END-TO-END QOS MANAGEMENT AND REAL TIME AGREEMENT PROTOCOL FOR RESOURCE RESERVATION FOR MULT IMEDIA MOBILE RADIO NETWORK

Sonia Ben Rejeb [1,2], *Zièd Choukair* [1], *Sami Tabbane* [1]

[1] Unité de recherche Médiatron

Ecole Supérieure des Communications de Tunis
Route de Raoued, km 3.5, 2083 Cité el Ghazala - Tunis

[2] Ecole Nationale Supérieure des Télécommunications de Bretagne
Technopole Brest-Iroise-Bp 832-29285-Brest Cedex - France
sonia.benrejeb@enst-bretagne.fr, sami.tabbane@supcom.rnu.tn

ABSTRACT

Third generation mobile networks is expected to provide services to users, while guaranteeing acceptable QoS (*Quality of Service*).
UMTS is a typical example of these new networks. Thus, this paper is focused on UMTS multimedia domain and its strategies of control of session and resources pre-reservation. In order to offer the various services, while respecting acceptable required level of QoS, it is necessary to guarantee minimum QoS for each service. Resources managers which we designed are then oriented towards the use of pre-reservation with policies to manage consensus. This orientation is also considered by 3GPP which adopted COPS and SIP protocols for the procedures of session authorization and domain resources IM of UMTS network.
In such context, the work developed in this paper is focused towards the setting up of a consensus protocol included inside management and check procedures for resource reservation for multimedia services in UMTS/IP network. The evaluation of the feasibility and of the consensus constitutes the main contribution of this work.

1. INTRODUCTION

Quality of service has a significant and direct impact on the performance of multimedia services. These applications generate flows having different outputs and can tolerate a certain threshold of time variation and gigue in the network. Nowadays, deployment of multimedia applications, which form most of traffic network, represent a great challenge due to their requirements in term of QoS. IP Protocol is known as best effort. It doesn't offer any guarantee of QoS for this type of application. That is to say that there is no mechanism in IP but windowing to control and manage flows which can cause a network congestion. Hence, the management of QoS is dealt with, in most cases, by applications themselves.

Different mechanisms were proposed to provide QoS for IP best effort. There is two distinct architectures: IntServ architecture (*Integrated Service*) [1], accompanied with its indication protocol RSVP (*Resource Protocol Reservation*) [2], and DiffServ architecture (*Differentiated Services*) [3].

In this paper we propose a control and pre-reservation model in DiffServ domain oriented towards multimedia flow. Our system provides QoS for multimedia flows which are adapted as well as possible to the conditions of a cellular network (bandwidth, rate of loss, rate of blocking). The model is established while taking into account the fact that, during a session, multimedia applications are flexible and tolerant compared to QoS networks parameters, in particular, packet loss, time variation and gigue, etc. In this prospect, our system exploits multimedia applications flexibility to the network conditions and adapts in a dynamic way QoS of the multimedia flow.

Our model is based on resource pre-reservation. This model is based upon a consensus algorithm. This algorithm is used to avoid arbitrary resources reservation. The consensus adopts alternatives when resources are not available in the same route and detects failures in terms of deadline overstepping. The allocation model manages consensual negotiation of resources according to SLA (*Service Level Agreement*) levels.

Next section presents the state of the art. Section 3 develops the topology of the adopted network. In section 4, we develop our consensus protocol inspired from *Losange S*, a Distributed asynchronous consensus model with deadline with failure detector. We end this paper by a conclusion and perspectives for this work.

2. STATE OF THE ART

An immediate approach to ensure QoS for multimedia sessions is based on a resource reservation mechanism. However, resources reservation requires a reservation protocol, an admission control mechanism and a general application control mechanism policy. These mechanisms are complex but they must not be intrusive at the level of current networks infrastructures, otherwise there will be an impact on network performance and scalability [4].

We will present some approaches for resources reservation and adaptation which take into account fluctuating conditions of the network. An architecture based on the concept of QoS directed applications is presented in [5]. A broker allows the applications and the suppliers of services to reserve the resources of the network. These brokers can be hierarchical, a broker requesting services from other brokers. This reservation diagram is static and the calculated status of network resources by the broker does not correspond exactly to the reality. The validity of the information on which the decisions are based is of a major importance. A key concept of their proposal is a feedback loop which controls the adjustment of the application to the conditions of the network. According to this proposal, the application must set up a monitor which takes regular measurements on QoS and an engine which reacts in response to the received information.

In [6], the application can adapt to execution conditions thanks to an adaptation system integrated in its execution environment. The system continuously controls the resources available and the behavior of the application in term of guarantee of QoS. It adjusts automatically this QoS according to available execution profiles. In [4], the adaptation is left to edge routers. Such routers allow the improvement or degradation of QoS of multimedia flow according to the state of the network.

In our system, resources reservation and adjustment are left to Resource Managers (RM) of each domain which take part to the application. Dynamic pre-reservation of multimedia flow resources consist in finding, when necessary, a road with an acceptable level of SLA.

3. PROBLEMATIC

If resources are not available or are not satisfactory for one of the RM, the system may undertake two actions :

- The system changes the path locally (inside the domain) and tries to negotiate another path in order to satisfy the user.
- The system offers a lower level of service which satisfies a lower level of SLA accepted by the user. The system will recover later the previous SLA.

The decision has to be shared by all the RM which domain is implied in the transfer of the media. We considered two models which correspond to a unique caller and many users participating to a collaborative or peer to peer application [13]-[14]-[15]:

- Distributed asynchronous consensus model with deadline with failure detector : *Losange S*.
- Centralized asynchronous consensus model with deadline : *DWD*.

It is difficult, in a centralized scheme, to ensure fairness in the performance centralized between nodes because of hidden node and other effects (such as channel capture) caused by radio propagation fluctuation. Clearly, these problems are most acute when there is high loading and therefore contention for network resources.

Our solution proposes to use distributed algorithm for setting up a consensus among participating resources managers of the impacted domains. This algorithm targets to make these managers converge towards a consensus on the reservation of a dynamic path with an acceptable level of QoS. The path reserved for the media should consider at the same time available resources in the network (contractual levels of service specified within the SLA) as well as requested QoS by each media.

4. OUR APPROACH

4.1. Graph theory

A grid system can be defined (modeled) as graph G=(V,E) with an ingress point (source) and an egress point (destination), where V represents adjacent domains group and E represents connections/path between radio operator mobile groups.

Arch (I, J) (band-width) represents the shortest path between a domain in I and the final domain in J;

We model our network according to following notations (see Figure 1) :

- We have an adjacent domains RM distributed on the coverage zone and offering SLA services, they represent SLA={x1, x2,…,xq };
- RM being provided with availabilities [b1, b2,…,bq] which represent maximum flows offered by each source (maximum capacity offered by a field);
- Each mobile requires a particular service from the P services available and each one requires a maximum of flow to be satisfied. The request of the users is comparable to the maximum of the expressed request using [a1, a2,…,aq];
- Bandwidth of each connection between different nodes from our network is comparable to the capacity of the archs;
- S_{ij} : series of the votes according to the number of voting domains;
- K : the number of voting domains;
- s_k : source domain;
- d_k : destination domain;
- $\varphi(i,j)$: the flow of data of the arch (I, J);
- c_{ij} : capacity of the archs.

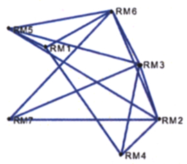

Figure 1. Network model

4.2. Problematic formulation

In the case of our network, we seek to satisfy several mobiles located in various zones of operator coverage. Various services (SLA contracts) are presumably offered by the operator to his subscribers.

Losange S algorithm is based on the network model which was presented previously. The series of the votes realized by the algorithm coordinator is expressed by the following expressions:

$$S_{ij} \quad Maximise \int_{(i,j)\ E} \int_{k\ 1}^{k} x_{ij}^{k} \tag{1}$$

with $\int_{j:(s_k,j)\ E}^{k} x_{s_k j} \ 1$; $\int_{i:(i,d_k)\ E}^{k} x_{i,d_k} \ 1$; k=1,…,K

The series S_{ij} converges for all $c_{ij} > 0$; there is an integer K such as for k<K :

$$S_{ij} \quad Maximise \int_{(i,j)\ E} \int_{k\ 1}^{k} x_{ij}^{k} \delta c_{ij} \tag{2}$$

5. A DISTRIBUTION AND ASYNCHRONOUS REAL TIME CONSENSUS MODEL : LOSANGE S

Based on the studies and the approaches adopted within [10]-[11]-[13]-[14]-[15], the system of *Losange S* consensus to simulate is thus formed by a set of nodes/votants (RM) connected through a DiffServ network. The nodes are synchronized and collaborate for an acceptable SLA level (see Figure 2).

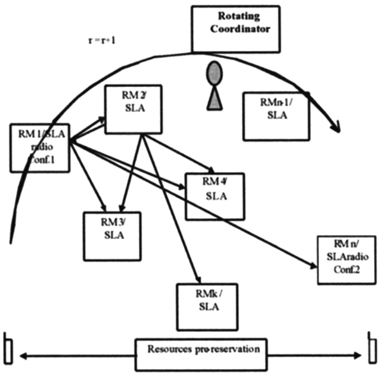

Figure 2. Consensus mechanism using a coordinator

Losange S algorithm uses three stages:

- Vote.
- Waiting.
- Decision.

and is represented by the following procedure:

Consensus procedure : ***Losange S*** **(** RM_i , r_{RMi} , S_{ij} , ts_{RM_i} **)**

Process RM_i *runs the following :*

$estimate_{RMi}$ μ S_{ij} $\perp$ denotes RM's estimate of the decision value

$state_{RMi}$ μ undecided

r_{RM_i} μ 0 $\perp$ r_{RMi} denotes the current round number

ts_{RM_i} μ 0 $\perp$ the round in which $estimate_{RMi}$ was last updated, initially 0

cobegin

*‖ **Task 1 :** $\perp$ **Rotate through coordinator until decision is reached***

while *$state_{RMi}$ = undecided*

r_{RMi} μ r_{RMi} +1

c_{RM_i} μ (r_{RMi} mod K)+1 $\perp$ K : is the RM's number (from source until destination)

$\perp$ c_{RM_i} is the current coordinator

Phase 1 :** $\perp$ **All resources managers K send** $estimate_{RMi}$ **to the current coordinator

RM_i sends (RM_i , r_{RMi} , $estimate_{RMi}$, ts_{RM_i}) to c_{RM_i}

Phase 2 :** $\perp$ **The current coordinator gather K – n estimates and proposes a new estimate

if *RM_i = c_{RM_i}* ***then***

wait until *[for K-n resource manager RM_j : received (RM_j , r_{RMi} , $estimate_{RMj}$, ts_{RM_j})* ***from*** *RM_j]*

$msgs_{RM_i}(r_{RM_i})$ μ $\perp$ (RM_j , r_{RMi} , $estimate_{RMj}$, ts_{RM_j})|

RM_i received(RM_j , r_{RMi} , $estimate_{RMj}$, ts_{RM_j}) ***from*** *RM_j*

tμ largest ts_{RM_j} such that (RM_j , r_{RMi} , $estimate_{RMj}$, ts_{RM_j}) $msgs_{RM_i}(r_{RM_i})$

$estimate_{RMi}$ μ select one $estimate_{RMj}$ such that (RM_j , r_{RMi} , $estimate_{RMj}$, t) $msgs_{RM_i}(r_{RM_i})$

RM_i sends (RM_i , r_{RMi} , $estimate_{RMi}$) to all resources managers

Phase 3:** $\perp$ **All resources managers wait for the new estimate proposed by the current coordinator

wait until *[received (c_{RM_i} , r_{RMi} , $estimate_{RMc}$)* ***from*** *c_{RM_i}]*

if *[received ((c_{RM_i} , r_{RMi} , $estimate_{RMc}$) from c_{RM_i}]* ***then***

$estimate_{RMi}$ μ $estimate_{RMc}$

ts_{RM_i} μ r_{RMi}

RM_i sends (RM_i , r_{RMi} ,ack) to c_{RM_i} $\perp$ RM_i suspects that c_{RM_i} can vote

else *RM_i sends (RM_i , r_{RMi} ,nack) to c_{RM_i} c_{RM_i} $\perp$ RM_i suspects that c_{RM_i} cannot vote*

Phase 4:** $\perp$ **The current coordinator wait for K-n replies. If these replies indicate that k-n resources managers adopted its estimate, the coordinator sends a request to decide

if *RM_i = c_{RM_i} then*

wait until *[for K-n resources managers RM_j : received (RM_j , r_{RMi} ,ack) or (RM_j , r_{RMi} ,nack)]* *if [for K-n resources managers RM_j : received $(RM_j, r_{RMi}$,ack)then RM_i recasts $((RM_i, r_{RMi},$ $estimate_{RMi}$,decide)]*

║ ***Task 2*** *:* $\perp$ ***When*** RM_i ***receives a decide message, it decides***

when RM_i *redelivers* $(RM_j, r_{RMj}, estimate_{RMj}$ *,decide) for some* RM_j

decide *on* $estimate_{RMj}$

$state_{RMi}$ μ *decided*

end.

At request time, the coordinator gets the user profile, the destination (IP address) and makes a local admission control within its domain. If resources are available, the RM establishes a path within its domain and then sends a Resources Allocation Vote (RAV) request to other domains to cross until the destination is reached.

The algorithm takes place within three asynchronous periods. Each period can be carried out by asynchronous rounds:

- First period: each RM can estimate a range of SLA levels (*w%*).
- Second period: a range of SLA is decided. This range will be locked by the coordinator. No other ranges of decision will be possible.
- The third period: the RM decides of the locked range.

5.1. Successive votes - alternatives paths - consensus convergence

In order to satisfy the media and to look for alternate paths, the consensus coordinator can then either:

- Recover the resources within the failing domains. It will thus try to negotiate the routers in order to decrease the resources allocated for other services;
- Negotiate resources from close domains.

During consensus elaboration, round robin coordinator decides of service SLA levels (*w%*) for each new media and then pre reserve them. Dynamic and temporary paths are thus allocated for each one.

Moreover, during consensus elaboration, round robin coordinator determines failing RM number and localization.

5.2. Results and experimentations

We developed simulations using the following system:
Application : video-telephony session;
K: numbers domains concerned : 10;
S_{ij} :series of the votes - successive votes on disjoined sets of voters;
δ :time of execution - end-to-end time on UMTS/DiffServ) < 150 ms.

During consensus, the coordinator decides of services levels SLA (*w%*) for the media for each session according to available resources. Dynamic temporary ways are thus allocated for each one. Moreover, during consensus execution, the coordinator fixes the number and the localization of failing RM.

During a requested session (see Figures 3 and 4), *Losange S* algorithm runs in order to satisfy it through contractual SLA (SLAmin, SLAavg or SLAmax). During a succession of votes for resource reservation, the algorithm converges towards a decision for each concluded contract SLA. Convergence allows a pre-reservation for those which voted resource allocation and, for domains where resources are unavailable, to consider alternative domains.

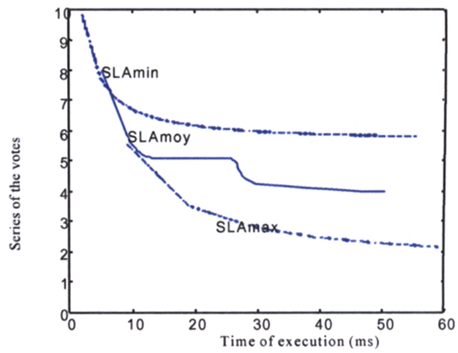

Figure 3. *Losange S* ; Convergence of consensus towards K fields concerned by the application - routing for SLAmin, SLAmoy, SLAmax

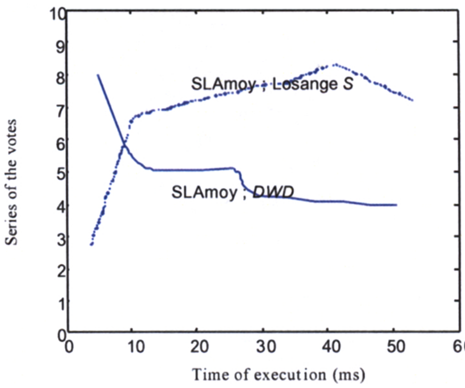

Figure 4. Comparison *Losange S* and *DWD*

In Figure 3, we notice that for contracts SLA, the number of the voters which vote to indicate resources availabilities, for SLAmin for example, is almost equal to 6 domains for all domains which take part to the application (10 domains). In this case we can set up a way composed by these 6 domains which can thereafter provide a path to transfer the data of this application.

Figure 4 shows the advantages to use *Losange S* compared to *DWD* considering that the first algorithm can be converged towards a decision required by this application whereas the second cannot be converged after a sum of the successive votes. In this case we cannot neither recognise the available domains, nor the unavailable domains.

Losange S algorithm is thus necessary to solve the problem of resources unavailability within some concerned domains. It uses a succession of successive votes on disjoined sets voters. It permits voting on several stages until convergence with pre-reservation for those which voted resource allocation and consequently to consider alternative domains in the case of the unavailable domains.

6. CONCLUSION AND PERSPECTIVES

Our work is aimed at setting up the infrastructure which helps the setting up of resource reservation policies in a consensual way by taking into account the constraints of each domain and the profiles and preferences of the users. Eventually, operator policies and strategies for resource allocation will be applied to optimize their network use while satisfying the users.

Our work makes it possible to put in place the infrastructure which contribute to establishing policies of resource pre-reservation in a consensual manner taking into account the constraints of each domain, and the profiles of the users.

The study and the setting up of the routing algorithm to ensure the connection of the ad hoc networks, is a rising challenge. Such environment is dynamic, and evolves over the course of time, with the topology of the network frequently changing. Thus, it seems significant to us to use our consensual approach. It is based on an asynchronous algorithm real time which is based on the votes for concluded contract SLA. It could be used to solve the problems encountered in the design of protocol of routing such as:

- The minimization of the network load: resource optimisation of network resources contains two other problems which are the avoidance of the routing loops, and the prevention of the concentration of the traffic around some nodes or bond.
- To ensure an optimal routing : the strategy of routing must create optimal ways and be able to take into account different metric from cost (band-width, a number of the bonds, resources of the network, time from end to end, etc.).
- Latency time : the quality of the way and latency times must increase if the connectivity of the network increases.

7. REFERENCES

[1] Wroclawski, J, "The use of RSVP with IETF Integrated Services," RFC2210, Sept. 1997.

[2] Braden, R and al., "Resource reservation protocol (RSVP)," version 1 Functional Specification, RFC 2205, Sept. 1997.

[3] Blake, S and, "An architecture for differentiated service," RFC 2475, Dece. 1998.

[4] Ahmed, T and al., "QoS dynamic management of video flow in IP DiffServ architector," in Proc. GRES'03, 2003, pp.53-65.

[5] Chang. F and al., "Automatic configuration and run-time adaptation of distributed application," Ninth IEEE Intl. Symposium on High Performance Distributed Computing (HPDC), Aug. 2000, pp.11-20.

[6] Chahdra. P and al., "Network Support for application-oriented QoS," Proc. Of Sixth International Workshop on Quality of Service (IWQoS'98), pp. 187-195.

[7] Ahmed, T and al., "Gestion dynamique de la QoS de flux sur l'architecture IP DiffServ," GRES Febr. 2003, Fortaleza-CE-Brazil.

[8] Misic, J and al., "Quality of Service Management for Wireless Networks with Heterogeneous Traffic," IEEE 1998.

[9] Misic, J and al., "Event based resource estimation in admission control for wireless networks with heterogeneous traffic, " In Mobile Computing and Communications Review, vol.1, no.4, 1997.

[10] Ben Rejeb, S, Choukair, Z and Tabbane, S, "Resource allocation and QoS management for cellular multimedia network, " in Proc. GRES'03, Brazil, Febr. 2003, pp.3-13.

[11] Ben Rejeb, S, Choukair, Z and Tabbane, S, "Modeling end-to-end QoS management and resource reservation for multimedia mobile radio network," in Proc. AICCSA'03, Jul. 2003.

[12] Ben Rejeb, S, Tabbane, S and Choukair, Z, "Mobility model for multimedia mobile radio network, " in Proc.MMNS'03, LNCS 2839 Sept. 2003, pp. 183-195.

[13] Hurfin, M and. Raynal, M, "A simple and fast asynchronous consensus protocol based on a weak failure detector," Distributed Computing, 1999, pp.209-223.

[14] Chandra, T.D, "Unreliable failure detectors for reliable distributed systems," Journal of the ACM, vol.43, n° 2, Mar. 1996, pp.225-267.

[15] Raynal, M, "Real time dependable decision in timed asynchronous distributed systems," Institue of Electical and Electronics Engineer 1997.