ICIC Express Letters ICIC International ©2019 ISSN 1881-803X
Volume **13**, Number **9**, September **2019** pp. **789–796**

A PATH SELECTION MODEL CONSIDERING PATH LATENCY IN THE COMMUNICATION NETWORK WITH GEOGRAPHICALLY CORRELATED FAILURES

Yong-Jin Lee

Department of Technology Education
Korea National University of Education
250 Taesungtapyon-Ro, Heungduk-Gu, Cheongju 28173, Korea
lyj@knue.ac.kr

Received February 2019; accepted April 2019

Abstract. *This paper presents path selection model in the communication network where geographically correlated failures take place. We consider both the proximity factor and the sharing factor while minimizing the path latency for effective path selection. In order to estimate the path latency, we present the iterative analytical model and algorithm. The proposed path latency model assumes the multiple packet losses and the narrowband environment including multi-hop wireless network, where fast retransmission is not possible due to small window. Our path latency algorithm also considers the initial congestion window size and the multiple packet loss in one congestion window. Computational experiments show that for small packet loss rate, our algorithm finds the least latency and for a given packet loss rate, round trip time and initial congestion window size mainly affect the path latency. The proposed path selection model and path latency algorithm are applied to building the sustainable communication network.*
Keywords: Path selection, Path latency, Congestion control, Geographically correlated failure

1. **Introduction.** Path or neighbor selection [1] is one of significant challenges when the abrupt disasters – especially, geographically correlated failures occur on the communication network. The improper selection of neighbor node in which multiple geographically correlated failures take place may make it impossible to transfer the important data between source node and destination node.

When the physical path information is known, Neumayer et al. [2] explore the impact of the geographically correlated failures, but do not analyze the impact to the communication networks. Kim and Venkatasubramanian [3] have proposed the proximity-aware neighbor selection method using the Euclidean distances between every physical node. Their simulation has shown that proximity-aware neighbor selection techniques can disseminate data to over 80% of reachable end destinations.

However, only proximity consideration may cause to choose a neighbor sharing a common router with other nodes. A geographically correlated failure that occurs at the sharing router may lead to the cutting off of communication. In addition, they take account of latency as a tie-breaker only when many nodes have the same minimum correlation (proximity), and therefore, cannot select the path with the least latency [4].

The path latency [5] is one of many important measures in the communication network since it is the main factor to affect end-to-end latency. Typically, path latency is affected by transfer object (file) size and transmission time according to transmission rate of link as well as by TCP congestion control mechanism. The common functions of TCP congestion control mechanism are slow start, congestion avoidance, timeout, and fast retransmission [6].

DOI: 10.24507/icicel.13.09.789

Previous related works [7-9] on analytical models of path latency over TCP assumed wideband network; therefore, they are not able to be applied directly to the narrowband network environment, which this paper considers. That is why narrowband environment including multi-hop wireless network does not allow fast retransmission of data due to the very small size of window [10].

Lee [11,12] found the lower bound of path latency when all the packet losses occur in the last window during slow start phase for the transfer completion. However, the lower bound is an extreme value, and we cannot use it in the path selection model. Thus, the contribution of our research is to propose more realistic model estimating path latency when packet losses are equally spread over the time. We can find path latency very easily by using our iterative algorithm based on the packet loss rate, the initial congestion window size, the link bandwidth, and round trip time (RTT).

Computational experiences show that when the packet loss rate is small (below 6%) at the given object size, the path latency is the least for selective acknowledgement (SACK) transmission; however, it increases sharply as the packet loss rate increases.

Our path selection model and path latency algorithm improving the previous related works [3,4,11,12] can be used in selecting the effective path with the least latency while avoiding the area with the geographically correlated failures; therefore, sustainable communication is maintained.

2. **Path Selection Model.** Previous path (neighbor) selection model [3] utilizes the proximity factor which indicates the closeness between two paths. Initially, proximity factors for every two paths from source node (X) to destination nodes (m, n) are set to zero. Euclidean distances between every node pair included in two paths are computed. The number that the Euclidean distance is less than the target distance (threshold) is counted, and proximity factor is increased by that number. However, selection of neighbor with the least proximity only may lead to the entire communication cut-off in the area where the geographically correlated failures occur.

Selecting the common router with the least proximity factor as a neighbor node in the failure area causes vulnerability to the entire communication shutdown. Thus, in such a case, we have to avoid sharing a common router on the path even if the proximity factor is very small.

We introduce sharing factor indicating that two paths share common router on each path. Initially, sharing factors for every two paths from source node (X) to destination nodes (m, n) are set to zero like proximity factor. If Euclidean distance between any two nodes pair on two paths is equal to zero, then we comprehend that two nodes included in the different two separate paths share common router. In such a case, we increase sharing factor by one.

Additionally, when there are several nodes having the same least proximity factor and sharing factor, path latency is used as a tie-breaker, and the node with the least latency is selected as a neighbor. Path latency estimation algorithm when object size and several parameters are given is discussed in Section 3.

Figure 1 presents path selection algorithm considering proximity factor, sharing factor, and path latency by modifying previous algorithms [3,4].

3. **Path Latency Modeling.** In the path selection algorithm of Figure 1, we used the path latency as a selection criterion. In this section, we describe how to estimate the path latency. To build a simple analytical model, it is assumed that packets are transmitted in units of the size of the congestion window. By receiving the transfer object size L bytes and the sender maximum segment size (SMSS) S bytes from Algorithm 1 of Figure 1, the number of packets included in the object is $N = \lceil L/S \rceil$. If the probability of packet loss

ALGORITHM 1. Path selection algorithm

INPUT: $N_{\max}$: target number of neighbor nodes (≥ 2), $D_{\max}$: target distance of a path
    X: source node to want find neighbors (path)
    M: set of random nodes
    P_m, P_n: physical path from X to m, n; where $P_m := \{R_{m1}, R_{m2}, \ldots, R_{mr}\}$,
    where $R_{mr} := r^{\text{th}}$ physical node on P_m
    latency(P_m): latency of path from X to node m ($\in M$), $X \neq m$
    L: transfer object size on P_m
    S: sender (X) maximum segment size on P_m
OUTPUT: Neighbor (path) from X
BEGIN
for all P_m and P_n, $X \in M$, $m \in M$, $n \in M$, $X \neq m \neq n$ **do**
  CALL path_latency function (P_m, P_n, L, S) of **ALGORITHM 2** in Section 3 and obtain latency for P_m, P_n;
  $\mathit{ProximityFactor}(P_m, P_n) = 0$;
  $\mathit{SharingFactor}(P_m, P_n) = 0$;
    for all $R_{mi} \in P_m$ **do**
      for all $R_{nj} \in P_n$ **do**
        if $D(R_{mi}, R_{nj}) < D_{\max}$ **then** $//D(R_{mi}, R_{nj})$ = Euclidean distance between R_{mi} and R_{nj}
          $\mathit{ProximityFactor}(P_m, P_n) = \mathit{ProximityFactor}(P_m, P_n) + 1$;
            if $D(R_{mi}, R_{nj}) = 0$ **then**
              $\mathit{SharingFactor}(P_m, P_n) = \mathit{SharingFactor}(P_m, P_n) + 1$;
            end if
        end if
      end all
    end all
end all
for all P_m and P_n, $X \in M$, $m \in M$, $n \in M$, $X \neq m \neq n$ **do**
      Ascending sort $\mathit{ProximityFactor}(P_m, P_n)$ and $\mathit{SharingFactor}(P_m, P_n)$
        with primary key = *ProximityFactor*;
end all
$\mathit{cnt} = 0$;   // the number of selected neighbors
while ($\mathit{cnt} < N_{\max}$)
  Select neighbor with the least *ProximityFactor* and the least corresponding *SharingFactor*;
  if there are two more neighbors satisfying the above condition, select the neighbors (m)
    with the least sum of *latency*(P_m);
$\mathit{cnt} = \mathit{cnt} + 1$;
end while
END

FIGURE 1. Path selection algorithm

is equal to p, the expected total number of lost packets by binomial distribution is equal to $\lceil Np \rceil$.

Thus, packet loss occurs during a slow start phase or congestion avoidance phase. Maximum number of packets to be transmitted (M_k, $k = 1, 2, \ldots, \varepsilon$) until the threshold ($\theta_k$, $k = 1, 2, \ldots, \varepsilon$) at which congestion avoidance begins and the expected number of packets sent before packet loss (E_k, $k = 1, 2, \ldots, \varepsilon$) can be compared to determine where packet loss occurs. That is, if $E_k \leq M_k$, a packet is lost during the slow start phase; otherwise, the packet is lost during the congestion avoidance phase.

For the data to be transmitted before the k^{th} packet loss, N_k ($N_k = N$ for $k = 1$), the expected number of packets sent including the lost packet until the packet loss is given by

$$E_k = \frac{1 - (1-p)^{N_k}}{p} + (1-p)^{N_k} + 1 \quad k = 1, 2, \ldots, \varepsilon \tag{1}$$

We consider the case such that the amount of retransmission can increase, even if we consider the multiple losses within one congestion window. In this case, the estimation of expected number of packets sent before the packet loss (E_k) in Equation (1) is evenly distributed over the transfer time. That is, packet errors are spread over time depending on the data amount to be transferred and the packet loss.

The initial value of congestion window (δ) is suggested as $2S$, $3S$, and $4S$. Initial threshold (θ_1) is set arbitrarily high (∞) and θ_k ($k \geq 2$) are set to

$$\theta_k = \max\left(\frac{F}{2}, 2S\right) \quad k = 2, 3, \ldots, \varepsilon \tag{2}$$

Here, F represents the data amount which has been sent but not yet acknowledged. In our paper, taking account of the worst case, F has been set to the congestion window size.

M_k ($k = 1, 2, \ldots, \varepsilon$) is the maximum number of packets to be sent until θ_k. Since $\theta_1 = \infty$, M_1 is also equal to ∞. Thus, E_1 is less than M_1. This shows that first packet loss ($k = 1$) must occur during slow start phase. Packets are transmitted in the manner δ, 2δ, 4δ, 8δ, $\ldots$ ($\delta = 2, 3, 4$) for $k = 1$ and δ, 2δ, 4δ, 8δ, $\ldots$ ($\delta = 1$) for $k \geq 2$, respectively. Thus, M_k is given by

$$M_k = \begin{cases} 2^{\left\lceil \log_2^{\theta_k+1} \right\rceil - \delta} & \text{if } \theta_k = 2^j \\ 2^{\left\lceil \log_2^{\theta_k+1} \right\rceil - \delta + \theta_k} & \text{if } \theta_k \neq 2^j \end{cases} \tag{3}$$

In order to completely send the object, we need α windows. Generally, α is expressed in terms of transmission data amount (Y) and initial window size (δ).

$$\alpha = \min\left\{i : \left(2^0 + 2^1 + \cdots + 2^{i-1}\right)\delta\right\} \geq Y = \left\lceil \log_2\left(1 + \frac{Y}{\delta}\right)\right\rceil \tag{4}$$

Since E_k is sent until the k^{th} packet loss, the window number (α_k) including E_k is given by

$$\alpha_k = \left\lceil \log_2\left(1 + \frac{E_k}{\delta}\right)\right\rceil \quad \begin{cases} \delta = 2, 3, 4 & \text{for } k = 1 \\ \delta = 1 & \text{for } k \geq 2 \end{cases} \tag{5}$$

When the window number is equal to α_k, congestion window size (C_k) corresponding to the window number is given by

$$C_k = 2^{\alpha_k - 1}\delta \quad \begin{cases} \delta = 2, 3, 4 & \text{for } k = 1 \\ \delta = 1 & \text{for } k \geq 2 \end{cases} \tag{6}$$

The maximum amount of packets sent before the α_k^{th} window (Q_k) is represented by

$$Q_k = \left(\sum_{j=0}^{\alpha_k - 1} 2^j\right) \times \delta = \left(2^{\alpha_k} - 1\right) \times \delta \quad \begin{cases} \delta = 2, 3, 4 & \text{for } k = 1 \\ \delta = 1 & \text{for } k \geq 2 \end{cases} \tag{7}$$

By taking account of the initial congestion size (δ), the number of receiver stalls (β) when the object contains an infinite number of segments is given by

$$\beta = \max\left\{i : \frac{S}{\mu} + rtt - 2^{i-1} \times \frac{\delta S}{\mu} \geq 0\right\} = \left\lfloor \log_2\left(1 + \frac{\mu \times rtt}{S}\right)\right\rfloor + 1 - \log_2 \delta \tag{8}$$

Here μ and rtt indicate the link bandwidth and round trip time between source and destination, respectively. Therefore, when the transmission data amount (Y) and the initial congestion window size (δ) are given, slow start time is represented by

$$T_{slow}^{Y} = \gamma\left(\frac{S}{\mu} + rtt\right) - \left(2^{\gamma} - 1\right) \times \frac{\delta S}{\mu} \tag{9}$$

where $\gamma = \min(\alpha - 1, \beta) = \min\left(\left\lceil \log_2\left(1 + \frac{Y}{\delta}\right)\right\rceil - 1,\ \left\lfloor \log_2\left(1 + \frac{\mu \times rtt}{S}\right)\right\rfloor + 1 - \log_2 \delta\right)$.

Table 1. Transmission, retransmission, and remaining data amount for GBN and SACK

Case	GBN			SACK		
	Y	R	N_{k+1}	Y	R	N_{k+1}
$E_k \leq Q_k \leq N_k, M_k$	Q_k	$Q_k - E_k$	$N_k - E_k$	Q_k	$cnt + 1$	$N_k - Q_k + cnt + 1$
$E_k \leq N_k \leq Q_k, M_k$	N_k	$N_k - E_k$	$N_k - E_k$	N_k	$cnt + 1$	$cnt + 1$
$E_k \leq M_k \leq Q_k, N_k$	M_k	$M_k - E_k$	$N_k - E_k$	M_k	$M_k - E_k$	$N_k - M_k + cnt + 1$

We now consider the transmission data amount (Y), retransmission data amount (R), and the remaining data amount for transmission before the next packet loss (N_{k+1}) when multiple packet loss occurs in one window after the k^{th} packet loss. We know that $E_k \leq M_k$, $E_k \leq N_k$, and $E_k \leq Q_k$ from Equations (2), (3), and (7). If the number of losses in one window is equal to cnt after E_k, there are three different cases as shown in Table 1. GBN and SACK represent Go-Back-N and selective acknowledgement as retransmission policy after timeout, respectively.

Therefore, path latency when the k^{th} packet loss occurs during slow start phase is the sum of slow start time of Y, transmission time of Y, and retransmission timeout as shown in Equation (10).

$$\Phi_k^{slow} = T_{slow}^{Y} + \frac{Y \times S}{\mu} + T_{out} \tag{10}$$

The retransmission timeout (T_{out}) is mostly given by $3/2 \times rtt$, which can be adjusted according to the actual environment. At the next step, we compute E_{k+1} in Equation (1) by using N_{k+1} given in Table 1. New slow start threshold, θ_{k+1} is given by

$$\theta_{k+1} = \max\left(\left\lceil \frac{C_k}{2} \right\rceil, 2S\right) \quad k = 2, 3, \ldots, \varepsilon \tag{11}$$

In addition, upon timeout, congestion window size must be set the loss window which is equal to one full-size segment. That is, δ must be set to S for $k \geq 2$.

We can know that the k^{th} packet is lost during congestion avoidance phase if $E_k > M_k$. At this time, E_k is composed of M_k sent until slow start threshold (θ_k) during slow start phase and $D(E_k - M_k)$ sent after threshold until the k^{th} packet loss during congestion avoidance phase. Congestion window is incremented by roughly S per round trip time (rtt) and each $cwnd$ needs acknowledgement, which requires one rtt in congestion avoidance phase. Thus, we have to determine the necessary number of rtt's (H) until the k^{th} packet loss. M is the largest value to satisfy Equation (12).

$$\sum_{j=0}^{H} (\theta_k + j) \leq D \tag{12}$$

Path latency when the k^{th} packet loss occurs during congestion avoidance phase is the sum of slow start time of M_k, additional $(H - 1)$ round trip time, transmission time of E_k, and retransmission timeout.

$$\Phi_k^{cong} = T_{slow}^{M_k} + (H - 1) \times rtt + \frac{E_k \times S}{\mu} + T_{out} \tag{13}$$

Here, T_{out} is $3/2 \times rtt$ and adjustable to real environment. The number of remaining data to be transmitted before the $(k+1)^{\text{th}}$ packet loss is N_{k+1} $(= N_k - E_k + 1)$ for both GBN and SACK. New slow start threshold, θ_{k+1} is given by

$$\theta_{k+1} = \max\left(\left\lceil \frac{\theta_k + H}{2} \right\rceil, 2S\right) \quad k = 2, 3, \ldots, \varepsilon \tag{14}$$

Since both fast retransmission and fast recovery are not possible, initial *cwnd* (δ) must be set to S for $k \geq 2$ after timeout.

After processing ε packet losses in either during slow start phase or congestion avoidance phase by the above model, there may still be data to be sent. At this point, since the remaining data ($N_{\varepsilon+1}$) is greater than zero and there is no longer any packet loss in Equations (10) and (13), a timeout (T_{out}) is not required. Therefore, we can simply find the transfer latency, T_{last}^{slow} during slow start phase (if $N_{\varepsilon+1} \leq M_{\varepsilon+1}$) or during congestion control phase (if $N_{\varepsilon+1} > M_{\varepsilon+1}$). $T_{last}^{slow}(N_{\varepsilon+1})$ and $T_{last}^{cong}(N_{\varepsilon+1})$ are given as follows:

$$\begin{aligned} T_{last}^{slow}(N_{\varepsilon+1}) &= T_{slow}^{N_{\varepsilon+1}} + \frac{N_{\varepsilon+1} \times S}{\mu} & \text{if } N_{\varepsilon+1} \leq M_{\varepsilon+1} \\ T_{last}^{cong}(N_{\varepsilon+1}) &= T_{slow}^{M_{\varepsilon+1}} + (H-1) \times rtt + \frac{N_{\varepsilon+1} \times S}{\mu} & \text{otherwise} \end{aligned} \tag{15}$$

To summarize Equations (10), (13), and (15), path latency for object with the number of packets, $N = \lceil L/S \rceil$ (object size = L and sender MSS = S) is

$$\Psi_L^S = \sum_{k=1}^{\varepsilon} \left[\rho \Phi_k^{slow} + (1-\rho)\Phi_k^{cong} \right] + \sigma T_{last}^{slow}(N_{\varepsilon+1}) + (1-\sigma) T_{last}^{cong}(N_{\varepsilon+1}) \tag{16}$$

where $\rho = 0$ or 1 and $\sigma = 0$ or 1.

4. Path Latency Algorithm.

Based on the model described and recent TCP congestion control standard (RFC-2581) [4], we propose an algorithm shown in Figure 2 to estimate the path latency. When the number of packets for an object is N, the complexity of each algorithm is $O(N)$. By applying Algorithm 2, we obtain Table 2 showing path latency when $\mu = 10$ Mbps, $rtt = 0.1$ sec, $\delta = 4S$ for varying packet loss rate ($p = 0$, $p = 0.01$, and $p = 0.05$) in GBN and SACK.

For the case of no packet loss ($p = 0$), path latencies for GBN and SACK are the same. However, all the path latencies for SACK become less than path latency for GBN as p increases. The reason is why GBN needs more retransmission (R) for the lost packets than SACK as shown in Table 2.

TABLE 2. Path latency when $\mu = 10$ Mbps, $rtt = 0.1$ sec, $\delta = 4S$

L (KB)	$p = 0.0$	$p = 0.01$		$p = 0.05$	
		GBN	SACK	GBN	SACK
1.35	0.001	0.017	0.017	0.017	0.017
13.5	0.027	0.053	0.042	0.090	0.058
135	0.127	0.299	0.280	0.789	0.664
1350	1.099	3.814	2.702	7.986	6.595

For more computational experiments, we firstly fixed round trip time (*rtt*) and initial congestion window (δ) for $k = 1$ as 256 ms and $2S$, respectively. And then, we changed packet loss rate (p) from 0 to 0.2. Two path latencies of SACK at 100 Mbps and 1 Mbps are less than GBN at 100 Mbps. In particular, it can be investigated that path latency does not significantly decrease even if the transmission speed of the link (μ) is increased from 1 Mbps to 100 Mbps. Secondly, we fixed the transmission rate of link (μ) and initial window (δ) as 10 Mbps and $2S$, respectively. We also varied packet loss rate (p) from 0 to 0.2. Path latency is greatly affected by *rtt* regardless of GBN and SACK. This is because the slow start time increases significantly when the round trip time is relatively large.

ALGORITHM 2. path_latency function (P_m, P_n, L, S)

FOR each P_m, P_n
INPUT: p: packet loss rate
rtt: round trip time
μ: link bandwidth
OUTPUT: Ψ_L^S: path latency for P_m
BEGIN
Compute the total number of packets included in an object, $N = \lceil L/S \rceil$
Compute the expected number of packet losses, $\varepsilon = \lceil Np \rceil$
Set $N_1 = N$, $\theta_1 = M_1 = \infty$;
Set $\Psi_L^S = 0$ and $k = 0$;
while (1)
  if $(k >= \varepsilon$ or $p = 0)$
    begin
      Compute $M_{\varepsilon+1}$ by using Equation (3)
      if $(N_{\varepsilon+1} \leq M_{\varepsilon+1})$
        Set $\Psi_L^S = \Psi_L^S + T_{slow}^{N_{\varepsilon+1}} + N_{\varepsilon+1} \times S/\mu$;
      else
        Set $\Psi_L^S = \Psi_L^S + T_{slow}^{M_{\varepsilon+1}} + (H-1) \times rtt + N_{\varepsilon+1} \times S/\mu$;
      end if
    end
  break;
  k++;
  Compute the expected number of packets sent until the packet loss (E_k) and the number of packets sent until slow start (M_k) by using Equations (1) and (3), respectively;
  if $(E_k \leq M_k)$
    begin
      if $(E_k \leq Q_k \leq N_k$ and $M_k)$ set $Y = Q_k$ and $N_{k+1} = N_k - E_k$; (for GBN)
        $N_{k+1} = N_k - Q_k + cnt + 1$; (for SACK)
      else if $(E_k \leq N_k \leq Q_k$ and $M_k)$ set $Y = N_k$ and $N_{k+1} = N_k - E_k$; (for GBN)
        $N_{k+1} = cnt + 1$; (for SACK)
      else if $(E_k \leq M_k \leq Q_k$ and $N_k)$ set $Y = M_k$ and $N_{k+1} = N_k - E_k$; (for GBN)
        $N_{k+1} = N_k - M_k + cnt + 1$; (for SACK)
      end if
      Compute the window number (α_k) and *cwnd* (C_k) using Equations (5) and (6), respectively;
      Set $\theta_{k+1} = \max(\lceil C_k/2 \rceil, 2S)$ by Equation (11);
      Set $\Psi_L^S = \Psi_L^S + T_{slow}^{Y} + Y \times S/\mu + 3/2 \times rtt$;
    end
  else
    begin
      Set $\Psi_L^S = \Psi_L^S + T_{slow}^{M_k} + (H-1) \times rtt + E_k \times S/\mu + 3/2 \times rtt$;
      Set $N_{k+1} = N_k - E_k + 1$;
      Set $\theta_{k+1} = \max(\lceil (\theta_k + H)/2 \rceil, 2S)$ by Equation (14);
    end
  end if
  end if
end while
Return path latency (Ψ_L^S) for P_m to **ALGORITHM 1**;
END

FIGURE 2. Path latency algorithm

5. **Conclusions.** This paper presents path selection model and algorithm in the communication network where geographically correlated failures take place. Our model deals with the proximity factor and the sharing factor simultaneously while minimizing the path latency. In order to utilize the path latency in the path selection model, we present an iterative algorithm to obtain the path latency in the narrowband network when packet losses are equally spread over the time.

Our path latency algorithm iteratively finds the latency based on the packet loss rate and the number of packets to be transmitted. It also considers the initial value of congestion window and multiple packet losses in one window which is especially useful for selective acknowledgement.

The proposed path latency algorithm for both GBN and SACK can easily find the path latency when the packet loss rate, object size, SMSS, RTT, and the link rate are given. Computational experiences show that the path latency is the least for SACK retransmission mechanism when the packet loss rate is small.

Our path selection and latency algorithm can be applied to selecting the effective path with the least latency in the communication network with geographically correlated failures in order to avoid the communication disruption. Future works include more accurate path selection model and path latency algorithm considering the probability distribution of burst errors in multiple user environment.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07045038).

REFERENCES

[1] J. Plachy, Z. Becvar and P. Mach, Path selection enabling user mobility and efficient distribution of data for computation at the edge of mobile network, *Computer Networks*, vol.108, pp.357-370, 2016.

[2] S. Neumayer, G. Zussman, R. Cohen and E. Modiano, Assessing the impact of geographically correlated network failures, *Proc. of IEEE Military Communications Conference*, San Diego, USA, p.16, 2008.

[3] K. Kim and N. Venkatasubramanian, Assessing the impact of geographically correlated failures on overlay-based data dissemination, *Proc. of IEEE Global Telecommunications Conference*, Miami, USA, pp.1-5, 2010.

[4] Y.-J. Lee, Neighbor selection considering path latency in overlay network, *Journal of Natural Science*, vol.2, pp.25-32, 2013.

[5] A. S. Asrese, S. J. Eravuchira, V. Bajpai, P. Sarolahti and J. Ott, Measuring web latency and rendering performance: Method, tools & longitudinal dataset, *IEEE Trans. Network and Service Management*, vol.16, pp.1-14, 2019.

[6] M. Allman, V. Paxson and E. Blanton, *TCP Congestion Control*, RFC-2581, IETF, 2009.

[7] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, Modeling TCP Reno performance: A simple model and its empirical validation, *ACM Trans. Networking*, vol.8, pp.133-145, 2000.

[8] N. Cardwell, S. Savage and Y. Anderson, Modeling TCP latency, *Proc. of IEEE INFOCOM Conference*, Tel Aviv, Israel, pp.1742-1751, 2000.

[9] J. Zhao, S. Zhang and Q. Zhou, An adapted full model for TCP latency, *Proc. of IEEE TENCON Conference*, Beijing, China, pp.801-804, 2002.

[10] D. Oliveria and R. A. Braun, A dynamic adaptive acknowledgement strategy for TCP over multihop wireless networks, *Proc of IEEE INFOCOM Conference*, Miami, USA, pp.1863-1874, 2005.

[11] Y.-J. 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.

[12] Y.-J. Lee, Least object transfer latency in the IoT with small congestion window, *Proc. of IEEE International Conference on Consumer Electronics*, Las Vegas, USA, pp.1-2, 2019.