Computing and Informatics, Vol. 23, 2004, 395–405

SERVICE NEGOTIATION MODEL FOR RESPONSE TIME IN DISTRIBUTED NETWORKS

Tae-Kyung Kim, Hyung-Jin Lim, Tai-Myoung Chung

Internet Management Technology Laboratory
School of Information and Communication Engineering
Sungkyunkwan University
Chunchun-dong 300, Jangan-gu, Suwon, Kyunggi-do
Republic of Korea
e-mail: {tkkim, hjlim}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr

Manuscript received 19 September 2004
Communicated by Jiří Šafařík

Abstract. The important thing of QoS is that response time of service is transparently suggested to resource management system and network users. This helps to schedule jobs or guarantee the service level agreement. This paper proposes to specify a negotiation policy for response time of distributed network using network latency function. To monitor and manage service response time in distributed network, we identified the relationships between network/application performance and QoS parameters. We also provided the statistical analysis on mapping user level response time to application and network level parameters. To show the validation of the network latency function, we used the NS-2 network simulator and showed the efficiency of that function. Finally we suggested the negotiation of policy for response time of requested service.

Keywords: Response time, specifying policies, negotiation

1 INTRODUCTION

In general, distributed network users can only think the performance attributes in terms of their needs for application performance. As analyzed in [14], the response time is the most significant performance attribute to the network users. Usually, 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].

Policy defined as means of applying QoS. Policy may specify what to do if a conflict with another policy is encountered or if an exception occurs in the exception of a policy rule. Policy also could be used to define mapping between high-level and lower-level policy rules. A policy specification at the application level permits the selection of appropriate network level parameters [3]. Therefore, 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 Section 2, related works are described. Section 3 presents the response time by mapping the user level parameters to application and network level parameters. In Section 4 we introduce the model of specifying policies for response time. Section 5 summarizes our work and identifies future research directions.

2 RELATED WORKS AND NEGOTIATION ISSUES

2.1 Related Works of Service Negotiation and Parameters Mapping

To support the negotiation of service and QoS of application, many projects are processed in this field. In grid networks, the network weather service [15] provides accurate forecasts of dynamically changing performance characteristics from a distributed set of metacomputing resources. In distributed network, 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. But these projects did not suggest the mapping parameters and relations among the different layers for negotiation of service or guaranteeing the QoS.

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]. Gilles Klein et al. use the mobile agent for the dynamic negotiation of SLA between a customer and several Internet Service Providers [16]. The agents able to take options on certain services to decide which provider will finally be chosen.

3 MAPPING PARAMETERS FOR RESPONSE TIME

In general, the response time of requested service can be estimated 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.

We can map the response time of user level to application level and network level elements. Elements of each layer are like these:

- 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, queueing delay, disk I/O, cpu processing, and system transaction 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, throughput, packet size, arrival rate, server, database, cpu and disk.

Figure 1 shows the relations among the user level, application level, and network level parameters for the response time of requested service.

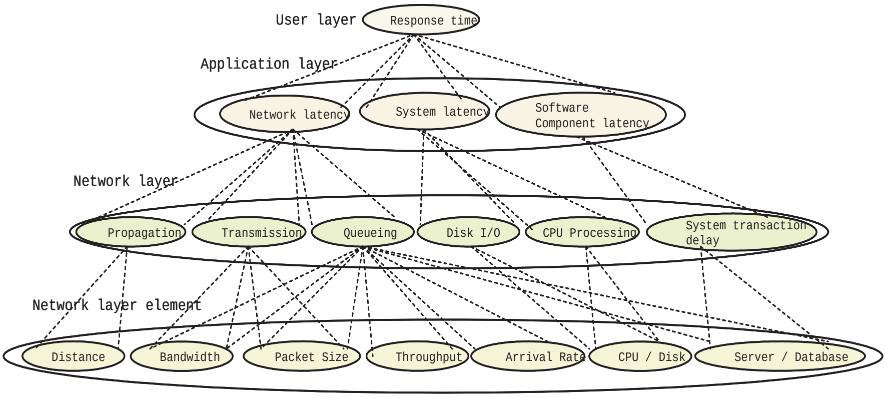

Fig. 1. Mapping parameters of response time

In this paper, we focus on the network latency for the modeling of distributed network and assume that system latency and software components latency is acquired using the real estimation. We can think of network 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.

In the network layer, the propagation delay is related with distance, 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. We can calculate the response time using network elements mapped with application elements.

An example of a distributed application running in the grid network is shown in Figure 2. We can see that there are four flows, namely F1, F2, F3, and F4. We analyze the response time by mapping application level parameters and network level parameters.

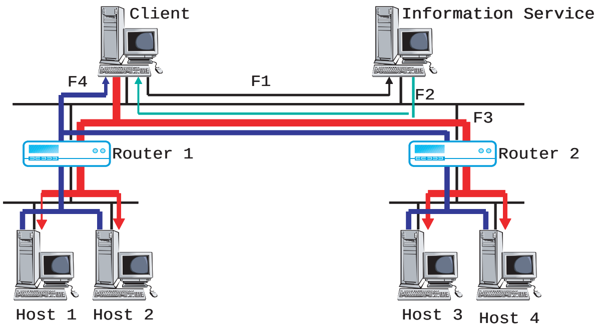

Fig. 2. Distributed application with process flows

F1: client requests information about distributed resources from information service to enable distributed application. Information service provides the information needed to perform dynamic resource discovery and configuration.

F2: Information service evaluates the request (F1) and locates appropriate resources.

F3: once the appropriate computational assets are allocated, the distributed application is processed in each host (Host 1–Host 4).

F4: then the results of distributed application are returned to client.

Therefore, 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.

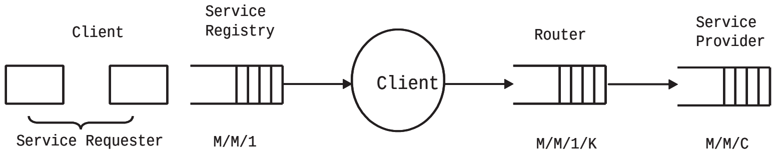

Fig. 3. The modeled distributed network

We also modeled this system as the M/M/1 queue for the process of getting the useful information of service in distributed network; is M/M/1/K for the model of router; M/M/C for the processing of service provider such as distributed computing node. Thus we can write the steady state probabilities as follows:

- Propagation delay: $\sum\limits_{j=1}^{j} \frac{D}{2.3\times 10^8}$
- Transmission delay: $\sum\limits_{k=1}^{k} \frac{\overline{M}}{B}$
- Queueing delay: M/M/1 [2], M/M/1/K - $\sum\limits_{l=1}^{l} \frac{\overline{M}}{B-\lambda\overline{M}}$

  M/M/C - $\sum\limits_{m=1}^{m} (\frac{\overline{M}}{c(B-\lambda\overline{M})} C(c,a) + \frac{\overline{M}}{B})$
- Network latency: $\sum\limits_{j=1}^{j} \frac{D}{2.3\times 10^8} + \sum\limits_{k=1}^{k} \frac{\overline{M}}{B} + 2\sum\limits_{l=1}^{l} \frac{\overline{M}}{B-\lambda\overline{M}} + \sum\limits_{m=1}^{m} (\frac{\overline{M}}{c(B-\lambda\overline{M})} C(c,a) + \frac{\overline{M}}{B})$ (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: 10 Mbps; packet size: 1500 byte; distance: 50 km; arrival rate: 42; service rate: 874. The arrival rate and service rate were calculated using $\mu = \frac{1}{T_S} = \frac{B}{M}$ and M/M/1 system queueing delay$\frac{\overline{M}}{B-\lambda\overline{M}}$.

- Propagation delay: 0.000217391
- Transmission delay: 0.0011444

Then, we calculated the value of queueing delay using the equation of (1). 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, M/M/C) and simulated the queueing delay. The Linux server used in this simulation has dual CPU of 1G and 256 MB RAM. We repeated this simulation 20 times to calculate the mean queueing delay and network latency time of simulation.

	M/M/1	M/M/1/K (router)	Number of host(c)	M/M/c	Network latency
Calculation	0.001202193	0.001202	2	0.001204519	0.004970705
			4	0.001144947	0.004911133
			6	0.001144412	0.004910598
			8	0.001144409	0.004910595
Simulation	0.001202	0.001197	2	0.001143	0.004903791
			4	0.001142	0.004902791
			6	0.001142	0.004902791
			8	0.001141	0.004901791
Error	0.016 %	0.42 %	2	5.38 %	1.36 %
			4	0.26 %	0.17 %
			6	0.21 %	0.16 %
			8	0.3 %	0.18 %

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

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) × 100

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

4 SPECIFYING POLICIES FOR RESPONSE TIME

The model of specifying policies for response time 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 others' 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 4 exemplifies response time negotiation system architecture.

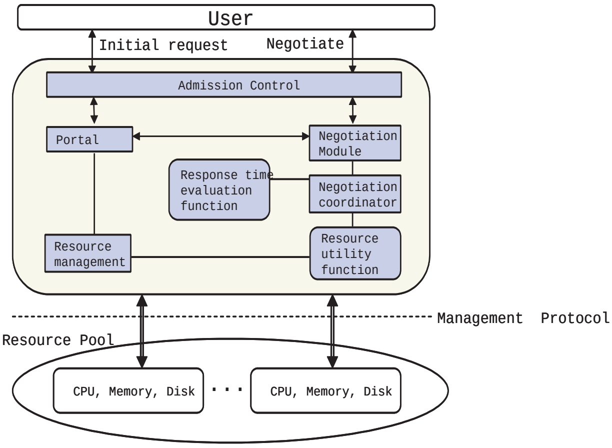

Fig. 4. Model of specifying policies for response time

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.

- Admission Control: In this model, admission control estimates the level of QoS that a new user session will need and whether there are enough network layer elements available to service that session. If network layer elements are available, the session is admitted.
- 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 straightforward: cleaning up all negotiation dependent objects and possibly canceling resource reservations, or passing the final contract on 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 management: Resource management establishes the interface to resource management. Also this module is able to handle reservation, information requests, and controls the resources of systems.

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

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. Example of an offer sequence during service negotiation

Response time is given in seconds, 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 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 calculate the response time and assign required system resources. In the service time, resource utility function checks the status of usage of system and network resources periodically to guarantee the QoS.

5 CONCLUSION AND FUTURE WORKS

The important thing of QoS is that response time of service is transparently suggested to resource management system and network users. To do this, we identified the relationships between network/application performance and QoS parameters. We provided the statistical analysis on mapping user level response time to application and network level parameters and suggested the network latency function which can estimate the network latency using some queueing models. To show the validation of the network latency function, we used the NS-2 and showed the efficiency of that function by comparing the results of simulation using NS-2 and error check function. Finally we suggested the negotiation of policy for response time of requested service. Using this approach, model of specifying policies for response time allows the specification of sophisticated negotiation behavior for response time in a manageable way.

Numerous challenges still remain in this area. There are other user level parameters like availability, reliability, etc., that we have not 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] Gimpel, H.—Ludwig, H.—Dan, A.—Kearney, B.: PANDA – Specifying Policies for Automated Negotiations of Service Contracts. ICSOC 2003, Trento, Italy, December 2003.

[2] Hua Liu, B.—Ray, P.—Jha, S.: Mapping Distributed Application SLA to Network QoS Parameters. IEEE 2003.

[3] Huard, J.-F.—Lazar, A. A.: 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] Fischer, S.—Keller, R.: 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] Jin, L.—Machiraju, V.—Sahai, A.: Analysis on Service Level Agreement of Web Services. Software Technology Laboratory, June 2002.

[6] Boutilier, C.—Das, R.—Kephart, J. O.—Tesauro, G.—Walsh, W. E.: Cooperative Negotiation in Autonomic Systems using Incremental Utility Elicitation. Proceedings of Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI 2003), Acapulco, 2003.

[7] Singh Dang, M.—Garg, R.—Randhawa, R. S.—Saran, H.: A SLA Framework for QoS Provisioning and Dynamic Capacity Allocation. Tenth International Workshop on Quality of Service (IWQoS 2002), Miami, May 2002.

[8] DaSilva, L. A.: 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] Yamazaki, T.—Matsuda, J.: On QoS Mapping in Adaptive QoS Management for Distributed Multimedia Applications. Proc. ITCCSCC'99, Vol. 2, 1999, pp. 1342–1345.

[10] Strobel, M.: Engineering Electronic Negotiations. Kluwer Academic Publishers, New York, 2002.

[11] Peterson, L.—Davie, B.: Computer Networks: A System Approach. Second edition. Morgan Kaufmann, 2000.

[12] NS Network Simulator: `http://www-mash.cs.berkeley.edu/ns`.
`http://www.cai.sk`.

[13] Su, S. Y. W.—Huang, C.—Hammer, J.: A Replicable Web-Based Negotiation Server for E-Commerce. Proceedings of the Thirty-Third Hawaii International Conference on System Sciences (HICSS-33). Maui, 2000.

[14] NSF CISE Grand Challenge in e-Science Workshop Report:
`http://www.evl.uic.edu/activity/NSF/index.html`, Jan 24, 2002.

[15] Wolski, R.—Spring, N. T.—Hayes, J.: The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. Future Generation Computing Systems, 1999.

[16] Klein, G.—Krief, F.: Mobile Agents for Dynamic SLA Negotiation. MATA '2003, 5th International Workshop on Mobile Agents for Telecommunication Applications, October 2003.

Tae-Kyung Kim received the BSc degree in mathematics education from Dankook University, Seoul, Korea in 1997, and the MSc degree in information and communication engineering from Sungkyunkwan University, Korea in 2001. He is a PhD candidate in computer engineering at Sungkyunkwan University. His research interests are IDS, Network Security, QoS management in wireless network, and Mobile grid.

Hyung-Jin Lim received the BSc degree in computer engineering from Hallym University, KyungGi, Korea in 1998, and the MSc degree in information and communication engineering from Sungkyunkwan University, Korea in 2001. He is a PhD candidate in computer engineering at Sungkyunkwan University. His research interests are VPN, IPv6, Adaptive QoS control, and AAA in mobile environment.

Tai-Myoung Chung received his first BSc degree in electrical engineering from Yonsei University, Seoul, Korea in 1981 and his second BSc degree in computer science from University of Illinois, Chicago, IL, USA in 1984. He received the MSc degree in computer engineering from University of Illinois in 1987 and the PhD degree in computer engineering from Purdue University, W. Lafayette, USA in 1995. He is currently a professor of information and communications engineering at Sungkyunkwan University, Suwon, Korea. His research interests are active network, IDS, VPN, network management, and network security.