MODEL FOR QUALITY OF SERVICE MENAGAMENT ON INTERNET BASED ON RESOURCE ALLOCATION

ZORAN PERISIC[(1)], ZORAN BOJKOVIC[(2)]
[(1)]Yugoslav Army,
Kneza Milosa 33, 11000 Belgrade,
YUGOSLAVIA,

[(2)]Faculty of Transport and Traffic Engineering,
University of Belgrade,
Vojvode Stepe 305, 11000 Belgrade,
YUGOSLAVIA,

Abstract: In order to achieve the users acceptance requirements and to satisfy some constrains on the multimedia systems, we need a QoS research. It includes modeling, architecture, traffic management protocols, etc. In these paper topics such as communication model and resource management architecture are analyzed. This allows to specify QoS parameters and therefore to control the resource allocation according to the quality desired by application. The adaptation of multimedia QoS to the dynamic operating system and network conditions is important as it allows a better use of resources.

Key-Words: -model, QoS, Internet, resource allocation.

1. Introduction

Model is based on dynamic and adaptive application framework with following characteristics:

♦ In system we have certain capacity of resources R and meny applications which sher resources.

♦ Every application generate new requests for resources.

♦ An application may require access to multiple resource types such as CPU, memory, bandwith etc.

♦ An application may need to satisfy many requirements: data quality, dependability, timelines, security, etc.

♦ An application requires a certain minimum resources allocation to be executed. It may also improve its performance with larger resource allocations. This improvement in performance is measured by utility function.

♦ First satisfies requirements of applications with largest priority.

♦ If request for resources is satisfied, performs resources allocation, vice versa request remain in undecided state whence execute (See figure 1.).

♦ A satisfied request signifies broadcast which is complete, whilst dissatisfied requests signifies broadcasts which is in proceedings or in queue.

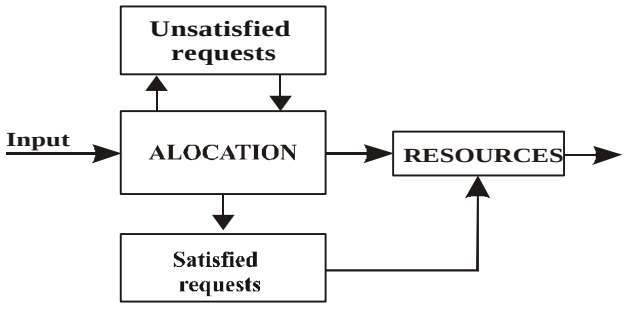

Fig. 1 Resource allocation in model

Model is based on fact that resources can be allocated to individual applications with goal of broadcasting optimization.

2. Definition of model

In this model following aspects of quality are considered: security, timelines, reliability and data quality like parameters of QoS.

It is important to emphasis that we consider system in which many applications with they requirements content for resources with the following characteristics:

♦ Each application may have minimum and/or maximum demands that relates to each parameter.

♦ Each resource allocation improves functionality of the system and application.

♦ System resources are limited so that the maximal demands of all applications often cannot be satisfied simultaneously.

With this model, decision about resource allocation will be made on a level of each application so that a global system goal will be maximized. Primary attention will be focus on resource allocation management for an audio flow in one node.

Parameters of QoS in system are end-to-end delay, which represents timelines and audio sample rate, which represents data quality.

Processing rate and audio sample rate can be changed independently of one another, and an increase in either leads to increases in utility of the system. Improvements in end-to-end delay from 350 ms to 70ms generally mast be perceived as much higher than improvements from 70 ms to 14 ms, i.e. [1].

Model will be defined in the following way. System consists of n applications $(r_1, r_{2,}..., r_n)$, $n \geq 1$, and m resources $(R_1, R_2,..., R_m)$, $m \geq 1$. Each resource R_j has a finite capacity and can be shared, either temporally or spatially. CPU and network bandwidth, for example would be time-shared resources, while memory would be a spatially shared resource

Let the portion of resource R_j allocated to application r_i be denoted as $R_{i,j}$. We enforce that $\sum_{i=1}^{n} R_{i,j} \leq R_j$. Two issues need to be noted in the context of real-time systems in particular:

♦ Utilization: The resource allocation to an application will be in terms of the utilization of a resource. This means that resources will be maximal utilized, bat without consequence on broadcasting in sense of delay [2].

♦ Schedulability: The constraint $\sum_{i=1}^{n} R_{i,j} \leq R_j$ implies that a resource can be fully consumed. That constraint is not every time fully correct but we assume like that because of simplicity of problem. A different maximal resource constraint beyond the scope of this paper must be used to support fixed-priority schemes.

Now we introduce some notions:

♦ The application utility, U_i, of an application r_i is defined to be the value that is accrued by the system when r_i is allocated $R' = (R_{i,1}, R_{i,j},..., R_{i,m})$. In other words, $U_i = U_i(R,)$, when U_i is denoted like utility function of r_i. This utility function defines a surface along which the application can operate based on the resources allocated to it.

Each application r_i has a relative importance specified by a weight w_i, $1 \leq i \leq n$.

♦ Total system utility $U(R^1, R^2,..., R^n)$ is defined to be the sum of the weighted application utility of the applications, $U(R^1, R^2,..., R^n) = \sum_{i=1}^{n} w_i U_i(R^i)$.

♦ Each application r_i needs to satisfy requirements along d QoS parameters ($P_1, P_2,..., P_d$), $d \geq 1$.

♦ An application r_i, has minimal resource requirements on QoS parameter P_k. These minimal requirements are denoted by $R_i^{\min_k} = \{R_{i,1}^{\min_k}, R_{i,2}^{\min_k},..., R_{i,m}^{\min_k}\}$ where $R_{i,j}^{\min_k} \geq 0$, $0 \leq j \leq m$.

♦ For an application r_i, is said that it is feasible if it is allocated a minimum set of resources on every QoS dimension. We denote the total minimum requirements by $R_i^{\min} = \{R_{i,1}^{\min}, R_{i,2}^{\min},..., R_{i,m}^{\min}\}$ where $R_{ij}^{\min} = \sum_{k=1}^{d} R_{i,j}^{\min_k}$, $1 \leq j \leq m$.

We assume that $m = 1$, i.e. only a single resource is being allocated.

2.1 Preliminary assumptions

Because of functionality of model we make following assumptions:

1. The applications are independent of one another.
2. The available system resources are sufficient to meet the minimal resource requirements of each application on all QoS parameters, $R_i^{\min}, 1 \leq i \leq n.$

3. Each application r_i, has a weight w_i denoting its relative importance.

We make the following observations concerning these assumptions. First, if assumption 1. does not hold, then resource allocation model still apply.

Second, if assumption 2. does not hold, then the minimal resource requirements cannot be met. If these requirements are not met, then some of the applications must be dropped. We can use a variety of techniques to determine which of the applications should be dropped.

Third, in view of third assumption we can now define a weight utility function for an application as $w_i * U_i$ and then solve the resource allocation problem for those weighted utility funcions.

Should be noted that U_i is not necessarily equal to $\sum_{j=1}^{m} U_{i,j}$. In other words, the utility obtained by an application r_i from a resource R_j may not be additive with respect to its utility from another resource. This is because the application may need two or more resources simultaneously to achieve a certain utility. For example, an audio-video application may need the CPU resources and bandwidth resources in order to satisfy even a minimal QoS requirement.

The goal of this model is to make resource allocations to each application such that the total system utility is maximized under the constraint that every application is feasible with respect to each QoS parameter. First, we need to determine$\{ R_{i,j}$, $1 \le i \le n, 1 \le j \le m \}$ such that $R_{ij} \ge \sum_{k=1}^{d} R_{i,j}^{\min_k}$ and U is maximal among all such possible allocations.

3. Resource allocation in the model

In this section, we derive some basic properties of the model defined in the previous section. We start with simple case of making allocations decisions where there is only a single resource type and a single QoS parameter. Then we extend this model to support multiple QoS parameters. In each case, we state a property that needs to be satisfied for maximizing the total system utility and/or present an algorithm which can find the optimal (or near-optimal) allocation.

3.1 A single resource and a single QoS parameter ($m=1$ and $d=1$)

In this case, we have $U_i \le U_i(R)$, $1 \le i \le n$, where R is the among of resource allocated to r_i. The minimum resource allocation needed to satisfy r_i is $R_i^{\min}$.

To illustrate this approach, we make the further assumption that utility function $U_i = U(R)$ are twice continuously differentiable and concave, that is $\frac{d^2 U_i}{dR^2} = U_i^{II} \le 0$ for $R > R_i^{\min}$. By convection, we assume $U_i(R) = 0$ for $0 \le R \le R_i^{\min}$.

It is very convenient to transform the resource allocation problem. Since we assume that all minimal application resource requests can be met, we can focus on the allocation of the excess resources available. Consequently, we can, without loss of generality, assume that $R_i^{\min} = 0$, $\forall i = 1$ to n and reduce the quantity of available resources by that amount. In our subsequent analysis, we assume that this transformation has been made and require only that $R_i \ge 0$ and $\sum_{i=1}^{n} R_i = R$, where R is the remaining quantity of resources left to allocate.

The goal is to determine the values $R_1, R_2, ..., R_n$ such that the total system utility, $\sum_{i=1}^{n} U_i(R_i)$, is maximized subject to the constraint $\sum_{i=1}^{n} R_i \le R_i$.

It should be noted that it is possible that all applications except one can receive zero resource allocations, and this one application consumes all the available resource quantity since the slope of its utility function is the highest.

3.2 A single resource and multiple QoS parameters ($m=1$ end $d>1$)

An application can have multiple QoS parameters ($d>1$)**.** For example, audio-video application has two QoS parameters, audio data quality (which increases with audio sampling rate) and end-to-end delay (which decreases with increases in processing rate) [3]. The resource allocation for systems with multiple quality dimensions depends upon the nature of the relationship between the dimensions themselves. In this section, we classify the relationship between QoS dimensions, discus their effects and study the resource allocation problem under various conditions.

3.3 Relationships between QoS parameters

The inter-relationship between QoS dimensions directly impacts the nature of the utility functions. We consider two kinds of relationship among QoS parameters.

Independent parameters: Two QoS parameters, P_a and P_b, are independent of one another if a quality increase along P_a, (P_b) does not increase the resource demands to achieve the quality level previously achieved along P_a, (P_b).

Dependent parameters: A QoS parameter P_a are dependent on another parameter P_b, if a change along the parameter P_b will increase the resource demands to achieve the quality level previously achieved along P_b. In the audio-video application if audio sampling rate is increased, the data volume increases and the CPU time needed to process the data increases.

Two QoS parameters P_a and P_b can both be depend on third parameter P_c. For example, if video quality is improved by increasing the size of the image, both processing capacity and network bandwidth demands would increase.

Suppose the CPU resource has to be allocated among 10 applications with specify utility curves. First, all 10 application will be allocated their minimum resource requirements. Next, additional resource allocations will be made only to the application with the highest utility slope. If any CPU cycles remain after that application reaches its maximum requirements, only then would they be allocated to the application with the next higher slope

4. Conclusion

Resource allocation model we presented are based on QoS that allows the utility derived from a system to be maximized by making resource allocations such that the different needs of concurrently running applications are satisfied. Each application has minimal resource requirements, but can adapt its behavior if given more resources and provide additional utility [4]. Each application also needs to satisfy QoS metrics along multiple dimensions such as timeliness, cryptography security, reliable packet delivery and data quality.

In this model emphasis is on the system with one node and that should be expand on distributed systems.

References

[1] T. Abdelzaher, “An Automated Profiling Subsystem for QoS-Aware Services”, *Second IEEE Real-Time Technology and Applications Symposiu*m, Nov. 2000.

[2] Foster, A. Roy, and V. Sander, “A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation”, *8th International Workshop on Quality of Servic*e, May 2000.

[3] F. Kon, R. Campbell, and K. Nahrstedt, “Using Dynamic Configuration to Manage a Scalable Multimedia Distributed System”, *Computer Communication Journal, Special Issue on QoS-Sensitive Distributed Network Systems and Application*s, pp. 156-161, Jan. 2000.

[4] D. Hull, A. Shankar, K. Nahrstedt, and J. W.-S. Liu, “An End-to-End QoS Model and management Architecture”, *IEEE Workshop on Middleware for Distributed Real-time Systems and Service*s, pp. 82–89, December 1997.