Specifying Policies for Service Negotiations of Response Time

T.K. Kim[1], O.H. Byeon[2], K.J. Chun[3], and T.M. Chung[1]

[1]Internet Management Technology Laboratory,
School of Information and Communication Engineering,
Sungkyunkwan University,
Chunchun-dong 300, Jangan-gu, Suwon, Kyunggi-do,
Republic of Korea
tkkim@rtlab.skku.ac.kr, tmchung@ece.skku.ac.kr
[2]Korea Institute of Science and Technology Information
ohbyeon@kisti.re.kr
[3]Sangmyung University
chunkj@smu.ac.kr

Abstract. The use of services in large-scale and cross-organizational environments requires the negotiation of agreements that define these services. This paper proposes to specify a negotiation policy for response time of distributed network. The analysis of distributed networks has become more and more important with the evolution of distributed network technologies resulting in increasing capacities. To monitor and manage service in distributed network, we must identify the relationships between network/application performance and QoS parameters. Therefore, we provide a statistical analysis on mapping user level response time to application and network level parameters by using some queueing models and suggest the negotiation of policy specification for response time of distributed network. Hence, the use of guaranteed services becomes feasible.

1 Introduction

The evolution of communication technologies results in continuously increasing capacities and higher concentration of traffic on relatively fewer network elements. Consequently, the failures of these elements can influence an increasing number of customers and can deteriorate the quality of service provided by the operator. The performance of distributed network is very important to network users for running their applications and to network manager for managing the distributed network. In general, the users do not know how to efficiently map their performance requirements to a complex QoS metric. Moreover, many of the sophisticated QoS and pricing mechanisms are complex to implement and therefore infeasible. As customers have begun to demand higher level of Quality of Service (as opposed to the best effort service) from the service providers, service level agreements (SLAs) between customers and service providers have become the norm. A service level agreement [5] is an agreement regarding the guarantees of a service. It defines mutual understandings and expectations of a service between the service provider and service consumers. These SLAs specify the quality of service and pricing information [7].

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 250–257, 2004.

Therefore, in distributed network, “best effort service” is no longer sufficient for guaranteeing QoS. Thus it is required to satisfy quality of service (QoS) in distributed network that a QoS specification at the application level permits the selection of appropriate network level parameters [3]. To allow the provision of a certain QoS, the parameters of one layer have to be mapped to those of other layers and of system resources [4]. For instance, typical QoS metrics include committed bandwidth, transit delay, packet loss rate and availability.

In this paper, we focus on network latency to represent the mapping of applications performance parameters to network performance parameters and negotiation of agreements that define this service. Negotiations are mechanisms that increase the flexibility of possible service contracts. In the context of dynamically setting up service relationships, it is important to use an efficient decision-making process that reduces cost and time of the setup. Decision-making is involved in deciding the acceptance of an offer and in the selection of an outgoing offer among multiple candidates [1].

In section2, related works are describes. Section 3 presents the response time by mapping the application level parameters to network level parameters. In section 4 we introduce the model of decision-making. Section 5 summarizes our work and identifies future research directions.

2 Related Works and Negotiation Issues

2.1 Related Works

To support the negotiation of service and QoS of application, many projects are processed in this field. Liu et al. [2] presented a formal statistical methodology for the mapping application level SLA to network level performance. They took the response time as the application level SLA and link bandwidth and router throughput/utilization at the network layer for their preliminary analysis and presented a function which directly links the response time at the application level to the network parameters and does not address the efficiency of a formal statistical methodology using simulation or real execution. Some other projects have presented QoS mapping in multimedia networks [8][9].The negotiation server by Su et el. uses rules to describe how to relax constraints defining acceptable offers in the course of the negotiation[13]. The complexity of utility functions and contract implementation plans is addressed by Boutilier et al. [6]. This approach is used for collaborative resource allocation within an organization and does not address negotiation across organizational boundaries.

2.2 Negotiation Issues

Negotiations are mechanisms that increase the flexibility of possible service contracts and negotiations are used as comprising all exchanges of messages, such as offers and acceptance messages, between two or more parties intended to reach an agreement. A common way of analyzing negotiations is differentiating the negotiation protocol, comprising the rules of the encounter, the negotiation object, which is the set of

negotiated attributes, and the decision making model. A number of simple negotiation protocols are used for match making reservations such as SNAP (Service Negotiation and Acquisition Protocol) has been proposed for resource reservation and use in the context of the Grid. A common problem in negotiations is the ontology problem of electronic negotiations [10]. It deals with the common understanding of the issues among negotiating parties. One approach of solving the ontology problem is the use of templates. Templates are partially completed contracts whose attributes are filled out in the course of negotiation [1].

3 Mapping Response Time to Application and Network Parameters

We can know the response time using the information of network latency, system latency, and software component latency. Network latency is composed of propagation delay, transmission delay, and queueing delay; system latency is composed of disk I/O, and CPU processing delay; software component latency is composed of server and database transaction delays.

$$Response\_time = Network\_latency + System\_latency + Software\_component\_latency$$

We will focus on the network latency, and assume system latency and software components latency are known. We can think of latency as having three components [11]. First, there is the speed-of-light propagation delay. This delay occurs because nothing can travel faster than the speed of light. Second, there is the amount of time it takes to transmit a unit of data. This is a function of the network bandwidth and the size of the packet in which the data is carried. Third, there may be queuing delays inside the network, since packet switches need to store packets for some time before forwarding them on an outbound link. So, we define the network latency as:

$$Network\_latency = propagation\_delay + transmission\_delay + transmission\_delay + queuing\_delay$$

We can map the network latency of user level to application level and network level elements.

- user layer: response time - the elapsed time from the user requests the service to the user accepts the results of application.
- application layer: network latency, system latency, software component latency
- network layer: propagation delay, transmission delay, and queueing delay. Network layer can be modeled as end-to-end latency partitioned into a speed-of-light propagation delay, a transmission delay based on the packet volume sent and the bandwidth, and queueing delay including host and router.
- network layer elements: distance, bandwidth, utilization, throughput, packet size, arrival rate, and number of servers.

Propagation delay is related with distance and transmission delay is related with bandwidth and packet size. And queuing delay is related with bandwidth, packet size, arrival rate, utilization, throughput, and number of servers. Figure 1 shows the relations among the user level, application level, and network level parameters.

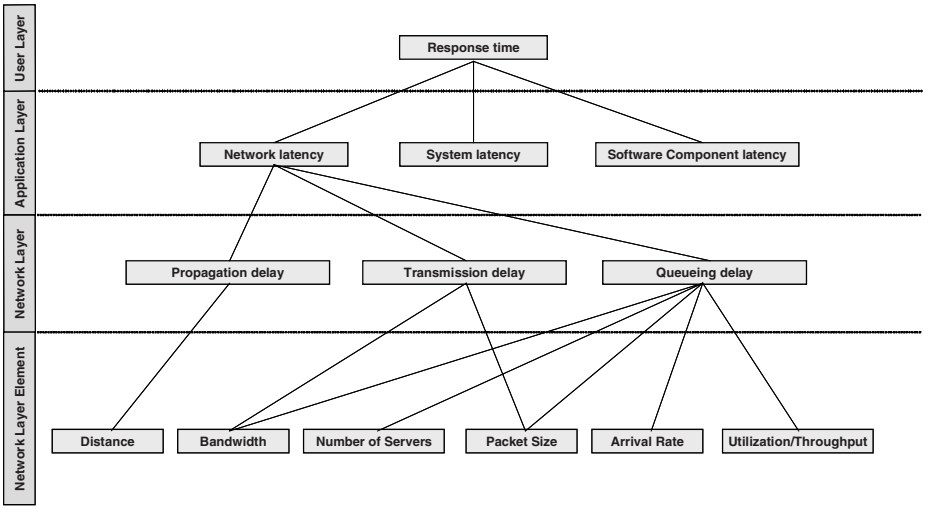

Fig. 1. Mapping of latency elements

We can calculate the response time using network elements mapped with application elements. So, we can statistically analyze the network element's contribution to the whole response time for an application running in a distributed computing environment. We assume the system to be a markovian system, which means the distribution of the interarrival times and the distribution of the service times are exponential distributions that exhibit markov property.

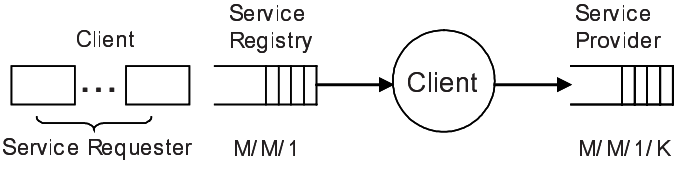

Fig. 2. The modeled distributed Network

We also modeled this system as the M/M/1 Queue for the process of getting the useful information of service; M/M/1/K for the processing of service provider.

Thus we can write the steady state probabilities as follows:

Propagation delay: $\sum_{j=1}^{j} \frac{D}{2.3\times10^{8}}$

Transmission delay: $\sum_{k=1}^{k} \frac{\overline{M}}{B}$

Queueing delay [2]: M/M/1, M/M/1/K - $\sum_{l=1}^{l} \frac{\overline{M}}{B-\lambda\overline{M}}$

$$Network\_latency = \sum_{j=1}^{j} \frac{D}{2.3\times10^{8}} + \sum_{k=1}^{k} \frac{\overline{M}}{B} + \sum_{l=1}^{l} \frac{\overline{M}}{B - \lambda\overline{M}} \quad (1)$$

where D is distance, $\overline{M}$ is the mean size of the packet, B is the bandwidth at which the packet is transmitted, and λ is the arrival rate of the client request.

We calculated the queueing delay and network latency. The parameters used in this calculation are like this: bandwidth: 10Mbps; packet size: 1500byte; distance: 50km; arrival rate: 42; service rate: 874. The arrival rate and service rate were calculated using $\mu = \frac{1}{T_S} = \frac{B}{\overline{M}}$ and M/M/1 system queueing delay $\frac{\overline{M}}{B - \lambda\overline{M}}$.

The propagation and transmission delay were calculated like these:

- Propagation delay: 0.000217391
- Transmission delay: 0.0011444

Then, we calculated the value of queueing delay using the equation of (1). Also, to check the value of calculation, we used the NS-2 (Network Simulator). NS-2 is an open-source simulation tool that runs on Linux [12]. We made Tcl script of each queueing model (M/M/1, M/M/1/K) and simulated the queueing delay. The Linux server used in this simulation has dual CPU of 1G and 256MB RAM. We repeated this simulation 20 times to calculate the mean queueing delay and network latency time of simulation.

Table 1. The value of calculating, simulation, and error of the network latency

	Propagation delay	Transmission Delay	M/M/1	M/M/1/K (K=50)	Network latency
Calculation	0.000217391	0.0011444	0.001202193	0.001202	0.003766
Simulation			0.001202	0.001197	0.003761
Error			0.016%	0.42%	0.133%

To compare the network latency of calculation value and simulation value, we calculated the average error for the above type of measurements as:

$$Error = average(\ abs(network\_latency\_function - queueing\_simulation)\ /\ queueing\_simulation) \times 100$$

The value of network latency calculation of (1) was relative similar to the value of queueing simulation and the error was relatively low. And distance, bandwidth, packet size, and the number of host are related to the network latency in distributed network.

4 Decision-Making Framework

The model of decision-making is designed as an object-oriented framework. The framework assigns evaluation function to check the response time, and specifies the object that can be accessed for rule-based reasoning.

Rules are expressed in a high-level language specified by an XML schema. This helps a business domain expert to specify negotiation strategies without having to deal with the programmatic implementation of the decision making system. The basic structure is a bilateral message exchange and follow-up messages are of the types accept, reject, offer, withdraw, or terminate. Accept leads to a contract based on the other's last offer, reject to the rejection of the last offer. Offer indicates that a filled template is sent as proposed contract, withdraw annuls the last offer, and terminate ends the entire negotiation process immediately [1]. Figure 3 exemplifies response time negotiation system architecture.

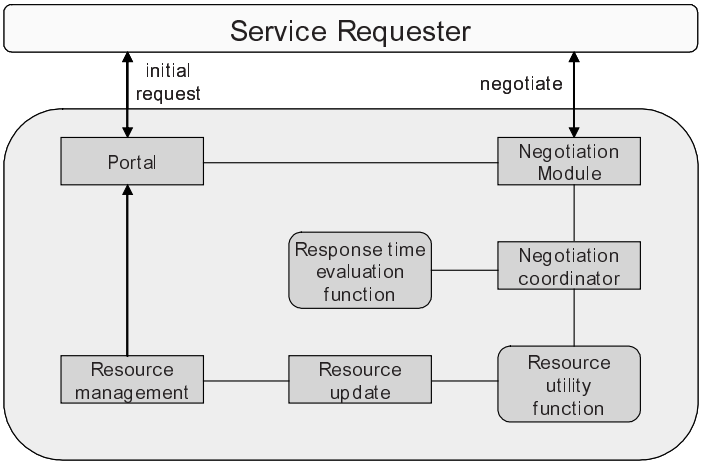

Fig. 3. Decision-maker components

A negotiation is initiated through the portal. The portal sends information about new interaction to the negotiation module and endows it with information on the service requester. After receiving new interaction, negotiation module takes control of the interaction with the service requester.

- Negotiation module: The negotiation module performs some administrative tasks such as checking the validity of the incoming message before proceeding to more sophisticated message handling. Upon reception of a termination or acceptance the procedure is straight forward: cleaning up all negotiation dependent objects and possibly canceling resource reservations, or passing the final contract on to for deployment. Otherwise, the negotiation module processes its rule corpus for producing a response message to send or for deciding to wait [1].
- Negotiation coordinator: The negotiation coordinator is designed to coordinate multiple negotiation modules and can measure the real-time response time of service using response time evaluation function. Resource utility function checks the status of usage of system and network resources. This coordinates the resource reservation to keep the contract of negotiation. If response time is more than that of contract, negotiation coordinator requests to assign more system resources to guarantee the contract.
- Resource update: resource update establishes the interface to resource management and is able to handle reservation and information requests. Resource management controls the resources of systems.

Table 2. Example of an offer sequence during service negotiation

Sender		…	Negotiation module	Service requester	Negotiation module	…
Message No.			3	4	5	
Response time			≥4	≤4	4	
Assignable resources	bandwidth		40-60%	80%	60%	
	cpu		40-50%	48%	48%	
	memory		50-55%	57%	57%	
Available resources	Bandwidth		75%	72%	73%	
	Cpu		60%	56%	55%	
	memory		68%	68%	71%	

Table 2 gives a snapshot of the offers exchange between service requester and negotiation module.

Response time is given in second, assignable resources are required to guarantee the response time of service. Available resources mean the status of system resources to be used.

The creation of negotiation module's number 5 goes as follows: negotiation module receives message 4, approves it as valid offer, and the starts processing its rule corpus containing a single rule set. Negotiation module invokes LEVEL_OF_DIFFERENT, which computes the difference of offers 3 and 4. After calculation of response time using resource utility function, negotiation module offers 4 seconds as message 5. The last offer makes it likely to be acceptable by service requester. The mapping information of network latency helps to calculate the response time and assign required system resources.

5 Conclusion and Future Works

The use of services in large-scale and cross-organizational environments requires the negotiation of agreements that define these services. Today, negotiation is complex and difficult to implement. In this paper, we provide a statistical analysis on mapping user level response time to application and network level parameters by using some queueing models and suggest the negotiation of policy specification for response time of distributed network. Using this approach, decision-making framework allows the specification of sophisticated negotiation behavior for response time in a manageable way. Also, we suggested a network latency function of (1) to calculate the network latency and showed the validity of the function by comparing the results of simulation using NS-2.

Numerous challenges still remain in this area. There are other user level parameters like availability, reliability, etc., that we haven't pursued in this paper. Future research will focus on presenting the user level parameters of SLA as numerical formula and extend negotiation framework can specify negotiation for all services in a concise, and easy way.

References

1. H. Gimpel, H. Ludwig, A. Dan and B. Kearney, “PANDA: Specifying Policies for Automated Negotiations of Service Contracts”, ICSOC 2003, Trento, Italy, December 2003.
2. B. Hua Liu, P. Ray, S. Jha, “Mapping Distributed Application SLA to Network QoS Parameters,” IEEE 2003.
3. J.-F. Huard, A. A. Lazar, “On QOS Mapping in Multimedia Networks”, Proceedings of the 21th IEEE International Computer Software and Application Conference (COMPSAC '97), Washington, D.C., USA, August 1997.
4. S. Fischer, R. Keller. “Quality of Service Mapping in Distributed Multimedia Systems”, In Proceedings of the IEEE International Conference on Multimedia Networking (MmNet95), Aizu-Wakamatsu, Japan, September 1995.
5. L. Jin, V. Machiraju, A. Sahai, "Analysis on Service Level Agreement of Web Services," Software Technology Laboratory, June 2002.
6. C. Boutilier, R. Das, J.O. Kephart, G. Tesauro, W.E. Walsh: Cooperative Negotiation in Autonomic Systems using Incremental Utility Elicitation. Proceedings of Nineteenth Conference on Uncertainty in artificial Intelligence (UAI 2003). Acapulco, 2003.
7. M. Singh Dang, R. Garg, R. S. Randhawa, H. Saran, "A SLA Framework for QoS Provisioning and Dynamic Capacity Allocation," Tenth International Workshop on Quality of Service (IWQoS 2002), Miami, May 2002.
8. Luiz A. DaSilva, “QoS Mapping along the Protocol Stack: Discussion and Preliminary Results,” Proceedings of IEEE International Conference on Communications (ICC’00), June 18-22, 2000, New Orleans, LA, vol. 2, pp. 713-717.
9. T. Yamazaki, J. Matsuda, “On QoS Mapping in Adaptive QoS Management for Distributed Multimedia Applications,” Proc. ITCCSCC’99, vol.2, pp.1342-1345, July, 1999.
10. M. Strobel: Engineering electronic negotiations, Kluwer Academic Publishers, New York, 2002.
11. L. Peterson, B. Davie, Computer Networks: A System Approach, Morgan Kaufmann, 2000, second edition.
12. NS network simulator. `http://www-mash.cs.berkeley.edu/ns`.
13. S. Y. W. Su, C. Huang, J. Hammer: A Replicable Web-based Negotiation Server for E-Commerce. Proceedings of the Thirty-Third Hawaii International Conference on System Sciences (HICSS-33). Maui, 2000.