Control Strategies of Subscription Notification Delivery in Smart Spaces

Dmitry G. Korzun[1(✉)], Michele Pagano[2], and Andrey Vdovenko[1]

[1] Petrozavodsk State University, Petrozavodsk, Russia
{dkorzun,vdovenko}@cs.karelia.ru
[2] University of Pisa, Pisa, Italy
m.pagano@iet.unipi.it

Abstract. The paper studies performance of the subscription operation in smart spaces for the case when the notification delivery to a client is subject to losses. We consider an active control of the check interval for the client to adapt to the observable loss rate. The adaptive strategy takes into account the number of lost notifications. The performance is analyzed and compared with other strategies under simplified assumptions about the loss distribution.

Keywords: Smart spaces · Subscription · Mobile clients · Control strategy · Performance · Simulation

1 Introduction

Information sharing in a smart space employs the subscription operation for detection of content changes and for subsequent delivery of notifications to clients [10,13]. Both change detection and notification delivery are subject to losses in networked environments [14], especially with the emergency of Internet of Things (IoT).

In existing solutions for smart spaces, the major role is played by information brokers [5,9]. In this paper, we study a solution closer to the Internet philosophy, in which the control of notification delivery is partially delegated to clients. The latter actively request the broker for new notifications, in addition to default passive waiting for incoming notifications from the broker. A key performance parameter is the check interval, whose length is adapted to the observable loss rate. This issue resembles the "classical" congestion control in Transmission Control Protocol (TCP) [1–3], by which the window size is reduced in case of losses and incremented otherwise.

Our study continues research [8,14]. We consider an active control of notification delivery for subscription operation for the case of notification losses. The problem was initially stated in [14]. In our solution, the client follows an adaptive strategy controlling the check interval based on the number of notifications lost in the latest window. This adaptive strategy was early experimented in [8]. In this extended paper we show that the strategy is a generalization of the

© Springer International Publishing Switzerland 2016
V. Vishnevsky and D. Kozyrev (Eds.): DCCN 2015, CCIS 601, pp. 40–51, 2016.
DOI: 10.1007/978-3-319-30843-2_5

TCP algorithm of additive–increase/multiplicative–decrease (AIMD). We perform simulation experiments to compare the performance with other strategies. Several different notification loss distributions and performance metrics are used to make this comparison more comprehensive.

The rest of the paper is organized as follows. Section 2 states the notification loss problem for subscription operation in smart spaces. Section 3 introduces strategies that a client can additionally use for controlling the notification delivery. Section 4 describes the results of our simulation experiments to evaluate and compare the performance of the considered control strategies. Finally, Sect. 5 concludes the paper.

2 Subscription Notification Delivery in Smart Spaces

The pub/sub model is widely used for organizing multi-agent interactions in distributed systems [4]. In this study we consider this model for the case of smart spaces [10,13].

In general, a smart space forms a sparse-connected multi-agent system deployed in a networked computing environment, typically with access to the Internet [7]. Such an environment consists of various digital devices, including the growing family of IoT devices. Software agents run on the devices and interact over the shared information content. This type of interaction involves, in parallel and asynchronously, a lot of informational sources and destinations. Information sharing makes the interaction indirect, based on a semantic information broker (SIB) [9,13]. The latter implements a shared information storage, serving requests from agents on read/write operations. In other words, SIB acts as an information hub maintaining knowledge of the whole environment and enabling the agents to construct digital services over this cooperative knowledge.

The subscription operation specifies a persistent query from an agent (a subscription client) to the SIB (a subscription server) for a particular part of the shared content [10]. Whenever the specified part is changed, the agent should receive the subscription notification. Changes are due to parallel activity of other agents, which act as publishers in this interaction (note that an agent may combine the roles of publisher and subscriber). SIB monitors subscriptions of all clients and maps all incoming content changes to the specified interests. Therefore, changes are controlled on the SIB side, and corresponding notifications are sent to the clients. SIB acts as a passive receiver, and we call such subscription notifications passive [14].

We employ Smart-M3 as a reference software platform for creating smart spaces [5,9]. For each subscription, the SIB maintains a network connection (e.g., a TCP connection) established by the client's request [10,11]. Knowing the set of all subscriptions, the SIB regularly checks that they are alive, removing the subscription if its network connection is lost.

Smart-M3 follows the best effort style in subscription notification delivery. A notification *should* be sent to a client if a related change in the content has happened. Some notifications can be unsent by SIB due to its overload or internal operability faults. SIB does not check delivery for already sent notifications, and a new notification can be sent although the underlying network connection is broken on the client side.

The above properties do not ensure the dependable notification delivery even if reliable network protocols are used, such as TCP. A possible solution is for a client to have an additional mechanism reducing the number of undelivered notifications. The obvious way is augmenting the passive notification delivery with an active control strategy that the client performs individually on its own.

Consider the following model to formalize the key properties of the subscription notification loss problem in smart spaces. Let $i = 1, 2, \ldots$ be the event-based time evolution on the client side, where i is the index of notification events. An event i is either a passive notification (i.e., received from SIB) or an explicit check of the notification delivery (made by the client within its active control).

Denote by t_i and k_i the time elapsed and the number of losses occurred between i and $i+1$, respectively. Assume that some initial value t_0 is always defined. The values for k_i are non-negative integers. We consider the following distributions to model the notification losses.

1. Let the time elapsed between consecutive losses follow a uniform distribution $\mathcal{U}\{0, \xi t_0\}$. Hence, the average number of losses in any check interval is proportional to its length t_i
2. Let k_i follow a Poisson process of parameter λt_i. Hence, the number of losses during t_i has the probability mass function
$$\mathbb{P}(k_i = k) = \frac{(\lambda t_i)^k}{k!} e^{-\lambda t_i} \tag{1}$$
3. Let k_i follow a two-state alternated Poisson process: the loss rate is λ_1 and λ_2 with probability p_1 and $p_2 = 1 - p_1$, respectively. The assumptions $0 < p_2 < p_1$ and $\lambda_1 \ll \lambda_2$ describe that the network typically operates with moderate losses (λ_1), while from time to time the network suffers from high losses (λ_2), e.g., due to burst overload.

3 Control Strategies

The notification loss problem is similar to the packet loss problem in TCP congestion control [1–3]. The additive–increase/multiplicative–decrease (AIMD) algorithm is a feedback control algorithm used in the TCP congestion avoidance mechanism. The AIMD algorithm calculates the size $\omega(\tau)$ of congestion window at time τ, after the initial slow start phase.

Denote by τ_i the end of the i^{th} TCP round, whose duration is then given by $t_i = \tau_i - \tau_{i-1}$ and by $\omega(\tau)$, where the latter is expressed in MSS units (maximum segment size). The following equation describes the process [2]:

$$\omega_{i+1} = \omega(\tau_i + 0) = \begin{cases} \omega(\tau_i) + 1 & \text{if no loss,} \\ \lfloor \frac{\omega(\tau_i)}{\kappa} \rfloor & \text{if losses are observed on } [\tau_{i-1}, \tau_i], \end{cases} \tag{2}$$

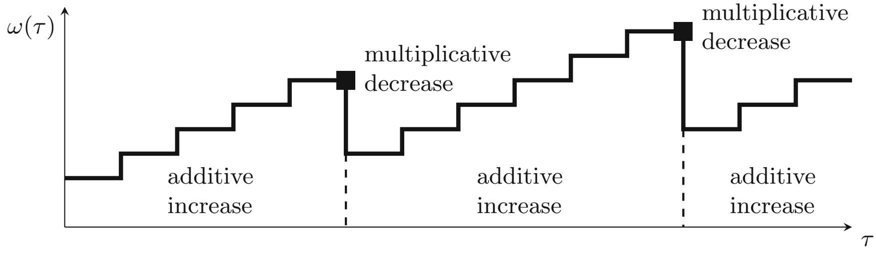

Fig. 1. Typical stepwise evolution of TCP congestion window.

The stepwise behavior of $\omega(\tau)$ is schematically depicted in Fig. 1.

The standard case is $\kappa = 2$, and the multiplicative reduction is applied as soon as a loss is detected (by triple duplicate acknowledgment). Although in TCP modeling the analysis is often carried out at round level, actual TCP implementations react at the arrival of each acknowledgement and not just at the end of round. Moreover, handling of multiple losses for congestion window is still an open issue and highly impacts on TCP performance, especially in wireless and satellite networks [3]. Consequently, the AIMD algorithm cannot be applied directly for the case of subscription notification losses.

For the case of subscription in smart spaces, let us introduce an analogue of the AIMD algorithm to perform active control of notification delivery. If the client has observed no losses during t_{i-1} (i.e., $k_{i-1} = 0$) then this observation indicates that the system state is "good". To save resources the client can increase additively the check interval, i.e. $t_i = t_{i-1} + \delta$ for a fixed parameter $\delta > 0$. As in the AIMD algorithm, the increment is conservative, since a high increase of t_i might determine a clear risk for suffering a burst of losses.

On the contrary, if the client has observed losses, i.e., $k_{i-1} > 0$, it must reduce t_i to decrease the number of losses in near future. The reduction is multiplicative, since the client is interested in fast achieving $k_i = 0$. Moreover, the reaction should take into account the amount of losses and the previous value of the control interval. We apply the multiplicative average

$$t_i = \alpha t_{i-1} + (1-\alpha)\frac{t_{i-1}}{k_{i-1}+1},$$

where $0 \le \alpha < 1$ is a fixed parameter (cf. Fig. 1 for $\kappa = k_{i-1} + 1$ and $\alpha = 0$).

We yield the recurrent system that describes an *adaptive strategy* [14], by which the check interval t_i is reduced (multiplicative decrease) in case of losses and incremented (additive increase) otherwise:

$$t_i = \begin{cases} t_{i-1} + \delta & \text{if } k_{i-1} = 0, \\ \dfrac{1 + \alpha k_{i-1}}{k_{i-1}+1} t_{i-1} & \text{if } k_{i-1} > 0. \end{cases} \quad (3)$$

Note that (3) is valid only for active control of subscription notifications. When a passive notification i is delivered, then the value of t_i is not set by the client.

This adaptive strategy refines the AIMD algorithm of TCP, i.e., providing a TCP-like control of notification delivery. One can interpret t_i in (3) as TCP congestion window. Fixed $\delta = 1$ means the additive increment by one full-sized segment. In the AIMD algorithm, $\alpha = 0$. TCP congestion implies $k_{i-1} = 1$, and $\kappa = k_{i-1} + 1 = 2$ halves the value of t_i, as it is described in (3).

In order to evaluate and compare the performance, we consider alternative control strategies and present some auxiliary analytical results. The simplest approach is the strategy of a *constant check interval*, i.e., $t_i = t_0$ for $i = 1, 2, \ldots$. The mean and variance of the random variable K, which describes the number of losses, depend only on the loss distributions introduced in Sect. 2. In the case of Poissonian losses, we have:

$$\mathbb{E}\Big[K\Big] = \mathrm{Var}\Big[K\Big] = \lambda t_0. \tag{4}$$

Since in many networked systems, randomness can improve the performance, an interesting strategy is *random selection* of the check interval. Intuitively, let t_i be chosen at random around t_0. In particular, we consider a continuous uniform distribution $\mathcal{U}\Big\{t_0 - \Delta, t_0 + \Delta\Big\}$ with $0 < \Delta < t_0$. In the case of Poissonian losses, applying the laws of total expectation and variance (e.g., see [15]), we have

$$\mathbb{E}\Big[K\Big] = \lambda t_0, \quad \mathrm{Var}\Big[K\Big] = \mathbb{E}_T\Big[\mathrm{Var}\Big[K|T\Big]\Big] + \mathrm{Var}_T\Big[\mathbb{E}\Big[K|T\Big]\Big] = \lambda t_0 + \frac{1}{3}\lambda^2\Delta^2. \tag{5}$$

The mean value is the same as in (4), while the variance is increased due to the variability of the interval length. Roughly speaking, in this case the randomness does not change the average performance indexes, but increases the probability that higher number of notifications can be lost.

Finally, we consider a semi-adaptive approach, in which the check interval is halved in case of losses and set to the initial (reference) value t_0 otherwise:

$$t_i = \begin{cases} t_{i-1}/2 & \text{if} \;\; k_{i-1} > 0, \\ t_0 & \text{if} \;\; k_{i-1} = 0. \end{cases} \tag{6}$$

We shall refer this strategy as *multiplicative–decrease*. The evolution of the check interval can be described by a discrete-time Markov chain [6]. State i corresponds to a check interval of length $t_i = 2^{-i} t_0$ and the only possible transitions from state i are back to state 0 (no loss is experienced) with probability p_i and to the following state $i + 1$ (the check interval is halved in case of losses) with probability $q_i = 1 - p_i$, as shown in Fig. 2.

Due to the particular structure of the Markov chain, the state probabilities can be easily written as a function of π_0 (local balance equations):

$$\pi_{n+1} = \pi_0 \prod_{i=0}^{n} q_i \tag{7}$$

with the additional normalization condition

$$\sum_{n=0}^{\infty} \pi_n = 1 \quad \Rightarrow \quad \pi_0 \Big[1 + \sum_{n=0}^{\infty} \prod_{i=0}^{n} q_i\Big] = 1. \tag{8}$$

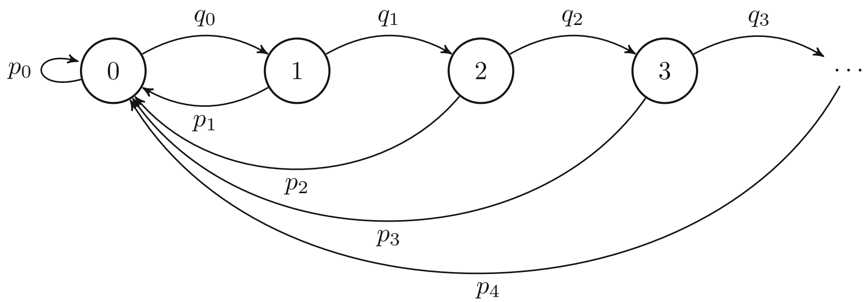

Fig. 2. Markov chain for the multiplicative–decrease strategy.

Note that the sum converges under reasonable assumptions over the loss processes. (It is enough to assume that the loss probability goes to zero as $n \to \infty$.) The state probabilities permit to find all the relevant statistics, such as the average check interval

$$\mathbb{E}\big[T\big] = \sum_{n=0}^{\infty} \pi_n t_n = t_0 \sum_{n=0}^{\infty} \pi_n 2^{-n}.$$

The numerical values of p_n and $q_n = 1 - p_n$ depend on the loss distribution. In the case of Poissonian losses, we have

$$\begin{cases} p_0 = \mathbb{P}(K = 0 | T = t_0) = e^{-\lambda t_0}, \\ p_n = \mathbb{P}(K = 0 | T = t_n) = {p_0}^{2^{-n}}, \quad n \geq 1. \end{cases}$$

In the case of uniformly distributed losses, the transition probability $q_n = 1 - p_n$ is proportional to the length of the check interval $t_n = t_0 2^{-n}$, i.e., $q_n = q_0 2^{-n}$. Then (7) is rewritten as

$$\pi_n \;=\; \pi_0 q_0^n \left(\frac{1}{2}\right)^{n(n-1)/2}$$

Taking into account the specific values of p_n and q_n as well as the normalization condition (8), it is possible to calculate the state probabilities (at least numerically). In the special case in which the loss probabilities do not depend on the duration of the check interval (i.e., $q_n = q \;\forall\, n$), the close-form expression for the state probabilities is $\pi_n \;=\; (1-q)q^n$ with the average check interval

$$\mathbb{E}\big[T\big] = \sum_{n=0}^{\infty} \pi_n t_n = \; t_0 \frac{1-q}{1-q/2} > t_0,$$

and the variance

$$\mathrm{Var}\Big[K\Big] \;=\; \mathbb{E}\big[T^2\big] - \mathbb{E}^2\big[T\big] \;=\; t_0^2 \frac{q(1-q)/4}{(1-q/4)(1-q/2)^2}.$$

The assumption that the number of losses does not depend on the length of the check interval might be unrealistic. Nevertheless, it provides a lower bound for the average (and an upper bound for the variance) w.r.t. the more realistic assumptions described in Sect. 2.

Moreover, a practical implementation does not use very short check intervals. This property corresponds to truncating the Markov chain at state N, remaining in that state in case of further losses (returning to state 0 with the probability p_N as usual). Hence, (7) is still valid for all the states up to $N-1$, while

$$\pi_N = \pi_0 \frac{1}{1-q_N} \prod_{i=0}^{N-1} q_i,$$

and the normalization condition (8) involves just $N+1$ terms.

4 Simulation Experiments

In the experiments, we use the three notification loss distributions (see Sect. 2). The key simulation parameters are summarized in Table 1. In all four control strategies, we take $t_0 = 20$ s. According to the selected parameters of the loss distributions, it means that every 20 s one notification is lost on average. The simulation parameters of the four control strategies are presented in Table 2.

The average check interval is exactly $t_0 = 20$ for the strategies of constant check interval and random selection. The adaptive strategy starts from t_0 and then forgets about t_0, balancing in accordance with the observed number of losses. The multiplicative–decrease strategy attempts to be below t_0 when losses are observed, leading to shorter check interval on average.

Experimental behavior of the four control strategies for the given three losses distributions is shown in Figs. 3, 4, and 5. From the visually comparison, it is immediately clear that the adaptive strategy makes the check interval longer compared with the other strategies. From the point of view of the performance, the adaptive strategy saves the resources, reducing the number of network requests from the client to SIB.

Consider quantitative performance indexes. The basic metrics are the average number of losses k_{avg} and the average length of the check interval t_{avg}:

$$k_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} k_i \ , \quad t_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} t_i .$$

Table 1. Simulation parameters of the notification loss distributions.

Uniform losses	Poissonian losses	Two-state alternated Poissonian losses
$\xi = 0.1$	$\lambda = 0.05$	$p_1 = 0.8$ and $\lambda_1 = 0.0375$, $p_2 = 0.2$ and $\lambda_2 = 0.1$

Table 2. Simulation parameters of the control strategies for notification delivery.

Adaptive	Constant check	Random selection	Multiplicative–decrease
$\alpha = 0.3$, $\delta = t_0 = 20$	$t_i = t_0 = 20$	$a = 10$, $b = 30$	$t_0 = 20$

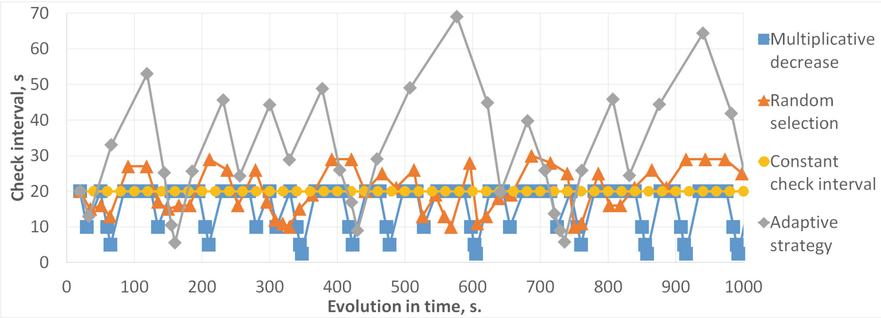

Fig. 3. Experimental evolution of the check interval: uniform losses.

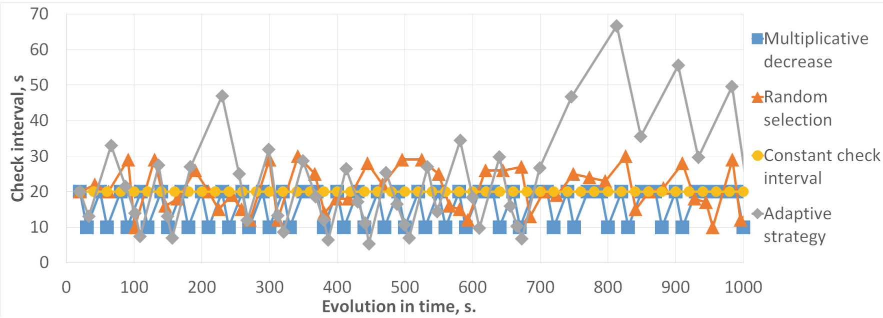

Fig. 4. Experimental evolution of the check interval: Poissonian losses.

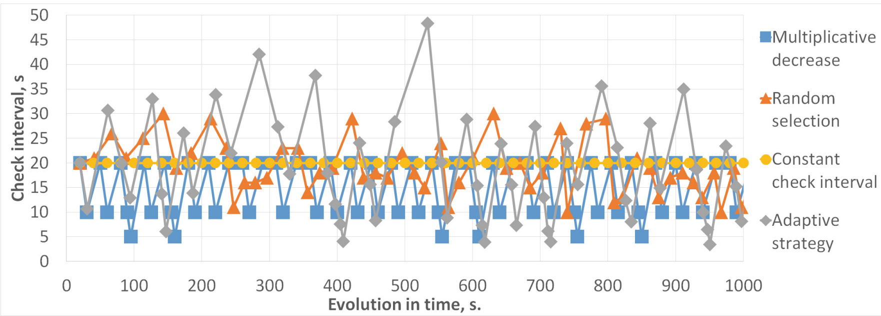

Fig. 5. Experimental evolution of the check interval: 2-state alternated Poissonian losses.

Table 3. Performance comparison based on the metrics Q_{tot} and Q_{avg}.

Loss distribution	Metric	Multiplicative–decrease	Random selection	Constant check interval	Adaptive strategy
Uniform losses	$k_{\text{avg}}/t_{\text{avg}}$	0.60/14.15	0.78/19.24	1.04/20.00	1.09/30.36
	Q_{tot}	0.043	0.041	0.052	0.035
	Q_{avg}	0.053	0.045	0.052	0.040
Poissonian losses	$k_{\text{avg}}/t_{\text{avg}}$	0.76/15.60	1.10/20.38	1.13/20.00	1.13/25.18
	Q_{tot}	0.048	0.054	0.057	0.044
	Q_{avg}	0.038	0.057	0.057	0.046
Two-state alternated Poissonian losses	$k_{\text{avg}}/t_{\text{avg}}$	1.01/14.25	1.66/18.81	1.54/20.00	1.33/20.81
	Q_{tot}	0.071	0.088	0.077	0.063
	Q_{avg}	0.060	0.087	0.077	0.067

Intuitively, the client is interested in reducing the losses (i.e., $k_{\text{avg}} \to \min$) for high values of the check interval (i.e., $t_{\text{avg}} \to \max$). Since these two requirements are in opposition, in analogy with the concept of power in computer networks defined as the ratio between the throughput and the delay [12], we use the ratio among the two, such as the following metrics:

$$Q_{\text{tot}} = \frac{k_{\text{avg}}}{t_{\text{avg}}} \to \min\,, \qquad Q_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n}\frac{k_i}{t_i} \to \min\,.$$

The performance comparison based on these metrics is shown in Table 3. The experiments indicate that the control by the multiplicative–decrease strategy and adaptation strategy achieves the lowest loss level. The efficiency of multiplicative decrease is due to frequent checks, which consumes resources of the client and the network. On the other hand, the adaptive strategy tries to increase the check interval length (on the risk of higher losses).

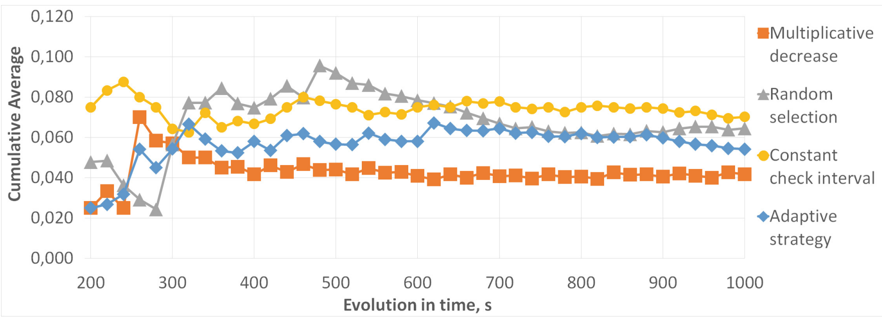

Fig. 6. Experiemental behavior of the cumulative average: Poissonian losses, $\beta = \gamma = 1$.

Table 4. Performance comparison based on the metric $C_{\text{cum}}(n; \beta, \gamma)$.

Loss distribution	Cost		Strategy performance in terms of $C_{\text{cum}}(n; \beta, \gamma)$			
	β	γ	Multiplicative–decrease	Random selection	Constant check interval	Adaptive strategy
Uniform losses	0.5	1.5	0.016	0.010	0.009	0.007
	0.75	1.25	0.028	0.021	0.022	0.016
	1	1	0.053	0.045	0.052	0.041
	1.25	0.75	0.102	0.098	0.126	0.109
	1.5	0.75	0.204	0.215	0.307	0.308
Poissonian losses	0.5	1.5	0.006	0.009	0.010	0.008
	0.75	1.25	0.015	0.023	0.024	0.0019
	1	1	0.038	0.054	0.057	0.046
	1.25	0.75	0.097	0.130	0.137	0.119
	1.5	0.75	0.251	0.329	0.340	0.325
Two-state alternated Poissonian losses	0.5	1.5	0.011	0.014	0.012	0.012
	0.75	1.25	0.025	0.034	0.030	0.028
	1	1	0.059	0.087	0.077	0.067
	1.25	0.75	0.146	0.224	0.201	0.170
	1.5	0.75	0.370	0.594	0.530	0.453

For further performance comparison, we consider the cumulative average

$$C_{\text{cum}}(j; \beta, \gamma) = \frac{1}{j} \sum_{i=1}^{j} \frac{k_i^{\beta}}{t_i^{\gamma}}, \quad j = 1, 2, \ldots, n, \quad \beta + \gamma = 2, \quad \beta, \gamma > 0$$

Parameters β and γ allow defining the performance tradeoff between losses and check interval. Clearly, the case of Q_{tot} and Q_{avg} corresponds to $\beta = \gamma = 1$, i.e., losses and check interval have the same cost influence to the performance. In particular, $C_{\text{cum}}(n; 1, 1) = Q_{\text{avg}}$. Experimental behavior of $C_{\text{cum}}(j; 1, 1)$ for the case of Poissonian losses is shown in Fig. 6.

If $\beta < 1 < \gamma$ then longer check intervals are preferable, i.e., many active checks degrade the performance more significantly than a high number of occurred losses. If $\beta > 1 > \gamma$ then less loss rate is preferable, i.e., many losses degrade the performance more significantly than a high rate of check requests. Our experiment results with varying β and γ are summarized in Table 4. When notification losses are considered less expensive then the multiplicative–decrease strategy becomes more efficient.

5 Conclusion

This paper has studied the problem of efficient notification delivery for the subscription operation in smart spaces. We considered different assumptions on the notification loss distribution in wireless networked environments. We introduced several control strategies, including a proposal for a TCP-like control for adaptation of the check interval to the observable loss rate. A client can implement such a strategy individually to increase the delivery efficiency. The presented strategies are provided with initial analytical estimates and simulation experiments for performance comparison.

Acknowledgments. This research is financially supported by the Ministry of Education and Science of the Russian Federation: project # 14.574.21.0060 (RFMEFI57414X0060) of Federal Target Program "Research and development on priority directions of scientific-technological complex of Russia for 2014–2020" and project # 1481 of the basic part of state research assignment for 2014–2016. The work of M. Pagano is partially supported by the PRA 2016 research project 5GIOTTO funded by the University of Pisa.

References

1. Allman, M., Paxson, V., Blanton, E.: TCP Congestion Control. RFC 5681 (Draft Standard), September 2009. http://www.ietf.org/rfc/rfc5681.txt
2. Bogoiavlenskaia, O.: Discrete model of TCP congestion control algorithm with round dependent loss rate. In: Balandin, S., Andreev, S., Koucheryavy, Y. (eds.) NEW2AN/ruSMART 2015. LNCS, vol. 9247, pp. 190–197. Springer, Heidelberg (2015)
3. Callegari, C., Giordano, S., Pagano, M., Pepe, T.: A survey of congestion control mechanisms in linux TCP. In: Vishnevsky, V., Kozyrev, D., Larionov, A. (eds.) DCCN 2013. CCIS, vol. 279, pp. 28–42. Springer, Heidelberg (2014)
4. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The many faces of publish/subscribe. ACM Comput. Surv. **35**, 114–131 (2003)
5. Honkola, J., Laine, H., Brown, R., Tyrkkö, O.: Smart-M3 information sharing platform. In: Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2010), pp. 1041–1046. IEEE Computer Society, June 2010
6. Kleinrock, L.: Queueing Systems: Theory, vol. 1. Wiley Interscience, New York (1975)
7. Korzun, D.: Service formalism and architectural abstractions for smart space applications. In: Proceedings of the 10th Central & Eastern European Software Engineering Conference in Russia (CEE-SECR 2014). ACM, October 2014
8. Korzun, D., Pagano, M., Vdovenko, A.: A TCP-like control of notification delivery for subscription operation in smart spaces. In: Proceedings of the 18th International Conference on Distributed Computer and Communication Networks: Control, Computation, Communications (DCCN-2015), pp. 10–18. ICS RAS, October 2015
9. Korzun, D.G., Kashevnik, A.M., Balandin, S.I., Smirnov, A.V.: The Smart-M3 platform: experience of smart space application development for internet of things. In: Balandin, S., Andreev, S., Koucheryavy, Y. (eds.) NEW2AN/ruSMART 2015. LNCS, vol. 9247, pp. 56–67. Springer, Heidelberg (2015)
10. Lomov, A.A., Korzun, D.G.: Subscription operation in Smart-M3. In: Balandin, S., Ovchinnikov, A. (eds.) Proceedings of the 10th Conference of Open Innovations Association FRUCT and 2nd Finnish-Russian Mobile Linux Summit, pp. 83–94. SUAI, November 2011

11. Morandi, F., Roffia, L., D'Elia, A., Vergari, F., Cinotti, T.S.: RedSib: a Smart-M3 semantic information broker implementation. In: Balandin, S., Ovchinnikov, A. (eds.) Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Tourism, pp. 86–98. SUAI, November 2012
12. Peterson, L.L., Davie, B.S.: Computer Networks: A Systems Approach. The Morgan Kaufmann Series in Networking, 5th edn. Morgan Kaufmann Publishers Inc., San Francisco (2012)
13. Smirnov, A., Kashevnik, A., Shilov, N., Oliver, I., Balandin, S., Boldyrev, S.: Anonymous agent coordination in smart spaces: state-of-the-art. In: Balandin, S., Moltchanov, D., Koucheryavy, Y. (eds.) ruSMART 2009. LNCS, vol. 5764, pp. 42–51. Springer, Heidelberg (2009)
14. Vdovenko, A.S., Korzun, D.G.: Active control by a mobile client of subscription notifications in smart space. In: Proceedings of the 16th Conference Open Innovations Framework Program FRUCT, pp. 123–128. IEEE, ITMO University, October 2014
15. Weiss, N., Holmes, P., Hardy, M.: A Course in Probability. Pearson Addison Wesley, New York (2005)