록 제안된 알고리즘이다. [24]

버퍼 할당 기법은 버퍼를 유용하게 쓸 수 있는 방법과 관련된 기능으로 예를 들어 FBA(Fair Buffer Allocation) 기법은 할당량보다 초과한 연결의 프레임에 대해 선택적으로 폐기함으로써 공정성을 높인다.

마지막으로, 스케쥴링 정책은 CBR, VBR, ABR, UBR 등 서로 다른 클래스간 대역폭의 결정과 관련되는 클래스 기반 기법과 동일한 클래스에서 VC간 대역폭 할당과 관련되는 VC 기반 정책이 있는다. 예를 들어, 전자는 CBR이나 VBR과 같은 우선 순위가 높은 클래스가 대역폭의 사용 후에 ABR 또는 UBR 연결이 얼마나 많은 대역폭을 할당 받을 수 있는가와 관련되며, 후자는 경쟁하는 UBR 연결간 대역폭을 어떻게 분배할 것인가와 관련된다 할 수 있다.

이러한 스위치 정책들은 전체적인 TCP/IP over UBR의 성능에 영향을 미친다. 하지만 이 논문에서는 스위치 정책에 대해서 고려하지 않고 TCP의 Congestion Avoidance 개선을 통해 성능향상을 제안한다.

5.2 TCP 혼잡제어 알고리즘 개선

기존의 TCP 혼잡제어 알고리즘을 개선 시키기 위한 연구는 (그림12)와 같다.

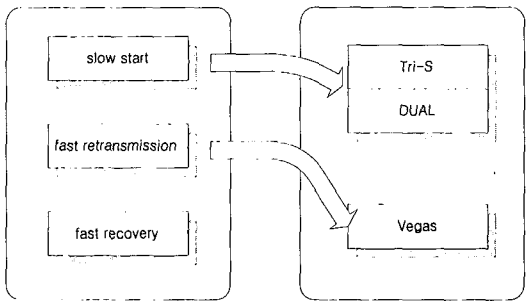


그림 12 Congestion Control Scheme for TCP/IP Network

Wang에 의해서 제안된 Tri-S(Slow Start Search)와 DUAL 알고리즘은 기존 slow start 알고리즘을 개선시켰고, Barkmo에 의해서 제안된 Vegas는 retransmission 알고리즘을 개선시켰다. Tri-s의 경우 혼잡상태를 전송량의 변화로 파악하여 혼잡 상태에서 최적의 전송량을 설정할 수 있도록 하였고, DUAL은 slow start시 RTT(Round Trip Time)을 이용하여 혼잡 상태를 파악하여 slow start때 전송량 sawtooth 형태로 진동하지 않도록 하였다.

반면 Vegas는 사전에 예측된 전송량과 실제 네트웍의 전송량의 차이를 이용해서 전송량을 증감시켜 네트웍이 혼잡 상태가 되지 않도록 하였다.

그러나, Vegas는 <표5>에 나타난 것처럼 증감 구간에 대해 단지 Linear한 형태만 고려하고 있다. 4장의

표 5 Vegas 와 Accelerated Vegas 비교

Section	Vegas	Accelerated Vegas
Boundary	$\alpha<\beta$	$\chi_1<\alpha<\beta<\chi_2$
Increase	Diff < w+α → *increase cwnd (linearly)*	*(w+χ1<Diff<w+α) → increase cwnd (linearly)* *(w+χ1> Diff) → increase cwnd (Exponential)*
Decrease	*Diff > w+β → decrease cwnd (linearly)*	*(w+β<Diff<w+χ2) → decrease cwnd (linearly)* *(Diff>w+χ2) → decrease cwnd (Exponential)*
Unchanged	*w+α<Diff< w+β*	

시뮬레이션 결과에서 나타났듯이 ATM망은 대역폭의 순간적인 변화가 크다. 그래서 Linear 및 Exponential 한 증감을 함께 제공할 수 있는 Accelerated Vegas Algorithm을 표6과 같이 제안한다.

<표5>는 Vegas와 Accelerated Vegas의 증감 구간에 대한 차이를 나타낸 것이다. Vegas는 예측 전송량과 실제 전송량 값의 차이를 Diff 변수로 나타내고, 이를 가상 Queue로 설정한다. Vegas에서 사용되는 변수는 다음과 같다.

C : 네트워 용량
BaseRTT : 최소 RTT(Round Trip Time)
CWND : 혼잡 윈도우 크기
w : CWND = C × BaseRTT

표 6 Accelerated Vegas: 개선된 혼잡회피 알고리즘

```
Procedure Congestion Avoidance
  Begin
      Base_RTT = MIN(All measured RTT);
      Expected = CWND / BaseRTT;
      Actual =  CWND / RTT;
      Diff = (Expexted – Actual) x BaseRTT;
      If (Current Status != SlowStart) then
        Begin                                          *
          set_threshold (BaseRTT)
          χ1=α/2; χ2=β/2;
          If (w+χ1 < Diff <w+ α ) Linear_Increase()
          else If (w+χ1 > Diff) Exponential_Increase();
          If (w+β < Diff < w+χ2) Linear_Decrease()
          else If (w+χ2 > Diff ) Exponential_Decrease();
        End
  End
```

* 기존 Vegas 의 CAM(Congestion Avoidance Mechanism)의 CWND 증감판단 부분을 수정하여 동적인 형태의 경계선 설정이 가능하도록 하여 linear 한 부분과 exponential 부분으로 나누어 빠른 안정상태회복 가능.

증감 구간에 대한 내용은 표5와 같이 Vegas의 경우 Diff<w+α 이면 CWND는 linear 한 증가하고, Diff<w+β 인 경우 CWND는 linear 한 감소를 하게 된다. 반면 w+α< Diff < w+β 인 경우 CWND는 변화가 없다. 이때 경계값 α, β는 2, 4 혹은 1, 3으로 제한해서 사용한다.

4.2절 의 실험결과 분석에서 알 수 있듯이 ATM 환경에서 TCP 전송률은 만족할 수 없는 결과를 나타내었다. 이는 기존 Vegas가 ATM망에서 대역폭의 변화에 효율적으로 CWND와 경계값 α, β를 대응시키지 못하기 때문이다.

새로 제안하는 Vegas 의 경우 BaseRTT 를 고려해서 유동적인 α, β 경계값 설정이 가능하도록 하고, 증감 구간을 세분화하기 위해서 $\chi 1 = \alpha/2$, $\chi 2 = \beta/2$ 로 설정할 것을 제안한다. 이렇게 함으로써 기존 Vegas의 문제점이었던 대역폭과 관계없는 경계값 설정으로 인한 네트워 자원의 낭비를 최소화할 수 있다.

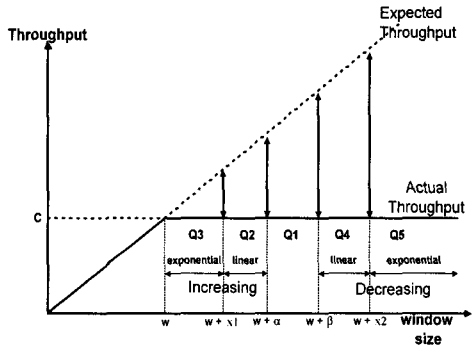


그림 13 Accelerated Vegas 증감

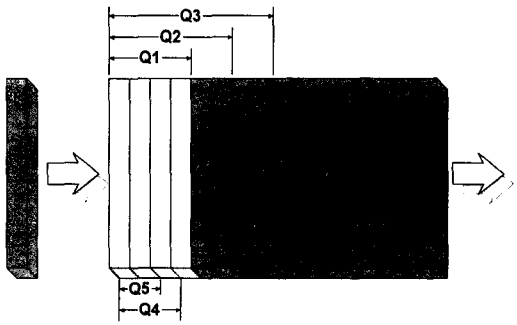


그림 14 Modeling Queue

(그림 14)은 (그림 13)의 Queue 상태를 나타낸 것이다. (그림 13)의 Q1은 CWND의 상태 변화가 필요하지 않은 안정적인 네트워 형태를 나타내는 것이다. 네트워 용량과 네트워 전송량이 가장 효율적으로 유지되는 상태를 나타낸다. 반면 Q2 상태는 CWND의 linear한 증가, Q3은 exponential한 증가 유도한다. 그리고 Q4와 Q5는 각각 linear 및 exponential한 증가를 유도한다.

제시된 accelerated Vegas를 통해서 셀 손실을 미리

방지할 수 있게 되고 ATM 네트워크 환경에서 TCP 전송시 성능에 가장 큰 문제점으로 지적된 셀 재전송에 의한 전송효율 문제를 해결할 수 있다. 그러므로 전체적인 네트웍 성능의 향상을 기대할 수 있다.

6. 결 론

본 논문에서는 ATM UBR 서비스를 통해서 TCP/IP를 수용하기 위해 TCP 측면에서의 성능 요구사항을 추출하고, 이를 ATM 망 및 패킷망에 적용시켜 시뮬레이션을 수행함으로써 성능을 분석하고 평가하였다.

언급한 바와 같이, ATM 망에서 TCP/IP 전송에 관한 문제는 ATM 스위치와 함께 TCP 구현과 관계되는 문제이다. 특히 TCP와 관련해서, 고정된 시간을 기반으로 한 재전송 메커니즘을 가진 기존 구현은 높은 대역폭의 네트웍에서 오히려 전송률 저하를 가져와 성능의 저하를 유발하므로, 이를 해결하기 위해 즉 TCP의 Congestion Avoidance 개선을 통해 성능향상을 위해 본 논문에서는 동적인 TCP 혼잡제어 알고리즘 개선 메커니즘을 제시하였다.

향후 연구 항목으로는 제안된 TCP 혼잡제어 알고리즘을 TCP Vegas에 적용시키는 것이다. 이와 관련해서 현재, 다양한 방법으로 이를 구현하기 위한 연구를 추진하고 있다.

참 고 문 헌

[1] The ATM Forum, "Traffic Management Specification Version 4.0," April 1996.

[2] ITU-T Recommendation I.356, "B-ISDN ATM Layer Cell Transfer Performance," October 1996.

[3] ITU-T Recommendation O.191, "Equipment to assess ATM Layer Cell Transfer Performance," April 1997.

[4] ITU-T Recommendation G.826: "Error performance parameters and objectives for international constant bit rate digital paths at or above the primary rate". [Revised draft, issued date: 14.03.1994]

[5] ITU-T Recommendation G.703, "Physical and electrical characteristics of hierarchical digital interfaces," 1991.

[6] ATM Forum contribution 96-0810R7, "ATM Forum Performance Testing Specification," December 1997.

[7] Raj Jain, "Performance Testing Effort at the ATM Forum : An Overview," IEEE Communication Magazine, October, 1996.

[8] J.Ahn, Pter Danzig, Zhan Liu, Elliot Yan, "Evaluation of TCP Vegas : Emulation and Experment,", In Proceeding of the ACM SIGCOMM '95 Symposim, 1995.

[9] Lawrence Brakmo, Sean W. O' Malley, Larry Peterson, " TCP Vegas: New Techniques for Congestion Detection and Avoidance," In Proceedings of the ACM SIGCOMM '94 Symposium, p 24-35, 1994.

[10] Lawrence Brakmo, Larry Peterson, "Performance Problems in 4.4BSD TCP," ACM Computer Communication Review, 25(5):69-86, October 1995.

[11] Shiv Kalyanaraman, "Performance and buffering requirments of Internet Protocol over ATM ABR and UBR services," IEEE Communication Magazine, October, 1997.

[12] Mohit Aron, "Analysis of TCP Performance over ATM Network," Master of Science Rice University.

[13] Teunis J. Ott, Neil Aggarwal, "TCP over ATM1: ABR or UBR ?," In Proceedings of the ACM SIGMETRICS '97 Conference, Seattle, WA, June 1997.

[14] Norman C. Hutchinson, Larry L.Peterson, "The x-kernel : An Architecture for Implementing Network Protocols," IEEE Transactions on Software Engineering, 17(1):64-76, January 1991.

[15] L. L. Peterson, Bruce S. Davie, Andrew C. Bavier, "x-kernel Tutorial," Jan 1996.

[16] L.L. Peterson, "x-kernel Programmer's Manual (Version 3.3)," Network Systems Research Group, Department of Computer Science, University of Arizona, June 1997.

[17] L. L. Peterson, "Getting Started with the x-kernel," Network Systems Research Group, Department of Computer Science, University of Arizona, Jan 1996.

[18] L.L. Peterson, B.S. Davie, "Computer Networks: A Systems Approach. Morgan Kaufmann Publishers," San Fransico, CA, 1996.

[19] L. L. Peterson, "x-Sim User's Manual (Version 1.0)," Network Systems Research Group, Department of Computer Science, University of Arizona, Jul 1997.

[20] W.R. Stevens, "TCP/IP Illustrated Volume 1 : The Protocols," Addison-Wesley, Reading, MA, 1994.

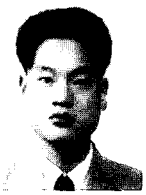
[21] G.R. Wright, W.R.Stevens, "TCP/IP Illustrated Volume 2 : The Implementation," Addison-Wesley, Reading, MA, 1995.

[22] Jeffrey C.Mogul, Observing TCP dynamics in real networks; Conference proceedings on Communications architectures & protocols , 1992, Pages 305 -317

[23] Steven M. Bellovin, "A Best-Case Network Performance Model," February 1992.

[24] G. Kim and A. Bestavros, "Preserving Bandwidth Through A Lazy Packet Discard Policy in ATM Networks," Tech. Rep. BUCS-TR-98-005, Boston University, Computer Science Department, February

1998.
[25] Zheng Wang and Jon Crowcroft, "A New Congestion Control Scheme: Slow Start and Search (Tri-S)," ACM Computer Communication Review, vol. 21, pp 32-43, Jan 1991
[26] Zheng Wang and Jon Crowcroft, "Eliminating Periodic Packet Losses in the 4.3-Tahoe BSD TCP Congestion Control Algorithm," ACM Computer Communication Review, vol. 22, pp. 9--16, Apr. 1992
[27] Lawrence S. Brakmo and Sean W. O'Malley, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," in SIGCOMM '94 Conference on Communications Architectures and Protocols, (London, United Kingdom), pp. 24-35, Oct. 1994.



안 성 수
1973년 8월 12일생. 1998년 부경대학교 정보통신공학과 학사. 1998년 ~ 현재 부경대학교 정보통신공학과 석사과정 재학중. 관심분야는 프로토콜 엔지니어링, 프로토콜 검증 및 시험, 컴퓨터 네트워크, TCP/IP Testing, Formal Description Technique

유 홍 식
1971년 6월 14일생. 1997년 경성대학교 전산통계학과 학사. 1999년 경성대학교 전산통계학과 석사. 1999년 ~ 현재 부경대학교 정보통신공학과 박사과정 재학중. 관심분야는 프로토콜 엔지니어링, 프로토콜 검증 및 시험, 컴퓨터 네트워크

황 선 호
1959년 10월 9일생. 1982년 경북대학교 전자공학 학사. 1985년 연세대학교 전자공학 석사. 1995년 연세대학교 통신공학 박사. 1983 ~ 1998 한국전자통신연구원. 1998 ~ 현재 한국전파기지국관리(주) 기술연구소 연구소장. 관심분야는 CDMA / IMT2000 이동통신시스템, 지상파 디지털 TV방송시스템

이 준 원
1976년 서울대학교 전자공학과. 1992년 충북대학교 전산과(이학석사). 1997년 충북대학교 전산과(이학박사). 1977년 ~ 1979년 삼성전기 기술개발실. 1980년 ~ 1998년 한국전자통신연구원 초고속망연구실장. 1987년 ~ 1988년 AT&T Bell 연구소 방문연구원. 현재 안동대학교 정보통신공학과 교수. 관심분야는 초고속정보통신망, 프로토콜엔지니어링, 소프트웨어 공학 등

김 성 운
1982년 경북대학교 전자공학과. 1990년 프랑스 국립파리 7 대학교 정보공학과(공학 석사). 1993년 프랑스 국립파리 7 대학교 정보공학과(공학 박사). 1982년 ~ 1985년 한국전자통신여구소 데이타통신 연구실(연구원). 1986년 ~ 1995년 한국통신 연구개발원(선임연구원). 1995년 ~ 현재 프랑스 전기통신기술연구소(초빙연구원). 1995년 ~ 현재 부경대학교 정보통신공학과(교수). 관심분야는 프로토콜 엔지니어링, 데이타통신 통신프로토콜시험, 컴퓨터 네트워크