Study of Active Subscription Control Parameters in Large-Scale Smart Spaces

Andrey S. Vdovenko, Olga I. Bogoiavlenskaia, Dmitry G. Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{vdovenko, olgbv, dkorzun}@cs.karelia.ru

Abstract—**The development of smart spaces for Internet of Things (IoT) environments meets the scalability challenge since many participants are involved into the distributed computation. In particular, many sensors continuously provide data, many reasoners analyze the sensed data to construct services, and many mobile clients regularly join and leave the smart space to consume the services. The network interaction is information-driven, using a semantic information broker, which implements a passive strategy for subscription. The strategy leads to performance bottleneck. We continue our study of the active control strategy, when a part of subscription processing is delegated to the subscribers. The client multiplicatively decreases its check interval, if subscription notifications are lost, and additively increases the interval, otherwise. We study the role of strategy parameters: the client can select their values preliminary and keep fixed, or the client can change them dynamically depending on the recent situation. With the aim we provide mathematical model which represents important performance metrics as a function of strategy parameters. Analytical result are validated by experimental evaluation. Additionally, our simulation experiments evaluate the scalability in dependence on the number of participants.**

I. Introduction

The evolving concept of Internet of Things (IoT) increases the role of distributed processing of the data from multiple heterogenous sources by multiple dynamic participants [1]. The smart spaces suit of technologies is used for creating a certain class of intelligent service-oriented environments [2], [3]. A shared view on information is provided to all participants via a semantic information broker (SIB). They interact by producing, processing, and consuming this shared information with focus on its semantics.

A core operation for this information-driven interaction is subscription [4]. The operation implements a persistent query for SIB to notify regularly about updates to all matched subscribers [5], [6]. When all processing for matching updates with any subscriber is delegated to SIB this passive strategy suffers from low performance, especially in large-scale IoT settings. In this paper, we continue our research on active control by mobile smart space participants for subscription [7]. A subscriber regularly checks for updates, in addition to passive responses from SIB.

Following the strategy, the client multiplicatively decreases its check interval, if subscription notifications are lost, and additively increases the interval, otherwise. This adaptive strategy was early experimented in [7] as a generalization of the TCP algorithm additive–increase/multiplicative–decrease (AIMD) [8]. Its convergence properties were studied in [9].

The dynamic changes the client performs depend on predefined strategy parameters (e.g., the multiplication level). This paper studies the role of strategy parameters: the client can select their values preliminary and keep fixed, or the client can change them dynamically depending on the recent situation. The study problem is to understand what parameters values for current situation lead to the better performance for entire system without significant losses for the client. For that purpose, we develop mathematical models to analytically estimate the dependency of performance on the strategy parameters. We choose a criteria by which the estimates should correspond to the upper bound of the experimental values.

For experimental evaluation, we consider a large-scale class of smart space applications, which now appears in many practical domains, e.g., see [4] and references therein. Participants are classified into the following groups: 1) sensors, 2) reasoners, and 3) mobile clients. We focus on the subscription performance for the latter group when every client uses active subscription control. Such a client adapts its check interval for requesting the SIB on updates. Evaluation of the performance scalability in these large-scale settings can be found in [10].

The rest of the paper is organized as follows. Section III defines the studied class of large-scale smart space applications. Section IV provides analytical performance estimates of active control at the individual level of the mobile client. Section V complements the theoretical study with simulation experiments. Section VI concludes the paper.

II. Related work

The problem of adaptation to data losses and development of its solutions for network performance improvement form a topical research direction in distributed systems. A particular reference area is Transmission Control Protocol (TCP). The protocol uses several algorithms for congestion control. Each aims at control of the sending rate by manipulating the congestion window that limits the number of outstanding unacknowledged bytes of data [8], [11]. A popular TCP algorithm is additive–increase/multiplicative–decrease (AIMD) [12]. The congestion window is increased by one maximal segment size per round resulting in linear increase. When the TCP transfer encounters congestion (packet losses are detected), the window is decreased. This algorithm cannot straightforwardly be used for notification losses in smart spaces. There is no congestion window for notifications, and agents implements active checks for content updates to detect losses. Nevertheless, many TCP algorithms are well established and used in other network performance problems [13]. Importantly that configuring the

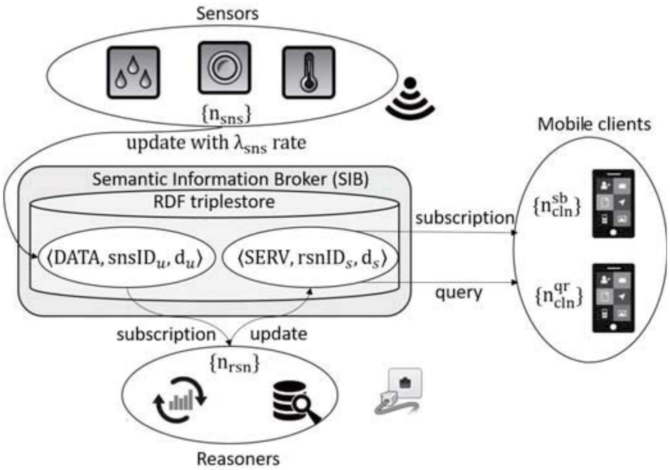

Fig. 1. A smart space application with $n_{\rm sns}$ sensors (data production), $n_{\rm rsn}$ reasoners (service construction), and $n_{\rm cln}$ mobile clients (service delivery)

TCP algorithms parameters based on mathematical modeling represents a promising approach in this area [14].

Another topical research and development direction considers performance of wireless connections in different conditions. In work [15], the author describes a backoff protocol to take into account the selfish user behavior and to implement a technique of an arbiter. The protocol improves the overall performance in the case of selfishness of end stations. In addition, the author suggests reputation metric functions to use in controlling the behavior of the stations. In work [16], the authors study the reliability and timeliness for publish/subscribe services over Wide Area Network (WAN). They use gossiping to retrieve missing packets in the case of incomplete information as well as network coding to reduce the number of retransmissions and, consequently, the latency.

In our previous work [7] on the problem of notification loss in a smart space, we study the subscription operation when the notification delivery to a client is subject to losses. We considered different assumptions on the notification loss distribution in wireless networked environments. We introduced several control strategies, including our proposal of the adaptive strategy where the client adapts its check interval to the observable loss rate, i.e., for TCP-like control. In other revious work [17], we analyzed the adaptive strategy of active control on the convergence property: the speed to reach the steady state when the notification loss distribution is fixed. Our analytical and experimental evaluation showed that the convergence speed is reasonable for such an IoT-enabled application domain as digital services in collaborative work environments. This study continues these previous works, and we focus on the problem of parameters selection and recalculation for the adaptive strategy.

III. Large-scale smart space application

The section describes experimental environment designed to research typical IoT smart space sensor based application. An IoT environment can include many devices from the Internet edges. Sizes of several hundred and thousand of devices are now typical, although even two or slightly more devices can form a small IoT environment. In general, multi-party networked interaction of heterogeneous devices (in small, medium, or large IoT environments) still needs more effective solutions [4]. In this study, we introduce an application model for a wide class of smart space applications deployable in IoT environments [18]–[20].

The following groups of participants are involved: 1) sensors, 2) reasoners, and 3) mobile clients, as depicted in Fig. 1. Sensors produce data flows, which lead to regular updates in the smart space. That is, a sensor represents a persistent data source. Reasoners detect appropriate updates and make data mining to deduce knowledge, which is then delivered to the users as a service. That is, a reasoner represents a primary service constructor. Mobile clients detect service construction activity of the reasoners and deliver the services to the end-users (e.g., visualization). That is, a client running on the personal mobile device represents an access&consumption service point to the end-user. The size of each group can be relatively large, (e.g., up to several thousand of sensors).

The previous study has demonstrated the outperformance of CuteSIB compared with other SIB implementations [21]. Nevertheless, the passive strategy for subscription still suffers from low performance. Earlier we showed that regular active requests from a mobile client improve the performance by reducing workload on SIB [22]. We focus on parameters of this active control, when the client follows the adaptive strategy to rationally adapt the check interval to recent update rates.

In the considered case, each sensor regularly updates its data value in SIB. The individual update rate is $\lambda_{\rm sns} > 0$ (update requests per second, $\rm s^{-1}$). The total number of sensors $n_{\rm sns}$ defines the primary size parameter for application scalability. For simplicity we assume that the time between two consecutive updates is selected uniformly at random. The sensed data are shared in the smart space to represent source information provided by all the sensors using the following triples ($n_{\rm sns}$ in total):

$$\langle \mathrm{DATA}, \mathrm{snsID}_u, d_u \rangle \ , \quad u = 1, 2, \ldots, n_{\rm sns}, \qquad (1)$$

where DATA shows that the triple represents sensed data, snsID_u is unique identifier of the sensor u, and d is the latest data value sensed by u.

Each reasoner corresponds one-to-one to a service s. The number of the reasoners $n_{\rm rsn}$ is comparatively small ($n_{\rm rsn} \ll n_{\rm sns}$) since a reasoner analyzes data produced by many sensors. Using its identifier rsnID_s the reasoner s determines the appropriate set of sensors u by computing their identifiers snsID_u. The sets are non-overlapped since the model assumption is that each service has own set of source data. Reasoner s subscribes to its set of the sensed data (using snsID_u). Whenever a sensor makes an update, the reasoner is notified to construct the service. Reasoner s reads the updated triples from the smart space, makes local processing, and shares the result in the smart space using the following triples ($n_{\rm rsn}$ in total):

$$\langle \mathrm{SERV}, \mathrm{rsnID}_s, d_s \rangle \ , \quad s = 1, 2, \ldots, n_{\rm rsn}, \qquad (2)$$

where SERV shows that the triple represents service data, rnsID_s is unique identifier of reasoner s (service), and d is the information inferred by s from the source sensed data. Note that reasoner s implements semantic linking (not explicitly represented in the smart space) between triples (1) and (2) by computing snsID_u locally.

Each mobile client is interested in any service s from the set constructed by the n_{rsn} reasoners. The worst-case model assumption is that a mobile client has to be able to access all the services. Service delivery is implemented using subscription to all triples in (2): a client is notified when a service s has been constructed. We divide the clients onto two groups: a) passive strategy for subscription notifications and b) active control strategy when the client periodically checks the service. There are $n_{\mathrm{cln}}^{\mathrm{sb}}$ and $n_{\mathrm{cln}}^{\mathrm{qr}}$ clients in total. In our application model, we assume the size relation

$$n_{\mathrm{rsn}} \ll n_{\mathrm{cln}}^{\mathrm{sb}} + n_{\mathrm{cln}}^{\mathrm{qr}} \ll n_{\mathrm{sns}}. \tag{3}$$

This model assumption makes $n_{\mathrm{cln}} = n_{\mathrm{cln}}^{\mathrm{sb}} + n_{\mathrm{cln}}^{\mathrm{qr}}$ the primary size parameter for subscription scalability. The environment described is used to develop analytical model of the adaptive strategy for active subscription control.

IV. Performance of active control

Following our previous work [7], we consider the adaptive strategy of active subscription control. It implements “adaptation to losses” when the client reduces its check interval if losses are observed and increases the check interval, otherwise. In fact, the adaptive strategy is a generalization of the TCP algorithm of additive–increase/multiplicative–decrease (AIMD).

Generalized AIMD-like adaptive strategy has the following form. Let $i = 1, 2 \dots$ be a sequence of the checks done by the client, t_i be the time period between consecutive checks $i-1$ and i, and k_i be the number of losses during t_i. At the end of t_i the client makes decision about the next t_{i+1} period using $t_{i+1} = f(t_i, k_i)$. In the simplest case, we straightforwardly apply the AIMD algorithm as follows.

$$t_{i+1} = \begin{cases} t_i/\alpha, & k_i > 0 \\ t_i + \delta & k_i = 0, \end{cases} \tag{4}$$

where $\alpha > 1$ stands for decrease and $\delta > 0$ for increase values of check interval length. Fig. 2 illustrates a typical evolution defined by (4).

Now we make a formal statement of the mathematical model, which describes behavior of t_i period. From this model we derive upper estimates of the expected length of check interval before a multiplicative decrease, the number of consecutive growths and losses metric for different types of a loss flow. Denote X_n the sequence of multiplicative decreases, i.e., $X_n = t_j$ if $t_{j+1} = t_j/\alpha$ for some j. A multiplicative decrease happens after each X_n. Consider the aggregated periods

$$S_n = \sum_{i=j}^{m} t_i,$$

where $j = i$ such that $t_i = X_n$ and $m = i$ such that $t_i = X_{n+1}$. Let us assume that the sequence $\{S_n\}_{n \le 0}$ forms a renewal process with an absolutely continuous renewal function $F(x)$ and $\mathsf{E}[S_n] = 1/\lambda^*$ [23]. Then the sequence $\{X_n\}_{n \ge 0}$ possesses the Markovian property. The following presentation holds (in accordance with [24], [25]).

$$\left(X_{n+1} + \frac{X_n}{\alpha}\right)\left(X_{n+1} - \frac{X_n}{\alpha}\right)\frac{1}{2\delta} = S_n\,. \tag{5}$$

Then

$$X_{n+1}^2 - \frac{X_n^2}{\alpha^2} = 2\delta S_n\,.$$

Now let us denote $Y_n = X_n^2$. Then (5) is transformed to

$$Y_{n+1} = \frac{Y_n}{\alpha^2} + 2\delta S_n\,. \tag{6}$$

According to [25], formula (6) is a particular case of stochastic linear difference equation with the stationary solution

$$Y_n^* = 2\delta \sum_{k=0}^{\infty} \frac{1}{\alpha^{2k}} S_{n-1-k}\,.$$

Under the assumptions above, starting from arbitrary window size Y_0, the sequence Y_n converges almost sure to the stationary solution, i.e.,

$$\mathbf{P}\left\{\lim_{n\to\infty} |Y_n - Y_n^*| = 0\right\} = 1.$$

To calculate expectation for the stationary solution we derive

$$\mathsf{E}[Y_n^*] = 2\delta \sum_{k=0}^{\infty} \frac{1}{\alpha^{2k}} \mathsf{E}[S_{n-1-k}] = \frac{2\delta}{\lambda^*} \sum_{k=0}^{\infty} \frac{1}{\alpha^{2k}}\,.$$

Then for $\mathsf{E}[Y_n^*]$ holds

$$\mathsf{E}[Y_n^*] = 2\frac{\alpha^2 \delta \mathsf{E}[S_n]}{\alpha^2 - 1}$$

and using Goelder’s inequality [26] one can obtain

$$T = \mathsf{E}[X_n^*] \le \alpha \sqrt{\frac{2\delta}{\lambda^*(\alpha^2 - 1)}}\,, \tag{7}$$

where T is the expected length of t_i before a multiplicative decrease.

Now we derive an upper bound for another important performance metric K that counts the average number of losses

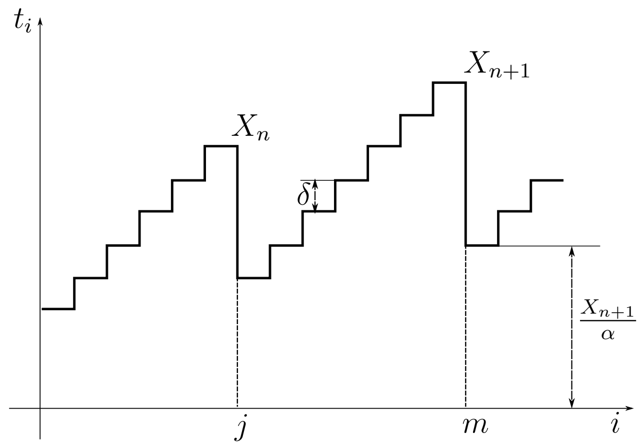

Fig. 2. Evolution of t_i in active control by (4).

at X_n interval. We consider three different types of a loss flow sensed by a single client.

1) Poisson flow with parameter λ. This model interprets that the losses visible at client side are provided by the different low intensity sources independently. Then superposition of such flows could be approximated by Poison flow with rate λ. In the case one can derive that
$$K \approx \lambda T \leq \lambda \sqrt{\frac{2\delta}{\lambda^*(\alpha^2-1)}}\,.$$
2) Bulk loss flow. This model assumes that delivery notification losses form bulk flow with arbitrary size n. The bulk size is described by the discrete distribution $\{p_n\}_{n=0}^{\infty}$ and bulk events form renewal flow $\{S_n\}$. Although loss events do not form bulks, this assumption does not reduce precision and generality of our analysis since losses happen at each X_n interval with probability 1 and intervals between losses within single X_n period are of no interest here. To compute K we denote
$$\mu = \sum_{n=0}^{\infty} p_n,$$
then $K = \mu$ and could be estimated from the observations.
3) Bernoulli losses. We assume that the notification losses happen independently with probability p. Therefore the expectation of losses in the sequence of N notification requests is Np and since a client keeps individual update rate λ_{cl} the expectation of the request number on X_n interval is $\lambda_{cl}T$ and hence $K \approx \lambda_{cl}Tp$.
Active strategy under consideration assumes that a client receives information about lost notifications only after request. If notifications were not received by the client during maximum request waiting time t_{max} and time is expired then all current notifications are considered as lost. Therefore if X_n period is longer than the maximum request waiting time then the client observes bulk notification losses. Therefore to remain within Bernoulli loss model and to avoid bulk losses the analyst is recommended to tune parameters α and δ so that T is not much greater t_{max}.

The number of consecutive growths is another important metric for the performance of adaptive strategy (4). The average number N is estimated as
$$N = \frac{1}{\delta}\mathsf{E}\left[\frac{X_n^*(\alpha-1)}{\alpha}\right] \leq \alpha\sqrt{\frac{2\delta(\alpha+1)}{\lambda^*(\alpha-1)}}. \qquad (8)$$
For the binary reduction case $\alpha = 2$ in (4) the following estimation is possible:
$$N = \frac{1}{\delta}\mathsf{E}\left[\frac{X_n^*}{2}\right] \leq \frac{\alpha}{\sqrt{2\delta\lambda^*(\alpha^2-1)}}\,.$$

V. Experimental evaluation

Our simulation testbed consists of a set of desktop computers. Each allows simulating many participants (small devices and their agents). A desktop computer hosts many parallel processes such that a process simulate the activity of a single participant (from one of the three groups). We use five modest-capacity computers to allocate the SIB, $n_{\rm sns}$ sensors, $n_{\rm cln}^{\rm sb}+n_{\rm cln}^{\rm qr}$ clients, and $n_{\rm rsn}$ reasoners. The testbed specification is summarized in Table I. Local wireless network is used except for the computer running reasoners (wired connection is used). In practice, reasoners typically need powerful machines equipped high-capacity communication channel, therefore our simulation testbed follow this feature. We experimented with CuteSIB version 0.5.0 (sourceforge.net/projects/smart-m3/), which is implemented using C++ and Qt.

Table I. Computers to allocate the SIB and simulated participants

Hosting	Workload	Characteristics
SIB	network connections with all participants	CPU Intel Core i3, CPU 1.90 GHz, RAM 4Gb, wired connection with 100 Mbps, Ubuntu 15.10
Sensors	$n_{\rm sns} \leq 10^4$ parallel processes	CPU Intel Dual Core, CPU 2.60 GHz, RAM 2Gb, wired connection with 100 Mbps, XUbuntu 16.04
Reasoners	$n_{\rm rsn} \leq 10^2$ parallel processes and passive subscriptions	CPU Intel Core i5, CPU 1.70 GHz, RAM 6Gb, wireless connection with 21 Mbps, Ubuntu 15.10
Clients	$n_{\rm cln} = n_{\rm cln}^{\rm sb} + n_{\rm cln}^{\rm qr} \leq 10^3$ parallel processes and active subscriptions	CPU Intel Core i5, CPU 2.50 GHz, RAM 3Gb, wireless connection with 21 Mbps, XUbuntu 16.04
Reference sensor, reasoner, and client	3 parallel processes	CPU Intel Core i5, CPU 2.50 GHz, RAM 3Gb, wireless connection with 21 Mbps, XUbuntu 16.04

All simulated participants (sensors, reasoners, and clients) are implemented using Python 2.7. The Smart-M3 PythonKPI library is used for SIB access primitives. In size relation (3), we fixed the case when
$$n_{\rm rsn} \leq 10^2,\;\; n_{\rm cln}^{\rm sb} = 0,\; n_{\rm cln}^{\rm qr} \leq 10^3,\;\; n_{\rm sns} \leq 10^4.$$
That is, the proportion $n_{\rm rsn} : n_{\rm cln} : n_{\rm sns} = 1 : 10 : 10^2$ is kept. The large number of sensors feeds the smart space with raw data. A small number of reasoners covers the whole data set. Each reasoner constructs its own service by tracking updates and processing the raw data, i.e., one service uses many sensors. The number of mobile clients is in the middle, i.e., one service targets several end-users.

The sum rate of operations for all three group (sensors, reasoners, clients) is 10 op/s (operations per second). We selected one reference participant from each of three groups for the evaluation. The reference sensor has individual rate based on random delay from 0 to 3 seconds (uniform distribution). For each experiment, 100 consecutive measurements are made. The reference client is used for performance evaluation. Other mobile clients generate background workload. The adaptive strategy for active control (4) uses $t_0 = 3$ s. Its parameters α and δ were varied to study their influence on the performance.

We compare the analytical estimates derived in Section IV with the simulation results. Using the experimental measurements we calculate sequence S_n and then predicted and experimental values of X_n expectation, where

- $T_{\rm pred}$ is calculated by (7),
- $T_{\rm exp}$ is the average of X_n^*,
- $N_{\rm pred}$ is calculated by (8),

Table II. COMPARISON TABLE OF ESTIMATIONS AND EXPERIMENTAL VALUES FOR DIFFERENT ACTIVE STRATEGY PARAMETERS

Strategy parameters	Evaluated values										
	$\mathsf{E}[S_n]$	T_{pred}	T_{exp}	N_{pred}	N_{exp}	λ_{cln}	$K_{\text{Poisson flow}}$	$K_{\text{bulk loss}}$	$K_{\text{Bernoulli}}$	K_{tot}	R_{exp}
$\delta = 0.5,\ \alpha = 1.5$	13	4.84	4.13	4.84	2.74	0.29	0.37	1.00	0.32	0.23	9
$\delta = 1,\ \alpha = 1.5$	10.36	6.11	4.79	3.05	1.55	0.25	0.57	1.00	0.42	0.28	1
$\delta = 2,\ \alpha = 1.5$	9.2	8.14	5.60	2.03	0.94	0.21	1.07	1.21	0.69	0.4	4
$\delta = 3,\ \alpha = 1.5$	9.2	8.14	5.60	0.99	0.32	0.21	3.34	2.64	1.34	0.4	4
$\delta = 0.5,\ \alpha = 2$	14.96	4.47	4.02	4.47	3.85	0.32	0.28	1.00	0.28	0.19	13
$\delta = 1,\ \alpha = 2$	9.74	5.1	4.16	2.55	2.10	0.32	0.52	1.00	0.50	0.31	27
$\delta = 2,\ \alpha = 2$	6.50	5.89	4.17	1.47	1.04	0.31	0.91	1.00	0.85	0.46	41
$\delta = 3,\ \alpha = 2$	9.2	8.14	5.60	1.15	0.77	0.21	2.63	2.22	1.21	0.4	4
$\delta = 0.5,\ \alpha = 4$	4.86	2.28	1.92	2.28	2.79	0.77	0.58	1.21	0.61	0.35	40
$\delta = 1,\ \alpha = 4$	4.84	3.21	1.94	1.61	2.82	0.77	0.82	1.22	0.69	0.28	32
$\delta = 2,\ \alpha = 4$	5.96	5.04	3.76	1.26	1.37	0.40	1.60	1.89	1.02	0.51	14
$\delta = 3,\ \alpha = 4$	5.00	5.66	3.84	0.94	0.94	0.39	1.82	1.61	1.16	0.53	16

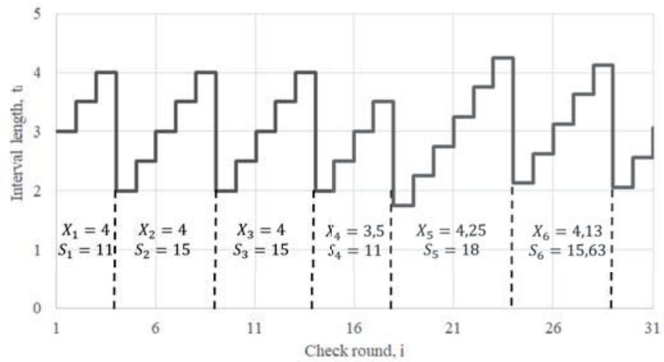

Fig. 3. Adaptive strategy for $\alpha = 2$ and $\delta = 0.5$ with $t_0 = 3$ s.

- N_{exp} is the average of X_i,
- K the average number of losses before an interval decrease (for different loss distributions),
- λ_{cln} is the rate of client requests checking for updates,
- K_{tot} is the proportion of update losses during the entire evolution interval,
- R_{exp} is the percent of requests with no update detection (redundant requests).

Basic experimental results summarized in Table II. They indicate good satisfaction with the analytical upper bounds which correlate with our criteria for analytical estimates. K metric that smaller or equal to 1 indicates that selected parameters are suitable with current situation. K_{tot} should strive to zero, but with current strategy behavior it's not possible, because increases are infinite until loss event happens. R_{exp} and λ_{cln} should be as small as possible without harm in the form of additional losses. In particular, Fig. 3 shows that values $\alpha = 2$, $\delta = 0.5$ provide a reasonable balance between redundant requests and update losses.

Fig. 4 shows experimental measurement of the average number of losses K (before an interval decrease) for different values of δ and for fixed $\alpha = 2$. The adaptive strategy behaves reasonably for different δ. Metric $K_{\text{bulk loss}}$ is totally equal to experimental values because of the calculation method, it can be used in situations when we have information about previous observation. Metrics $K_{\text{Bernoulli}}$ and $K_{\text{Poisson flow}}$ are close to $K_{\text{bulk loss}}$ at values such as $\delta = 0.5, 1, 2$. The observed behavior is due to the rate of sensor $\lambda_{sns} = 1.5$, which is much bigger or comparable with values of δ mentioned above. When interval increases further and decreases smaller than sensor rate, it is in-effective leading to bigger losses, e.g., for values of $\delta = 3, 4, 5, 6$. The adaptive strategy needs to make several rounds to stabilize interval value but on small number of rounds.

The client request rate λ_{cln} defines how many requests was made per time unit. Fig. 5 demonstrates behavior of λ_{cln} for different values of δ and α as well as interdependence of losses percent at all intervals. Higher rate leads to smaller percent of losses. It is at the expense of frequent requests with small delays between them. Note that at same time too big δ and α values lead to worse losses percent. The case $\alpha = 2$ provides balanced values of the metrics, i.e., the binary reduction (halving) is preferable in the active control.

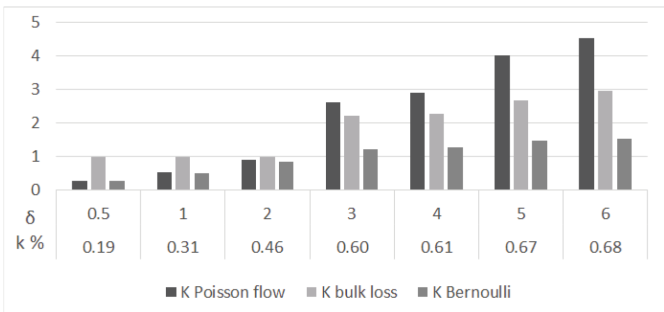

Fig. 4. Comparison of different types of losses for $\alpha = 2$.

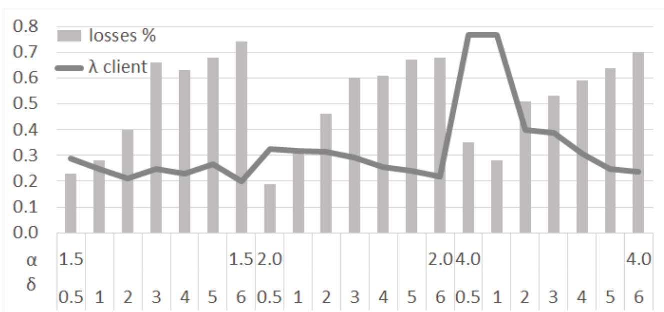

Fig. 5. Client request rate λ_{cln} for different values of δ and α.

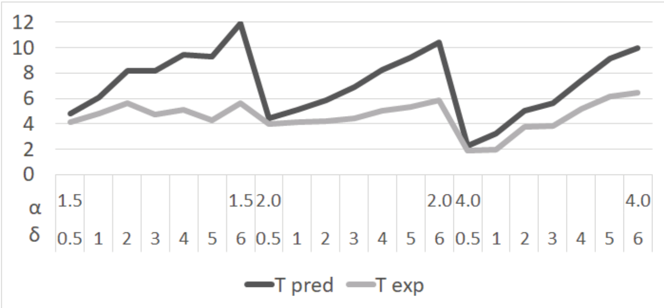

Fig. 6. Length of interval t_i before a decrease by analytical estimations and experiments for different values of δ and α.

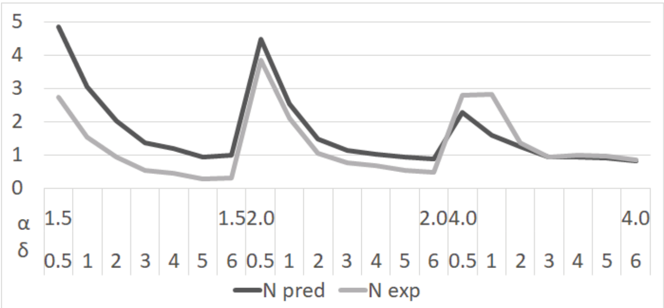

Fig. 7. The number of consecutive growths N by estimations and experiments for different values of δ and α.

Fig. 6 shows comparison of the interval length estimations with experimental results for different values of δ and α. In this case, we observe the same situation as with the K metrics: When $\delta > 3$ for $\lambda_{\text{sns}} = 1.5$ then the analytical estimations give much bigger values that experimental.

Fig. 7 shows comparison of estimation N_{pred} and experimental N_{exp} values of the number of consecutive growths N. As we can see the estimation provides a high bound for experimental values in most of cases. It can be used to determine the average length of interval before a decrease.

In sum, the achieved measurement results can be used to set up the strategy parameters in order to provide better performance. The results point out that the adaptive strategy needs to be customized during its operation to react on changes of rates in system. This customization can be achieved by appropriate modification of of α and δ with use of previous observation and knowing estimations for K metrics. As we can see from the experiments, values $\alpha = 2$, $\delta = 0.5$ provide a satisfactory balance between extra requests and update losses at $\lambda_{\text{sns}} = 1.5$.

VI. Conclusion

This paper continues our study of the active strategy implemented by a client in the Large-Scale Smart Spaces. We proposed adaptive AIMD-like algorithm to control notification check interval. We derive analytical estimates of the important performance metrics of the strategy. Namely, we obtain estimations for the expected length of check interval before a multiplicative decrease, the number of consecutive growths and losses metric for different types of a loss flow. The estimates obtained have simple closed representation, which makes them easily applicable. Also, we provide experimental evaluation of smart space application operation in the Large-Scale IoT environment that provides interaction of many sensors, reasoners and clients. Our simulation experiments showed that the analytical estimates could be applied for tunning parameters of the active strategy. Experimental evaluation demonstrated the active strategy could improve efficiency of notification delivery. Dynamic control of strategy parameters can lead to better performance and it can be achieved with use of estimations and predicting situations in system.

Acknowledgment

This research is financially supported by the Ministry of Education and Science of Russia within project # 2.5124.2017/8.9 of the basic part of state research assignment for 2017–2019. The work is implemented within the Government Program of Flagship University Development for Petrozavodsk State University in 2017–2021.

References

[1] V. Cristea, C. Dobre, and F. Pop, *Internet of Things and Intercooperative Computational Technologies for Collective Intelligence*. Springer Berlin Heidelberg, 2013, ch. Context-Aware Environments for the Internet of Things, pp. 25–49.

[2] J. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent environments: a manifesto," *Human-centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–18, 2013.

[3] D. G. Korzun, S. I. Balandin, A. M. Kashevnik, A. V. Smirnov, and A. V. Gurtov, "Smart spaces-based application development: M3 architecture, design principles, use cases, and evaluation," *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 8, no. 2, pp. 66–100, 2017.

[4] D. Korzun, A. Kashevnik, S. Balandin, and A. Smirnov, "The Smart-M3 platform: Experience of smart space application development for Internet of Things," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems. ruSMART 2015*, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds., vol. 9247. Springer, Aug. 2015, pp. 56–67.

[5] A. Smirnov, A. Kashnevik, N. Shilov, I. Oliver, S. Balandin, and S. Boldyrev, "Anonymous agent coordination in smart spaces: State-of-the-art," in *Smart Spaces and Next Generation Wired/Wireless Networking. Proc. 9th Int'l Conf. NEW2AN'09 and 2nd Conf. on Smart Spaces ruSMART 2009. LNCS 5764*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 42–51.

[6] A. A. Lomov and D. G. Korzun, "Subscription operation in Smart-M3," in *Proc. 10th Conf. of Open Innovations Association FRUCT and 2nd Finnish–Russian Mobile Linux Summit*, S. Balandin and A. Ovchinnikov, Eds. SUAI, Nov. 2011, pp. 83–94.

[7] D. Korzun, M. Pagano, and A. Vdovenko, "Control strategies of subscription notification delivery in smart spaces," in *Distributed computer and communication networks*, ser. Communications in Computer and Information Science (CCIS), V. Vishnevsky and D. Kozyrev, Eds. Springer International Publishing, 2016, vol. 601, pp. 40–51.

[8] C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "A survey of congestion control mechanisms in linux tcp," in *Distributed Computer and Communication Networks*. Springer, 2014, pp. 28–42.

[9] D. Korzun, A. Vdovenko, and O. Bogoiavlenskaia, "Convergence evaluation of adaptation to losses: The case of subscription notification delivery to mobile users in smart spaces," in *Distributed Computer and Communication Networks*, ser. Communications in Computer and Information Science. Springer International Publishing, 2016, vol. 678, pp. 504–522.

[10] A. Vdovenko, D. Korzun, and O. Bogoiavlenskaia, "Performance evaluation of active subscription control for service delivery in internet of things applications," in *Distributed computer and communication networks: control, computation, communications (DCCN-2017): proceedings of the 20-th international scientific conference*, vol. 4, Sep. 2017, in press.

[11] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The newreno modification to tcp's fast recovery algorithm," Tech. Rep., 2012.

[12] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," RFC 5681, September, Tech. Rep., 2009.

[13] L. Vasilescu, V. Olteanu, and C. Raiciu, "Sharing cpus via endpoint congestion control," in *Proceedings of the Workshop on Kernel-Bypass Networks*. ACM, 2017, pp. 31–36.

[14] N. S. Rao, Q. Liu, S. Sen, D. Towsley, G. Vardoyan, I. Foster, and R. Kettimuthu, "Tcp throughput profiles using measurements over dedicated connections," in *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 2017, pp. 193–204.

[15] A. Lukyanenko *et al.*, "Multi-user resource-sharing problem for the internet," 2010.

[16] C. Esposito, M. Platania, and R. Beraldi, "Reliable and timely event notification for publish/subscribe services over the internet," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 230–243, 2014.

[17] D. Korzun, A. Vdovenko, and O. Bogoiavlenskaia, "On convergence of active control strategies for subscription notification delivery in smart spaces," in *Distributed computer and communication networks: control, computation, communications (DCCN-2016): proceedings of the 19-th international scientific conference*, vol. 3, Nov. 2016, pp. 281–288.

[18] S. Balandin and H. Waris, "Key properties in the development of smart spaces," in *Proc. 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI '09). Part II: Intelligent and Ubiquitous Interaction Environments, LNCS 5615*, C. Stephanidis, Ed. Springer-Verlag, Jul. 2009, pp. 3–12.

[19] E. Ovaska, T. S. Cinotti, and A. Toninelli, "The design principles and practices of interoperable smart spaces," in *Advanced Design Approaches to Emerging Software Systems: Principles, Methodology and Tools*, X. Liu and Y. Li, Eds. IGI Global, 2012, pp. 18–47.

[20] D. Korzun, S. Balandin, and A. Gurtov, "Deployment of Smart Spaces in Internet of Things: Overview of the design challenges," in *Internet of Things, Smart Spaces, and Next Generation Networking*, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds., vol. 8121. Springer, Aug. 2013, pp. 48–59.

[21] F. Viola, A. D'Elia, D. Korzun, I. Galov, A. Kashevnik, and S. Balandin, "The M3 architecture for smart spaces: Overview of semantic information broker implementations," in *Proc. of the 19th Conference of Open Innovations Association FRUCT*, S. Balandin and T. Tyutina, Eds. IEEE, Nov. 2016, pp. 264–272.

[22] A. Vdovenko, I. Galov, and D. Korzun, "Performance scalability study of the smart-m3 cutesib implementation," in *Proc. 20th Conf. Open Innovations Framework Program FRUCT*, Apr. 2017, pp. 727–730.

[23] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. John Wiley & Sons London-New York-Sydney-Toronto, 1968, vol. 1.

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, pp. 303–314, 1998.

[25] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 231–242, 2000.

[26] D. S. Mitrinovic, J. Pecaric, and A. M. Fink, *Classical and new inequalities in analysis*. Springer Science & Business Media, 2013, vol. 61.