Quantitative analysis of access strategies to remote information in network services

Olsen, Rasmus Løvenstein; Schwefel, Hans-Peter; Hansen, Martin Bøgsted

Published in:
Global Telecommunications Conference, CLOBECOM 2006 - IEEE

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Quantitative analysis of access strategies to remote information in network services

Rasmus L. Olsen[1], Hans-Peter Schwefel[1], Martin B. Hansen[2]

[1]Center for TeleInFrastruktur, Aalborg University, Niels Jernes Vej 12, 9220 Aalborg, Denmark
[2] Dept. of Mathematical Sciences
Email:[1]{hps|rlo}@kom.aau.dk, [2]mbh@math.aau.dk

Abstract— **Remote access to dynamically changing information elements is a required functionality for various network services, including routing and instances of context-sensitive networking. Three fundamentally different strategies for such access are investigated in this paper: (1) a reactive approach initiated by the requesting entity, and two versions of proactive approaches in which the entity that contains the information element actively propagates its changes to potential requesters, either (2) periodically or triggered by changes of the information element (3). This paper first develops a set of analytic models to compute different performance metrics for these approaches, with special focus on the so-called mismatch probability. The results of the analytic models allow for design decisions on which strategy to implement for specific input parameters (change rate of the information element, network delay characterization) and specific requirements on mismatch probability, traffic overhead, and access delay. Finally, the analysis is applied to the use-case of context-sensitive service discovery.**

Keywords: Distributed systems, remote access, performance modelling, context-sensitive networking

I. Introduction

Timely, remote access to dynamically changing information elements is a common problem for a large range of functionalities in different layers of modern telecommunication networks:

- On the link-layer, efficient radio-resource management at base-stations requires information about channel state and buffer filling as measured in mobile devices.
- On the network layer, routing decisions require the knowledge about the existence and the characteristics of links between remote intermediate nodes. This is particularly relevant when topology changes are rather frequent such as in wireless multi-hop networks[1].
- Network Services, such as dynamic distributed data-bases as used in certain name-services in mobile networks, require knowledge about (remotely performed) updates of the name to address mapping [2].
- Context-sensitive services require access to typically remotely obtained context information. Context information may thereby be used both during service execution [3] as well as for the service discovery process [4].
- For highly dependable networks and services, resilience is obtained by replication of services, which requires state-updates at remote replicants in order to avoid inconsistency [5], [6], [7] .

Common to all these use-cases of access to remote information is that basic design decisions on how to efficiently implement such access need to be taken. Efficiency is thereby typically measured by access delay, probability of using 'correct' information, and network traffic overhead created by the remote access strategy. Two basic types of solutions exist:

1) **Reactive**, 'on-demand' access: Whenever a certain piece of remote information is needed at the processing entity, it is actively obtained (request) from the remote entity that has access to this information. This in principle implements a client-server architecture.
2) **Proactive** distribution of information: The entity that has control of the information element will pro-actively distribute updates of its value to potential 'requesters'. Thereby, two underlying sub-strategies can be distinguished
    a) **Event-Driven** proactive updates: Whenever the information element changes value, an update is triggered. For a further differentiation with respect to the semantics of these updates, see Sect. III-C.
    b) **Periodic** proactive updates: After certain time-intervals, the current value of the information element is distributed to potential request processes.

In this paper, we provide the methodology and the results of the quantitative analysis of different performance metrics, including in particular the so-called mismatch probability for the different remote access strategies. Section III introduces the different analytic models, while the quantitative results of these analytic models and their validation via simulations are discussed in Sect. IV. Finally, the analysis is applied to a use-case scenario of context-sensitive service discovery in Personal Networks in Sect. V.

II. Problem formalization and performance metrics

This section provides an abstracted description of the access procedures to remote information using stochastic processes. This description allows to analytically obtain different performance metrics, in particular including the so-called mismatch probability.

The simplified model contains three parts:

- The information element is maintained by a remote node (information provider) and it dynamically changes

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

its value at certain points in (continuous) time. It is assumed here that these changes will always result in a value previously not observed, e.g. as it is the case for monotonic changes. Since the actual value of the information element is not relevant in this paper (only the fact whether it has changed), we will use a point process $\mathcal{E} = \{E_i,\ i \in \mathbf{Z}\}$, where E_i is an increasing sequence of event times numbered such that E_0 is the event just before 0. The process $\mathcal{E}$ is called the **event process**. $E(t)$ denotes the value of the (monotonically increasing) information element at time t, see the Appendix.

- The remote information element is required by a certain entity (the requester/client) at certain moments in time, identified by the **request process**, $\mathcal{R} = \{R_k,\ k \in \mathbf{Z}\}$, which in turn is a point process denoted in the same way as the E_k's. Depending on the selected update strategy, an event of the request process may trigger an actual request to the remote server (reactive approach), or it may lead to an instantaneous access to the local replication of the information element in the pro-active approaches.
- Communication between requesting entity and server is described by stochastically varying delays, the upstream delays, $\{U_k,\ k \in \mathbf{Z}\}$ between requester and server (only in case of pro-active approaches), and the downstream delays, $\{D_k,\ k \in \mathbf{Z}\}$. Messages are never lost, however, these delays are potentially unbounded. Messages are identified by sequence numbers, so that out-dated updates can be detected and discarded.
  In this paper, we will limit our discussion to independent, identically distributed (iid) delay processes, corresponding in practice to cases in which the inter-message times are larger than the time-scales at which queues build up and drain in the network due to congestion. For fast core-networks, such an iid delay assumption is realistic. Furthermore, some of the performance metrics below, in particular the stationary mismatch probability in the reactive case, are insensitive to correlation properties of the delay processes.
  Random variables with the upstream and downstream delay distributions are denoted generically as $\mathcal{U}$ and $\mathcal{D}$, respectively. These delay distributions correspond to the end-to-end delays between information provider and requester, hence e.g. cases of wireless multi-hop communication can be included via appropriate choice of $\mathcal{U}$ and $\mathcal{D}$. Also, message drops can be included via degenerated distributions (with probability mass at infinity).

In this paper, we consider the following three performance metrics, where focus is put on the mismatch probability:

1) Network overhead: The amount of data transmitted on the network in the remote access strategy.
2) Access delay: The time interval from the moment when a certain piece of information is needed at the requester until it is finally available for use. For the pro-active access strategies, this delay is zero. Processing times are neglected in this paper.
3) Mismatch probability: The probability that the value of the information element that is used at the requester does not match the current true value at the remote location. The consequence of such a mismatch depends on the specific application, see e.g. Sect. V.

III. Analytic Models

A. Reactive, on-demand access

Figure 1 illustrates the message flows in the reactive approach. In this scheme a request is initiated by the client at

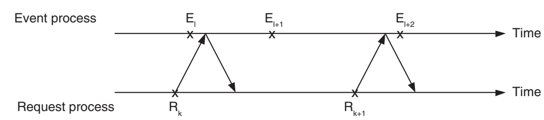

Fig. 1. Reactive access: In the example, the k-th access, R_k, leads to a 'correct' value, while the $k+1$th access causes a mismatching value.

time R_k, which is received by the provider at time $R_k + U_k$. The provider creates a response message containing the value of the requested information element which is received at the requester at time $R_k + U_k + D_k$. In the shown example, the k-th request leads to a correct value, since no changes of the information element occurred, while the response is being transmitted. The $(k+1)$-th request on the other hand leads to a mismatching value. For the assumption that the event process, $\mathcal{E}$, is a Poisson process with rate λ, the mismatch probability can be calculated as follows, see the appendix for the derivation:

$$\text{mmPr}_{react}(\lambda, \mathcal{D}) = 1 - \mathcal{L}\{f_{\mathcal{D}}\}(\lambda),$$

where the last term is the Laplace transform of the density of the down-stream delay, $\mathcal{D}$, evaluated at value λ. Note that the mmPr is independent of the request process, $\mathcal{R}$. However, $\mathcal{R}$ will influence statistical properties of corresponding estimators of mmPR. Note also, that the mmPr is not depending on the upstream delay process. Two different cases for the downstream delay are interesting and considered later in this paper:

- Constant (deterministic) delay: $\mathcal{D} \equiv c$

$$\text{mmPr}^{(det)}_{react}(\lambda, c) = 1 - \exp(-\lambda\, c). \tag{1}$$

- iid exponentially distributed delay with rate ν:

$$\text{mmPr}^{(exp)}_{react}(\lambda, \nu) = 1 - \frac{\nu}{\lambda + \nu} = \frac{\lambda}{\lambda + \nu}. \tag{2}$$

Note that the mmPr assuming an exponential delay is in fact smaller than in the deterministic setting with $c = 1/\nu$ for all values of λ, and ν (since $e^{-x} \geq (1+x)^{-1}$ for all values of x). The network overhead, $V_{react}(T, s, \mu)$, in a time interval of duration T is depending on the message sizes $s \in \{s_u, s_d\}$ (upstream and downstream, respectively), and the rate μ of the (not necessarily Poisson) request process: $V_{react}(T, s, \mu) = \mu T(s_u + s_d)$. The average access delay is $E\{\mathcal{U}\} + E\{\mathcal{D}\}$.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

B. Proactive - Periodic update

For the proactive case, no request messages are needed, but the remote note sends updates to the requester. First we discuss the 'periodic' version, i.e. the update is sent after some (potentially stochastically varying) time interval independent of event and request processes. See Figure 2 for an illustration. Assuming a Poisson process for the event process (with rate λ), iid exponentially distributed downstream delays with rate ν, and that the 'periodic' updates are determined by a third independent Poisson process with rate τ, the mismatch probability can be computed through the steady-state probabilities of the following 3-state continuous time Markov chain:

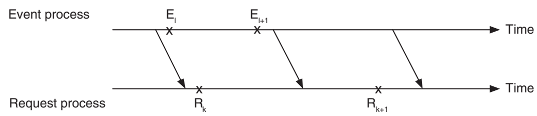

Fig. 2. Proactive periodic update using a deterministic period: R_k results in mismatching value, while R_{k+1} leads to a correct value.

State 1:	Correct value at requester
transitions:	event→S2, Update generated→S1, Update arriving→S1
State 2:	mismatch at requester, no correcting update in transit
transitions:	event → S2, Update generated → S3
State 3:	mismatch, correcting update in transit
transitions:	event→S2, Update generated→*S3, Update arriving→S1

Note that 'outdated' updates (which were sent out before the last event occurred) are irrelevant. If the delay-distribution is probabilistic, multiple updates in transit can occur, which would need to be counted in the state-space (transition marked with * above in the table). For simplicity, the table above and the generator matrix below do not implement this counting of updates in transit, although the numerical results in the subsequent sections include them. Under the Poisson assumptions on Event process, downstream delay, and update sending period, the state transitions can be described by the following generator matrix:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 \\ 0 & -\tau & \tau \\ \nu & \lambda & -\nu-\lambda \end{bmatrix}.$$

The mismatch probability is then the steady-state probability that the Markov process is in States S2 or S3, which has the following closed-form solution:

$$\text{mmPr}^{(exp)}_{proact,perodic}(\lambda,\nu,\tau) = \frac{\lambda\left[(\nu+\lambda)(2\tau+\nu+\lambda)+\tau^2\right]}{(\tau+\lambda)(\nu+\tau+\lambda)(\nu+\lambda)} \quad (3)$$

See [8] for the detailed analysis of the case with multiple updates in transit. The overhead can be computed as follows:

$$V_{proact,periodic}(T,s,\tau) = \tau T s_d.$$

The (average) access delay is 0.

C. Proactive - Event based update

In the proactive event driven update scheme, the provider sends an update to the client node, whenever an event has happened, i.e. when the information element has changed value, see Figure 3. In order to investigate the mismatch

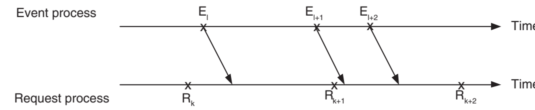

Fig. 3. Proactive event driven update: the request at time R_k results in a correct value, while R_{k+1} leads to a mismatch, since the updated value is in transfer when the user accesses the current value.

probability, two different cases with respect to the semantics of the update messages have to be distinguished:

Case I: Incremental Updates: In this scenario, the requester only accesses the correct information, if all update messages from previous events have been successfully received. These messages can be re-ordered by the network, but through the use of sequence numbers, the requester is able to put them back in the correct sequence. In this case, a mismatch would occur, if any of the update messages is still in transit. This is equivalent to the probability that an $\mathcal{E}/\mathcal{D}/\infty$ queue is in a busy period (a customer being served in the queue is equivalent to an update in transit). Hence, the mismatch probabilities can be computed as

$$\text{mmPr}^{(GI)}_{proact,incr}(\mathcal{E},\mathcal{D}) = \text{Pr}(\mathcal{E}/\mathcal{D}/\infty \text{ queue is busy}),$$

which reduces under Poisson assumptions for $\mathcal{E}$ (with rate λ) and General Independent (GI) assumptions for the downstream delay $\mathcal{D}$ (with mean $\bar{D}$) to

$$\text{mmPr}(\lambda,\bar{D}) = 1-\exp(-\lambda\bar{D}).$$

The downstream delay $\mathcal{D}$ can be GI, since the steady-state queue-length probabilities for the M/GI/∞ queue are identical to the M/M/∞ queue, see [9]. In case of constant (deterministic delay), the above mmPr is equivalent to the reactive case. Hence, for constant delay, re-active and proactive, incremental, event driven access strategies lead to the same mmPr.

Case II: Full updates: If a single update message contains all information so that previous updates are not needed at the requester, it is only important that the update message of the last event has reached the requester. Hence, the mmPr can be derived from a similar mapping to a queueing model, but here, only the last customer (event) is relevant. Hence, instead of an $\mathcal{E}/\mathcal{D}/\infty$ queue as in Case I, we are now in the setting of a finite $\mathcal{E}/\mathcal{D}/1/1$ queue with pre-emptive service and only a single customer in the system (a customer in service is pushed out and discarded by a newly arriving customer). Appendix B derives a general formula for the mmPr in that case, with the following two special cases:

- Constant (deterministic) delay: $\mathcal{D} \equiv c$

$$\text{mmPr}^{(det)}_{proact,full}(\lambda,c) = 1-\exp(-\lambda\, c). \quad (4)$$

- iid exponentially distributed delay with rate ν:

$$\text{mmPr}^{(exp)}_{proact,full}(\lambda,\nu) = 1-\frac{\nu}{\lambda+\nu} = \frac{\lambda}{\lambda+\nu}. \quad (5)$$

In both cases, the mmPr is identical to the corresponding reactive setting. In both cases, the overhead follows as:

$$V_{proact,event}(T, s, \lambda) = \lambda T s_d.$$

However, note that typically the message size for the incremental updates is (much) smaller than for the full updates. This difference depends on the complexity of the data structure of this information element, which is outside the scope of this paper. The (average) access delay is 0.

IV. Analytic results and validation via simulation

A summary of selected analytic result is given in the following table: Figure 4 shows the results for the mmPr

	reactive	proact. event full update	proact. event incremental	proact. periodic
mmPr				
Exp. Delay	$\frac{\lambda}{\lambda+\nu}$	$\frac{\lambda}{\lambda+\nu}$	$1 - e^{-\lambda/\nu}$	$\approx$ Eq. (3)
Det. Delay	$1 - e^{-\lambda c}$	$1 - e^{-\lambda c}$	$1 - e^{-\lambda c}$	see[8]
overhead	$\mu(s_u + s_d)$	$\lambda s_d^{(a)}$	$\lambda s_d^{(i)}$	$\tau s_d^{(a)}$
access delay	$E(\mathcal{U}) + E(\mathcal{D}) > 0$	0	0	0

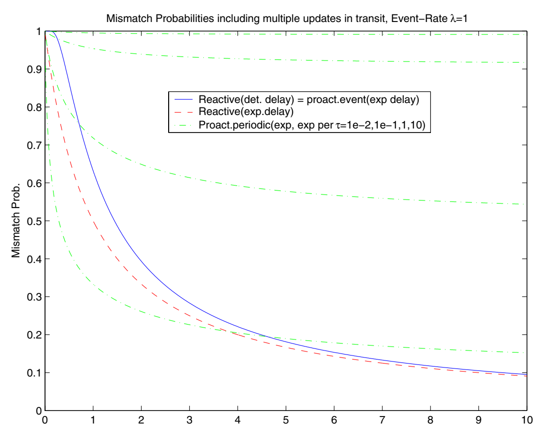

Fig. 4. Comparison of mismatch probabilities in the different remote access strategies:

as computed by the analytic models for the different remote access strategies, for the assumption of a Poisson event process with rate $\lambda = 1$ and an iid exponentially distributed downstream delay with rate ν. In the proactive periodic case, the period is iid exponentially distributed with varying rate $\tau = 10^{-2}, ..., 10$. The table and the figure allow to draw the following conclusions:

- For each of the cases, exponential delay and deterministic delay, the reactive approach leads to the same mmPr as the corresponding pro-active event driven approach with full updates. Hence, the latter is not shown in the figure.
- The reactive strategy in the case of deterministic downstream delays, $\mathcal{D} \equiv 1/\nu$, (solid line) leads to a higher mmPr than in the case of an exponentially distributed delay with same mean (dashed line). In contrast to intuition from other analytic models, e.g. in queueing models in which deterministic delays typically lead to shorter waiting times, here the deterministic case is not the best case scenario!
- For scenarios of long exponentially distributed delays, $\nu \to 0$, the derivative $d/d\nu$ of mmPr at the value $\nu = 0$ is $-1/\lambda$ for the reactive approach, while the derivative is zero for the pro-active event-driven incremental approach. Hence, for small values of ν (corresponding to long delays), the reactive approach is always creating a smaller mmPr.
- For very short downstream delays (large ν) both the reactive and the pro-active event-driven strategies decay asymptotically as mmPr λ/ν for both deterministic and exponential delays, and also independently of incremental or full updates. Hence, asymptotically for $\nu \to \infty$, all proactive event-driven and reactive strategies behave equally.
- for large $\nu \to \infty$, the pro-active periodic approach shows a limit of $\lim_{\nu\to\infty} \text{mmPr}(\lambda, \nu, \tau) = \lambda/(\lambda+\tau) > 0$. Consequently, for large τ eventually, the periodic approach will at some point always perform worse than the event-driven and reactive approaches.

Validation by simulation

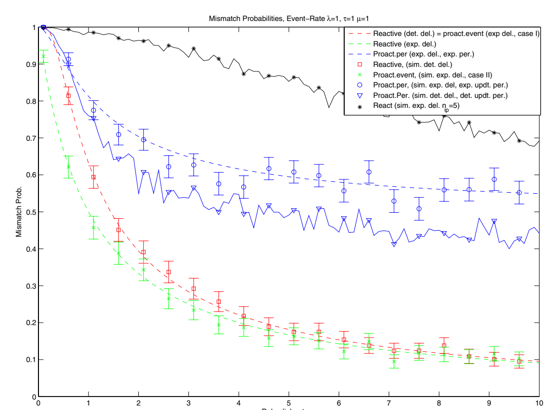

Fig. 5. Theoretical results vs. simulated results varying the down link rate. 95% confidence intervals are provided for most of the simulation estimates, however, those are obtained without consideration of correlation properties.

Figure 5 shows a comparison of the mismatch probability as estimated from simulation experiments for the various strategies for varying downstream delay rates, ν. The results are obtained from simulating 1000 requests and the comparison of simulation and analytic results validates the analytic formulas and provides a visual illustration of variance properties of the corresponding simulation estimator. In addition to the validation of the analytic results, the simulations in Figure 5 show also two cases that have not been treated analytically:

- The case of periodic updates with deterministic downstream delay and update time interval ($\tau = 1$ sec.)

(triangle): In contrast to the periodic update strategy with an exponential delay and update time interval, the deterministic delay and update time interval leads to a reduced mmPr.

- The case of multiple information providers: In this scenario, the information element is a tuple of which the different elements are provided at different entities. This setting will also be considered in the next section. If the different elements change according to independent Poisson processes, the tuple changes also according to a Poisson process with the sum of the individual rates. Detailed analytic modelling of this scenario is outside the scope of this paper, but the black star curve in Fig. 5 shows the simulation result for a 5-tuple with distributed providing entities and exponential downstream delays. See the next section for more discussion on this scenario.

V. Application to Context-Sensitive Service Discovery/Routing

In this section, we discuss the impact of mismatching remote information for the example of context-sensitive service discovery in Personal Networks. Thereby, we also extend the performance metric from the pure mismatch probability to expected values of observed information deviation. The latter requires a semantic description of the information element, which here will be done based on a simplified setting for illustration purposes.

A. *Context sensitive service discovery*

A Personal Network (PN) [10] is a logical private network that interconnects the user's Personal Area Network (PAN) with remote nodes; the latter are typically grouped in so-called clusters. Figure 6 shows an example of a PN which consists of two interconnected clusters (one of which is the user's PAN). Since PNs may be large and geographically distributed,

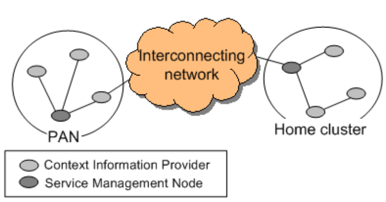

Fig. 6. Context aware service discovery in Personal Network

and furthermore they could contain many devices and hence potentially many services, context-dependent service ranking may strongly increase the user-friendliness of the service-discovery process. An example of a hierarchical service discovery architecture is illustrated, [4], in in Figure 6. The user submits a service discovery request which is sent to the Service Management Node (SMN) in the PAN. The SMN obtains then the context information from information providers in the PAN or in other clusters using one of the three strategies discussed in earlier parts of this paper. After the collection of context information, the SMN can rank and filter the service(s), e.g. based on a calculated score. If the calculated score is below a certain threshold, the service is considered not relevant and is not shown to the user. An example for such a score-function is

$$\text{score}(t) = \frac{\sum_{n=1}^{N} w^{(n)} f^{(n)}(E^{(n)}(t), E_{ref}^{(n)})}{\sum_{n=1}^{N} w^{(n)}}, \qquad (6)$$

where $f^{(n)}$ is an a pre-defined function, which determines a score for the matching of the n-th context field, $E^{(n)}(t)$, in comparison to a reference value, $E_{ref}^{(n)}$. Here $w^{(n)}$ denotes a weighting factor for the different context elements. As Equation (6) shows, the score is based on up to n context values, which are processed at the SMN. There can be a mismatch between the context value used at the SMN for service ranking and the true value at the remote node, which in turn will lead to a wrongly calculated score value, leading to a possible wrong service (de)selection.

B. *Impact of mismatch probability on score function*

We use a simulation model that simulates up to n context providing nodes, in which the context information is assumed to be monotonically increasing. The same simple, linear scoring function $f^{(n)}(t) = 4E^{(n)}(t)$ is used for all context values with same weights $w^{(n)} = 1$. For each context access strategy and each parameter setting, 100.000 service discovery requests are simulated, and the average of the absolute error in score value, is calculated. All results are obtained using an exponentially distributed link delay. The proactive, event-driven scheme is based on full updates (Case II). The simulated results for the average error for different link delay rates are shown in Figure 7: Although the proactive event-driven strategy with full updates and the reactive strategy show the same mismatch probability, the average error for the reactive strategy is higher. This can be explained by the histograms of the error distribution in Fig. 8; the mismatch probability only indicates whether there is a deviation (height of bar at left hand). As for the mismatch probability, the periodic strategy leads to a higher average score error for small delays at the right end of Fig. 7, but it actually outperforms the reactive strategy for long delays.

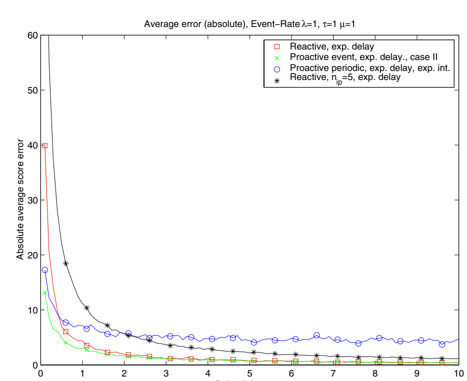

Fig. 7. Average score error using the three update mechanisms.

Furthermore, the case of multiple (five) information providers increases the average score error for the reactive

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

strategy, while the impact on the other strategies seems less pronounced (not shown here). More detailed analysis of such scenarios with multiple information providers will be done in the future.

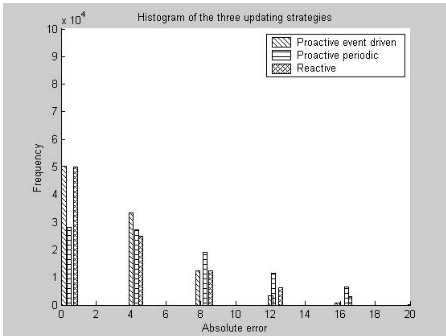

Fig. 8. Histogram of the absolute errors of the three updating strategies.

VI. Summary and Outlook

This paper presents analytic models for the mismatch probability of the following different access strategies to dynamically changing, remote information elements: (1) Reactive access; (2) proactive, periodic access; (3) proactive event-driven access with the two sub-cases incremental and full update messages. The discussion of the analytic results focuses on information elements that never change back to a previous value, the change events form a Poisson process, and network delays are described by an exponentially distributed or constant random variables. The actual request process is irrelevant for the value of the mismatch probability but instead it has an impact on statistical estimation properties and on other performance metrics, such as generated network traffic. Simulation results are subsequently used to validate the analytic results and to provide quantitative results for the scenarios outside the scope of the analytic models treated in this paper. This includes multiple information elements provided by different entities in the network. Finally, the mismatch probability and its impact is discussed in a use-case of context sensitive service discovery in Personal Networks. The analytic models will be extended in future papers to include more general settings and in order to cover the scenario of multiple information sources. Furthermore, additional use-case scenarios such as link-state information in ad-hoc routing, binding tables in dynamic name services, and replication for resilience purposes will be investigated.

Acknowledgments

The authors would like to thank Yaoda Liu (AAU), Homare Murakami (NICT, Japan), and Thibault Renier (AAU), for their helpful comments. Parts of this research is performed in the framework of the EU-FP6 projects MAGNET (www.magnet-ist.org) and HIDENETS (www.hidenets.aau.dk).

References

[1] Y. Liu and H.-P. Schwefel, "An enhanced optimized link state routing with delayed event driven link state updating," *16th IEEE Proceedings of Personal, Indoor and Mobile Radio Communications (PIMRC)*, September, 2005.

[2] H. Murakami, R. Olsen, H. P. Schwefel, and R. Prasad, "Managing personal network specific addresses in naming schemes," *Proceedings of WPMC'05, Aalborg, Denmark*, 2005.

[3] S. Panagiotakis and A. Alonistioti, "Intelligent service mediation for supporting advanced location and mobility aware service provisioning in reconfigurable mobile networks," *IEEE Wireless Communication*, vol. 9, no. 5, pp. 28–38, 2002.

[4] M. Ghader, R. L. Olsen, M. G. Genet, and R. Tafazolli, "Service management platform for personal networks," *Proceedings of IST Summit 2005, Dresden*, 2005.

[5] T. Renier, E. V. Matthiesen, H. P. Schwefel, and R. Prasad, "Inconsistency evaluation in a replicated IP-based call control center," *To be published in the proceedings of ISAS'06*, Helsinki, Finland, May 2006.

[6] W. Chen, S. Toueg, and M. K. Aguilera, "On the quality of service of failure detectors," *IEEE Transactions on Computers*, vol. 51, no. 5, May 2002.

[7] M. Božinovski, "Fault-tolerant platforms for IP-based session control systems," Ph.D. dissertation, Aalborg University, 2004.

[8] M. B. Hansen, R. L. Olsen, and H.-P. Schwefel, "General scenarios of remote access strategies to information elements," *Submitted to Performance Evaluation*.

[9] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Upper Saddle River, NJ: Prentice Hall, 1989.

[10] I. Niemegeers and S. H. de Groot, "From personal area networks to personal networks: A user oriented approach," *Kluwer Journal*, May 2002.

Appendix

A. Derivation of the mmPr for the reactive case

Assume $\mathcal{E}$ is a stationary point process, then define the excess $Y = R_1$ and the age $U = -R_0$ and their distribution functions by $B(t) = P(R_1 \leq t)$ and $A(t) = P(-R_0 \leq t)$, [9]. The density functions are denoted by a and b. Furthermore, construct the stochastic process $E(t) = k,\ t \in [E_k, E_{k+1})$. Assume that event process $\mathcal{E}$ is a Poisson process with intensity λ, then by stationarity we have the following probability of mismatch upon reception of the message for any request at time R_k

$$\begin{aligned} &P\left(E(R_k + U_k + D_k) \neq E(R_k + D_k)\right) \\ &\quad = \quad P\left(E(D_k) \neq E(0)\right) \\ &\quad = \quad 1 - \int P(E(D_k) = E(0) | D_k = t) f_D(t) dt \\ &\quad = \quad 1 - \int e^{-\lambda t} f_D(t) dt = 1 - \mathcal{L}\{f_D\}(\lambda). \end{aligned}$$

As the mismatch probability does not depend on R_k we can define the mismatch probability in the reactive case to be

$$\text{mmPr}_{react} = 1 - \mathcal{L}\{f_D\}(\lambda).$$

B. Derivation of the mmPr for the pro-active, event-driven strategy with full updates

The probability of mismatch for the requesting time R_k is derived by conditioning on the situation that no event has happened in the interval $[t, R_k]$ and that the message is not delayed more than $R_k - t$ time units, consequently by stationarity

$$\begin{aligned} \text{mmPr}_{proact,full} &= 1 - \int_0^\infty P(\mathcal{D} \leq t | U = t) a(t) dt \\ &= 1 - \int_0^\infty P(\mathcal{D} \leq t) a(t) dt. \end{aligned}$$