A FRAMEWORK FOR THE DYNAMIC CONFIGURATION OF ADAPTIVE TRANSPORT PROTOCOLS

Application to QoS Requirements

F. Armando[1], N. Van Wambeke[1], C. Chassot[1,2], E. Exposito[1,2] and K. Drira[1]

[1]LAAS-CNRS, [2]Université de Toulouse / INSA – Toulouse – France

Keywords: Dynamically configurable transport protocol, Policy-based management, QoS provisioning, model-based decision.

Abstract: Self-adaptation of communication protocols is a major issue in the conception of future QoS-oriented services for the ambient Internet. Our approach is based on behavioural and architectural adaptation properties of dynamically configurable Transport protocols. This paper proposes an architecture for the QoS provisioning at the Transport level. To fulfil this provisioning, the decision process follows a policy-based framework, using different external models in order to have an extensible design. We illustrate the use of this framework in a case study for the QoS optimization of a mobile user, roaming from a wired network to a wireless network.

1 INTRODUCTION

Recent advances in computing and wireless networking technologies allow considering the deployment of complex wireless, mobile and cooperative applications within a fully pervasive, mobile and heterogeneous Internet. For instance, military emergency operation management systems are a typical example of such applications. They support mobile users cooperating applications for crisis management in linked to variable communication resources that change depending on the investigated field, the role of the participants in the operation, etc.

From a communication point of view, multiple user and application time-varying requirements have to be satisfied. They depend on the communication tools used by the participants (e.g. interactive audio/video, sms ...); they also depend on the evolution of the activity which can make it different the interactions (e.g. nature, priority) between the participants, for instance when one of them discovers a critical situation. By the way, multiple time-varying constraints (e.g. power, bandwidth) are also to be considered, depending on machine and network resources which are used by the participants.

In such a complex and dynamic context, future communication systems are expected to have behavioral and architectural self-adaptation properties, aimed at tackling different kinds of dynamic requirements, still considering dynamic constraints associated with the network/machine environment.

Accordingly, several solutions have already been proposed in the literature; they differ in several points related to the targeted goals (e.g. QoS, security …), the addressed levels (Application, Transport …), the adaptation actions and their properties. Particularly, protocols whose internal architecture may be dynamically composed appear to be very suitable for both behavioral and architectural self adaptation (Bridges et al., 2001, Exposito et al., 2003, Mocito et al., 2005, Wong et al., 2001). Our approach is based on adaptation of such protocols at the end-to-end level (Transport level - TCP level -and above). The goal is to match as best as possible dynamic application QoS requirements, while taking into account network resource and context changes.

Dynamic configuration of adaptive protocols raises different classes of problems that we address for the Transport level adaptation.

Problem 1: Design and evaluation of *new Transport mechanisms and protocols*, allowing optimizing a given QoS requirements with regard to the network resource constraints.

Armando F., Van Wambeke N., Chassot C., Exposito E. and Drira K. (2007).
A FRAMEWORK FOR THE DYNAMIC CONFIGURATION OF ADAPTIVE TRANSPORT PROTOCOLS - Application to QoS Requirements.
In *Proceedings of the Second International Conference on Wireless Information Networks and Systems*, pages 259-267
DOI: 10.5220/0002151202590267

Problem 2: Elaboration of *provisioning rules* for the selection and adaptation of the Transport mechanisms to be (re)composed / (re)parameterized.

A major difficulty, without generic solution at the current time, is to ensure coherency of the composition / parameterization choices, both within and between the considered adaptation levels. Tackling these needs by means of informal models may lead to non generic or suboptimal solutions due to the complexity of the problem. In front of this limit, formal model-based design constitutes a promising approach (Farkas et al., 2006, Landry et al., 2004), particularly studied in our work for architectural self adaptation at different levels of the end-to-end communication stack (application, middleware and transport levels) (Chassot et al., 2006a). In (Van Wambeke et al., 2007), we propose an analytical model aimed at helping the decision process at the Transport level was designed.

Problem 3: *Design of an architecture for the provisioning process, and then the enforcement* of the operational rules on the active communication elements (e.g. adaptive Transport protocols).

It is the major goal of the NETQoS project to design such architectures. The proposed approach starts from the policy-based network management (PBNM) model, which is extended to reach several goals:

- management of several kinds of actor's policy: operator, service provider, user and application;
- dynamic adaptation, not only at the network level but also at the Transport level.

The main contribution of this paper has been performed within the NETQoS project, and mainly deals with the third problem exposed hereafter. We present a framework for a model-based provisioning and enforcement of configuration / adaptation rules for adaptive Transport protocols. This framework is then illustrated with a case study.

The rest of the paper is structured as follows. Section 2 describes related work. Section 3 describes the elements of the framework and the provisioning process. Section 4 provides details about the elaborated models for Transport-level adaptation. Section 5 provides a case study illustrating the provisioning process within the proposed framework. Section 6 provides conclusion and future work.

2 RELATED WORK

There are many existing solutions for context adaptation. A complete study and classification can be found in one of our previous works (Chassot et al., 2006b). In what follows, this classification is summarized and dynamically configurable transport protocols are presented.

2.1 Classification of Context Adaptation Solutions

Adaptation objectives, techniques and properties are among the main facets of adaptability. They are studied and classified in this section.

2.1.1 Adaptability Objectives

Adaptability targets several objectives. QoS aspects such as connectivity or access bandwidth issues in roaming are considered in (Kaloxylos et al, 2006). End to end QoS optimization in the Best Effort Internet makes heavy use of adaptation techniques (Akan and Akyildiz, 2004). Security in wireless networks, such as firewalls activation and deactivation, can also benefit from adaptability (Perez and Skarmeta, 2004). Resources optimization related to device power, computation or storage capability are presented in (Marshall and Roadknight, 2001).

2.1.2 Adaptation Techniques

Adaptation techniques target all layers of the OSI model.

Application layer – (Wu et al., 2001) addresses adaptation of video streaming applications for the Best-Effort Internet. The proposed techniques are based on two mechanisms: an applicative congestion control (rate control, rate-adaptive video encoding) and time aware error control with FEC.

Middleware layer - Reflexive architectures such as OpenORB or Xmiddle (Capra et al., 2003) are good supports for adaptation as they allow run-time modification of the architecture.

Transport layer - TCP's congestion control is a well-known adaptation example. The IETF DCCP protocol allows users to choose the congestion control. SCTP targets adaptation to network failures using multi homed associations. (Akan and Akyildiz, 2004) studies various types of mobile applications in wireless Internet. Adaptation consists in parameterization of congestion control mechanisms using context information. (Bridges et al., 2001, Exposito et al., 2003, Mocito et al., 2005, Wong et al., 2001) study the architectural adaptation of transport protocols by dynamic composition of protocol modules. Next section (2.2) is dedicated to these frameworks illustrating the modular architecture concept targeted by our work.

Network layer – (DaSilva et al., 2004) addresses QoS-aware routing problems within ad-hoc mobile networks. In (Wong et al., 2001), dynamic and secure provision of IP services for military wired/wireless networks is considered. In a policy-based networking management context, the need for self-adaptation is considered in (Samaan and Karmouch, 2005), using a learning-based approach.

MAC layer - The solutions handle connection and access QoS problems for mobile users using different terminals and roaming protocols. (Kaloxylos et al., 2006) provides a solution for optimizing the handover latency but the other QoS requirements are not considered.

2.1.3 Adaptation Properties

The adaptation solutions suggested in the literature are defined in various ways.

The adaptation is *behavioral* when the execution of a service can be modified without modifying its structure. TCP and specific protocols such as the ones in (Akan and Akyildiz, 2004) provide behavior-based adaptation. It is easy to implement but limits the adaptability range. Indeed, the addition of new behaviours requires the component to be recompiled and the adaptation can no longer be performed during run-time.

The adaptation is *architectural* when the structure of adapting services can be modified. The replacement of a component by another can be implemented following a plug and play approach where the new component has the same interfaces as the replaced one.

Finally, adapting components can reside on a single machine or be distributed. In the first case, adaptation is *vertical* and changes are performed only locally. In the second case, it is horizontal and synchronization between peer adapting entities has to be managed.

2.2 Dynamically Configurable Protocol Architectures

Dynamically configurable protocol architectures are based on the *protocol module* concept. A protocol module is a primitive building block (Hutchinson and Peterson, 1991) resulting from the decomposition of the protocol's complexity into various successive elementary functions. A protocol is then viewed as the composition of various protocol modules in order to provide a global service.

These architectures can be refined into two different categories depending on their internal structure: the event based model (followed by Coyote and Cactus) and the hierarchical model (X-Kernel (Hutchinson and Peterson, 1991) and APPIA). ETP follows an hybrid approach combining both models (Exposito, 2003).

These protocol architectures appear as a good choice for future communication protocol's *self-adaptation* as they are capable of *run-time* architectural adaptation, meaning that the modules composing them can change during the communication. This *run-time* architectural adaptation raises many problems such as: (1) synchronization of adapting peers; and (2) the choice of the best composition guided either by the user's requirements or by the modification of the context.

3 THE FRAMEWORK COMPONENTS AND THE PROVISIONING PROCESS

The NETQoS IST project is addressing the problem of QoS in IP Networks from Policy based network management architecture, to provide flexible and adaptive end to end QoS provision, unlike number of solutions provided by bottom-up approach. The NETQoS system introduces the notion of predefined policies as a promising solution to address the needs of QoS traffic management.

In order to address the policy based QoS management, policy architecture was defined. Policies prescribe a set of rules based on the users/application requirements, which are transformed into high level network level policies. Since the network level-policies are derived from business objectives, users and applications requirements described in the SLA, these policies have to evolve and to be adapted to the changes in these objectives and requirements in a timely manner. An autonomous self-adaptable policy-based management framework with inherent dynamic capabilities to best manage, customize and extend the underlying complex infrastructure of communication systems resources in response to the continuously changing business objectives and users requirements, is the main goal of the NETQoS project. The enforcement policies thus generated are translated into network device dependent rules to configure the network

3.1 The framework for Dynamic Policy Provisioning

The general architecture of the NETQoS system distinguishes four main entities or systems (Fig. 1):

- the *Policy Description* (PD) implements a set of ontologies used to specify the actor-level policies, the operational policies, etc;
- the *Actor Preference Manager* (APM) provides NETQoS GUI/API allowing users to define actor-level dynamic policies to the NETQoS system based on ontologies. These policies (e.g. requirements, preferences, profile, quality reporting…) may be expressed before or during the communications;
- the *Automated Policy Adaptor* (APA) is the central entity of the NETQoS system. It does not provide QoS by itself, but provides and dispatches operational policies (namely Network and Transport level policies) that allow the communication system to take into account the actor-level dynamic policies;
- the *Monitoring and Measurement* (MoMe) implements all the monitoring and measurement tasks associated with: (1) context evolution, e.g. actor's policies evolving, end systems/network resource changing …; and (2) evaluation of the operational policy efficiency and the reporting of quality information to the actors.

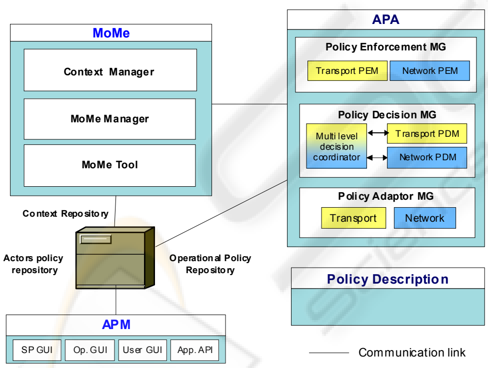

Figure 1: The general NETQoS architecture.

3.1.1 The Automated Policy Adapter

The APA is aimed at deciding, dispatching and adapting the operational policies that allows the communication system to take into account the actors' level dynamic policies taking into account the context evolving. The APA is composed of three main components:

- the *Policy Decision Manager* (PDM) has in charge deciding the set of operational policies that implements the actor-level policies. This mapping may change depending on context-related information;
- the *Policy Enforcement Manager* (PEM) has in charge the deployment of the policies decided by the PDM on their Network/Transport-level enforcement points;
- the *Policy Adaptation Manager* (PAM) has in charge the adaptation of policies, individually or by grouped, when the success criteria associated with a policy is not reached.

Policy decision manager. The PDM is aimed at deciding an optimal set of policies to be settled at the Network and/or at the Transport level to satisfy the set of actor-level policies. This *policy provisioning* may be performed using rules taking into account the dynamicity of the actor's policies and changes in the context information; these rules are elaborated using a model-based approach (see section 3.3.2). The policy provisioning may lead to policy conflicts that the PDM has to solve, for instance when actor-level policies cannot be reached as required.

Each time the PDM decides a (new) operational policy, it provides the corresponding rules to the PEM. If the policy is enforceable, the PDM informs the PAM of the enforcement of a new policy.

Policy enforcement manager. The PEM is in charge of dispatching the operational policies decided by the PDM to the actual policy enforcement point (PEP). For instance, for a Transport level adaptation, the PEM dispatches the Transport protocol configuration rules to be applied on the end nodes.

The PEM is independent of the Network and Transport technologies that are used to really enforce the policies, i.e. the PEM provides the rules to be performed in a language that is independent of the ones used by targeted QoS-oriented communication system. Consequently, the PEM provides generic interfaces allowing the different entities to communicate with the actual PEP, and adaptors have to be implemented, for instance on the PEP themselves, to translate the generic PEM rules into specific technology-dependant rules.

Policy adaptation manager. The PAM has in charge the adaptation of a policy when its success criteria are not reached. The PAM may decide to adapt the policy or an associated subset of relevant policies. When adaptation is not possible, the PDM is informed of the current policy failure, possibly associated with a diagnosis.

The PAM mainly acts in the following situations:

- when the PAM is informed by the PDM that a new operational policy is enforced, the PAM stores the policy in the operational policy repository together with some success criteria (e.g. end-to-end packet loss must be smaller than 5%).
- when the PAM receives alarms from the MoMe, it tries to adapt the failed policy and informs the PDM of the result.

3.3.2 Policy Provisioning

Two different *policy enforcement levels* are considered: the *transport level* and the *network level*. For both levels, *policy provisioning* deals with the way the operational policies are elaborated, selected and adapted.

This process is complex as it requires knowledge of many context aspects such as access network or actors' preferences. The component that performs policy provisioning is the central element of the NETQoS framework.

Various techniques exist for the provisioning process. The simplest one consists in having a set of predefined rules which are hard coded into the deciding component producing a set of predefined responses that depend on the environment. The main aspect (among many others) that makes this simple technique inefficient is its lack of extensibility. Indeed, being hard coded, the policies that manage the decision process are not changeable during run-time.

In order to have an extensible decision process, it is possible to guide it using different external models such graph-based models presented in (Chassot et al., 2006b) where all aspects of communication can be represented and the different evolutions of the system are characterized as elements of a graph grammar. In such models, the adaptation is thought ahead of time providing fast response to changes. Moreover, these models can be stored in an external repository and may change during the communication. By doing so, the set of responses that the decision component may have is not statically predefined at design-time and can be further extended by refining the models that manage the system evolving during run-time.

Moreover, different models can be used at different abstraction levels. Hence, the cooperation and interaction description can be implemented by a labelled and directed graph model that may be transformed following a set of graph grammar rules into a transport connection model. At the transport level, specialized models such as analytical models can be used in order to get a policy response which is best suited to the environment while taking both actor requirements and preferences into account.

In the NETQoS system, the component in charge of this decision process is the APA. Various sub components take place in this provisioning process, the PDM is responsible for performing the provisioning. The PAM then takes the necessary re-provisioning actions in order to perform adaptation.

1) **Provisioning.** In order to perform provisioning at the Transport and Middleware levels, policies in place at the network level have to be known. Due to this constraint, the natural order for the provisioning consists in initially performing network provisioning prior to transport provisioning. Once these two steps are done, middleware provisioning can start.

2) **Adaptation.** Once the adequate communication services have been selected and deployed, specific adaptation actions are performed in order to maintain the QoS required. The adapting conditions are expressions based on the current media flows and applications regarding the QoS goals computed during the decision process. One possibility is to adapt the middleware level first then the transport level and finally the network level.

3.3.3 Policy Provisioning Process

In order to perform provisioning at both the network, transport and middleware levels, the PDM component has been divided in two different components that are responsible for (N-PDM) network level provisioning, (T-PDM) transport level provisioning and (M-PDM) middleware level provisioning.

The policy provisioning process can be lead using different models at these various levels. In order to support this adaptation, various architectures are possible. The configuration and outsourcing models defined by the IETF (Boyle et al., 2000) are suitable for such task.

For instance, for an interactive and high priority flow, a network service characterized by low delay and high reliability could be selected in a first time (e.g. EF service). In a second time the selection of a basic UDP transport service without additional middleware mechanisms would be enough to guarantee the satisfaction of the user. In contrast, if the EF service has not been declared as available for this user (e.g. following the service provider policies), a Best-Effort network service could be selected, combined with an UDP or DCCP transport service and completed with an error control mechanism implemented at the middleware level.

The design can be done as illustrated by the figure 2 below. In this architecture, the outsourcing model is followed; the PDM component represents any of the N-PDM, T-PDM or M-PDM. The decision is not directly performed by the PDM but it is delegated to external modules (represented by squares on the diagram). In each of these modules, a specific decision algorithm is implemented in order to construct a potentially partial policy. The successive invocation of various modules (TD...ND) will lead to a valid policy being generated.

For each of the outsourced modules, the access method can be of any kind (from simple RPC to XML/SOAP Web Services). This design architecture allows easy extension of the system by simply adding new decision components. By such, any new model that is produced by the different actors (Operator/Service provider) can be deployed and used without requiring modification of the PDM itself. The PDM acts here as a system kernel managing the process by delegating tasks to the various external elements successively.

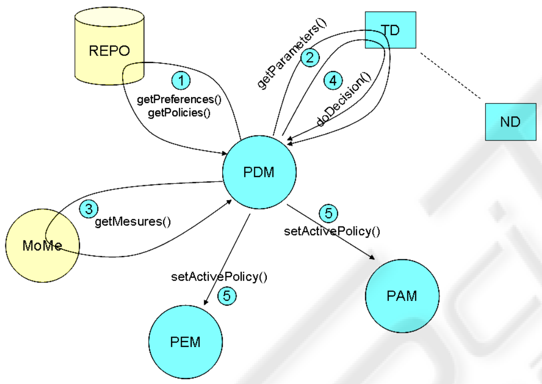

Figure 2: Policy provisioning process architecture.

The above figure presents an outsourcing based model for the PDM decision process. When necessary (e.g. when a new connection is to be established in the NETQoS domain), the PDM retrieves the information regarding the actor preferences and associated policies from the repositories. These policies contain information that allows the PDM to contact the appropriate software module to perform the decision. Technologies such as RMI, Web Services or CORBA may be used for this task. As this software module is external to the PDM, this model is largely extensible by the simple addition of new modules and simply describing them in the repositories. This action also allows users to choose and classify the adaptation actions provided by these new modules in their preferences.

In order to support different interfaces, the invocation takes place in two stages. Stage 1 allows for discovering the parameters required by the outsourced modules (i.e. inputs to the decision model they hold inside). The PDM is then responsible for retrieving the necessary information from the other components in the NETQoS system (e.g. MoMeTool or Context Manager). Once these parameters have been retrieved, the outsourced decision service can be invoked by the PDM returning the decided policy. The PDM could then continue invocation following the users' preferences (i.e. perform adaptation both at Middleware and Network levels simultaneously) which would lead to further modification of the policy being decided upon. At the end of this process, the PDM has constructed the policy to be activated and deployed in order to support the new communication. This decision is then transmitted to the PAM and PEM in order for them to take the necessary actions to respectively, monitor and react to adaptation needs by enforcing and deploying the policy via the PDPs.

4 THE PROPOSED MODELS FOR THE TRANSPORT-LEVEL ADAPTATION

4.1 Adaptation at the Transport Level

Transport-level adaptation consists in dynamically selecting and adapting parameters and/or internal architecture of the adaptive Transport protocol used to transfer an application data flow. The goal is to match as best as possible dynamic and hierarchical requirements, i.e. an actor's policy, associated to a given application, taking the current communication context (e.g. access networks used by the terminals hosting the applications):

- the processing of an actor's policy, typically a user's policy for a given application when this one is launched, may lead to Transport-level adaptation for the connection underlying this application only. It may also lead to Transport-level adaptation for other connections when hierarchical relationships (e.g. priorities) have been defined between the actor's policies, for instance when a user has defined priorities between his/her applications;
- the communication context may evolve depending on several factors, for instance when the hosting terminal is moving from an access network to another.

To handle dynamic and hierarchical policies, together with an evolving communication context, two kinds of Transport policies, namely per connection handling, and per group of connections handling, may be applied:

- *per connection handling* means that the adaptation rules are applied independently, connection per connection;
- *per group of connections handling* means that several connections may be coordinated by the adaptation rules, allowing taking into account hierarchical policies, for instance expressed by a user on his/her applications.

Both policies are refined in operational rules that consist in composition and parameterization rules of the transport protocol implementing each connection.

4.2 The Proposed Models for Policy Provisioning

In what follows, a model for automatically selecting the best composition of protocol modules at the transport/middleware level is presented. This model has been previously presented in (Van Wambeke et al., 2007) and is only briefly recalled here.

We model a protocol module (A) by the following matrix:

$$A = [\, A^P \, A^R \,]$$

$$A^P{}_i = \begin{cases} 1 \text{ if A produces variable } i \\ 0 \text{ otherwise} \end{cases} \qquad A^R{}_i = \begin{cases} 1 \text{ if A requires variable } i \\ 0 \text{ otherwise} \end{cases}$$

Given this description, we are able to describe a valid composition as respecting the following constraints:

$$C\text{-valid} \Leftrightarrow \begin{cases} (\sum_i (A^P)_i)_k \leq 1 & \forall k \in [1..N] \\ \sum_i (A^P_k)_i > 0 & \forall k : \sum_i (A^R_k)_i > 0 \\ \forall i : A_i \in D_i \text{ then} & \nexists A_j \in C : (A_j \in D_i) \end{cases}$$

Additionally, for each module, efficiency vectors are defined which allow for evaluating the module's efficiency in the various contexts that are considered.

Based on the above, the decision process comes to solving the following optimization problem:

$$max \quad (E = \sum_k (A^E_i)_k)$$
$$\text{under} \begin{cases} \textit{c-valid} \\ \textit{additional constraints} \end{cases}$$

The additional constraints are produced by the refinement of the policies. For example, a user might specify that he is using a low memory device which would be refined into a constraint that limits the number of unused variables production.

$$min(\sum_k (\sum_i (A^P_k)_i - \sum_i (A^R_k)_i)_k)$$

5 CASE STUDY: QOS OPTIMIZATION FOR MOBILE USERS IN NETQOS

5.1 Scenario Description

This scenario aims at illustrating Transport adaptation by instantiating the above presented model in the decision process.

<u>*Assumption*</u>

For this scenario, let's assume that

- the user's terminal and the server are connected to high speed wired access networks, interconnected by a Best Effort Internet domain;
- the user's policies (applicable to the present context) are as follow:
  - "when I'm on a wired connection, I would rather have ensured QoS of Gold quality for all my applications";
  - "when I'm on a wireless network, I would rather have my communications classified by decreasing level of quality requirements as follow: Audio call, Video call, VoD movie, Internet Radio, Web, Mail and similar, File transfers".

While the user is connected to the wired network, the service provider's policies are such that the maximum amount of bandwidth that the user might use is 10 Mbps, thus, resource reservations will be performed and updated until this limit is reached. Note that these reservations are handled by any QoS system in place such as the one proposed in the IST FP6 EuQoS project.

Let's now assume that:

- at t = t0, the user starts to work at his office desk;
- at t = t1 > t0, the user disconnects his laptop and the wireless interface becomes the default interface (this is handled by the OS itself).

5.2 Interaction of Policy Actors and NETQOS Components

The NETQoS components that are involved in this scenario are:

- MoMe (Monitoring and Measurement): Context Manager (CM) and MoMeTool,
- APA (Automated Policy Adapter): PDM, PEM and PAM (for the Transport only, i.e. adaptation at the network level is not considered in this scenario),
- REPO (Policy Repository).

The following illustrate the main scenario steps.

1) At the NETQOS system initialization, the APA subscribes to the CM for a set of events: application launch, policy violation, context modification etc.

2) A t = t0, the launch of various applications is detected by the CM; for each one:

- the CM informs the PDM that an application is launched;
- the PDM retrieves from the REPO the policies related to the identified user and application; these policies contain information that allows the PDM to contact the appropriate external software module that decides the Middleware/Transport protocol configuration to be selected with regard to the policy defined for the ftp application;
- the PDM contacts the outsourced software module, and provides it with the necessary information to decide the rules to be applied;
- the appropriate rules for Transport protocol configuration are returned to the PDM by the outsourced software module following the model presented above;
- those rules are then transmitted to the PAM and PEM in order for them to take the necessary actions to respectively, monitor and react to adaptation needs and enforce and deploy the policy via the PEP at the Transport layer;
- if the MoMeTool, via the CM informs the PAM that success criteria associated to the selected policy are no longer met (via alarms); the PAM may try to adapt the policy and may also inform the PDM if necessary. For instance, the adaptation action can consist in changing parameters of the micro-protocols.

3) A t = t1, the change of network context is detected by the CM

- the CM informs the PDM that the user is moving to a different network;
- the PDM retrieves from the REPO the policies related to the identified user and application for the new context; these policies contain information that allows the PDM to contact the appropriate external software module that implements the model presented in the previous section in order to decide on the Middleware/Transport protocol configuration to be selected with regard to the policy defined for the currently running applications.
- At this step, two possibilities may be considered.

  <u>1st possibility</u>

  - the outsourced software module provides the PDM with the set of rules that can be applied for the application;
  - taking into account its current execution context (here, various active applications for which the QoS requirements have changed), the PDM/PAM selects the Transport protocol configuration / adaptation to be applied for each connection.

  <u>2d possibility</u>

  - the appropriate rules for Transport protocol configuration/adaptation for the new connection and possibly existing connections are returned by the outsourced software module;
  - those rules are then transmitted to the PAM and PEM in order for them to take the necessary actions to respectively, monitor and react to adaptation needs and enforce and deploy the policy via the PEP at the Transport layer;
- if the MoMeTool, via the CM informs the PAM that success criteria associated to the selected policy are no longer met (via alarms); the PAM may try to adapt the policy and may also inform the PDM if necessary.

When an application is stopped, the PAM and PDM may also apply reconfiguration actions.

6 CONCLUSION

In this article, we presented a framework for the dynamic configuration of adaptive Transport protocols in order to support policy-based QoS provisioning for heterogeneous, mobile and cooperative activities in a pervasive Internet. The main components of the general policy architecture have been described, especially the APA component, responsible for the policy decision process of adaptation actions, in particular at the Transport level. A case study aimed at displaying the framework's usage and internal interactions in the case of mobile, multi-network, continuous, seamless communications has been presented. The models which govern the policy-based decision and provisioning process have been illustrated in this context.

Future works on the topic include, but are not limited to, the implementation of the presented framework. Moreover, evaluation of the efficiency of the different mechanisms that compose the current architecture has to be carried on.

ACKNOWLEDGEMENTS

Part of this work presented has been carried on as part of the European IST FP6 NetQoS project. In conjunction with this, several aspects of this works are funded by DGA grant.

REFERENCES

Bridges, P. G., Chen, W.-K., Hiltunen, M. A., Schlichting, R. D., 2001. Supporting coordinated adaptation in networked systems, In *8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Elmau, Germany*.

Exposito, E., Diaz, M., Senac, P., 2003. Fptp: the xqos aware and fully programmable transport protocol, In *11th IEEE International Conference on Networks (ICON'2003), Sydney (Australie) - pp.249-254*.

Mocito, J., Rosa, L., Almeida, N., Miranda, H., Rodrigues, L., Lopes, A., 2005. Context adaptation of the communication stack, In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05), Colombus, Colombus, OH, USA*.

Wong, G. T., Hiltunen, M. A., Schlichting, R. D., 2001. A configurable and extensible transport protocol, In *Proceedings of INFOCOM, Anchorage, Alaska, USA, pp. 319–328*.

Farkas, K., Wellnitz, O., Dick, M., Gu, X., Busse, M., Effelsberg, W., Rebahi, Y., Sisalem, D., Grigoras, D., Stefanidis, K., Serpanos, D., 2006. Realtime service provisioning for mobile and wireless networks, In *Elsevier Computer Communication Journal Vol. 29, No. 5, 540–550*.

Landry, R., Grace, K., Saidi, A., 2004. On the design and management of heterogeneous networks : A predictability-based perspective, In *IEEE Communications Magazine Volume 42, Issue 11 Page(s):80 – 87*.

Chassot, C., Lozes, A., Racaru, F., Auriol, G., Diaz, M., 2006a. A user-based approach for the choice of the ip services in the multi domains diffserv internet, In *1st International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNET'06)*.

Van Wambeke, N., Armando, F., Chassot, C., Exposito, E., 2007. A model-based approach for self-adaptive Transport protocols, *to appear in Elsevier Computer Communications, Special issue on End-to-end Support over Heterogeneous Wired-Wireless Networks*.

Chassot, C., Guennoun, K., Drira, K., Armando, F., Exposito, E., Lozes, A., 2006b. Towards autonomous management of qos through model driven adaptability in communication-centric systems, In *International Transactions on Systems Science and Applications 2 (3) 255–264*.

Kaloxylos, A., Lampropoulos, G., Passas, N., Merakos, L., 2006. A flexible handover mechanism for seamless service continuity in heterogeneous environments, In *Elsevier Computer Communications Volume 29, Issue 6, 717–729*.

Akan, O. B., Akyildiz., I. F., 2004. ATL: An adaptive transport layer suite for next-generation wireless internet, In *IEEE Journal on Selected Areas in Communications Vol. 22 (No. 5) 802–817*.

Perez, G., Skarmeta, A. G., 2004. Policy-based dynamic provision of ip services in a secure vpn coalition scenario, In *IEEE Communications Magazine Volume 42, Issue 11: 118 – 124*.

Marshall, I., Roadknight, C., 2001. Provision of quality of service for active services, In *Elsevier Computer Networks Volume 36, Issue 1 75–85*.

Wu, D., Hou, Y. T., Zhu, W., 2001. Streaming video over the internet: Approaches and directions, In *IEEE Transactions on Circuits and Systems for Video Technology vol. 11, no. 1*.

Capra, L., Emmerich, W., Mascolo, C., 2003. Carisma: Context-aware reflective middleware system for mobile applications, In *IEEE Transactions on Software Engineering 29 (10) 929–945*.

DaSilva, L. A., Midkiff, S. F., Park, J. S., Hadjichristofi, G. C., Davis, N. J., Phanse, K. S., Lin, T., 2004. Network mobility and protocol interoperability in ad hoc networks, In *IEEE Communications Magazine Volume 42, Issue 11 88–96*.

Samaan, N., Karmouch, A., 2005. An automated policy-based management framework for differentiated communication systems, In *IEEE Journal on Selected Areas in Communications vol. 23, no. 12*.

Hutchinson, N. C., Peterson, L. L., 1991. The x-kernel: An architecture for implementing network protocols, In *IEEE Transactions on Software Engineering 17 (1) 64–76*.

Exposito, E. J., 2003. Design and implementation of quality of service oriented transport protocol for multimedia applications, *Ph.D. thesis, National Polytecnic Institute of Toulouse, Network and Telecommunications*.

Van Wambeke, N., Armando, F., Chassot, C., Exposito, E., 2007. Architecture and Models for self-adaptability of Transport protocols, to appear in *Proceeedings of the 2nd International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE'07)*.

Boyle, J., Cohen, R., Durham, D., Herzog, S., Rajan, R., Sastry, A., 2000. The COPS (Common Open Policy Service) Protocol, *RFC 2748*.