

사물인터넷에서 객체전송지연을 계산하기 위한 수리적 모델링 및 휴리스틱 알고리즘의 개발

이용진*

한국교육대학교 기술교육과 교수

Analytical Modelling and Heuristic Algorithm for Object Transfer Latency in the Internet of Things

Yong-Jin Lee*

Professor, Dept. of Technology Education, Korea National University of Education

요약 본 논문은 평균 객체 전송 지연 시간에 대한 기존의 모델들을 하나의 프레임워크로 통합하고 실제 계산 경험을 통해 결과를 분석하는 것을 목표로 한다. 해석적 객체 전송 지연 시간 모델은 다중 패킷 손실과 작은 혼잡제어 윈도우로 인해 빠른 재전송이 불가능한 멀티홉 무선 네트워크를 위시한 사물 인터넷(IoT) 환경을 가정한다. 이 모델은 또한 초기 혼잡 윈도우 크기와 하나의 혼잡 윈도우에서의 다중 패킷 손실을 고려한다. 성능평가에 의하면, 전송 객체 크기와 패킷 손실률이 작은 경우 평균 객체 전송 지연의 하한값과 상한값은 거의 동일하다. 그러나 패킷 손실률이 커지면 초기 혼잡 윈도우의 크기와 왕복 시간이 평균 객체 전송 지연의 상·하한값에 영향을 미치는 것으로 나타났다.

주제어 : 객체 전송 지연, TCP 혼잡제어, 패킷 손실, 사물인터넷

Abstract This paper aims to integrate the previous models about mean object transfer latency in one framework and analyze the result through the computational experience. The analytical object transfer latency model assumes the multiple packet losses and the Internet of Things(IoT) environment including multi-hop wireless network, where fast re-transmission is not possible due to small window. The model also considers the initial congestion window size and the multiple packet loss in one congestion window. Performance evaluation shows that the lower and upper bounds of the mean object transfer latency are almost the same when both transfer object size and packet loss rate are small. However, as packet loss rate increases, the size of the initial congestion window and the round-trip time affect the upper and lower bounds of the mean object transfer latency.

Key Words : Object Transfer Latency, TCP Congestion Control, Packet Loss, IoT

1. 서론

객체 전송 지연은 종단 간 지연(end-to-end delay)에 영향을 미치는 인터넷의 많은 측정값들 중 하나이다.

일반적으로 객체 전송 지연은 TCP 혼잡 제어 메커니즘뿐만 아니라 링크의 전송 속도, 데이터 크기, 그리고 전송 시간의 영향을 받는다. TCP 혼잡 제어 메커니즘의 일반적인 기능은 느린 시작(slow start), 혼잡 회피, 시간

이 논문은 한국교육대학교 2019학년도 연구년교수 학술지원비 지원을 받아 수행한 연구의 결과임.

*교신저자 : 이용진(lvj@knu.ac.kr)

접수일 2020년 8월 7일 수정일 2020년 8월 27일 심사완료일 2020년 9월 4일

초과, 그리고 빠른 재전송이다[1].

TCP를 사용한 객체 전송 지연의 해석적 모델들[2,3]은 광대역 네트워크를 가정했기 때문에 본 연구가 고려하는 사물인터넷(IoT) 환경[4] 등의 협대역 네트워크에는 적용할 수 없다. 왜냐하면, 멀티홉 무선 네트워크를 포함한 협대역 환경에서는 혼잡 윈도우의 크기가 매우 작아서 데이터의 빠른 재전송이 이루어지기 어렵기 때문이다[5].

Lee[6]는 느린 시작 단계에서 객체의 전송지연을 구하였으나 성능 평가를 제시하지 않았다. [7]에서는 혼잡 회피 단계를 고려한 전송 지연의 하한 시간(LB: lower bound)을 구하였다. 하한 시간은 느린 시작 동안 마지막 윈도우에서 모든 패킷 손실이 일어나는 경우의 전송지연이다. [8]에서는 이웃 경로 선택 문제를 풀기 위해 경로 전송 시간 함수를 제시하였다. 이 함수는 패킷 손실이 전 기간에 걸쳐 균등하게 분산될 때를 가정하였으므로 전송 지연의 상한 시간(UB: upper bound)을 구한 것으로 간주할 수 있으나 구체적인 성능 평가 결과를 제시하지 않았다.

따라서 본 논문의 목적은 이전의 모델들에 대해 동일한 프레임워크를 사용한 성능평가를 통해 전송지연의 하한 시간(LB: lower bound)과 상한 시간(UB: upper bound)의 차이를 분석하는 데 있다.

2. 객체전송지연 모델

이 장에서는 기존의 평균 객체 전송 지연에 대한 모델들[6-8]을 하나의 프레임워크 내에서 통합적으로 서술한다.

먼저, 모델을 단순화하기 위해 전송 객체의 크기가 동일하고 패킷이 혼잡 윈도우의 크기 단위로 전송된다고 가정하자. 이 가정의 배경은 다음과 같다.

첫째, 객체의 크기가 동일하지 않고 임의의 확률 분포를 갖는 경우를 고려할 수 있다. 인터넷에서의 객체의 크기는 헤비-테일 분포(heavy-tailed distribution)의 특성을 갖는 파레토 분포 (Pareto distribution)[9-11]를 따른다. 모델링을 위해 전송 객체의 크기(θ)를 파레토 분포의 평균으로 가정하더라도 θ 값은 상수가 된다. 따라서 전송 객체의 크기가 동일한 경우의 수학적 모델링과 같아진다. 참고로, 파레토 분포의 확률 분포($f(x)$)와 평균($E(x)$)은 식 (1)과 같다. η 와 ν 는 각각 형상(shape)과 위치(position) 파라미터이다[12].

$$f(x) = \frac{\eta v^\eta}{x^{\eta+1}} (x > v; \eta > 0) \quad (1)$$

$$E(x) = \theta = \frac{\eta v}{\eta - 1} (\eta > 1)$$

둘째, 실제 인터넷에서 객체는 크기와 관계없이 트랜스포트 계층에서 혼잡 윈도우 크기로 분할되어 패킷 단위로 전송된다.

이제, 전송 객체의 크기를 θ 바이트, 최대 송신 세그먼트 크기를 S 바이트로 가정하자. 그러면, 객체에 대해 전송할 패킷 수는 $N = \theta/S$ 이다. 패킷 손실 확률이 p 일 때 예상되는 총 패킷 손실 수는 이항 분포에 의해 $a = Np$ 이다.

협대역 네트워크에서는 빠른 재전송과 빠른 복구가 불가능하다. 따라서 모든 패킷 손실은 느린 시작 또는 혼잡 회피 기간 동안에 발생한다.

임의의 k 번째 패킷 손실이 느린 시작 단계에서 발생했을 때, 평균 객체 전송 지연은 식 (2)와 같다[6].

$$\xi(\text{slow} : k) = T_{\text{slow}}(Y) + Y \times S/\mu + \text{Timeout} \quad (2)$$

여기서 Y 는 전송 데이터의 양, S 는 최대 송신 세그먼트의 크기, μ 와 rtt 는 각각 송수신자 사이의 링크 대역폭과 왕복 시간이다. Timeout 은 $3/2 \times rtt$ 이며 조정이 가능하다.

$T_{\text{slow}}(Y)$ 는 전송 데이터의 양이 Y 이고, 초기 윈도우 크기가 W 일 때의 느린 시작 시간으로 식 (3)과 같다.

$$T_{\text{slow}}(Y) = \rho(rtt + S/\mu) - (2^\sigma - 1)W \times S/\mu \quad (3)$$

$$\text{where } \rho = \min(\sigma, \epsilon - 1)$$

초기 윈도우 크기를 고려하고 객체가 무한대의 세그먼트들을 갖는다고 가정하면, 수신자에서 기다리는 패킷의 수(σ)는 식 (4)와 같다[13].

$$\begin{aligned} \sigma &= \max[\gamma : S/\mu + rtt - 2^{\gamma-1}W \times S/\mu \geq 0] \quad (4) \\ &= \lceil \log_2(1 + \mu \times rtt/S) \rceil + 1 - \log_2 W \end{aligned}$$

객체의 전송을 완료하기 위해서 ϵ 개의 윈도우가 필요하다고 하면, ϵ 는 식 (5)에서와 같이 전송 데이터의 양(Y)과 초기 윈도우 크기(W)의 향으로 표현된다.

$$\begin{aligned} \epsilon &= \min[\gamma : W(2^0 + 2^1 + \dots + 2^{\gamma-1}) \geq Y] \quad (5) \\ &= \lceil \log_2(1 + Y/W) \rceil \end{aligned}$$

다음으로, 혼잡 회피 단계 중에 k 번째 패킷 손실이 발생할 때의 평균 객체 전송 지연은 임계값 (R_k)까지 전송 가능한 최대 패킷의 개수(Ψ_k)의 느린 시작 시간, 추가되는 $(\delta-1)$ 개의 왕복 시간, 손실 이전에 전송되는 예상 패킷 개수(O_k)의 전송 시간, 그리고 재전송 타임아웃의 합으로 식 (6)과 같다[7].

$$\xi(\text{con} : k) = T_{\text{slow}}(\Psi_k) + (\delta-1)rtt + O_k S/\mu + \text{Timeout} \quad (6)$$

패킷들은 $W, 2 \times W, 4 \times W, 8 \times W, \dots (W=1,2,3,4)$ 순으로 전송되므로 Ψ_k 는 식 (7)과 같이 된다. 초기 느린 시작의 임계값(R_1)은 임의의 큰 값(∞)으로 설정되므로 Ψ_1 역시 ∞ 이다.

$$\Psi_k = \begin{cases} 2^{\lceil \log_2 \frac{R_k}{2} \rceil} - W & \text{if } R_k = 2^j \\ 2^{\lceil \log_2 \frac{R_k}{2} \rceil} - W + R_k & \text{if } R_k \neq 2^j \end{cases} \quad (7)$$

식 (7)에서 k 번째 임계값($R_k: k \geq 2$)은 식 (8)과 같다.

$$R_k = \max(Z/2, 2S) \quad (8) \\ k = 2, 3, \dots, \alpha$$

여기서, Z 는 전송되었지만 아직 응답되지 않은 데이터의 양이다.

패킷 손실률이 p 이고, k 번째 패킷 손실 이전에 전송된 데이터의 양이 N_k 일 때 O_k 는 식 (9)와 같이 표현된다.

$$O_k = [1 - (1-p)^{N_k}] / p + (1-p)^{N_k} + 1, \quad k = 1, 2, \dots, \alpha \quad (9)$$

혼잡 윈도우는 왕복 시간당 S 씩 증가하며, 각 혼잡 윈도우는 $1 \times rtt$ 가 걸리는 응답이 필요하다. 따라서 우리는 식 (6)에서 k 번째 패킷 손실까지 필요한 rtt 의 개수(δ)를 결정해야 한다. δ 는 식 (10)을 만족하는 가장 큰 값이다.

$$\sum_{j=0}^{\delta} (R_k + j) \leq O_k - \Psi_k \quad (10)$$

Timeout 은 $3/2 \times rtt$ 이며 조정이 가능한 값이다.

위의 식들을 사용하여 α 개의 패킷 손실이 느린 시작 또는 혼잡 회피 중에 처리된 이후에도 데이터가 전송되지 못하고 남아 있을 수 있다. 이 경우 패킷 손실은 없으므로 식 (2)와 식 (6)에서의 타임아웃은 필요하지 않다.

따라서 느린 시작 단계의 전송시간, $Q_{\text{slow}}(N_{\alpha+1} \leq \varphi_{\alpha+1})$ 또는 혼잡 회피 단계의 전송시간, $Q_{\text{con}}(N_{\alpha+1} > \varphi_{\alpha+1})$ 을 계산하면 된다. $Q_{\text{slow}}(N_{\alpha+1})$ 및 $Q_{\text{con}}(N_{\alpha+1})$ 는 식 (11)에 의해 얻는다.

$$\begin{aligned} & \text{if } N_{\alpha+1} \leq \Psi_{\alpha+1}, \\ & Q_{\text{slow}}(N_{\alpha+1}) = T_{\text{slow}}(N_{\alpha+1}) + N_{\alpha+1} S/\mu \\ & \text{else} \\ & Q_{\text{con}}(N_{\alpha+1}) = T_{\text{slow}}(\Psi_{\alpha+1}) + (\delta-1)rtt + N_{\alpha+1} S/\mu \end{aligned} \quad (11)$$

패킷 손실률(p)이 0일 때, 데이터 전송은 느린 시작 단계($N < \varphi_1 (= \infty)$ 일 경우)에서 완료된다. 따라서 평균 객체 전송 지연은 식 (12)와 같다.

$$\begin{aligned} \xi(\text{slow} : p = 0) &= Q_{\text{slow}}(N) \\ &= T_{\text{slow}}(N) + NS/\mu, \quad \text{if } N < \varphi_1 (= \infty) \end{aligned} \quad (12)$$

패킷 손실 이후, α 개의 패킷들은 마지막 단계에서 오류 없이 느린 시작이나 혼잡회피 도중에 전송될 수 있다. 따라서 객체 크기가 θ 이고 송신자 MSS가 S 일 때, 객체 전송 지연의 하한값(LB)은 식 (13)으로 주어진다[7]. $p=0$ 일 때 식 (13)은 $\alpha=0, \gamma=1, \text{Timeout}=0$ 이기 때문에 식 (12)와 같아진다.

$$LB(S; \theta) = T_{\text{slow}}(N) + \gamma Q_{\text{slow}}(\alpha) + (1-\gamma) Q_{\text{con}}(\alpha) + (N+\alpha) S/\mu + \text{Timeout} \quad (13)$$

where $\gamma = 0$ or 1

식 (2), 식 (6)과 식 (11)로부터, $N = \theta/S$ 일 때 객체 전송 지연의 상한값(UB)은 식 (14)와 같다[8].

$$UB(S; \theta) = \sum_{k=1}^{\alpha} \beta \xi(\text{slow} : k) + (1-\beta) \xi(\text{con} : k) + \gamma Q_{\text{slow}} + (1-\gamma) Q_{\text{con}} \quad (14)$$

where $\beta = 0$ or 1 and $\gamma = 0$ or 1

3. 성능평가

2장의 모델에 의해 평균 객체 전송 지연의 LB와 GBN (Go-back-N)과 SACK (Selective Acknowledgement)

재전송기법[14,15]에 대해 UB를 계산하는 다음의 알고리즘을 구현할 수 있다.

알고리즘: 평균 객체전송 지연계산

입력: 전송객체 크기(θ), MSS(S), 패킷 손실률(p), 초기 윈도우크기(W), 링크 전송률(μ), 왕복시간(RTT)

출력: 전송지연 하한($LB(S, \theta)$), 상한($UB(S, \theta)$)

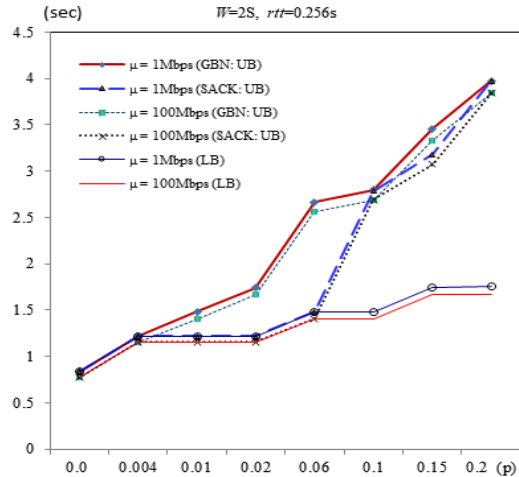
임시변수: $\tau = 0$ (지연 누적)

- 01: BEGIN
- 02: 전송 객체에 포함된 패킷개수($N=\theta/S$) 계산
- 03: 손실 패킷의 기댓값 ($a=Np$) 계산
- 04: for each 손실패킷 ($k=1$ to a)
- 05: 재전송을 포함한 전송 패킷 개수(Y) 계산
- 06: if (패킷손실 위치 == 느린시작 단계)
- 07: $\tau = \tau +$ 느린시작 단계의 지연 (식 (2))
- 08: else (패킷손실 위치 == 혼잡회피 단계)
- 09: $\tau = \tau +$ 혼잡 회피 단계의 지연 (식 (6))
- 10: end if
- 11: $\tau = \tau +$ 마지막 단계의 지연 (식 (11))
- 12: end for
- 13: if (모든 패킷 손실이 마지막 윈도우에서 발생)
- 14: τ 를 이용하여 $LB(S, \theta)$ 계산 (식 (13))
- 15: else if (패킷 손실이 전체 구간에서 균등발생)
- 16: for each GBN and SACK
- 17: τ 를 이용하여 $UB(S, \theta)$ 계산 (식 (14))
- 18: end for
- 19: else
- 20: end if
- 21: END

일반적으로 웹에서의 평균 객체 크기는 13.5KB인 것으로 알려져 있다. 송신자 MSS(S)는 WAN의 경우 536B이다[13]. S 가 1095B보다 작거나 같은 경우, W 는 최대 4S로 설정된다[1].

먼저, 왕복 시간(rtt)과 초기 혼잡 윈도우(W)를 각각 256ms와 2S로 고정하고, 패킷 손실률(p)을 0에서 0.2까지 변경했다.

Fig. 1은 링크 전송률(μ)이 1Mbps와 100Mbps일 때 GBN과 SACK에 대한 평균 객체 전송 지연의 하한값(LB)과 상한값(UB)을 나타낸다.

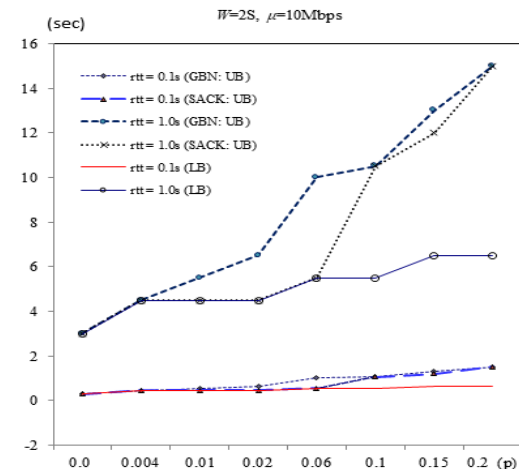


[Fig. 1] Mean object transfer latency for varying the link speed(μ)

그림에서 보듯이 100Mbps와 1Mbps에서 SACK의 두 개의 전송 지연 시간은 100Mbps에서의 GBN보다 작다. 특히 링크의 전송 속도를 1Mbps에서 100Mbps로 늘렸음에도 평균 전송 지연은 크게 줄지 않는다.

1Mbps와 100Mbps의 LB와 UB(SACK)는 $p \leq 0.06$ 일 때 거의 같다. 그러나 모든 경우의 LB와 UB의 차이는 $p > 0.06$ 일 때 빠르게 증가한다.

다음으로, 링크 전송률(μ)과 초기 윈도우(W)를 각각 10Mbps와 2S로 고정하고, 패킷 손실률(p)은 0에서 0.2까지 변화시켰다.



[Fig. 2] Mean object transfer latency for varying round trip time(rtt)

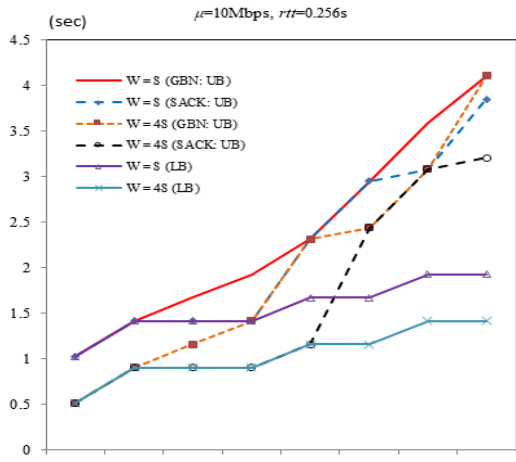
Fig. 2는 왕복 시간(rtt)이 0.1초와 1.0초일 때 GBN과 SACK에 대한 평균 객체 전송 지연의 LB와 UB를 나타낸다.

그림에서 볼 수 있듯이, 평균 객체 지연은 GBN과 SACK에 관계없이 왕복 시간의 영향을 많이 받는다. 그 이유는 왕복 시간이 상대적으로 클 때 느린 출발 시간이 크게 늘어나기 때문이다.

$p \leq 0.6$ 이고 $rtt=0.1$ 초일 때의 LB는 $rtt=1$ 초일 때의 LB보다 크다. $rtt=0.1$ 초일 때와 $rtt=1$ 초일 때의 LB는 $rtt=0.1$ 초일 때의 UB(SACK)와 $rtt=1$ 초일 때의 UB(SACK)와 같다. 그러나 $p > 0.6$ 일 때, LB와 UB의 차이는 급격히 증가한다.

마지막으로, 링크의 전송률(μ)과 왕복 시간(rtt)을 각각 10Mbps와 0.256초로 고정시킨 후 패킷 손실률(p)을 0에서 0.2로 변화시켰다.

Fig. 3은 초기 윈도우 크기(W)가 5와 45일 때 GBN 및 SACK에 대한 평균 객체전송 지연의 LB 및 UB를 보여준다.



[Fig. 3] Mean object transfer latency for varying initial window (W)

그림에서 보듯이, $W=45$ 일 때 SACK의 평균 전송 지연이 가장 작고 $W=5$ 일 때 GBN의 평균 전송 지연이 가장 크다. 특히 GBN의 평균 전송 지연($W=45$)은 $p < 0.1$ 의 경우 SACK ($W=5$)보다 작지만, 전자는 $p < 0.1$ 의 경우 후자보다 크다. $W=5$ 일 때의 LB와 $W=5$ 일 때의 UB(SACK)는 거의 같다. $W=45$ 일 때의 LB와 $W=45$ 일 때의 UB(SACK)도 $p \leq 0.06$ 일 때는 거의 같다.

그러나 $W=5$ 와 45일 때의 UB는 $p > 0.02$ ($W=5$)일 때

와 $p > 0.06$ ($W=45$)일 때 모든 LB보다 커지므로, 큰 W 를 가진 SACK은 최소 지연의 UB를 갖는다고 말할 수 있다.

4. 결론

본 연구는 다중 패킷 손실과 멀티홉 무선 네트워크를 포함한 사물인터넷(IoT) 환경에서 평균 객체 전송 지연에 대한 기존의 하한값 및 상한값 모델을 하나의 프레임워크에 통합하고 계산 경험 및 분석을 제시한 것이다.

계산 경험에 따르면 객체 크기와 패킷 손실률이 작을수록 평균 객체 전송 지연의 하한값과 상한값은 거의 같아지고, 특히 선택적 재전송 기법(SACK)을 사용했을 때의 상한값과 하한값의 차이는 매우 작았다. 그러나 패킷 손실률이 증가할수록 초기 혼잡 윈도우의 크기와 왕복 시간이 평균 전송시간의 상·하한값에 영향을 미친다.

이 모델은 단일 사용자를 가정하고, 모델을 단순화하기 위해 하한값과 상한값이라는 두 가지 극단적인 패킷 손실 사례만을 다루었다. 향후 연구에서는 다중 사용자 환경에서 버스트 오류를 고려하면서 보다 정확도를 높이는 모델의 개발이 기대된다.

ACKNOWLEDGMENTS

이 논문은 한국교원대학교 2019학년도 연구년교수 학술지원비 지원을 받아 수행한 연구의 결과임.

REFERENCES

- [1] M.Allman, V.Paxson and E.Blanton, TCP Congestion Control, RFC-2581, 2009.
- [2] J.Padhye, V.Firoiu, D.Towsley and J.Kurose, "Modeling TCP Reno performance: A Simple Model and its Empirical Validation," ACM Transactions on Networking, Vol.8, pp.133-145, 2000.
- [3] J.Jiong, Z.Shu-jing and Qi-gang, "An Adapted Full Model for TCP Latency," IEEE TENCON Conference, pp.801-804, 2002.
- [4] R.Buyya and A.V.Dastjerdi, Internet of Things-Principles and Paradigms, Elsevier, 2016.
- [5] D.Oliveria and R.Braun, "A Dynamic Adaptive Acknowledgement Strategy for TCP over Multi-hop Wireless Networks," IEEE INFOCOM Conference, pp.1863-1874, 2005.

- [6] Y.Lee, "Analytical Model for Mean Web Object Transfer Latency Estimation in the Narrowband IoT Environment," Journal of The Korea Internet of Things Society, Vol.1, pp.1-4, 2015.
- [7] Y.Lee, "Lower Bound for Mean Object Transfer Latency in the Narrowband IoT Environment," International Journal of Applied Engineering Research, Vol.12, pp.3365-3369, 2017.
- [8] Y.Lee, "A Path Selection Model Considering Path Latency in the Communication Network with Geographically Correlated Failures," ICIC Express Letters, Vol.13, pp.789-796, 2019.
- [9] M.Hachol-Balter, Performance Modeling and Design of Computer Systems, pp.352-355, Cambridge Press, 2013.
- [10] A.Leon-Garcia, Probability, Statistics, and Random Processes for Electrical Engineering, pp.187-188, Pearson, 2009.
- [11] S.Ross, Introduction to Probability and Statistics for Engineers and Scientists, pp.179-182, Hong-Reung, 2017.
- [12] W.Stallings, High-speed Networks, pp.191-192, Prentice-Hall, 1998.
- [13] J.Kurose and K.Ross, Computer Networking, pp.272-279, Prentice-Hall, 2003.
- [14] B.A.Forouzan, Data Communications and Networking, 5th Edition, McGrawHill, pp.733-747, 2013.
- [15] L.Zhao, C.Ladas and R.Edwards, "A Selective-ARQ Scheme for Improved TCP and UDP Performance over Wireless Networks," Proceedings of AIMEE, 2017.

이 용 진(Yong-Jin Lee)

[종신회원]



- 1995년 2월 : 고려대학교 전산과 학과 (이학박사)
- 1995년 3월 ~ 2005년 8월 : 우송대학교 컴퓨터정보학과 교수
- 2005년 9월 ~ 현재 : 한국교원대학교 기술교육과 교수

〈관심분야〉

인터넷 기술, 모바일 컴퓨팅, 성능 평가