Class-Based Context Quality Optimization For Context Management Frameworks

Shawky, Ahmed Sherif Mahmoud; Olsen, Rasmus Løvenstein; Pedersen, Jens Myrup; Schwefel, Hans-Peter

Published in:
Computer Communications and Networks (ICCCN), 2012 21st International Conference on

DOI (link to publication from Publisher):
10.1109/ICCCN.2012.6289287

Publication date:
2012

Document Version
Early version, also known as pre-print

Class-Based Context Quality Optimization For 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

Hans-Peter Schwefel
FTW - Forschungszentrum
Telekommunikation Wien, and
Dept. Electronic Systems
Aalborg University
Email: hps@es.aau.dk

***Abstract*—Context-awareness is a key requirement in many of today's networks, services and applications. Context Management systems are in this respect used to provide access to distributed, dynamic context information. The reliability of remotely accessed dynamic context information is challenged by network delay, packet drop probability, information dynamics and the access strategies taken. QoS classification and system configuration of context management traffic is in this aspect important in order to efficiently balance generated access traffic between network delay, packet loss probability, information dynamics and reliability requirements to the information from the applications. In this paper we develop a QoS Control network concept for context management systems. The concept includes a soft realtime algorithm for model based context access configuration and QoS class assignment, and allows to put probabilistic bounds on the information reliability (the so-called mismatch probability).**

I. Introduction

The ability of applications to adapt to the users environment is often referred to as context awareness, [1], and is becoming a key factor for users to be able to efficiently interact with applications and platforms in a highly dynamic world. Context information is gathered from the surroundings via dedicated subsystems, so-called context management systems, which ensure easy access to distributed, dynamic context elements. The reliability of the accessed information is the key to the success of any context aware applications, since application adaptation should respond to current events and not earlier ones. The reliability of context information is challenged by network delay, information dynamics, the access strategy used and their respective parameter settings. We use the notion mismatch probability (mmPr) [10] as the reliability parameter for accessed dynamic context information. [11] states that high mmPr is a cause of service degradation in context management framworks.

This paper investigates the possibility of increasing context reliability and reducing context information mmPr by using QoS traffic classification. The paper presents a new QoS class assignment and configuration mechanism that optimizes access traffic to dynamic context information coming from context management systems to efficiently meet reliability requirements coming from the applications. whith this model we are able to assign QoS classes intelligently to the generated context management traffic.

The paper is divided into four parts. Section II gives a brief description of Context Management Systems and the various access strategies, and provides a short overview of the new model used later on. Section III introduces the QoS model and explains its functionality. Section IV presents simulations scenarios and results. Section V provides a conclusion and an outlook on future work.

II. Context Management Framework and Background

The Context management framwork formulas are based on work from [12]. For applications to be able to adapt to their environment, the access to dynamic context information [1], which describes the current situation, is required. Context Management systems offers flexible access often via dedicated query languages, e.g. [5], [2].

Figure 1 shows a scenario where a group of network devices such as smarthpones, pc's and smart TVs connect to a context manager that has access to a variety of context information that allows application adaptation to it's environment. Context agents are deployed at the sources which enables any type of needed interaction with context managers in the network, such as information discovery, security and access control, query distributions, negotiation and control of access mechanisms as described later.

A. Background

A specific problem of access to remote dynamic information is that the required communication and processing delays makes it possible for the accessed information to become mismatching to its true value when used in the application, leading to potential erroneous application behavior. In our earlier work, we introduced the notion mismatch probability (mmPr), [10], as *the probability that at the time instant of using a certain information for processing in the context-sensitive application, this information does not match the value at the (physical) source*. While the reliability of context information has previous been acknowledged and considered an important part of the quality of context, [3] or [4], this current

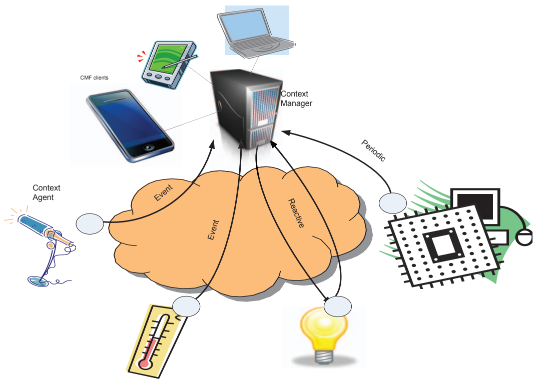

Fig. 1. Example of Context Management Node that allows the mobile device access to a variety of dynamic information in the network useful to adapt the operation of running applications

paper uses the quantitative metric for intelligent choices of access strategies.

Table I shows the different access strategies mmPr formulas, based on work from [10] and [11].

The mismatch probability is dependent not only on the two stochastic processes a) network delay and b) information change process, but also on the strategy by which the information is accessed. The common model parameters are:

- Network up- and downstream delay: i.i.d exponential distributed with rate ν $[sec^{-1}]$
- Event inter arrival times: i.i.d. exponential distributed with rate λ $[events/sec]$
- Message loss probability: $p_{loss} \in [0..1]$
- Information request rate: μ $[req/sec]$
- Message size: U $[bytes]$
- Link Utilization ρ $[Mbit/sec]$

B. Network model

The Network model is used to translate utilization into mean delay and packet loss probability. For its simplicity an M/M/1/K model is used for testing the algorithm. The model has a buffer of size K. The M/M/1/K mean delay and its packet loss equations at utilization ρ are found in any standard queuing theory book:

$$PacketLoss = \frac{((1-\rho)\rho^K)}{(1-\rho^{K+1})}$$

$$QueueLength = \frac{\rho}{(1-\rho)} - \frac{(K+1)(\rho^{K+1})}{(1-\rho^{K+1})}$$

$$Delay = \frac{QueueLength}{\lambda(1-P_{loss})}$$

III. System Architecture and Execution

A. Context Quality Model

To determine the suitable class and the suitable access strategy, different model parts needs to be combined as illustrated in Figure 2: based on the parameters of the context elements accessed (Change event rate λ_i and Update Size U_i) in the upper left and based on the request rate from the subscription μ_i, the additionally generated class traffic can be calculated in the upper left block. The additional increase calculated for each class traffic is then used as input to the network model, which in return calculates performance parameters delay and packet loss for the communication path between context subscriber and context provider. For this paper the results from the network model block are mean delays and mean packet loss although it is also possible to use more complex network models.

The Class delay and packet loss values combined with the context and subscription parameters are then used as input for mmPr calculation block. The class mmPr results are passed to a mmPr bound filter where classes that do not meet the mmPr bound are rejected. Finally minimization of global mmPr is done and the class with lowest GmmPr is selected.

TABLE I
CONTEXT INFORMATION PARAMETERS, WHEN ALL PROCESSES ARE ASSUME EXPONENTIALLY DISTRIBUTED

Access Strategy	Mismatch probability
Reactive	$mmPr_{rea,exp} = 1-(1-p_{loss})^2\frac{\nu}{\lambda+\nu}$
Proactive Event Driven	$mmPr_{evn,exp} = 1-(1-p_{loss})\frac{\nu}{\lambda+\nu}$
Proactive Periodic Update	$mmPr_{per,exp} = \phi^{\psi}\frac{\Gamma(\phi+\psi)}{\psi\phi^{\phi+\psi}}F_{\Gamma(\phi+\psi,\psi)}$ $\phi=\lambda/\nu,\ \psi=(1-p_{loss})\tau/\nu$ $F_{Gamma(a,b)}$ is the gamma distribution with parameters a and b
Access Strategy	**Network overhead**
Reactive	$\eta_{rea,exp} = \mathbb{E}(U_{rqs}+U_{rsp})/\mu$
Proactive Event Driven	$\eta_{evn,exp} = \mathbb{E}(U_{rsp})/\lambda$
Proactive Periodic Update	$\eta_{per,exp} = \mathbb{E}(U_{rsp})/\tau$
Parameter	**Definition**
Class mismatch Probability	$CmmPr = \sum_{i=0}^{n}\frac{mmPr_{i(E_i,D_n,P_n,\tau_i)}}{n}$
Global mmPr	$GmmPr = \sum_{c=0}^{n}\frac{CmmPr_n}{n}$

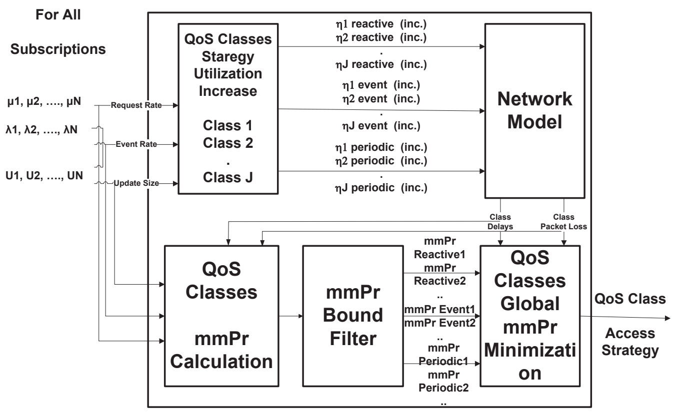

Fig. 2. Model Overview and interaction. Calculations will be executed on the CMN.

B. Execution Sequence

Having explained the models functionality to assigning subscriptions to the appropriate QoS class while optimizing each class mismatch probability of context access, this model now can be used in an online algorithm for context access configuration and quality optimization. The algorithm is executed on the Context-Management-Node (CMN), which is the node that receives all context subscriptions and manages the interaction with the context providers, the latter are implemented via so-called Context Agents (CAs), which can reside on any mobile device or anywhere in the network.

Basic principle for message flow that occurs during the registration and subscription phases:

1) All CAs register at the CMN. They thereby provide information about the type of context elements they can offer and the relevant parameters. Particularly, they provide information on the temporal dynamics of the context element, which in our simplified setting is the change rate λ_i of the context-element i. Furthermore, also information on the size of update messages of the context element are provided.
2) When a certain application or middle ware function on a node is interested in a certain context element, it sends a subscription to the CMN. In the subscription it may specify quality requirements, such as bounds on mismatch probability or bounds on access delays to the context element. We will however not use these in the current algorithms but minimize what we call a GmmPr as introduced in the previous sections. Extensions to constraint-based mmPr minimizations are however straightforward.
3) The CMN then evaluates which context access procedures with what parameter settings are best to implement for this new subscription. In order to do so, it may interact with Quality of Service functions in the network. Having determined the most suitable configuration, the CMN informs the subscriber and the CA about this configuration and starts receiving context updates (in the proactive strategies) respectively relaying requests for context elements in the reactive strategies. All interaction between subscriber and CA is performed via the CMN in order to allow abstraction and efficient processing.
4) The CMN keeps track of all active subscriptions. Hence, as a variant, when a new subscription comes in, the CMN may not only determine an optimal access configuration for this newly arriving subscription, but it may in addition identify that it is beneficial to change some of the configurations of the already active subscriptions. Both cases we will specify and evaluate further below.

As mentioned the algorithm is capable of intelligent class assignment as well as context-access configuration control for incoming context subscriptions. The algorithm implements a GmmPr minimization function, which iterates over current active subscriptions when a new subscription arrives and determines if an other configurations would lead to a lower GmmPr.

C. Algorithm: Context Quality Optimization

The algorithm is executed at the start of a new subscription. The algorithm then checks the available QoS classes and calculates the mmPr for all access strategies for all QoS classes. The algorithm then selects the class and the strategy with the lowest mmPr. As new subscriptions arrive, the algorithm calculates the GmmPr using all access strategies for all QoS classes, then it would select the QoS class with lowest GmmPr after reconfiguring all existing subscriptions. The pseudocode is found below.

Algorithm 1 Base Algorithm

```
 function determine_QoS class and access configuration_of_new_subscription
 Input: Current network load L and available network (bottleneck) resources R
               in the path context agent - CMN - subscriber.
        Request Rate mu_i of new subscription.

Available Information (persistently maintained at CMN)
           From registration: Change rate lambda_i, and update size u_i of
           context element of new subscription.
           From previous subscriptions: Parameters and configurations of
                                        currently active subscriptions.

Output: QoS Class for new access strategy
             access strategy for new subscription
        parameterizations of the access strategy
        (here: rate of proactive periodic strategy)
        resulting new GmmPr best_mmPr.

Begin
for all classes
class_mmPr=1.0;
class_strategy=not_defined;
For all access strategies in (None, reactive, proactive event-driven,
            proactive periodic_rate1, ... proactive periodic_rateK);
      calculate additional class load Lc_inc created by NEW subscription
      when using this access strategy (see Tabel I);
      Call network model to calculate class performance metrics (Dc, Pc)
      at class load Lc+Lc_inc for resources R;
      Calculate new subscription resulting mmPr for class, Reject access strategy
      if mmPr< mmPr bound.
      If calculated mmPr < best_mmPr;
               Set currently considered access strategy and parameters
               as best one for current class class;
               Update class_mmPr;
     end-if
end for-loop;
end for-loop;
Calculate Link Global mmPr (GmmPr) for all classes;
Select class, strategy and parameter that minimizes GmmPr;
If Subscription_number > 1 for selected class
Run Reconfigure subscriptions and recalculate GmmPr;
end function;
```

IV. EVALUATION

To evaluate the algorithm's performance a series of simulations were carried out. These simulations are a method of evaluation of the algorithms and a way to compare the algorithm against the context strategy optimization assignment used in [12]. The simulation considers the arrival of N context access requests at the CMN. All parameters and their values in the simulation are listed in Table II.

The CMN is assumed to have perfect knowledge of the average request rates for the individual subscriptions.

For the network, we assume that the context access traffic of all N subscriptions share the same network bottleneck, for instance if the CMN issues these N subscriptions over the same wireless access interface. We consider thus a WLAN access scenario with a link capacity of 10Mb/s. The buffer-size in the used network model (M/M/1/K) is set to 100 messages. It is furthermore assumed that other (non-context) traffic coming to/from the CMN is utilizing the same wireless network, creating a 'base load' taking two different values, 50% and 85% respectively.

The mmPr models of Section III is then used in the simulations and the GmmPr is calculated for each service class and compared to the classless optimization algorithm found in [12].

Table II shows the context information parameters and the constant M/M/1/K parameters used for the two different scenario's. The context parameters are thereby uniformly distributed in the range specified in the table.

TABLE II
CONTEXT INFORMATION PARAMETERS

Context Parameter	Value
Context Request Rate (μ)	[0.2-5] Requests/Sec
Context Event Rate (λ)	[0.2-5] Events/Sec
Context Request Size (U_{rqs})	200 Bytes
Context Update/Response Size (U_{rsp})	[800-1200] Bytes/Update
Classless Buffer Size (K)	100 Packest
Class 1 Buffer Size (K_1)	100 Packets
Calss 2 Buffer Sive (K_2)	30 Packets
Classless Maximum Link Capacity (μ_{link})	10 Mbit/s
Class 1 Maximum Link Capacity (μ_1)	9 Mbit/s
Class 1 Maximum Link Capacity (μ_2)	1 Mbit/s

For the classfull scenarios the background traffic is only utilizing Class 1. Class 2 traffic however is close to zero and is used for context quality optimization.

In the following we describe our findings.

Scenario with 50% utilization from other traffic: Figure 3 shows the Comparison between classless context optimization and classfull context optimization with a baseload of 5 Mbit/s on the Classless link and class 1 of of the classfull link. The classfull optimization scenario is without any mmPr bounds i.e $mmPr_{Bound} = 1$.

The results for the 50% backround traffic scenario shows:

1) That the classfull scenario start with a lower GmmPr and both are assigned to class 2, however as the number of subscriptions goes over 10, GmmPr for class 2 becomes higher than that of the classless optimization due to the spliting of the link bandwidth and therfore continues to rise.
2) Figure 3 also shows that class 1 with no mmPr bound has very high GmmPr,this is due to the fact for spliting the link bandwidth, however this problem can be addressed by adding a mmPr bound.

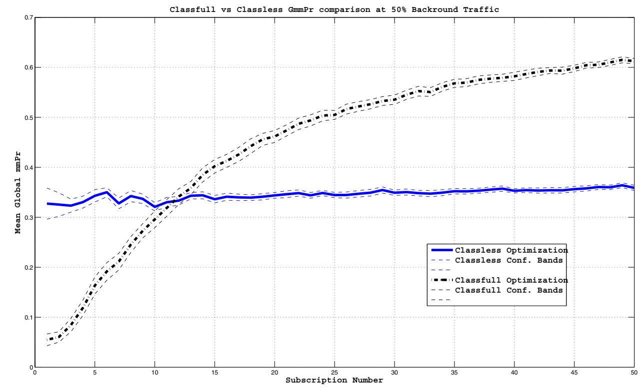

Fig. 3. Classfull vs Classless Comparison at 50% Background Traffic

Scenario with 85% utilization from other traffic: Figure 4 shows the Comparison between classless context optimization and classfull context optimization with a baseload of 8.5 Mbit/s on the Classless link and class 1 of of the classfull link. The classfull optimization scenario is without mmPr bounds i.e $mmPr_{Bound} = 1$.

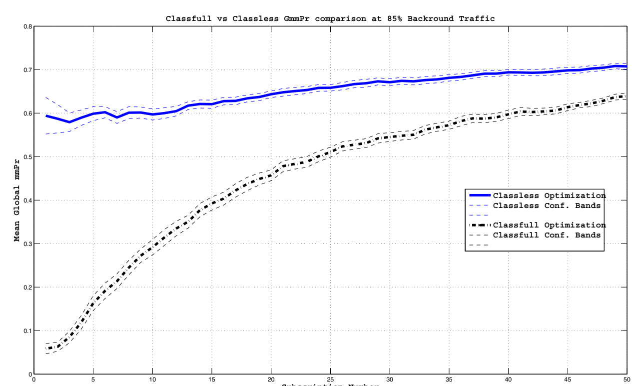

Fig. 4. Classfull vs Classless Comparison at 85% Background Traffic

The results for the 85% background traffic scenario shows:

1) That for higher utilization classfull optimization GmmPr is lower than the GmmPr of a classless context optimization, this is due to the senstivety of mmPr to service rate and delay and thus using a less utilizaed class would benfite under higher utilization.
2) The classfull optimization benfites the GmmPr results due to the high utilization on the classless optimization.

V. Conclusions and Outlook

This paper investigated the possibility of increasing context information reliability by using QoS classification. The paper presented a new model-based approach that maximizes context reliability by assigning dynamic context subscriptions to an appropriate QoS service class. The concept defines and evaluates an algorithm which is intended to be used as a part of a context QoS control framework. The Model's algorithm is executed on the Context-Management-Node (CMN), which is the node that receives all context subscriptions and manages the interaction with the context providers. The algorithm showed that for lower utilizations a classless environment would serve best the so called GmmPr metric, however for high utilization scenarios the algorithm manages to reduce GmmPr as incoming subscriptions would be assigned to a better serving class with lower GmmPr. It was also shown that subscriptions assigned to class 1 traffic yields very high GmmPr, yet this problem could be solved by introducing mmPr bounds that keeps the GmmPr below a certain threshold. The findings can be used in the future as a base for a QoS control framework where dynamic QoS classes are configured to increase context information reliability even further.

Acknowledgment

This work was partially supported by the EU ICT FP7 project 'Open Pervasive Environments for iNteractive migratory services - OPEN', see www.ict-open.eu. The Telecommunications Research Center Vienna (FTW) is supported by the Austrian Government and by the City of Vienna within the competence center program COMET.

References

[1] Dey, A.K., Providing Architectural Support for Building Context-Aware Applications, Ph.D. thesis, Georgia Institute of Technology, 2000

[2] Luis Sanchez, Rasmus L. Olsen, Martin Bauer, Marc Girod Genet: A Generic Context Management Framework for Personal Networking Environments, First International Workshop on Personalized Networks, 2006

[3] T. Buchholz, A. Kupper and M. Schiffers, Quality of context: What it is and why we need it, in Proceedings of the 10th Workshop of the OpenView University Association: OVUA'03, Geneva, Switzerland, (July 2003).

[4] 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

[5] A. Devlic, E. Klintskog, Context Retrieval and distribution in a mobile distributed environment, Proceedings of 3rd workshop on context awareness for proactive systems, Guildford, United Kingdom, 2007

[6] 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

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

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

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

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

[11] Shawky, A.; Olsen, R.; Pedersen, J.; Rasmussen, J.G.; , "Service Degradation in Context Management Frameworks," Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on , vol., no., pp.1-6, July 31 2011-Aug. 4 2011 doi: 10.1109/ICCCN.2011.6006060

[12] Shawky, A.; Olsen, R.; Pedersen, J.;Schwefel H.; , "Optimizing the Quality of Dynamic Context Subscriptions for Scarce Network Resources". Accepted in the proceedings of. AppRoaches to MObiquitous Resilience (ARMOR).Sibiu, Romania. 2012.