韓國 OA學會 論文誌
第7卷 第4號, 2002. 12.

2002-7-4-3-1

TCP-RLDM: Congestion losses과 Wireless losses 구별을 통한 수신측 기반 혼잡제어 방안

노 경 택* 이 기 영**

TCP-RLDM: Receiver-oriented Congestion Control by Differentiation for Congestion and Wireless Losses

Kyung-Taeg Rho* Ki Young Lee**

요 약

본 논문은 수신측이 네트워크 혼잡도를 측정 참여하여 송신측으로 하여금 네트워크 상태에 따른 윈도우 크기를 조절하는데 있다. TCP-RLDM은 기존 TCP 의 Additive Increase/Multiplicative Decrease 방법의 단점을 보완하는 데이터 수신율을 기초로 하는 측정기반 전송 전략을 채택하였다. 이는 유무선 망에서나 지연에 민감하거나 용인하는 응용들로 구성된 이질적 환경에서 동적으로 대응하기 위해 에러손상의 성질에 따른 즉, 혼잡에 의한 손실인지 전송상 일시적 손실인지에 따라서 적절히 대처함으로써 성능을 높일 수 있게 되었다. 수신측으로부터의 데이터 수신율과 에러발생 원인에 대한 정보를 이용, 송신측의 wave 전송방식과 가급적 혼잡이 발생하기 이전에 혼잡회피전략을 적용함으로써 가변적인 네트워크환경에 잘 대처하도록 하였다.

Abstract

This paper aims to adjust the window size according to the network condition that the sender determines by making the receiver participating in the congestion levels. TCP-RLDM has the measurement-based transmission strategy based on the data-receiving rate complementing TCP with the property of Additive Increase / Multiplicative Decrease. The protocol can make an performance improvement by responding differently according to the property of errors - whether congestion losses or transient transmission errors - to confront dynamically in heterogeneous environments with wired or wireless networks and delay-sensitive or -tolerant applications. By collecting data-receiving rate and the cause of errors from the receiver and by enabling sender to use the congestion avoidance strategy before occuring congestion possibly, the protocol works well at variable network environments.

*, ** 서울보건대학 사무자동화과 교수

논문접수 : 2002. 9. 10
심사완료 : 2002. 11. 25

Ⅰ. 서론

TCP는 신뢰적인 윈도우 기반 흐름제어와 혼잡제어 프로토콜이다. 송신측인 세그먼트의 전송을 조절하기 위해 ACK 신호를 이용하고 혼잡상태에 빠져있다는 것을 알기 위해 3 duplicated ACK과 timeout을 이용한다. TCP 송신측은 혼잡상태를 발견했을 때 윈도우 크기를 줄임으로써 전송률을 감소시킨다.

본 논문에서는 window의 크기가 작을 때 timeout이 자주 발생하는 문제점을 해결하기 위해서, window의 크기를 일정한 임계치이상으로 유지하면서 네트워크의 상태에 따라서 전송량을 조절하려고 한다. 네트워크의 상태를 판단하기 위해서 wave 메커니즘을 사용하였으며, wave 메커니즘은 receiver가 수신한 같은 wave에 속한 패킷들의 도착시간을 기초하여 데이터수신율을 계산하여 네트워크의 상태를 파악함으로써, backward path에 대한 네트워크의 정보를 배제하고 오직 forward path에 대한 네트워크 정보만을 활용하여 네트워크의 상태를 측정함으로써 ACK 기반의 방식보다 좀더 정확하게 네트워크의 상태를 파악 할 수 있기 때문이다. 또한 wave 메커니즘은 라우터의 도움을 필요로 하지 않기 때문에 TCP의 종단간 신뢰적인 전송이라는 원리에 위배되지 않는다.

TCP는 동적으로 통신채널 대역폭에 적응하지만 네트워크의 상태를 정확하게 측정하지 못하여 윈도우크기를 잘못 조절한다. 맹목적으로 (blindly) 윈도우 크기를 확대하는 방법은 혼잡을 일으키게 하고 에러회복을 불러일으킨다. 혼잡 (congestion) 은 손실된 세그먼트에 의해 탐지된다 [1]. 윈도우크기를 확대/축소 방법은 네트워크 혼잡상태에 있을 때나 대역폭을 이용가능 할 때 동적으로 전송률의 균형을 유지하기 위해 고안되었다. 네트워크 상태가 좋을 때 윈도우 크기를 점차적으로 증가시키고 패킷이 손실되었을 때 윈도우 크기를 축소하는 방법으로 [2]에서 제안된 AIMD (Additive Increase/Multiplicative Decrease) 알고리즘에 의해 이러한 균형이 이루어진다.

TCP의 급격한 윈도우 크기 축소와 점차적 확대 방법은 TCP 기반 응용들과 지연상태에 민감한 응용들의 성능을 저하시킬 것이다. 본 연구목적은 다음과 같이 두 가지에 초점을 두고 있다. 첫째, 혼잡에 의한 손실이 아닌 무선 환경에서 손실이라는 것을 파악하여 그릇된 혼잡처방을 피하는 것이다. 둘째, 혼잡상태일 동안 중요한 손상을 피하는데 있다. 일부 손상이 혼잡상태일 동안 벌어질지라도 차츰 윈도우 사이즈를 줄이는 것이 애플리케이션 성능을 향상시킬 가능성이 있다. TCP-friendly 프로토콜은 TCP(α, β)의 multiplicative factor를 나타내는 β와 additive increase factor를 반영하는 α를 상쇠 함으로써 이루어진다. 즉 혼잡상태에서 윈도우 크기를 보다 크게 줄이고 난 후에 적절한 비율로 윈도우를 확대시킨다.

수신측 기반 정보를 이용한 수신측기반 (Receiver-oriented) 에러제어는 데이터 전송률과 데이터 손실정도를 정확하게 결정하기 위해 수신측의 도움을 필요로 한다. 이는 손실되거나 지연된 ACK 신호에 기인하는 송신측의 잘못된 평가를 제거한다. 효율적인 접근 방법은 패킷전송상태를 기반으로 한다. 네트워크 혼잡수준을 평가하는 것은 조기에 혼잡회피를 하기 위함이다. 필요한 대역폭을 얻기 위한 경쟁과 큐잉 지연에 관련한 패킷손실은 네트워크 혼잡에 기인한다. 최소 지연으로 전송되는 데이터윈도우로부터의 패킷손실은 비트에러일 가능성이 높다. 네트워크 상태에 대한 평가는 이전과 현재 그리고 다음 RTT 동안에 기대치와 실제 전송률을 비교하면서 보다 정교하게 얻어질 수가 있다. 또한 보다 정교한 에러상태 분류와 대응 프로토콜을 얻을 수 있다.

본 논문의 나머지 부분은 다음과 같이 구성하였다. 2장에서는 TCP의 성능을 향상시키기 위하여 기존 접근방안과 그들의 문제점을 살펴보고 3장에서는 제안방법의 기초가 되는 wave 에 대한 개념을 설명하고, 이를 이용한 우리의 제안 알고리즘을 설명하겠다. 끝으로 4장에서 결론 및 향후 계획으로 맺으려고 한다.

Ⅱ. 관련 연구

무선망에서 패킷손실은 혼잡 (congestion) 에 의한 loss가 아닌 무선링크 특성상 발생하는 것이기 때문에 기존의 혼잡회피 (congestion avoidance) 알고리즘을 수행하게 되면 throughput이 많이 떨어지게 된다. 따라서, 유선 망에 적합하게 발전해 온 TCP 프로토콜을 무선망

의 특성에 맞게 효과적으로 사용하도록 하는 것이 중요한 문제로 인식되고 있다. 이러한 문제를 해결하기 위해서 TCP 프로토콜을 유무선 통합 망 환경에 적응하도록 하는 여러 가지 방법들이 제안되었다.

TCP_Westwood은 유무선 환경에서 TCP 의 성능을 향상시키기 위해서 TCP 혼잡제어 알고리즘의 송신측을 변경하였다 [3]. TCP의 주요한 문제점인 패킷 loss의 원인을 식별하기 위해서 종단간 (end-to-end) 대역폭 검사를 사용하였다. 이들의 특징은 중간노드에서 TCP 패킷을 검사하거나 interception 하는 것을 필요로 하지 않는다. 더군다나 이것은 종단간 TCP 디자인 원리에 완전히 부합된다. 이들의 주요한 아이디어는 돌아오는 ACK rate를 모니터링 함으로써 connection rate를 TCP source가 꾸준히 측정하는 것이다. 이러한 검사는 congestion window를 계산하는데 사용되며, congestion episode 후에 임계치를 slow start시킨다. 이러한 전략의 원리는 간단하다; three duplicate ACK 후에 congestion window를 맹목적으로 반으로 줄이는 TCP와는 다르게 TCP westwood는 slow start threshold와 congestion window를 선택하도록 시도한 것이다. 그러나 ACK의 속도에 기초한 전송률 조절은 잠재적으로 reverse path의 비대칭적인 특성을 반드시 반영하게 된다. 그러므로 송신측의 전송률은 항상 forward path의 성능만을 반영하지는 않는다. 이것은 사용 가능한 대역폭이 이용되지 않고 남아 있으므로 효율에 직접적인 영향을 준다.

TCP Vegas는 잘 디자인되고 측정을 기초로 한 TCP이다 [4]. BaseRTT는 모든 측정된 RTT 중에서 최소값을 갖는 RTT 이다. ExpectedRate는 congestion window와 BaseRTT의 비율이다. 송신측은 샘플 RTT를 이용하여 ActualRate를 측정한다. 만약 ExpectedRate와 ActualRate간의 차이값이 임계치 α 보다 작다면 윈도우 크기를 다음 RTT 동안에 선형적으로 증가시킨다. 차이값이 또 하나의 임계치 β를 초과한다면 다음 RTT 동안 윈도우 크기를 선형적으로 감소시킨다. Vegas는 무선상의 에러와 비대칭적 패스의 문제점들을 다루지 않았지만 측정을 기초로 한 윈도우 크기 조절방법으로서 중요한 메커니즘이다.

TCP- Real [5] 는 네트워크의 상태를 좀 더 정확하게 판단하기 위하여 수신측이 받은 패킷의 정보를 송신측에 보내서 네트워크의 상태를 판단하며, 네트워크의 상태가 좋아지면 전송량을 증가시키고, 네트워크의 상태가 이전과 비슷하면 전송량을 유지하고, 네트워크의 상태가 이전에 비해 많이 나빠지면 전송량을 줄이는 방안을 제안하였다. 또한 TCP-Real은 네트워크의 상태가 나쁠 때 발생한 손실은 혼잡에 의한 손실로 여겨서 전송량을 감소시키고, 네트워크의 상태가 좋을 때 발생한 손실은 무선망의 높은 비트 오류율로 인한 손실로 보고 전송량을 줄이지 않는 방안을 제안하였다. 그러나 TCP-Real은 링크를 공유하는 TCP연결의 수가 많아질 경우, flow들간의 경쟁으로 인하여 자주 손실이 발생할 경우 성능에 많은 저하를 가져온다. 윈도우 크기가 fast retransmit 와 fast recovery하기에 작은 수의 패킷을 가진 경우 빈번히 timeout이 발생함으로써 TCP-Real의 성능은 저하될 것이다.

Ⅲ. TCP-RLDM: Receiver-Oriented Loss Discriminating Mechanism

수신측이 네트워크의 상태를 정확히 판단하고 패킷 손실을 정확히 관찰하기 위해서, 송신측은 잘 알려진 통신 패턴으로 패킷을 전송해야 한다. 우리는 이러한 통신 패턴을 wave라고 부르며, 이것은 논문 [8,9] 에서 소개되었다. 고정된 크기의 data segment들의 그룹으로 이루어진 wave는 연속적으로 보내어진다. 실질적으로 TCP wave는 추가적인 정보를 가지는 congestion window로 볼 수 있다. 이 wave의 크기에 대한 정보는 송신측과 수신측이 공유하게 된다. TCP의 전송 패턴이 ACK-clocking mechanism에 의해서 조정되기 때문에 패킷 또는 ACK가 지연되거나 손실될 때 congestion window는 연속된 패킷들의 작은 그룹으로 분할될 것이다. 이들 그룹은 wave에 해당한다. 그 결과 wave 패턴은 하나의 congestion window 안에 연속적으로 보내질 수 있는 부분이라 말할 수 있다.

Wave를 이용하여 통신의 양측에 잘 알려진 패턴으로 데이터를 교환함으로써 수신측은 네트워크의 상태를 정확히 판단할 수 있다. 패킷간 간격 (Inter-packet gap) 과 wave delivery time은 네트워크 안에 경쟁상태 (contention)를 알려주는 정보로 에러구분 (error classification) 과 적절한 회복 (appropriate recovery) 을 가능하게 해준다. bottleneck이 걸린 라우터 상에서 여러 flow들이 경쟁함으로써 생기는 패킷들의 interleaving 패턴으로 인하여 결정되는 해당 wave의

데이터 수신율 (data-receiving rate) 을 수신측은 계산을 한다. 기대되는 데이터 수신율이 낮다는 것은 bottleneck router에서의 congestion level이 높다는 것을 의미하며, 전송률을 낮출 것을 제안하게 된다. 이와 반대로 기대되는 데이터 수신율이 높다는 것은 네트워크 내의 congestion level이 낮다는 것을 의미하며, 전송률을 높일 것을 제안하게 된다. congestion 동안에는 데이터 수신율은 bottleneck router 상에서 발생하는 패킷상실 (packet drop) 으로 인하여 극동적으로 변하게 될 것이다. 그러나, 만약 패킷상실이 wireless error에 의한 것이라면, 데이터 수신율은 영향을 받지 않을 것이다. 이러한 관찰은 일시적인 에러를 congestion으로부터 구별하기 위한 방안으로 사용할 수 있다. 이들 방안은 현재의 rate와 이전 rate를 비교를 통한 다음 RTT에 적용한다.

비록 large window가 error detection과 rate estimation을 위한 좋은 샘플을 제공하고 패킷간 갭을 발견할 가능성이 있지만, small window에 의한 판단은 에러나 혹은 우연에 의한 일치를 나타나게 된다. 이러한 불확실성을 내포할 때는 [6]의 에러상태 구별방법을 가미한 수정된 standard AIMD를 적용한다.

개선된 error classification과 congestion estimation은 에러의 성격에 기초한 error recovery를 가능하게 된다. 수신측은 패킷손실을 윈도우 크기 조절로부터 분리 할 수 있다. 즉 수신측은 data receiving rate가 높을 때 잃어버린 패킷의 random error로 인한 손실로 여겨서 불필요하게 congestion window의 크기를 줄이지 않으므로 성능을 향상시킨다. 또한 네트워크의 상태를 미리 판단하여 상태가 이전 보다 나빠진다고 판단이 되면, 수신측은 송신측의 전송률을 조절하도록 지시한다. 그러므로, congestion avoidance 기간동안 drop되는 패킷을 줄일 수 있다. *Receiver-oriented control*은 *asymmetric* path의 문제를 극복할 수 있다. 즉 수신측은 reverse path의 특징으로 인한 문제점을 배제하고 rate adjustment 할 수 있다.

3.1 Packet Inter-Arrival Time as the Discriminator

그림 1을 참고할 때 path 의 마지막 링크는 무선링크이고, 이 무선링크는 연결에 있어서 bottleneck 이고, 송신측은 bulk 데이터 전송을 한다고 가정하자. 그림 1(a) 경우에 수신측은 모든 패킷을 패킷간의 간격이 무선링크 상에 하나의 패킷을 전송하는데 요구된 시간 T로 받는다. 그림 1(b)는 패킷 2를 잃는 경우로 두 패킷의 도착시간 간격은 2T이다. 왜냐하면 패킷 2가 전송에러로 인하여 시간 T만큼 이용하기 때문이다. 그림 1(c)는 패킷 2가 중간 라우터에서 혼잡으로 인하여 분실된 경우로 패킷 1과 패킷 3 사이에 도착시간 간격은 시간 T 이다. 이때 패킷 3은 out-of-order 패킷이다.

Tmin은 connection 동안 수신측에서 관측된 최소 도착시간 간격이라 하자. Po는 수신측이 받은 out-of-order 패킷이라 하자. Pi는 Po 이전에 순서적으로 도착한 마지막 패킷이라 하자. Tg는 Po와 Pi 의 도착시간 차이 값이다. Po와 Pi 사이에 분실된 패킷의 수를 n이라 하자. 만약 $(n+1)Tmin \langle = Tg \langle = (n+2)Tmin$ 이라면 n개의 분실된 패킷은 무선상의 전송에러에 기인한다고 간주하고, 그렇지 않을 경우는 혼잡으로 인한 분실로 간주한다 [6]. 패킷 손실을 무선상의 손실로 여길 때는 아주 엄격해야한다. 무선상의 손실을 혼잡에 의한 손실로 간주하는 경우가 그 반대의 경우보다 더욱 성능을 떨어지게 하기 때문이다.

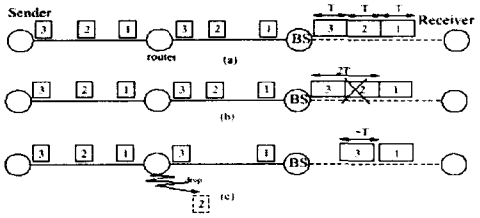

Figure 1. Inter-arrival gap

3. 2 Congestion Avoidance and Control Algorithms of TCP-RLDM

그림 2를 참고로 할 때 수신측은 각 wave의 데이터 수신율을 측정하고, [6]에서 제안된 방법을 이용하여 현재 상태가 정상적인 흐름인지, congestion error 인지, wireless error 인지를 파악한다. 그 결과를 ACK에 실어서 송신측으로 보내어 전송률을 조절하도록 한다. new data에 대한 ACK을 수신하고 cwnd (congestion window) 를 조절할 때 현재의 데이터 수신율을 이전 것과 비교한다. 수신율이 감소되지 않는다면 RTT마다 (α =1) 1 MSS만큼 cwnd를 증가시킨다. 만약 감소치가 작다면 cwnd는 일시적으로 변하지 않는다. 감소치가 높다면 송신측은 cwnd를 1/8 만큼 multiplicatively 하게 줄인다. 여기서 cwnd가 congestion이 발생하기 전에 줄

인다는 것이 중요하다. wave size가 너무 작을 - 4 세그먼트 보다 작음 - 경우 측정된 데이터 수신율을 standard TCP와 유사하게 처리하지만 수신측의 정보(receiver_info) 에 따라서 라인 16과 라인 21과 같이 congestion warning 값 혹은 transmission warning 값을 증가시킨다.

송신측은 wave 안에 데이터와 wave 순서번호와 wave 크기 정보를 TCP header option을 이용해서 전송한다 [5]. 수신측은 현재 wave에 일치하는 데이터 패킷을 모아서 데이터 전송률을 계산한다. 데이터 수신율은 wave 크기와 wave receiving time 간의 비율이다. 무선 통신에서 발생한 패킷손실은 multiplexing/contention 의 정도를 측정하기 위한 rate 계산에 영향을 주지 않는다. congestion에 기인한 패킷손실은 데이터수신율의 변화로 발견할 수 있다. congestion epoch은 윈도우 크기가 중단없이 증가하는 기간이다. 즉 이것은 윈도우 크기가 β w 개의 패킷으로 시작해서 매 RTT 마다 α 패킷만큼 증가시켜 패킷손실이 처음으로 발생할 때인 윈도우 크기 w 패킷 수에 도달할 때까지 증가시킨다 [7].

각 epoch에서 송신측은 가장 높고 낮은 데이터 수신율을 기록한다. 그림 3 혼잡제어알고리즘을 볼 때 라인 2 3 4에서 데이터 수신율의 변동폭이 크고, 현재 수신율이 비교적 낮고, congestion warning 값이 transmission warning 값보다 큰 경우에 congestion으로 간주한다. 송신측은 TCP Reno와 같이 cwnd를 1/2로 줄인다. 그렇지 않은 경우 패킷손실을 무선상 손실로 간주하고 cwnd를 줄이지 않는다. 라인 7 8은 송수신측이 네트워크 정보를 충분히 갖고 있지 않거나 congestion warning 값이 transmission warning 값보다 큰 경우 congestion으로 간주한다. 그렇지 않고 duplicate ACK이 3 보다 같거나 크고 transmission warning 값이 congestion warning 값보다 큰 경우에 일반 TCP와 같은 방법을 적용한다.

TCP-RLDM은 standard TCP에 error/contention 발견 전략을 적용한 것이다. 비대칭적 링크에서 윈도우 크기를 잘못되게 줄이는 것을 피하기 위하여 송신측은 timeout 메커니즘과 RTT를 윈도우 크기와 분리시킨다. standard TCP에서 ACK신호 손실은 timeout 기간을 늘리고 cwnd 크기를 줄인다. TCP-RLDM은 timeout 기간은 늘리지만 윈도우 크기를 같거나 오히려 증가시키기까지 한다.

```
whenever the sender receives an ACK:
01  if(the ACK contains data-receiving rate of the next wave)
02      if(the wave size is larger than 4 packets){
03          previous_rate = current_rate;
04          current_rate = the data_receiving rate in the ACK
05          numberofrateupdates ++;
06          if(receiver_info == 1) {
07                  congestion_warning_num++;
08                  cwnd -= 1 / cwnd;
09          }
10          else if(receiver_info == 2) {
11                  transmission_warning_num++;
12          }
13      }
14      else{
15          previous_rate = current_rate;
16          if(receiver_info == 1) {
17                  congestion_warning_num++;
18                  if(cwnd >= 2)
19                    cwnd -= 1 / cwnd;
20          }
21          else if(receiver_info == 2) {
22                  transmission_warning_num++;
23      }
24      if(duplicate ACK is received) {
25          dupacks += 1;
26          if(dupacks >= 3)
27                  call Congestion Control Algorithm;
28      }
29      if(the ACK acknowledges new data) {
30          rate_ratio = current_rate / previous_rate;
31          if(rate_ratio >= 1.0)
32           cwnd += 1/ cwnd;  // additive increase
33          else if(rate_ratio >= 0.8)
34                  cwnd += 0;
35          else //measurement-based congestion avoidance
36                  if(receiver_info == 1) {
37          cwnd -= 1/8; //apply multiplicative decrease to cwnd
38                  }
39      }
```

Figure 2. Congestion Avoidance Algorithm

```
Whenever 3-dacks are received, or a timeout occurs, do the following
01  congestion = 0;
    // a boolean indicating whether or not congestion occurs
02  if((highest_rate > lowest_rate * c) AND
03  (current_rate - lowest_rate < highest_rate - current_rate) AND
04  (congestion_warning_num >= transmission_warning_num)) {
05          congestion = 1;
06  }
07  if((numberofrateupdates < threshold) OR
08  (congestion_warning_num >= transmission_warning_num))
09          congestion = 1;
10  if(congestion)
11  slowdown(); // backoff: Slow start (if timeout) or
    fast recovery (if 3dacks)
12  if((dupacks == 3) AND
    (transmission_warning_num > congestion_warning_num)) {
13          retransmit a missing packet;  // wireless loss
14          cwnd = cwnd + 3;
15  }
16  if((dupacks > 3) AND (transmission_warning_num >
    congestion_warning_num))
17          cwnd = cwnd + 1;
```

Figure 3. Congestion Control Algorithm

Ⅳ. 결론 및 향후 계획

송신측은 데이터 전송률 조절을 위해 수신측의 도움을 필요로 한다. 수신측의 패킷간의 도착시간을 이용하여 무선 상에 발생하는 일시적 에러 (transient errors) 인지 유선망의 혼잡상태에 따른 손실인지를 파악하는

receiver_info 정보와 wave 크기와 wave receiving 시간을 이용한 데이터 전송률 정보를 결합 적용함으로서 보다 정확하게 에러상태 분류를 얻을 수 있었다. 이 정보를 이용하여 혼잡상태로 인한 에러인 경우에 윈도우 크기를 standard TCP와 같은 1/2 크기로 줄이고, 무선 상태에서 발생한 일시적인 에러일 경우 윈도우 크기를 같은 크기로 유지하거나 오히려 1 MSS만큼 늘림으로써 대역폭의 낭비를 줄일 수 있어 효율적인 네트워크 이용이 가능할 수 있게 되었다. 또한 혼잡이 발생되기 이전에 혼잡 가능성이 있으면 먼저 윈도우 크기를 하나씩 줄임으로써 전체적 성능을 효율적으로 유지한다. 여기서 패킷간의 도착시간에 따른 에러분류 방법은 일반적인 경우로 무선망에 접하는 라우터에서 상당한 경쟁이 벌어질 때 보다 효율적이지만 그렇지 않은 경우에 대한 실험이 필요하다고 생각한다. 제안된 방법에 대한 시뮬레이션과 그 결과로 나타나는 문제점을 해결하려 연구중이다.

참고문헌

[1] M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", RFC2581, April 1999.

[2] D-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", Computer Networks and ISDN Systems, 17(1): 1-14, June 1989.

[3] C. Casetti, M. Gela, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", In Proceedings of ACM Mobicom 2001, July 2001

[4] L.S. Brakmo and L.L. Peterson, "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communications, 13(8): 1465-1480. Oct 1995

[5] V. Tsaoussidis, A. Lahanas and C. Zhang, "TCP-Real: Receiver-oriented Congestion Control", The Journal of Computer Networks (COMNET), 2002

[6] Saad Biaz, Nitin H. Vaidya, "Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver", 1998

[7] S. Floyd, M. Handley and J. Padhye, "A Comparison of Equation-Based and AIMD Congestion Control", May 2000

[8] V. Tsaoussidis, H. Badr and R. Verma, "Wave and Wait Protocol: An energy-saving Transport for Mobile IP-Devices", In Proceedings of the 7th International Conference on Network Protocols, Oct 1999

[9] V. Tsaoussidis, A. Lahanas and C. Zhang, "The Wave & Probe Communication Mechanisms", Journal of Supercomputing, Kluwer Academic Publishers, September 2001

저자소개

노 경 택

현재 서울보건대학 사무자동화과 조교수

한국OA학회 논문지 제7권 제2호 참조

이 기 영

현재 서울보건대학 사무자동화과 부교수

한국OA학회 논문지 제3권 제4호 참조