

Push-N-Scheme with Timeout for Content Delivery of Social Networking Services

Kyungkoo Jun*

Abstract—Widely spreading smart devices have become an important information sharing channel in everyday life. In particular, social networking services (SNS) are the hub for content creation and sharing. Users post their contents on SNS servers and receive contents of interest. Contents are delivered in either pull or push. Regarding delivery, cost and wait time are two important factors to be minimized, but they are in a trade-off relationship. The Push-N-scheme (PNS) and timeout-based push scheme (TPS) have been proposed for content delivery. PNS has an advantage in cost over TPS, whereas TPS has an edge in terms of the wait time over PNS. We propose a hybrid push scheme of PNS and TPS, called push-N-scheme with timeout (PNT), to balance the cost and the wait time. We evaluate PNT through simulations, with the results showing that PNT is effective in balancing PNS and TPS.

Keywords—Push Scheme, Content Delivery, SNS, Wait Time

1. INTRODUCTION

The widespread adoption of smart phones allows users to communicate through the Internet -- overcoming time and space barriers -- thanks in particular to the advancements in wireless technologies. Social networking services (SNS) such as Twitter and Facebook have been establishing as the most popular online services. Over several hundred million users are using such services daily, and its growth is expected to continue for the time being [1].

The basic content delivery mechanism of SNS is subscribe-and-publish. Users post their user-created contents on SNS servers, which then deliver the contents to other subscribers who have expressed interest. Content delivery can be achieved by either pull or push. The pull mechanism requires users to connect explicitly to the SNS servers and download the contents under their subscription. During this process, it takes time to set up the connection to the servers and to download the content. On the other hand, the push mechanism has the servers literally pushing contents to the users and actually transmitting them if some conditions are met. The advantage of push over pull is that it takes less time because it foregoes the need for connection setup and also consumes less energy through smart phones.

Regardless of the mechanisms used to deliver the contents, it is important to consider both delivery cost and wait time. Cost consists of time and energy. Wait time is defined as the time that elapsed since an article is posted on the servers before it is pushed or pulled. Note that there is a trade-off between them. For example, efforts to reduce cost by transmitting multiple

※ This work was supported by Incheon National University Research Grant in 2013.
Manuscript received March 22, 2013; accepted July 21, 2013.

Corresponding Author: Kyungkoo Jun (kjun@incheon.ac.kr)

* Dept. of Embedded Systems Engineering, Incheon National University, Incheon, 406-772, Korea (kjun@incheon.ac.kr)

contents by single delivery translate into increased wait time. On the other hand, an attempt to decrease the wait time by frequent pushes hikes up the cost.

A push-N scheme (PNS) [2] requires the SNS servers to collect at least N contents before transmitting them by setting up a single connection. Its performance actually depends on the choice of parameter N . The optimal value of parameter N cannot be calculated analytically; hence its limited applicability. Moreover, it may suffer from starvation -- as long as less than N contents are collected, there is no push. Of course, pull delivers the contents even though there are less than N contents.

To address the problem of PNS, a timer-based push scheme (TPS) [3] is proposed wherein the SNS servers begin to collect the contents with a timer set and transmit them when the timer expires. The advantage of TPS is that the maximum wait time can be bound by timeout parameter T . Note, however, that TPS is subject to inflexibility when timeout T is fixed. Thus, TPS is not appropriate when data traffic by delivery varies frequently and rapidly. A large amount of network resources may be required momentarily to transmit the contents upon timer expiration, resulting in network bottleneck.

In general, we expect PNS to be superior in cost to TPS, which in turn has an advantage in the wait time over PNS. Since PNS guarantees somewhat the number of carried contents in a single connection, cost efficiency is expected to be high. On the other hand, TPS has the bound wait time thanks to the timeout, resulting in a predictable wait time.

Since the cost and the wait time are in a trade-off relationship [4] [5], we aim to develop a new scheme that can balance them. We propose a hybrid scheme that combines PNS with TPS, called Push-N scheme with timeout (PNT). It acts in the same way as PNS, but timeout is set whenever contents are posted. If the timer expires, it immediately transmits the contents even though there are less than N contents. It can improve the wait time of PNS and also avoid starvation thanks to the timeout. Unlike TPS, it can improve the cost because the average number of contents in a single connection is expected to be more than that of TPS.

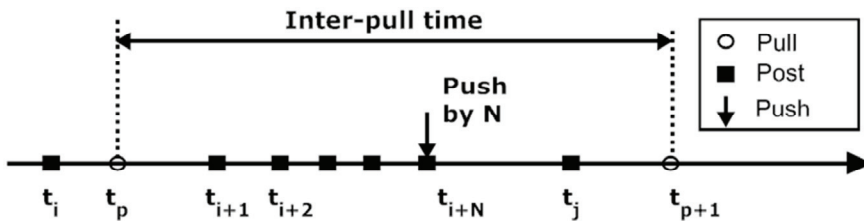


Fig. 1. Example showing how PNS determines push

The rest of the paper is organized as follows: Section 2 evaluates PNS and TPS through simulations, presents their characteristics, and provides ideas to improve them; Section 3 describes the hybrid scheme PNT; Section 4 evaluates the proposed scheme by comparing with PNS and TPS. Section 5 presents the conclusion.

2. EVALUATION OF PNS AND TPS

In this section, we describe how PNS and TPS work and analyze the cost and the wait time. Likewise, we evaluate both of them through simulations. We expect the evaluation to aid in understanding the characteristics of PNS and TPS as well as the pros and cons.

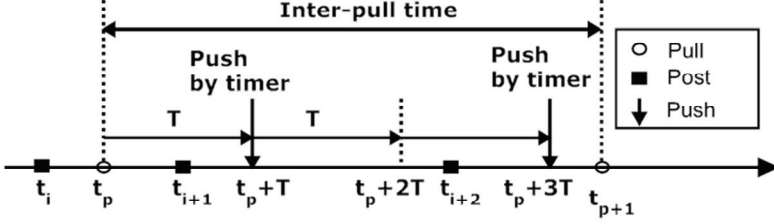


Fig. 2. Example showing how TPS determines push

Fig. 1 shows how PNS determines when push should take place. We set the push threshold as N . Thus, content is posted at time t_{i+1} and is pushed along with the following $N - 1$ contents at t_{i+N} . On the other hand, content posted at time t_i or t_j is pulled at t_p and t_{p+1} .

Fig. 2 shows how TPS determines push using timeout. We set timeout as T . A timer with timeout T is set when pull occurs at t_p . The timer expires at time $t_p + T$, and the content posted at $t_{i+1} < t_p + T$ is pushed. On the other hand, there is no push at time $t_p + 2T$ because no content is posted during $[t_p+T, t_p + 2T]$. Note that the timer is reset at every pull, e.g., at t_p and t_{p+1} .

We analyze the cost and the wait time of PNS and TPS. The assumptions for the analysis are as follows, which are the same as in [6] [7] [8] [9]. The interval between two consecutive pulls -- called inter-pull time $t_a = t_{p+1} - t_p$ -- is assumed to follow an exponential distribution with mean $1/\lambda_a$. The interval between two consecutive posts, called inter-post time $t_b = t_{i+1} - t_i$, is supposed to follow a Gamma distribution with mean $1/\lambda_p$ and variance V_p .

We first analyze the cost efficiency. A more straightforward way of putting it is that the greater the number of contents carried in a single delivery, the higher the cost efficiency is. Let n_{push} be the number of contents delivered by a single push and n_{pull} be the number of contents by a single pull. Then, we can define cost efficiency C as the sum of the expected number of n_{push} and n_{pull} as follows:

$$C = E(n_{push}) + E(n_{pull}) \quad (1)$$

Then, for PNS, $E(n_{push}) = N$ because push occurs if and only if N contents are collected and $E(n_{pull}) = \frac{N-1}{2}$ because the maximum number of contents delivered by pull is $N - 1$ and the minimum number of contents is 0. Thus, the cost efficiency of PNS, C_{PNS} , is as follows:

$$C_{PNS} = \frac{3 \cdot N - 1}{2} \quad (2)$$

The cost efficiency of TPS, C_{TPS} , is as follows. The average number of contents posted during T is $\lambda_p \cdot T$ by Little's law. Thus, it is obvious that $E(n_{push}) = \lambda_p \cdot T$. On the other hand, the average number of pulled contents, i.e., $E(n_{pull})$, can be obtained by subtracting the total number of pushed contents from the total number of contents posted between pulls as follows:

$$E(n_{pull}) = \frac{1}{\lambda_a} \cdot \lambda_p - \left[\frac{1}{\lambda_a} \cdot \frac{1}{T} \right] \cdot E(n_{push}) \quad (3)$$

Therefore,

$$C_{TPS} = \frac{1}{\lambda_a} \cdot \lambda_p + \left(1 - \left[\frac{1}{\lambda_a} \cdot \frac{1}{T} \right] \right) \cdot \lambda_p \cdot T \quad (4)$$

From (3) and (4), we can see that C_{PNS} is determined by N only. On the other hand, C_{TPS} is controlled by λ_a and λ_p as well as timeout T . It implies that PNS has advantage over TPS in terms of the cost because PNS can have complete control over the cost efficiency unlike TPS.

We define the wait time of a content as $t_k - t_p$ if the content is posted at t_p and pushed at t_k . Let w_{PNS} be the average wait time of PNS. Since PNS collects N contents before pushing them, the first posted content should wait until the following $N - 1$ contents are posted. Therefore, its wait time is $(N - 1) \cdot \frac{1}{\lambda_p}$. Similarly, k th content should wait $(N - k) \cdot \frac{1}{\lambda_p}$.

Therefore, w_{PNS} becomes

$$w_{PNS} = \frac{\sum_{i=1}^N (N - i) \cdot \frac{1}{\lambda_p}}{N} = \frac{(N - 1)}{2 \cdot \lambda_p} \quad (5)$$

Let w_{TPS} be the average wait time of TPS. The average number of contents posted during T is $\lambda_p \cdot T$. We assume that those contents are uniformly distributed over T . Then the first content is posted at time past $T/(T \cdot \lambda_p + 1)$ from the beginning of the timer. Thus, the first content should wait $T - T/(T \cdot \lambda_p + 1)$. Similarly, k th content should wait $T - k \cdot T/(T \cdot \lambda_p + 1)$. Therefore, w_{TPS} becomes

$$w_{TPS} = \frac{\sum_{i=1}^{\lambda_p \cdot T} T - i \cdot T/(T \cdot \lambda_p + 1)}{\lambda_p \cdot T} = \frac{T}{2} \quad (6)$$

Regarding the maximum wait time, PNS is bound by the maximum inter-pull time. On the other hand, TPS is bound by T . From (5) and (6), w_{TPS} is controlled by T only, whereas PNS is affected by λ_p . Therefore, TPS is expected to have more control than PNS over the wait time.

To verify the analytical models for the cost and the wait time, we evaluate the performance of PNS and TPS using simulations. We assume that $\lambda_a = 1/60$ and $\lambda_p = 5/60$ and $V_p=1$. In the

simulations, we measure the cost as follows:

$$Cost = C_{setup} + n \cdot C_{delivery} \tag{7}$$

where n is the number of contents and C_{setup} and $C_{delivery}$ are the costs for the connection setup and transmission, respectively. Pull incurs higher cost than push because users need to connect to the SNS servers before requesting content delivery. Nonetheless, we assume that such difference is ignorable compared to the overall cost. Thus, we assume that push and pull have the same delivery cost. In the simulations, we set $C_{setup}=1.0$ and $C_{delivery}=1.68$ as in [3]. We simulate the usage pattern wherein one user utilizes an SNS server for 30 days. In addition, we attempt to use various N_s of PNS and T of TPS to observe the effects of those parameters on the cost and the wait time.

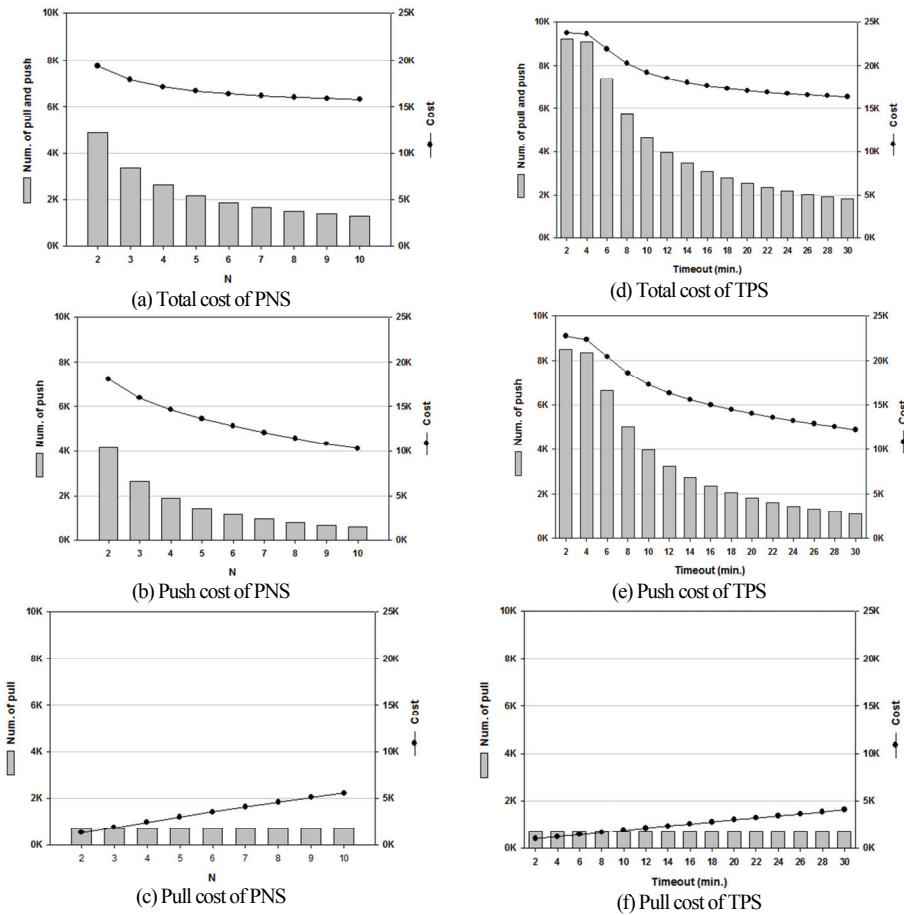


Fig. 3. Comparison of the costs of PNS and TPS along with the number of pushes and pulls. The results of PNS are (a), (b), and (c), whereas (d), (e), and (f) are those of TPS

Fig. 3(a) shows the total number of pushes and pulls of PNS and the corresponding cost as N changes. We can see that the occurrences of push and pull decrease as N increases, and so does the cost. To study the decreases in detail, we look into the statistics of push and pull in Fig. 3(b) and Fig. 3(c), respectively. In Fig. 3(b), we observe that the number of pushes and the cost decrease as N increases. This is because longer wait time is required until N contents are collected, resulting in less frequent push. Moreover, the cost obviously drops if the number of pushes decreases. On the other hand, we see in Fig. 3(c) that the number of pulls remains unchanged regardless of N . This is because the occurrences of pull in the simulations are governed by Gamma distribution irrespective of N . Note, however, that the cost increases in proportion to N . As explanation, if N becomes larger, push occurs less frequently. It means that the number of contents waiting to be pushed increases, and that there are more contents to be pulled by users unlike when N is smaller. Thus, the overall cost of pull grows as N increases.

Regarding TPS, Fig. 3(d) shows the total number of pushes and pulls of TPS and the corresponding cost as T changes. The total number decreases as T increases, similar to PNS, and so does the cost. To study further the effect of T on push and pull separately, we present the measurement of push and pull in Fig. 3(e) and Fig. 3(f), respectively. Fig. 3(e) shows the number of pushes and the cost. The number of pushes is inversely proportional to T because the longer T becomes, the less push occurs. Obviously, the decreasing cost is attributed to the decreasing number of pushes. Fig. 3(f) shows the number of pulls and the cost according to T . We can observe that the number of pulls is steady, whereas the cost increases as T increases for the same reason as Fig. 3(c).

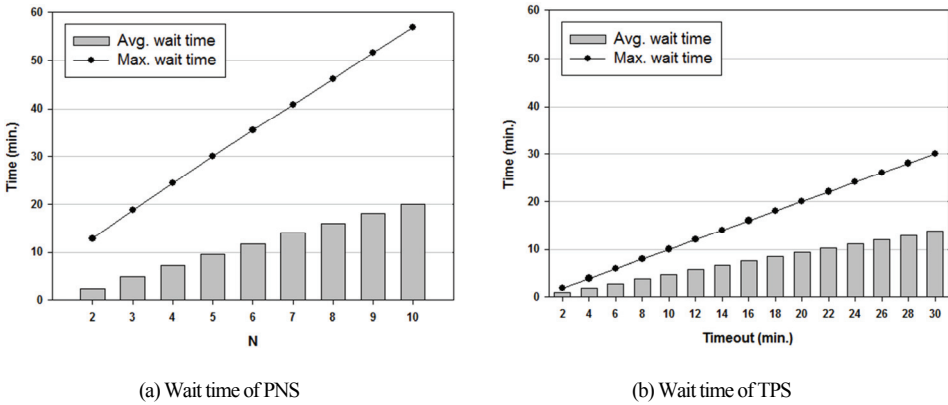


Fig. 4. Average and maximum wait times of PNS and TPS

Fig. 4 shows the average and maximum wait time of PNS and TPS. In Fig. 4(a), the average wait time of PNS increases in proportion to N because PNS should wait longer as the required number of contents triggering push increases, and so does the maximum wait time. In Fig. 4(b), the average and maximum wait times of TPS increase as T increases for the same reason.

Overall, from the simulation results of Fig. 3 and Fig. 4, we can see that PNS has an advantage over TPS in terms of the cost, but TPS is more competitive when it comes to the wait time. PNS improves the cost by 7% on the average compared to TPS, whereas TPS enhances the wait time by 42%. Note that, since there is a trade-off between the cost and the wait time, an advantage in one aspect results in the degradation of the other aspect.

From the perspective of network operators and users, giving priority to either the cost or the wait time at the expense of the other poses a dilemma. Note, however, that it is hard to develop a scheme that improves both of them at the same time. Therefore, it is best to find a way to balance them. We propose a method that can achieve the average performance of PNS and TPS in the following section.

3. PUSH-N SCHEME WITH TIMEOUT

We propose a novel push scheme that aims to secure balanced performance between PNS and TPS, called push-N scheme with timeout (PNT). Basically, as a hybrid scheme of PNS and TPS, PNT uses both the threshold and the timeout. PNT operates just as PNS, but the former does not wait indefinitely until the next post occurs. Instead, PNT uses the timer just as TPS. When the timer expires, the collected contents are pushed even though they are less than N . On the other hand, PNT works just as TPS, but the former resets the timeout at every new post, having the effect of postponing the timeout. In summary, the push of PNT occurs in one of two cases: when the number of collected contents reaches N or when the timer expires.

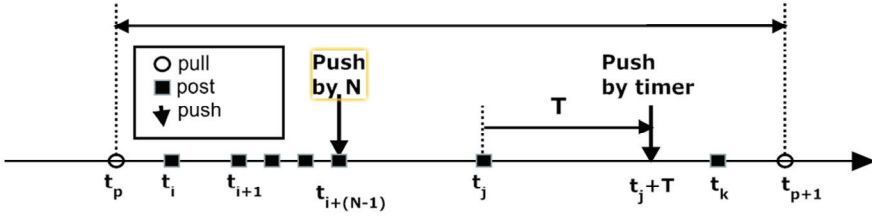


Fig. 5. Example showing how PNT determines push

Fig. 5 shows how PNT works. Suppose a user pulls contents at t_p and t_{p+1} and content i is posted at t_i . Then, because N contents are collected at $t_{i+(N-1)}$, all of them are pushed. On the other hand, the content posted at t_j is pushed at time t_{j+T} when the timer expires because there is no following post before the timeout. Content k is pulled by the user at t_{p+1} .

We expect PNT to have improved wait time compared to PNS because the former can limit the wait time for the next post. On the other hand, PNT can improve the cost compared to TPS. Because PNT postpones the timeout at every new post, its single push can carry more contents than TPS, resulting in the reduced number of pushes.

We analyze the cost efficiency of PNT stochastically. Let t_p be the time that elapsed until the next post. Then, the probability that N contents are collected and pushed without timer expiration is $P(t_p < T)^{N-1}$. On the other hand, the probability that $1 \leq k < N$ contents are pushed by the timeout is $P(t_p < T)^{k-1} \cdot P(t_p \geq T)$. Therefore, $E(n_{push})$ is as follows:

$$\begin{aligned}
 E(n_{push}) &= N \cdot P_1^{N-1} + \sum_{k=1}^{N-1} K \cdot P_1^{k-1} \cdot P_2 \\
 &= \frac{P_2 \cdot (1 - P_1^{N-1})}{(1 - P_1)^2} - \frac{(N-1) \cdot P_1^{N-1} \cdot P_2}{1 - P_1} + N \cdot P_1^{N-1}
 \end{aligned} \tag{8}$$

where $P_1 = P(t_p < T)$ and $P_2 = P(t_p \geq T)$.

Likewise, $E(n_{pull}) = \frac{N-1}{2}$ because the maximum number of contents delivered by pull is $N-1$ and the minimum number is 0. Therefore, the cost efficiency of PNT C_{PNT} is as follows:

$$C_{PNT} = \frac{P_2 \cdot (1 - P_1^{N-1})}{(1 - P_1)^2} - \frac{(N-1) \cdot P_1^{N-1} \cdot P_2}{1 - P_1} + N \cdot P_1^{N-1} + \frac{N-1}{2} \tag{9}$$

We analyze the average wait time of PNT w_{PNT} stochastically. We consider the push by the threshold and the push by the timer separately. In the case of push by the threshold, the first posted content should wait until the following $N-1$ contents are posted. Then the wait time of the first content is upper bound by $(N-1) \cdot T$ since timeout T is renewed at every post. Similarly, the wait time of the i th content is bound by $(N-i) \cdot T$. Thus, the average wait time of the contents when push by the threshold occurs, w_N , is bound by the following, and its probability is P_1^{N-1} :

$$w_N \leq \frac{\sum_{i=1}^{N-1} (N-i) \cdot T}{N} = \frac{T \cdot (N-1)}{2} \tag{10}$$

In the case of push by the timer, we consider separately the cases by the number of contents in the push. If there is only content, the average wait time is T because there is no following post anymore within T . Likewise, its probability is P_2 . Similarly, if there are two contents, the average wait time is bound by $2 \cdot T / 2$ with probability of $P_1 \cdot P_2$. Then, for $i \leq N-1$ contents, the average wait time is bound by $i \cdot T / i$ with probability of $P_1^{i-1} \cdot P_2$.

Thus, the average wait time of PNT w_{PNT} is upper bound by

$$w_{PNT} \leq w_N \cdot P_1^{N-1} + \sum_{i=1}^{N-1} T \cdot P_1^{i-1} \cdot P_2 \tag{11}$$

Overall, C_{PNT} and w_{PNT} are stochastically dependent on N and T . In particular, P_1 and P_2 are affected by T . This implies that the deterministic analysis of PNT is hard and impractical. Thus, in the next section, we evaluate PNT through simulations in comparison with PNS and TPS.

4. PERFORMANCE EVALUATION

We evaluate the performance of PNT under the same simulation environment of Section 2 as well as the parameters of the probability distribution. We measure the cost and the wait time while varying N and T .

Fig. 6 shows the cost of PNT as N and T change, represented as contour lines. The numbers on the contour lines mean the cost. For example, when $N = 8$ and $T = 20$, the cost is approximately 16,000. We can see how rapidly the cost changes by observing the distance between the contour lines. The greater the distance is, the slower the changes are. We investigate the effect of T on the cost. Once T becomes larger than 8, it does not affect the cost. In other words, the probability that N or more contents are posted and subsequently pushed before the timer expires increases as T grows, having the effect of T less. This implies that, once the timeout is longer than 8 minutes, most of the content delivery is carried by the push by the threshold or the pulls. In this case, N is the dominant factor for determining the cost. On the other hand, when T is less than 8, it affects the cost significantly. We can see it from the shortened distances between the contour lines. This is because push by timeout is more likely to happen with shorter T s. In this case, N has less effect on the cost.

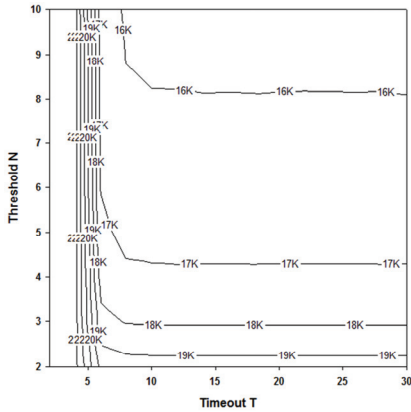


Fig. 6. Cost of PNT in contour form

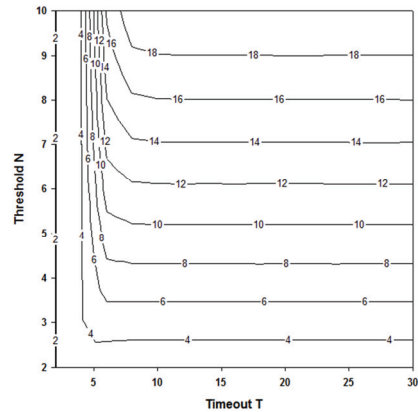


Fig. 7. Wai time of PNT in contour form

We also observe that, when T is over 10, the distance between the contour lines becomes greater as N increases but is short when N is between 2 and 4. In other words, the change rate of the cost decreases as N grows. This coincides with the results of Fig. 3(a), i.e., the cost begins to stabilize once N becomes larger than 6. This implies that increasing N by more than 6 is unnecessary to improve the cost.

Fig. 7 shows the average wait time as the contour lines when N and T change. Similar to Fig. 4(a), the wait time is dominantly affected by N when $T > 8$. We observe that the distance between the contour lines is uniform as N increases, which is different from Fig. 6. It coincides with the linear increase of the wait time in Fig. 4(a), implying that the average wait time increases in linear proportion to N until it is upper bound by the inter-pull time.

From these results, we can see that the optimal combination of N and T , which balances the cost and the wait time, is around $N = 6$ and $T = 10$. We compare the cost and the wait time of PNT with those of PNS with $N = 6$ and TPS with $T = 14$ because those parameter settings of

PNS and TPS show the balanced performance between the cost and the wait time.

Fig. 8 compares the costs of PNT, TPS, and PNS. The cost of PNT improves 7% compared with that of TPS. This implies that the use of the timeout helps improve cost efficiency. On the other hand, compared to PNS, the cost is degraded, but it is less than 1%. This shows that PNT is able to improve the cost of TPS without loss of the advantage of PNS.

Fig. 9 compares the average wait times of PNS, TPS, and PNT. PNT improves the average wait time compared to PNS by 4% and the maximum wait time by 16.5%. The use of the timeout is a key factor to the improvements. Compared to TPS, however, the average and maximum times degrade by 67% and 11%, respectively. As an explanation, the effective timeout of PNT is as large as N times longer than TPS even though we set the same T for PNT and TPS. This is because the timeout is canceled and reset at every new post, and the push by the thresholds needs to collect N contents. Therefore, PNT -- which in fact has very long timeout -- cannot avoid the degradation of the wait time. We believe it can be improved further by studying the adjustment of the timeout in future works. Nevertheless, we can see that PNT provides a proper method of balancing the cost and the wait time, combining the advantages of PNS and TPS.

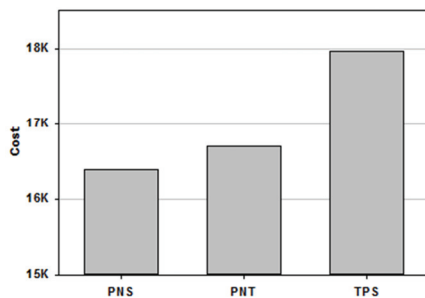


Fig. 8. Comparison of costs of PNS, TPS, and PNT

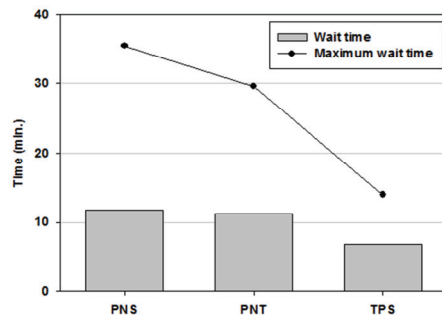


Fig. 9. Comparison of wait times of PNS, TPS, and PNT

5. CONCLUSIONS

We propose PNT, which is a hybrid scheme of PNS and TPS, for the content delivery of the SNS servers. From the simulation results and the stochastic analysis of PNT, we can see that PNT provides a method that balances the cost and the wait time. In other words, PNT is able to combine the advantages of PNS and TPS. Note, however, that PNT is not able to improve both the cost and the wait time because they are in a trade-off relationship. Therefore, PNT degrades the wait time while improving the cost of TPS and sacrifices the cost of PNS for the improvement of the wait time. From the simulation results, the average and maximum wait times of PNT improve 4% and 16.5%, respectively, compared to PNS, whereas the cost is improved 7% over TPS. Note, however, that PNT degrades the cost of PNS by 1% and the cost of TPS by 11%. Overall, PNT provides a solution that is able to reach a compromise between the cost and the wait time.

REFERENCE

- [1] Facebook, "Statistics." <http://www.facebook.com/press/info.php?statistics>
- [2] C. Chuang, Y. Lin, and Y. Yeh, "Performance of linear mobile data transmission," *IEEE Trans. Wireless Commun.*, vol. 10, no.8, Aug. 2011, pp.2451-2455.
- [3] H. Ko, S. Pack, and W. Lee, "Timer-based push scheme for online social networking services in wireless networks," *IEEE Commun. Letters*, vol. 16, no. 12, Dec. 2012, pp.2095-2098.
- [4] Y. Lin, W. Lai, and J. Chen, "Effects of cache mechanism on wireless data access," *IEEE Trans. Wireless Commun.*, vol. 2, no. 6, Nov. 2003, pp.1247-1258.
- [5] Y. Fang, Z. Haas, B. Liang, and Y. Lin, "TTL prediction schemes and the effects of inter-update time distribution on wireless data access," *ACM Wireless Network*, vol. 10, no. 5, Sep. 2004, pp.607-619.
- [6] S. Yang, "Dynamic power saving mechanism for 3G UMTS system," *ACM/Springer Mobile Network Applications*, vol. 12, no. 1, Jan. 2007, pp.5-14.
- [7] Y. Lin, "Eliminating tromboning mobile call setup for international roaming users," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, Jan. 2009, pp.320-325.
- [8] P. Escalle, V. Giner, and J. Oltra, "Reducing location update and paging costs in a PCS network," *IEEE Ttrans. Wireless Commun.*, vol. 1, no. 1, Jan. 2002, pp.200-209.
- [9] S. Sou and Y. Lin, "Modeling mobility database failure restoration using checkpoint schemes," *IEEE Trans. Wireless Commun.*, vol. 6, no. 1, Jan. 2007, pp.313-319.



Kyungkoo Jun

He received a Ph.D. degree in Computer Science from Purdue Univ. in 2001. He has been an associate professor at Incheon National Univ. since 2004. His research interests are in the area of network systems, embedded systems and mobile services.