ATM에서 TCP의 효율적 동작에 관한 연구 †

이재원○ 윤현수

한국과학기술원 전산학과

Efficient TCP Traffic Control Over ATM Networks

Jae-Won Lee○ Hyunsoo Yoon

Dept. of Computer Science, KAIST

e-mail : {jaewon,hyoon}@camars.kaist.ac.kr

요 약

컴퓨터가 발전해오고 그 활용폭이 넓어지면서 컴퓨터 네트워크의 중요성이 점차 증가하고 있다. 단독으로 실행되는 컴퓨터 그 자체만으로는 활용에 제한이 있다. 다른 컴퓨터들과 유기적으로 연결된 상태에서 데이타의 공유와 작업의 분산을 통해 좀 더 효율적인 작업을 할 수 있는 것이다. 따라서 컴퓨터를 논함에 있어서 네트워크는 결코 빼질 수 없는 존재가 된 것이다. 이에 우리는 차세대 네트워크라 불리우는 ATM network에 대해 알아보고 여기에서의 현안 문제의 하나인 TCP의 동작에 대해 논하기로 한다.

1. 서론

ATM(Asynchronous Transfer Mode) network은 ATM switch를 사용하여 저 용량 저속 데이타 전송에서부터 고속 광대역 통신에 이르기까지 높은 수준의 통신 품질을 보장할 수 있어 차세대 network으로 주목 받고 있다.[1, 2] 미래의 network에서는 많은 양의 멀티미디어 데이타가 전송될 것이고 ATM network은 이러한 요구를 충분히 소화해낼 수 있고, 또한 기존의 네트워크와 연계하여 사용할 수 있기 때문에 앞으로 ATM network의 역할이 중요해 지리라 쉽게 예상할 수 있다.

그러나 여태까지의 많은 연구에 의해 ATM network 상에서 TCP (Transmission Control Protocol)를 이용한 데이타의 전송은 상당히 비효율적이라고 알려졌다.[3, 1] TCP의 특성상 ATM switch의 장점을 제대로 활용하지 못하기 때문에 기존의 packet-switched network상에서 보다 나쁜 효율을 보인다. 하지만 현존하는 대부분의 network 응용 프로그램들이 TCP/IP를 사용하므로 이 프로그램들을 ATM network에서 그대로 사용하기 위해서는 이 문제를 해결해야만 하고 또한 이 문제들은 당분간 지속될 것이다.

TCP가 ATM 상에서 비효율적으로 동작하는데는 많은 이유가 있지만 본 연구에서는 주로 네트워크 상에서 congestion control로 인한 성능 저하에 대해 알아보고 기존에 제안된 여러 해결책들을 알아본 뒤, 이를 개량한 새로운 방법을 제시하여 그에 대한 성능 평가를 해보도록 한다.

†본 연구는 한국과학기술원 정보시스템연구소의 지원을 받아 연구됨

1.1 ATM network의 원리와 그 특징

미래의 네트워크에서는 음성, 영상, 문자 등 멀티미디어 데이타를 효율적으로 전송 시킬 수 있는 능력이 요구될 것이다.[4] ATM은 이들을 일원적으로 다룰 수 있어 현재로서 가장 많이 주목 받고 있는 기술 중 하나이다.

기존의 네트워크는 크게 circuit-switched network과 packet-switched network 두 가지로 구분 될 수 있다. ATM은 이들 방식의 장점을 모은 중간적 교환 방식이다. 두 단말이 서로 통신하기 위해서 우선 단말간의 virtual path를 설정한다. 이후 모든 통신은 이 virtual path를 통하게 된다. 전송되는 모든 데이타는 53-byte의 cell 단위로 쪼개지고 이들 cell이 virtual path내의 virtual channel을 통해 전달 됨으로써 데이타의 전송이 이루어 진다. Virtual path에는 여러 개의 virtual channel이 있을 수 있고, 이 virtual channel들은 전송되는 데이타의 성격에 따라 서로 다른 bandwidth를 갖을 수 있다. 전체 ATM network 상에는 많은 ATM switch가 있어 cell들을 원하는 곳까지 갈 수 있도록 switching을 해준다. ATM switch는 고속 전송을 위하여 cell을 53-byte로 고정시키고 하드웨어 적으로 간단히 routing을 하게 된다.

1.2 ATM에서 TCP를 사용했을 때의 문제점

먼저 기본적인 TCP의 동작에 대해 알아보자. TCP는 connec-tion-oriented protocol 로서 unreliable internetwork 상에서 reliable end-to-end byte stream을 제공하기 위해 만들어졌다.[5] Network 상에서 packet들이 이동하는 도중 destination 까지 가지 못하는데는 두 가

지 경우가 있을 수 있다. 첫째는 network의 noise로 인한 packet loss가 있을 수 있고, 둘째로는 network 상의 congestion으로 인해 전송이 지연되는 경우이다. 이 경우 destination 쪽에서는 이 packet을 무한정 기다릴 수 없으므로 어느 일정한 시간(time out)이 지난 뒤에는 source에서 retransmission을 해야 한다. 기술의 발전으로 network의 품질도 좋아짐으로써 요즘엔 noise로 인한 loss는 거의 없으므로 congestion에 의한 delay를 없애는 데 관심이 집중되고 있다.

TCP packet이 ATM network 상에서 전송되기 위해서 다시 53-byte cell 단위로 쪼개져야 한다. 그 뒤 ATM switch가 이 각각의 cell들을 전송하게 된다. 여기서 문제점이 생길 수 있다. 한 packet의 어떤 한 cell만 loss되도 나머지 cell들은 쓸모 없어지게 된다. 오히려 이들은 network의 bandwidth만 소비하는 결과를 낳는다. 이러한 현상은 ATM switch 내의 buffer가 full됐을 때 switch에서는 그 이후의 cell들을 모두 drop시키는 방법을 사용하기 때문에 일어나게 된다. 즉, 한 packet이 전송될 때 이를 구성하는 cell중 일부가 switch에 도착 했을 때 buffer가 full 되면 그 뒤의 cell들은 모두 버려지기 때문에 packet의 일부가 손실 되고, 그 packet은 더 이상 쓸모 없어지게 되는 것이다. 따라서 기존의 network에선 보이지 않던 packet loss 현상이 일어나게 되고 이것이 TCP의 congestion control로 인해 어떤 time interval 동안 packet이 도착하지 않을 때의 retransmission과 합쳐져 쓸모 없이 traffic을 증가시키게 되는 것이다.

본 연구에서는 이 time interval을 network delay에 맞춰 dy-namic하게 변화 시킴으로써 불필요한 packet retransmission을 줄이는데 중점을 두었다. 또한 기존에 ATM 상에서 TCP의 효율을 증대시키기 위한 다른 방법들도 혼용하여 더욱 향상된 성능을 보이는지 실험하였다. 실험에는 YATS network simulator를 사용하였다. 여기에는 여러 가지 network 환경의 parameter를 사용할 수 있고 ATM에서 TCP/IP 전송 simulation도 제공하고 있다. 실험 환경은 SPARC Station 10 / Solaris 2.5 상에서 YATS simulator를 사용하였다.

2. 관련연구

ATM switch에서 buffer full이 일어나면 무조건 buffer에 빈자리가 생길 때까지 들어오는 cell들을 drop하게 된다. 이렇게 되면 어떤 TCP packet의 일부 cell이 drop됐을 때 packet에 손상이 가게 되므로 그 packet은 더 이상 쓸모 없게 된다. 그 뒤 TCP 자신의 error handling으로 그 packet을 retransmission 하게 된다.

Worst case에서는 cell 하나만 계속 drop될 경우 network의 효율은 급격히 떨어지게 될 것이다. 이를 방지하기 위한 여러 가지 방법들이 연구됐고 몇몇 알고리즘들은 성능이 우수하여 실제로 사용되고 있는 것들도 있다. 여기서는 그 중 대표적 두 가지 방법을 제시하고 각각의 장단점을 살펴 보도록 한다.

2.1 Partial packet discard

먼저 partial packet discard(PPD) 방법이 있다. 이는 ATM에서 buffer full이 일어나면 나중에 buffer에 빈 공간이 생기더라도 그 packet에 해당하는 모든 cell들을 버린다.

ATM switch에서 buffer full이 일어났을 때 cell을 drop시키는 것은 ATM 자체의 방법이므로 다른 방법이 없다. 다만 이렇게 cell drop으로 더 이상 쓸모 없어진 packet의 나머지 cell들도 따라서 모두 버림으로써 불필요한 cell들의 전송을 막자는 것이 본 방법의 취지이다.

PPD를 구현하는 것은 그리 어렵지 않다. Virtual channel(VC)에서 한번 cell drop이 일어나면 ATM switch는 해당 VC에서 packet의 끝을 알리는 ATM-layer-user-to-user (AUU) parameter set을 만날 때까지 계속해서 cell drop을 해나간다. 하지만 여기에도 더 이상 넘지 못하는 한계가 있다.

Buffer full이 일어나기 직전까지 보냈던 cell들은 그 이후의 cell들이 전송되지 못함으로써 쓸모 없게 되므로 차라리 전송하지 않는 것이 좋았을 것이다. 먼저 전송된 쓸모 없는 cell들은 network의 bandwidth를 낭비하는 결과를 초래한다. 이를 해결하기 위한 방법이 다음에 소개될 early packet discard method이다.

2.2 Early packet discard

Romanow와 Floyd의 연구에서는 ATM network 상에서 con-gestion이 일어났을 때 각각의 cell들을 drop시키는 것보다 TCP packet 전체를 drop시키는 Early packet discard(EPD) 방식을 제안했다.[6] 이 방법으로 중간에 drop된 cell들로 인해 쓸모 없어진 packet들을 전송함으로써 빚어지게 되는 network bandwidth의 낭비를 막을 수 있다. Buffer full이 일어날 것을 예상하고 buffer full을 일으키게 할 packet 전체를 drop하게 되면 쓸모 없는 cell들의 전송을 최소화 할 수 있을 것이다.

하지만 문제는 어떻게 buffer full이 일어날 것을 예상할 수 있는가 하는 것이다. 이 문제는 buffer size에 대해 threshold를 정하고 buffer에 그 값 이상 cell이 들어오면 그 이후의 packet들을 버리게 된다. 이렇게 함으로써 threshold를 넘기 전까지 들어온 packet들은 모두 안전하게 전송될 수 있고, 이후의 packet들은 buffer 내에서 잘릴 것이 예상되므로 전체를 버리게 된다.

EPD mechanism은 ATM switch와 무관하게 동작할 수 있으므로 end-to-end congestion control을 사용하지 않는 UDP 같은 곳에서도 효과를 볼 수 있다.

2.3 PPD와 EPD에 대한 성능 평가

먼저 TCP가 ATM이 아닌 일반 packet-switched network을 통해 전송될 때의 효율을 살펴보자. 그림 1은 buffer size에 대한 전송 효율을 나타낸 것이다. Buffer size가 작아도 좋은 전송 효율을 보임을 알 수 있다.

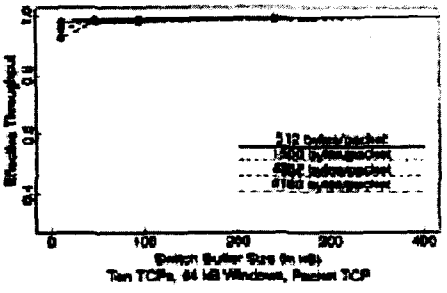

그림 1: non-ATM 상에서 TCP의 효율

그림 2는 TCP가 ATM 상에서 전송될 때의 효율을 나타낸다. non-ATM에 비해 상당히 효율이 떨어짐을 한

눈에 알 수 있다.

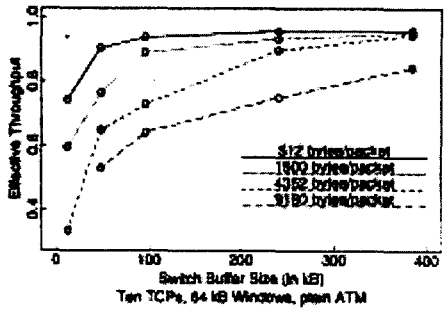

그림 2: ATM 상에서 TCP의 효율

그림 3은 ATM 상에서 PPD를 사용하는 경우이다. PPD를 사용할 경우 packet size가 커질수록 그 효과가 더 커진다는 것을 알 수 있다.

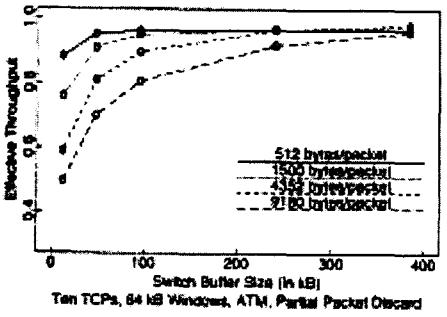

그림 3: ATM 상에서 PPD를 사용하는 경우의 효율

그림 4는 EPD를 사용하는 경우이다. Threshold는 buffer size의 반으로 잡은 경우이다. 이 경우에서는 전반적으로 PPD보다는 효율적이지만 packet size가 작을수록 효과가 커진다. 이는 threshold를 설정해서 평균적으로 전체 buffer size 전체를 충분히 사용하지 못하기 때문인 것 같다.

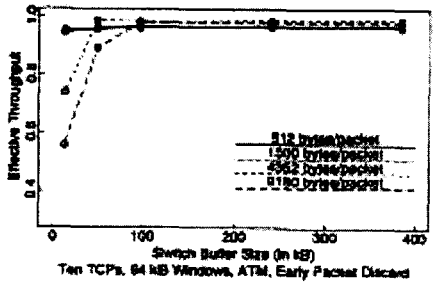

그림 4: ATM 상에서 EPD를 사용하는 경우의 효율

3. 시뮬레이션

3.1 YATS - Yet Another Tiny Simulator

YATS는 ATM network을 위한 cell-level 시뮬레이터이다. 이벤트 스케쥴러, 심볼 메니져/스케너/파서로 구성된 커널을 가지고 있다. 입력 화일은 시뮬레이션을 위한 network 환경 모델을 기술하며, 시뮬레이션을 실행하고, 그 결과를 분석할 수 있다. 이 파일은 간단한 스크립트 언어로 작성되며, 시뮬레이션 설정 뿐만 아니라, 루프, 매크로, 기본적인 산술연산 등을 수행할 수 있다. C++로 작성된 이 시스템은, 모든 network 노드를 object로 구현하며, 이들간의 통신은 표준화된 message를 사용한다. Graphic object를 사용하여 시뮬레이션 결과를 화면상으로 직접 출력하는 것도 가능하다. 현재 YATS의 버전은 0.1로, 지원하는 기능은 다음과 같다.

- variety of cell source types
- delay lines, different shapers, policing function
- multipexers (including WFQ, EPD) / demultiplexers + switches are easily composed from mux / demux
- ABR entities (source, multiplexer, and sink according to ATMF TM 4.0)
- AAL 5 end system entities
- TCP / IP end system entities
  + TCP/IP, AAL5, and ABR entities can be stacked
- measurement devices, and graphical online displays

아직은 기능이 부족한 면이 있지만, 조만간에 보완되리라 본다.

3.2 시뮬레이션 setup

High bandwidth에서 low bandwidth인 network으로 데이터를 전송할 때, switch에서 발생하는 병목현상에서 시뮬레이션을 하려고 하였으나, YATS에서는 switch에서 bandwidth를 조절할 수 가 없으므로, packet을 전송하는 호스트의 수를 늘려 switch가 bottle neck이 되도록 하였다.

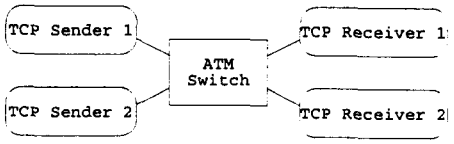

그림 5: 시뮬레이션 구성도

그림 5와 같이 TCP sender 1과 2는 계속적으로 TCP Receiver 1과 2로 packet을 보내며, 이는 ATM switch를 통하여 routing된다. 이때, ATM switch에서는 병목현상이 일어나게 되며, switch의 buffer가 다 차게 되고, 손실되는 packet이 발생할 것이다. 위의 모델을 YATS에서 기술하는 network 노드로 바꾼 것이 그림 6이다.

- CBRFrame 2개의 CBRFrame을 사용하는데, CBRFrame은 일정 시간을 간격으로 data를 만들어내는 data source이다. 이 시뮬레이션에서는 100 Tick(2ms)마다 9180 byte의 data(4.38MB/s)를 만들어 내도록 하였다.
- Senke Sink이다.
- TCP/IPsend TCP/IP entity를 시뮬레이션 하는 노드이며 보내기만 하는 node이다. 현재는 CBRFrame에서 만들어낸 data를 받아서 AAL5send로 연결시킨

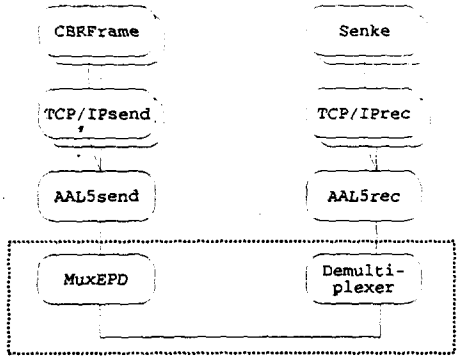

그림 6: YATS 네트워크 노드

다. 이 node에는 여러 옵션이 있는데, 아래와 같이 설정하였다.

Buffer Size	64,000 Bytes
MTU	9,180 Bytes
Time Stamp Option (RFC1323)	On
Fast Retransmission and Recovery	Off
TICK	1
NAGLE (RFC1122)	Off

[표 1]

- TCP/IPrec TCP/IP entity를 시뮬레이션 하는 노드이다. 받기만 하는 node이다. 현재는 AAL5Rec에서 받은 data를 sink로 보내어 소멸시킨다.
- AAL5Send AAL5를 시뮬레이션 하는 노드이다. 보내기만 하는 node이다. 현재는 TCPIPsend에서 만들어낸 data를 받아서, AAL5send로 연결시킨다.
- AAL5Rec AAL5를 시뮬레이션 하는 노드이다. 받기만 하는 node이다. 현재는 switch(mux/demux)에서 받은 data를 sink로 보내어 소멸시킨다.
- MuxEPD/Demultiplxer 2가지가 합쳐 하나의 switch를 simulation 한다. MuxEPD는 Early Packet Discard를 하는 mux이다.

RTOMIN을 2가지의 경우에 대하여 simulation을 하였다. 450ms와 100ms일 경우에 대하여 각각의 자료를 뽑았다.

3.3 시뮬레이션 디자인

TCP가 ATM에서 나쁜 효율을 보이는 이유 중 retransmission으로 인한 traffic 증가와 bandwidth 소모가 가장 크다. 이를 최소화 하기위해서 우리는 ATM의 빠른 전송 능력을 최대한 활용하기 위한 방안을 모색해 봤다. 그 결과 TCP가 retransmission 하기까지 기다리는 시간 즉, retransmission time-out(RTO)를 줄여 보기로 했다.

ATM은 기존의 network에서 보다 bandwidth가 높고 전송 속도 또한 빠르기 때문에 기존의 time-out 시간은 너무 길다고 판단 되었다. 위에서 살펴본 바와 같이 ATM switch 내에서 cell dropping이 일어나면 packet을 retransmission 해야만 한다. 이 때 sender가 time-out이 지날 때 까지 기다렸다가 receiver로부터 acknowledgement가 오지 않았을 경우 retransmission을 해야 하는데 이 기다리는 시간을 줄이면 전체적인 성능이 증가될 것이다.

4. 실험 결과 및 분석

4.1 실험 결과

4.1.1 configuration 1)RTOMIN = 450 ms

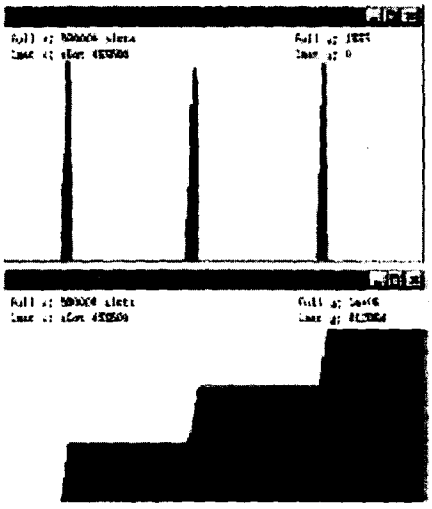

그림 7: RTOMIN을 450 ms로 했을 때의 결과

4.1.2 configuration 1)RTOMIN = 100 ms

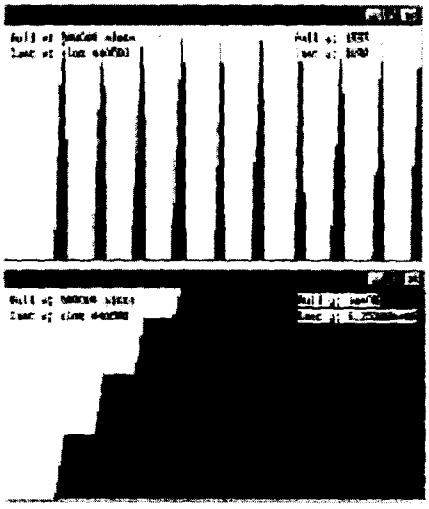

그림 8: RTOMIN을 100 ms로 했을 때의 결과

4.1.3 결과 데이타

Configuration 1) RTOMIN = 450 ms
average throughput : 1.1938505859e+04
load in abrmux : 5.1519400000e-01

loss in abrmux : 1251

Configuration 2) RTOMIN = 100 ms
average throughput : 2.4008369141e+04
load in abrmux : 6.0467100000e-01
loss in abrmux : 4205

4.2 결과 분석

그림 7, 8은 500,000 time slot 동안의 ATM switch의 buffer의 사용량(위)과, TCP Sender1이 실제로 전송한 자료의 양(아래)을 Minimal Retransmission Time Out이 450ms일 경우와 100ms일 경우에 대하여 나타낸 그래프이다.

그림 7의 위의 그래프를 살펴보자. ATM switch의 buffer가 TCP sender의 burst input에 의하여 순간적으로 차며, buffer가 넘치는 순간에 이르면 buffer의 사용량은 다시 순간적으로 줄어들며, 오랜 시간 동안 buffer가 비어있다가, 다시 차 들어가기 시작하며, 이 과정이 계속 반복적으로 일어나고 있다. 처음에 buffer가 차 들어가는 것은 TCP sender가 보내는 data에 의한 것이며, buffer가 찰 경우 packet loss가 일어나서 TCP는 현재 보낼 수 있는 packet만을 더 전송하고, receiver로부터 loss가 일어난 packet에 의해 더 이상의 ACK 신호를 받을 수 없으므로 전송을 중단하게 된다. 그러므로, buffer의 사용량은 순간적으로 다시 줄어들며, RTO (Retransmission Time Out)이 될 때 까지 아무런 작업을 하지 않은 채, bandwidth를 낭비하고 있는 것이 보인다. Graph에서 이 간격은 RTOMIN(450ms)와 같은 것을 알 수 있다 (1 Tick=20us).

그림 8는 RTOMIN을 100ms로 설정하고 simulation한 결과이다. 이미 예측했듯이, packet loss가 일어난 시점에서 retransmission이 시작된 시기가 현저히 줄어들어 전송률이 훨씬 좋아진 것을 알 수 있다. 실제 전송률과 switch의 load를 볼 때도, RTOMIN이 적을 때, 효율이 더 높으며 load도 늘어났다. Load가 늘어난 것은 실제 더 많은 데이터를 전송하기 때문에 당연하다고 할 수 있다. 또 그에 따른 buffer pool의 회수도 많아지며, cell loss도 많아짐을 알 수 있다. 결론적으로 Minimal RTO를 줄이면 더 congestion상태에서 더 좋은 효율을 볼 수 있다고 할 수 있다.

5. 결론

TCP가 ATM 상에서 사용될 때는 기존의 다른 network에서 보다 그 효율이 많이 떨어진다. 특히 congested network 상에서 그런 경우가 심하다. 그 이유는 congested 된 switch에서는 buffer full이 일어날 가능성이 크고, 그렇게 됐을 경우 switch는 full된 이후의 cell들을 모두 drop시키는데 그렇게 할 경우 TCP packet에 손상을 가져오게 되므로 순간적인 buffer full로 인해서도 packet 전체를 retransmission 해야 하기 때문이다. 이럴 경우 ATM과 같은 고속 통신망에서는 packet retransmis-sion time-out을 적게 잡을수록 그 효과가 크다는 것을 알았다. 하지만 그 값을 너무 작게 잡을 경우 불필요한 retransmission을 하게 될지도 모른다. 이런 경우는 약간의 congestion은 일어났지만 buffer full로는 이어지지 않아 cell dropping이 일어나지 않을 경우다.

따라서 RTO 값을 적정하게 결정하는 것이 무엇보다 중요하다. 분명한 것은 ATM network에서는 기존의 network에서보다는 작은 값을 사용하는 것이 더 효율적이다라는 점이다. 그러나 현재 network 응용프로그램들이 사용하고 있는 TCP에서는 RTO 값을 300 450ms 정도로 밖에 사용하고 있지 않다. 따라서 이들을 ATM network 상에서 사용하기 위해서는 새로 TCP를 implement 해야 하는 숙제가 남아있게 된다.

이번 연구에서는 단지 RTO를 기존의 TCP가 사용하던 값보다 줄여서 실험을 했을 뿐이다. 하지만 무조건 작은 값이 좋다고 만은 할 수 없을 것이다. 실제 network에는 수많은 변수가 있을 것이고 이들이 어떻게 영향을 미치느냐에 따라서 RTO가 더 큰 경우가 좋을 수도 있을 것이다.

가장 바람직한 경우는 network의 상태에 따라 RTO 값을 dynamic 하게 변화 시켜줄 수 있는 방법을 찾는 것이라 하겠다. 이렇게 network의 상태에 따라 가장 효율적인 RTO 값을 찾는 것은 상당히 어렵다고는 하지만, ATM은 circuit-switched network이란 점과 switch 내부에서 최소한의 작업만을 하기 때문에 기존의 network에서 보다 변수가 작으므로 가능성은 있다고 본다.

참고 문헌

[1] G. Minden, V. Frost, J. Evans, and B. Ewy. TCP/ATM Experiences in the MAGIC Testbed. 1995.

[2] Rainer Handel, Manfred N. Huber, and Stefan Schroder. *ATM Networks: Concepts, Protocols, Applications.* Addison-Wesley, 1994.

[3] Allyn Romanow. *TCP over ATM: Some Performance Results.* ATM Forum/93-784, 1993.

[4] Martin de Prycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN.* Prentice-Hall, 1991.

[5] A.S. Tanenbaum. *Computer Networks 3rd.* Prentice-Hall, 1996.

[6] A. Romanow and S. Floyd. The Dynamics of TCP Traffic over ATM Networks. *ACM SIGCOMM Computer Communications Review*, OCT 1994.