Service Degradation in Context Management Frameworks

Shawky, Ahmed; Olsen, Rasmus Løvenstein; Pedersen, Jens Myrup; Rasmussen, Jakob Gulddahl

Published in:
Proceedings of the International Conference on Computer Communications and Networks

DOI (link to publication from Publisher):
10.1109/ICCCN.2011.6006060

Publication date:
2011

Document Version
Accepted author manuscript, peer reviewed version

Service Degradation in Context Management Frameworks

Ahmed Shawky
Networks and Security
Dept. Electronic Systems
Aalborg University
Aalborg, Denmark
Email: asms@es.aau.dk

Rasmus olsen
Networks and Security
Dept. Electronic Systems
Aalborg University
Aalborg, Denmark
Email: rlo@es.aau.dk

Jens Pedersen
Networks and Security
Dept. Electronic Systems
Aalborg University
Aalborg, Denmark
Email: jens@es.aau.dk

Jakob Gulddahl Rasmussen
Department of Mathematical Science
Aalborg University
Aalborg, Denmark
Email: jgr@math.aau.dk

***Abstract*—Context aware network services are a new and interesting way to enhance network users experience. A context aware application/service enhances network performance in relation to dynamic context information, e.g. mobility, location and device information as it senses and reacts to environment changes. The reliability of the information accessed is a key factor in achieving reliable context aware application. This paper will review the service degradation in Context Management Frameworks (CMF) and the effect of high network utilization, with particular focus on the reliability of the accessed information. The paper considers a developed framework from the ICT project, OPEN, and investigates the impact of applying Differentiated Services (DiffServ) Quality of Services (QoS). The paper finally provides insight in how the insight gained can be utilized to ensure reliable remote accessed context information.**

I. Introduction

An Important feature in todays applications and services would be its ability to sense and react to environment changes such as location changes, weather changes or even network changes. An example on such environment-aware i.e. context-aware service would be application migration from one network device to another. But context information is not only utilized by applications so they can adapt to their environments, for example in [6] triggers for service migration are based on the contextual situation of the user and the involved devices. Via the migration framework running applications may shift between devices, e.g. switch a video-stream from the mobile phone to the large TV screen. A Context Management Framework (CMF) as described in [5] offers the service of providing the information needed to perform context aware actions. A CMF's main goal is to manage information distributed in the network such as collecting, storing, processing and delivering relevant information from different sources of context to functions in need of information, which can be either directly measurable or inferred/processed information. The performance of such framework is therefore closely linked to the performance of context aware applications. For example, to carry out timely correct triggers to migrate services, the context information is required to be timely correct as well. The dynamics of context information in combination with any delays leads to potential use of mismatching information, giving grounds to incorrect execution of context aware applications and triggers. In this paper, the delay of information occurs due to communication between a context consuming node and a context providing node. In order to obtain the missmatch probability (mmPr) [11] the delay Distribution should be obtained. The main research questions in this paper are:

1) How much utilization will result in sever degradation?
2) What is the impact of the delay estimation on the mmPr?
3) How can this knowledge be used in context management?

This paper is divided into five parts; the first part describes the CMF and the parameters causing the degradation in CMF functionality. The second part describes the test bed used to test the CMF degradation and the impact QoS has on it. The third part presents the results found. The fourth part mentions how to improve service degradation. Finally an overall conclusion is presented.

II. Context Management Degradation Parameters

A. Background and state of the art

European projects like MAGNET Beyond [1], SPICE, [2] or E-SENSE, [3], and others outside Europe, have been researching and developing concepts for context management for some time, whereas reliability indicators of context information just recently caught the attention e.g. in the project SENSEI, [4]. Using time as an indicator for the reliability of the information provided to the application is often used, e.g. in [8], [9], [10] where the notion up-to-dateness or freshness is used, which may be useful in time synchronized networks, however without caring much of the dynamics of the information or the access method, which has been shown to be a key problem for reliable context information in e.g. [11]. Therefore, we focus our performance investigations on the reliability of context information. However, in that investigation we need to understand the realistic properties of context management which is why we carry out the experiments later on.

B. Brief Introduction to Context Management Systems

In the paper we focus on the specific implementation of [1], for which the conceptual structure is shown in Fig. 1 . This consists of two or more Context Agents that is capable of exchanging information regarding what context information is available on which node. One of the nodes acts as a anchor point, the Context Management Node, which contains an overview of all other Context Agents.

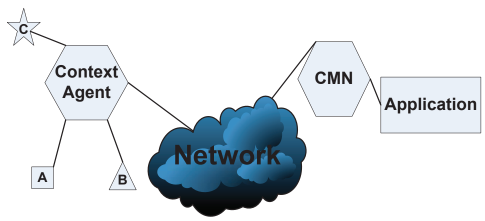

Fig. 1. CMF Framework

In Fig. 1 an application is gaining access to the remote, dynamic context elements A, B and C over the network, in order for it to perform a context aware reaction upon a change in the contextual situation. Each context agent which has access to context information, has a so-called retriever installed, which ensures the local data abstraction and interfacing between the local source of information and the context management system. the communication between CMN and CA is based on xml-rpc, which is TCP based protocol.

C. Degradation Parameters of Context Management Systems

In this section context information is collected via a reactive access strategy as show in Fig. 2. The figure gives an abstracted view of the communication between the two Context entities in order for the application to obtain the required context information. The defined performance parameters will in the following relate to this figure.

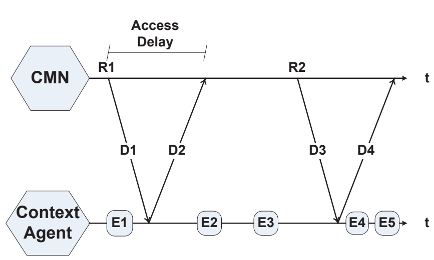

Fig. 2. Access Delay

1) Network Overhead: Network over-head affects the service degradation since the amount of bytes/second generated depends on the request process, the size of the request and response messages. The focus is therefore to measure the sizes of the request and response messages only, whereas the actual network overhead can easily be calculated.

2) Access Delay: Fig. 2 shows that the CMN sends a request (D1) to the CA to collect context info. The CA then sends response (D2) to the CMN with the requested context information. The access delay is then defined as the time at which a request is sent till a response is received, as seen in Eq. 1.

$$t_{access} = D_1 + D_2 \tag{1}$$

Since, later on mainly the response time is of interest, we assume symmetrical delays.

3) Mismatch probability: We call the probability of using mismatching information for mismatch probability (mmPr), [11]. From [11] we know the mmPr is expressed as shown in Eq. 2.

$$mmPr_{full} = \frac{1}{\mathbf{E}(E)} \int_0^{\infty} \bar{F}_D \bar{F}_E dt \tag{2}$$

with E as the event process, F_D the forward access delay cdf of D_2, F_E the forward event time cdf (the over-line indicates 1-F). For the context management framework it is reasonable to assume that meta information regarding the dynamics of the context information is available as this may be a part of the context description itself. Hence, it is assumed that every response received from the context provider, contains not only the value of the information, but also complete meta information about the distribution, F_E. The delay distribution, relates mainly to the network, congestion level and QoS as will be a central part of the evaluation and influencing factor on the mismatch probability.

III. Test bed Setup

To evaluate the CMF degradation the test bed setup shown in Fig. 3 is used. The CA is seen on the left side of the

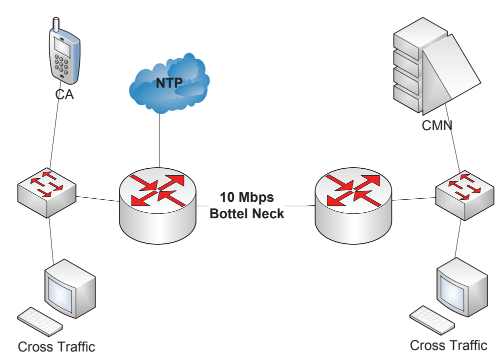

Fig. 3. Test Bed Setup

network and collects some context info such as location, noise or temperature. The CMN is located on the right side of the network. The CA and CMN are connected to two Cisco 3950 switches each switch is then connected to a Cisco 3620 Router finally the two routers are connected to each other with a 10 mbps bottle neck link. On each side of the network there are two sources that generate cross traffic. The traffic generated is constant bit rate traffic in order to keep the network constantly busy. A series of tests with varying network utilization can now take place, the network traffic between the CMN and CA has two senarios, one with Best Effort (BE) and one with QoS priortization. For each test 50 requests from the CMN are carried out i.e. 50 requests/responses. The test scenario will introduce different traffic utilizations between 80% to 97% utilization.

A. Result Evaluation

1) Network Traffic: We use Wireshark to collect the information for the evaluation. The TCP streams between the CA and the CMN are filtered out for each direction of communication, and statistics about data exchange and the overhead information, including TCP overhead are extracted. The statistics are normalized to the 50 collected samples for each test, so the results are mean values for a single request or response, while in fact according to network utilization sample retransmission and duplicate segments may occur. To evaluate the impact of context state sizes, a reconfigurable retriever is used which allows setting the state size of the accessed information. The impact of state size on the traffic is evaluated for the range 10B - 10KB state information i.e. each bandwidth utilization test will be carried out on state sizes varying between 10B - 10KB.

2) Delay: For each of the samples time stamps are logged and used to examine the delay. The system time is logged just before making a request call to the CA, and when this returns with a value, the time is logged again. The time difference is used as an indication of the access delay time for the reactive strategy. The impact of the state size on the delay is evaluated for the range 10B - 10KB of state size for both scenarios, BE and QoS.

3) mmPr: To evaluate mmPr the Kolmonogorov-Smirnov test [13] is a nonparametric test used to compare a sample with a reference probability distribution. Applied in a reverse way, this can be used to put confidence on an empirically constructed cdf, which is useful to determine when enough samples are obtained in-order to provide an accurate mmPr estimate of the accessed information. The cdf is constructed as shown in Eq. 3

$$F_n(x) = \frac{1}{N}\sum_{i=1}^{N}[x_i < X] \tag{3}$$

Defining $D_N = sup_x|F_N(x) - F(x)|$, i.e. the least difference between the observed distribution from an assumed distribution, then the Kolmonogorov-Smirnov tests states that under null hypothesis that the samples comes from a particular distribution $F(x)$, then the value $\sqrt{n}D_N$ converges in probability Kolmonogorov distribution which does not depend on $F(x)$, [12] [13]. The hypothesis is rejected at some level α for $\sqrt{n}D_N > K_\alpha$ where K_α is found from $Pr(K < K_\alpha) = 1-\alpha$.

Applying in reverse, provides confidence to the estimated delay distribution, and eventually provides confidence to the mmPr estimate. By looking entirely on the hypothesis relation, it can be seen that the e-cdf is bounded by the value $\frac{K_\alpha}{\sqrt{n}}$ for any distribution that $F(x)$ may attain. The impact of this distribution boundary on the mismatch probability can be obtained from Eq. 2, and turns out to be exactly $\frac{K_\alpha}{\sqrt{n}}$. A simulation example to support this, has been carried out and plotted in Fig.7, where it is easily seen that use of many samples for the CDF is useful. However, usually also at the cost of time and effort.

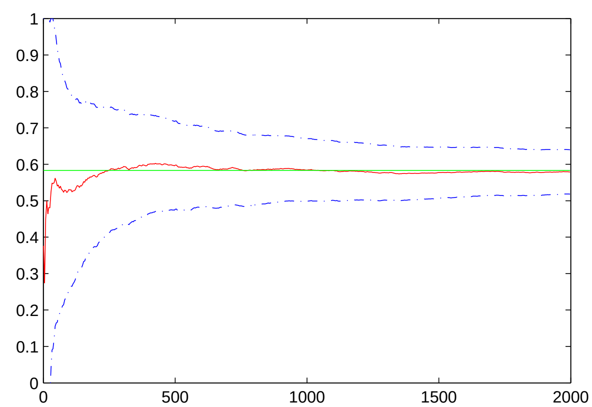

Fig. 4. Bands of confidence around an mmPr estimate as a function of samples, K_α = 1.3581 with 95% confidence Interval.

IV. Results

The following results are based on the test bed described in Section III.

A. Network Traffic

The results from the analyzed network traffic for each test are presented in the following tables. Table I shows the analyzed network traffic between the CMN and CA, where the CMN and CA packets are marked with BE i.e. there is no prioritization. Table II shows the analyzed network traffic between the CMN and CA, where the CMN and CA packets are marked with EF QoS i.e. the CA and CMN packets are prioritized over other packets and a bandwidth reservation of 10% will be allocated in case of congestion. The notation (1) and (2) are used such as (1) represents traffic in the direction from CA to CMN and (2) represents traffic in the direction from the CMN to CA.

As it can be seen from the data from the tables, over head for smaller data-state sizes is quite large. Further analysis of the traffic shows that the cause of increased traffic overhead as a function of increased state size is due to the three main factors of TCP retransmission, lost TCP segments and duplicate TCP

TABLE I
NETWORK TRAFFIC IN BYTES FOR BE TRAFFIC

State Size	Data (1)	Data (2)	Over-head (1)	Over-head (2)	Total
10B	1009	1238	304	442	2993
1000B	2066	1238	297	487	4088
2000B	3132	1238	315	530	5215
5000B	6264	1238	247	602	8351
10000B	11590	1238	257	844	13929
Avg.	4812	1238	284	581	6915.2

TABLE II
NETWORK TRAFFIC IN BYTES FOR EF QOS TEST

State Size	Data (1)	Data (2)	Over-head (1)	Over-head (2)	Total
10B	1009	1238	337	292	2876
1000B	2066	1238	337	351	3992
2000B	3132	1238	338	411	5119
5000B	6264	1238	272	470	8244
10000B	11590	1238	272	734	13773
Avg.	4812	1238	284	581	6800.8

acks, this is due to the nature of the TCP protocol. Fig. 5 shows the different factors of increased overhead for five different state sizes varying from 10 - 10000 Bytes. It would be of intrest to see the impact of implementing the CMF functionality using UDP in order to see the impact on service degradation.

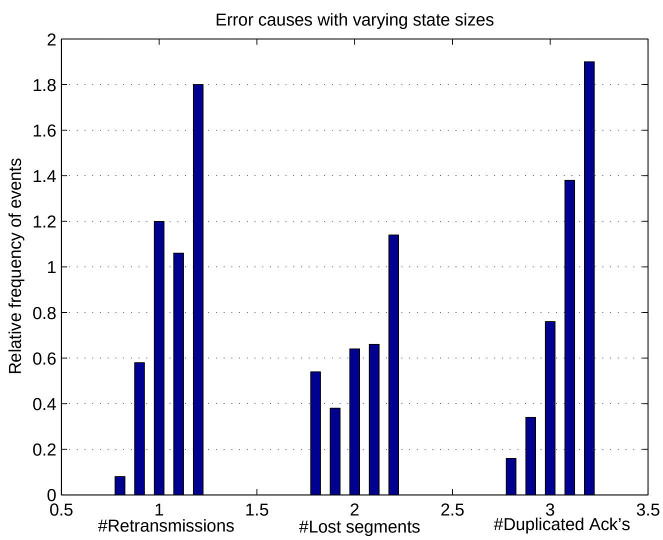

Fig. 5. Factors of Increased Overhead

B. Mean access delays due to Utilization

Fig. 6 and Fig. 7 shows the mean access delay time for the best effort traffic and EF traffic respectively. The Ebar indicates mean time between events is set to 5 seconds and Rbar indicates the mean time between requests is also set to 5 seconds and the state size is 1KB.

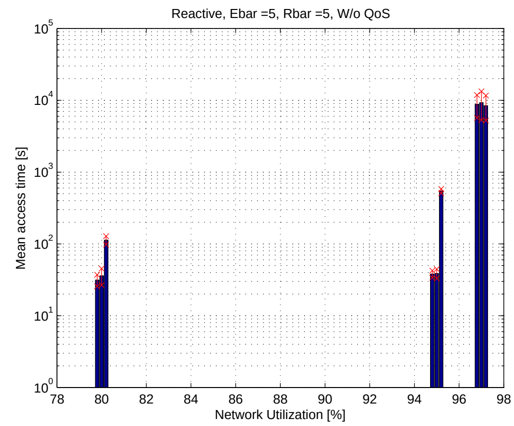

Fig. 6. Best Effort Mean Access Times with varying Utilization for 10 byte, 2KB and 10Kb state sizes, respectively.

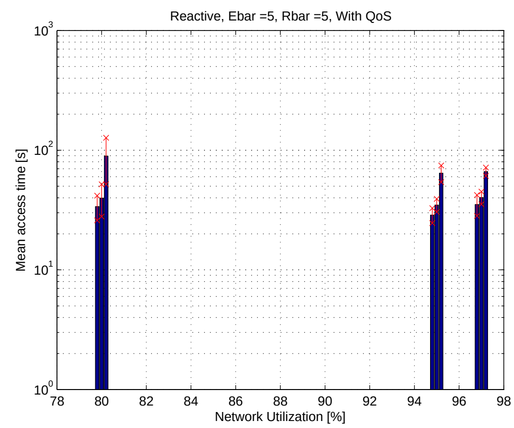

Fig. 7. EF Mean Access Times with varying Utilization for 10 byte, 2KB and 10Kb state sizes, respectively.

C. Access Time distributions

The Cumulative Distribution Function (CDF) for access times with best effort under different network utilizations is shown in fig. 8 and fig. 9. Fig. 10 shows the CDF for access times with EF under different network utilizations.

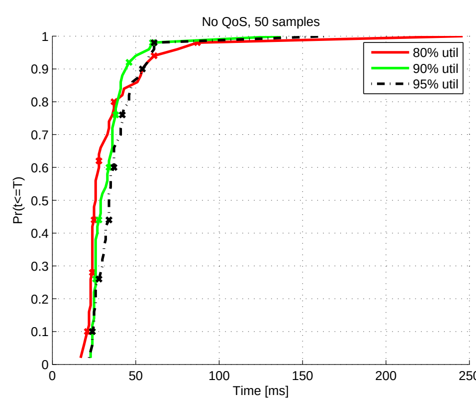

Fig. 8. Best Effort Access Times CDF, 96% - 97% util

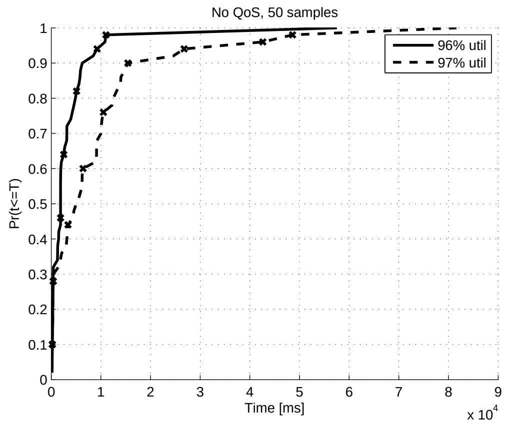

Fig. 9. Best Effort Access Times CDF, 96% - 97% util

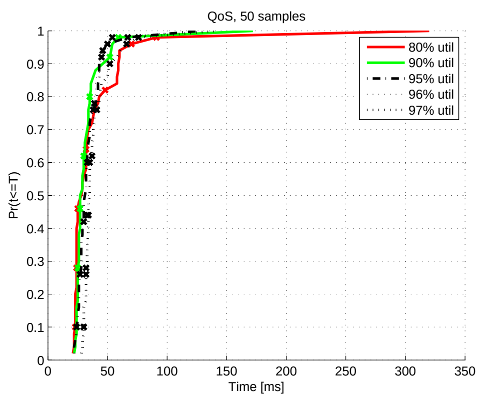

Fig. 10. EF Access Times CDF, 80% - 97% util

As it can be seen from the figures there is very little difference in the access times considering the network utilization interval 80%-90% utilization. This is valid as long as the active QoS class is not saturated i.e. if the CMN should need to exchange large amount of information in this case exceeding the 10% reserved bandwidth it may need to classify the CMN and CA traffic with more than one class in order to distribute the load.

D. Mismatch probability

Recall that the mismatch probability depends on the CDF's of both the event process and delay process as shown in Equation (2). For the event process, this may be given as a priori knowledge from context providers, but for the network this is not possible due to time variation in the network load. From the measurement campaign, we can now plot the resulting mismatch probability with confidence bounds. Figure 11 shows the result for two different state sizes, 2KB and 10KB, respectively, how the information dynamics influences the mismatch probability given the delay distributions shown in Figure 8 and Figure 10. Thus, the two utilization factors representing the cases using normal and prioritized traffic via QoS marking.

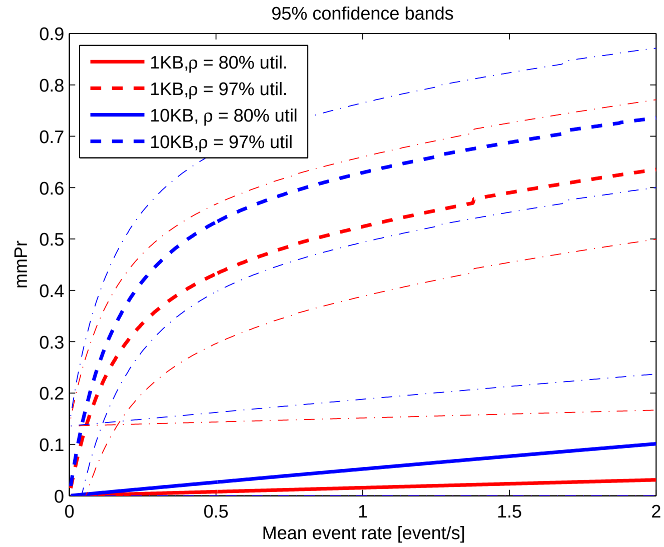

Fig. 11. MmPr plots of two dynamic information types with varying state size under different network conditions. MmPr's are plotted with

From Figure 11 it is quite obvious that prioritizing the context management traffic higher than other traffic positively impacts the reliability of the information being accessed. Although not easy to see in Figure 11, the impact of state size is most significant in low congested scenarios. Then increase in mismatch probability for the case with λ = 2 events/sec., is from around 0.03 to 0.1 for an increase of 8KB in state size, where in the highly congested scenario the increase is from 0.63 to 0.74 for the same information dynamics.

V. Context Management Service Improvment

In this section we want to illustrate how to use the type of investigations we did. Assume that it is possible to reduce the state size in a response message, either by a query language that efficiently reduces the size, or by any compressing techniques, then Figure 11 implies that mismatch probability may be reduced simply by focusing on reducing transported state size. Figure 12 shows resulting mismatch probability for the 80% utilized network scenario with 1KB, 5KB and 10KB state size, respectively. Usually, query processing or compression techniques also takes additional time, which on top of the network delay poses a challenge to the reliability as we showed in Figure 11. We now add an additional processing component in the time delay, such that Eq. 1 which we for simplicity assume deterministic:

$$t_{access} = t_{D1} + t_{processing} + t_{D2} \tag{4}$$

Figure 12 now shows the mismatch probability as a function of this added processing time.

If the compressing algorithm or query processing, manage to reduce a 10KB data element into a 5KB data element, then as long it takes less than 30 msec, this will have a positive impact on the reliability. If it takes roughly 30msec, reliability will

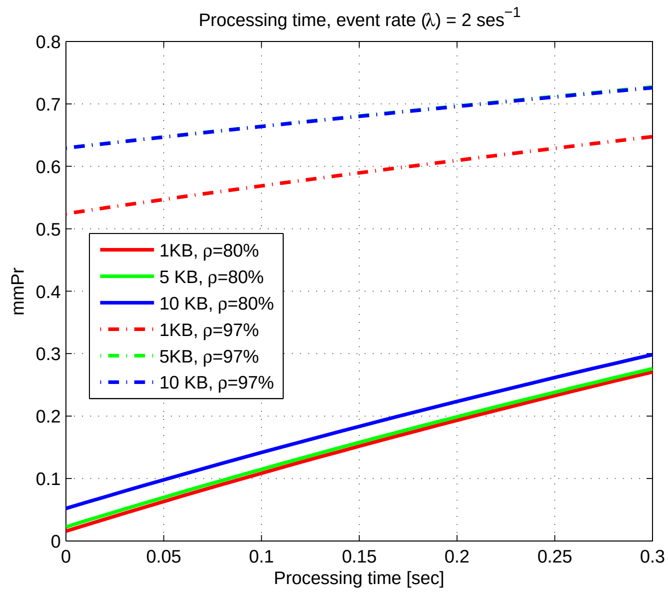

Fig. 12. Impact of processing time on the reliability of the accessed information

not be impacted, but if it takes longer, then there will be a loss in reliability at the gain of less traffic being generated. If the compression algorithm is able to reduce data size even further, to e.g. 1KB, then in the case shown in Figure 12 this will have an additional 10msec to do so in order to gain the benefits of improved reliability and decreased network overhead as described. If there are strict requirements to the reliability, e.g. 0.4, we can see that an upper cap of processing time can be found in similar way. In the case of 12 for a 10KB state, this will give the context provider 200msec to process the data before it impacts the reliability. It is important to be aware that 12 is restrictive in that the processing time is deterministic, access delay as mentioned is based upon measurements with the before mentioned confidence bands, and an information type which has exponentially distributed time interval between events.

For the case where the network is congested, i.e. here 97%, the mismatch probability is much higher than the previous case. Notice, the 5KB and 10KB state size practically leads to the same mismatch probability, hence there is no need to do much additional processing. The reliability gain is simply lost in the higher network delay. However, spending time effort to reduce the data element to a 1KB size may be worth pursuing. Adding less than 270 msec of processing delay to reduce the state size from 10KB or 5KB to 1KB will reduce the mismatch probability, hence more time may be spend.

VI. CONCLUSION

This paper stated the issue of service degradation in CMFs. The paper gave a brief description of a CMF and the parameters affecting CMF performance were found to be network overhead, access delay and mmPr. The results of the constructed test bed were investigated and the results for network overhead showed that overhead in small data sizes was rather large and that it's main cause was the number of TCP retransmission, lost TCP segments and duplicate TCP acks. A simulation is intended to show the impact of using a UDP based CMF to try and improve the degradation parameters, however TCP reliability is then sacrifice. it was also clear the prioritized context traffic resulted in a slightly lower overhead. The mean access delay results stated that for BE traffic a significant increase in access delay can be seen in network utilizations between 96%-99%, this is not a surprise however prioritizing CMF traffic gave a degree of normalization and control of access delay in higher utilizations. By applying the Kolmonogorov-Smirnov test to mmPr CDFs it was seen that prioritized context management traffic added more reliability and validity to the context information used, impact of state-size on mmPr however was more significant in lower congested scenarios. Finally it was shown that taking time to compress the response state has a direct effect on overhead, access delay and mmPr and will be considered in future work.

REFERENCES

[1] R.Prasad (Ed.), My Personal Adaptive Global Net (MAGNET): Signals and Communication Technology. Springer Science+Business Media, 2010, DOI: 10.1007/978-90-481-3437-3

[2] http://www.ist-spice.org/

[3] http://www.ist-e-sense.org/

[4] http://www.sensei-project.eu/

[5] L.Sanchez, R.L.Olsen, M.Bauer, M.G.Genet, A generic context management framework for Personal Networking environments, First International Workshop on Personalized Networks, 2006, IEEE (explorer) - Paper number 0-7803-9792-4,

[6] A.Nickelsen, F.Paterno, A.Grasselli,K.U.Schmidt,M.Martin,B.Schindler, F.Mureddu, OPEN: Open pervasive environments for migratory interactive services, Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, 2010, p. 637-644

[7] R.L.Olsen, J.Figuerias, J.G.Rasmussen, H.P.Schwefel, How precise should localization be? A quantitative analysis of the impact of the delay and mobility on reliability of location information, Globecom. I E E E Global Telecommunications Conference. 2010, 6 s.

[8] Z. Abid, S. Chabridon, and D. Conan, A Framework for Quality of Context Management, QuaCon 2009, LNCS 5786, pp. 120-131, 2009, Lecture notes published by Springer-Verlag Berlin Heidelberg 2009

[9] S. McKeever, J. Ye, L. Coyle, and S. Dobson, A Context Quality Model to Support Transparent Reasoning with Uncertain Context, QuaCon 2009, LNCS 5786, pp. 65-75, 2009. Lecture notes published by Springer-Verlag Berlin Heidelberg 2009

[10] K. Henricksen, J. Indulska, A. Rakotonirainy, Generating Context Management Infrastructure from High-Level Context Models, Proceedings of 4th International Conference on Mobile Data Management, 2003

[11] Martin Bogsted, Rasmus L. Olsen, Hans-Peter Schwefel, Probabilistic models for access strategies to dynamic information elements, Performance Evaluation 67 (2010), pp. 43-60

[12] Stuart, Alan; Ord, Keith; Arnold, Steven [F.] (1999). Classical Inference and the Linear Model. Kendall's Advanced Theory of Statistics. 2A (Sixth ed.).

[13] Lopes, R.H.C., Reid, I., Hobson, The two-dimensional Kolmogorov-Smirnov test, XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research (2007).