An Efficient Negotiation Protocol for Real-Time Multimedia Applications Over Wireless Networks

Xinbing Wang
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695
Email: xwang8@ncsu.edu

Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695
Email: wwang@ncsu.edu

Abstract— **A new negotiation protocol is proposed to reduce network resources waste for real-time multimedia services over wireless networks. Existing negotiation protocols for wireless communications can be classified into two categories: network-oriented negotiation and application-oriented negotiation. Currently, the research on the two categories are separated. This causes inefficient resource utilization, especially for real-time services, which have stringent quality of service (QoS) requirements. In this paper, a new negotiation protocol is presented, which combines application-oriented and network-oriented negotiation to achieve higher resource utilization. We design a protocol architecture and negotiation messages for three scenarios. Simulation results show that our protocol achieves higher bandwidth efficiency and shorter negotiation delay.**

I. INTRODUCTION

Real-time multimedia services over wireless networks are experiencing rapid development due to the proliferation of video applications on the World Wide Web and the emergence of broadband wireless networks. However, Quality of Service (QoS) could not be guaranteed from end-to-end up until now, although technologies such as IntServ [1], DiffServ [2], RSVP [3], MPLS [4] are proposed. This results in a serious waste of network resources, because the actual QoS parameters, such as transmission rate, are determined by the narrowest link of the entire route, which degrades the resources utilization dramatically. In the case of a real-time multimedia service, resource utilization becomes even worse during network congestion, because it requires an higher transmission rate. Furthermore, when we consider wireless environments, one terminal may use several channels (up to eight channels according to [5]) for a higher transmission rate. During network congestion, only a part of the whole capacity is utilized and several channels' capacity is wasted.

As one of the underlying technologies, negotiation is the first step for a mobile terminal to acquire network resources and service QoS parameters [6], [7], [8], [9]. Currently, the research for two categories of negotiation protocols has been conducted.

- *Application-oriented negotiation* (end-to-end negotiation) [6] [7]: In order to acquire application level quality of service (QoS) from end to end, the negotiation should result in an agreement on certain capability between each end through the exchange of service and capability information. Therefore, we name it "service-oriented negotiation". Since this is achieved by the negotiation of two ends, we also call it "end-to-end negotiation". In this paper, we use both names to emphasize different features. Capability negotiation for MPEG-21 peer-to-peer communication is explored in [7]. And SDPng is proposed as an negotiation method for Multiparty Multimedia Conference [6]. Another example is Session Initiation Protocol (SIP) [10], which is designed to set up session parameters with one or more parties.
- *Network-oriented negotiation* (end-to-net negotiation) [8], [9], [11]: In order to achieve transmission with certain QoS, end users should negotiate with a network for network resources. Correspondingly, we call it "network-oriented negotiation", which occurs between end users and networks, so called "end-to-net negotiation". Dynamic Service Negotiation Protocol (DSNP) is proposed to negotiate the Service Level Specification (SLS) in IP layer [8]. It can negotiate services from a host to a network, and a network to another network, which is particularly suitable to a wireless environment. In addition, COPS_SLS is proposed to extend the COPS [12] for intra- and interdomain network negotiation [9].

We can see that the objectives of application-oriented negotiation and network-oriented negotiation are different, which results in the separation of the two categories of protocols. However, from an end user's point of view, they are closely related. When a user launches a network-oriented negotiation, it does not mean that he just wants network access, say, 100 $Kbps$ or 1 $Mbps$ to the Internet. What the user indeed wants to obtain is an on-line application service such as on-line video, music or games. Therefore, the ultimate goal of the network-oriented negotiation is to obtain network resources to guarantee the service requirements determined by application-oriented negotiation. Currently, the QoS cannot be guaranteed end-to-end, which may cause network resource waste. Hence, in order to achieve network resource efficiency for real-time multimedia service over wireless networks, the application-oriented negotiation and network-oriented negotiation should collaborate. The existing solutions, for example SIP plus COPS, do not take this into account. We therefore propose a new negotiation protocol, which combines application-oriented and network-oriented negotiations.

The remainder of this paper is organized as follows. In Section II, the system architecture is proposed. Negotiation messages and negotiation procedures are presented in Section III. We describe our test-bed and provide simulation results in Section IV. This paper is concluded in Section V.

II. System Architecture

In this section, we will describe our system architecture, the major functions of each entity, and the relationships between each entity.

We deploy a scalable distributed architecture to solve the scalable problem shared by centralized architecture [8] [9]. In [8], a mobile terminal (MT) needs to communicate with a global QoS server (GQS) to obtain the service level agreement (SLA). The drawback is that GQS's burden is very heavy, because it must communicate with each MT and manage its QoS parameters. In addition, GQS should know the information of each local QoS server (LQS) in order to allocate proper resources within the LQS. Then, the scalability problem shared by [9] occurs, because with the increasing number of MTs, all requests from MTs within a large-scale network will be sent to this GQS. Hence, GQS will become a bottleneck of the system. In our architecture displayed in Fig. 1, each domain has one resource manager (RM), which distributes the workload of a GQS. We avoid using a global server to eliminate the bottleneck. Another benefit for the distributed architecture is the shorter negotiation delay, because to communicate with the centralized server may incur more time compared to the local server.

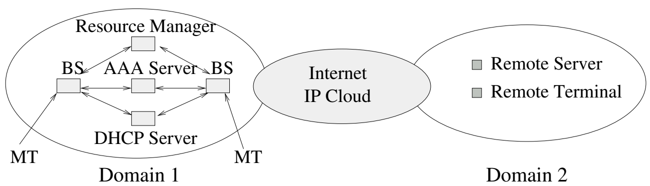

Fig. 1. System Architecture.

In our architecture, the global network consists of many domains. Each domain has one Resource Manager (RM), one Authentication, Authorization, and Accounting Server (AAA server [13]), one DHCP Server and multiple Base Stations (BSs). The end users are represented by MTs, which are the devices that communicate with a BS. We note that AAA server in this paper is only for authentication and other services of AAA server is not discussed. A BS is the bridge between MTs and networks. An RM determines whether or not the resource can be allocated according to network status. In our scheme, an RM is located in each domain. The AAA server in our architecture is only for authentication purposes. Other functions of the AAA server are beyond our topic. An MT negotiates with a Remote Terminal (RT) or a Remote Server (RS), both are located on the other side of the Internet. The difference between RT and RS is that the relationship between RT and MT is peer-to-peer, while RS and MTs is client-server mode. We note that in our architecture we keep the DHCP server to be compatible with Mobile IP requirement. In reality, it may not be necessary for certain system. For example, the negotiation in UMTS do not require a DHCP server involved.

III. New Negotiation Protocol

In this section, we will introduce our negotiation messages and procedures to meet real-time wireless service requirements and save network resources as well.

A. Signaling Messages

There are two groups of messages in our protocol: end-to-net negotiation messages (with a "+" in front of the messages) and end-to-end negotiation messages (with a "*" in front of the messages).

+ **nego_request**: This message is sent by an MT to a BS, to request a negotiation. We assume the MT has its predetermined QoS profile, which is also called Service Level Specification [14]. At the initial phase, the MT will request as the highest QoS profile as possible.
+ **nego_respond**: A BS sends this message to the MT in response to the nego_request message. If the BS cannot provide the QoS required by the MT due to the lack of resources, it returns the highest QoS profile it can provide.
+ **AAA_request**: This message is sent by a BS to a AAA server, to authenticate the MT. When a BS receives a new request from an MT, BS should first authenticate the MT before providing service to it. All the authentication information within one domain is stored in a AAA server. Hence, the BS consults the AAA server to acquire the authentication information.
+ **AAA_respond**: This message is sent by an AAA server to a BS in response to the authentication request.
+ **res_request**: This message is sent by a BS to an RM, to consult whether the resource is issued or not. The RM decides how much the resource is allocated according to the resource utilization.
+ **res_respond**: This message is sent by an RM in response to the resource_request message. In this message, the QoS profile available for the MT is included.
* **session_QoS_request**: This message is sent by an MT to an RT or an RS to request the QoS parameters for this session. In our approach, QoS profile should be negotiated per session. The RT or RS can measure the actual transmission rate and other QoS profile from end-to-end through the techniques such as calculating throughput by round trip time (RTT) divided by packet data size, which can be used to roughly estimate the actual network throughput from end-to-end. Based on the service QoS requirement, the RT or RS can determine the QoS profile for this session. We note that session_QoS_request is an abstract message, mainly for getting the QoS parameters

of this session. For different application layer protocols, it may negotiate different parameters.

* **session_QoS_respond**: An RT or an RS sends this message in response to session_QoS_request message including the QoS parameters for this session back to the MT. Then, the MT can use these parameters to renegotiate with the BS to reduce bandwidth waste.

+ **re_nego_request**: This message is sent by an MT to a BS to renegotiate a new QoS profile, which will meet the actual requirement. During one session, the MT can dynamically send this message to upgrade QoS or downgrade QoS with the service requirements. For example, when the RS sends session_QoS_respond back to the MT to inform the servicing QoS profile downgrade, the MT can renegotiate with BS to save bandwidth. This message can be sent by a BS to an MT as well, when BS needs to downgrade the QoS of an MT, for example, to meet the requirement of a higher priority MT.
+ **re_nego_respond**: This message is sent by a BS to an MT to respond to the re_nego_request message, which tells the MT the actual QoS profile during this secession.

B. Negotiation Procedures for Real-Time Applications

There are three types of real-time applications over wireless networks: (I) transmission between multiple mobile terminals without any dedicated server. Such types of applications can be peer-to-peer transmission between two terminals; (II) transmission between one server and one mobile terminal. The application can be single-user on-line games; and (III) transmission between one server and multiple users such as multi-user on-line games. Type II and type III applications are both client-server mode, while the difference between type II and type III is that for type III, users may have different capabilities. In order to promote fairness among all the end users, the servers need to choose the minimum QoS profiles of all of the users as the session QoS profile. Next, we introduce a scenario for each type of application.

1) Procedures for Type I Applications: Fig. 2 shows the negotiation procedures between an MT and an RT. There are three phases in our protocol. During the first end-to-net negotiation, the MT is authenticated by networks and knows the available network resources. In the second phase, end-to-end negotiation, the MT can obtain the service QoS profile for this session and evaluate the real-time network condition. As we state in the introduction, the actual network QoS parameters such as bandwidth are determined by the narrowest link of the entire route. Moreover, differences may exist between the service QoS parameters and the actual network QoS profile. In order to eliminate such differences, in the third phase, end-to-net negotiation once more, the MT calculates QoS requirements and the actual network QoS profile, then applies the minimum QoS profile to reduce network resources waste. Furthermore, we can see that there are two session_QoS_request messages. One is sent by the MT to the RT, and the other is sent by the RT to the MT, because type I application is a peer-to-peer communication. Therefore,

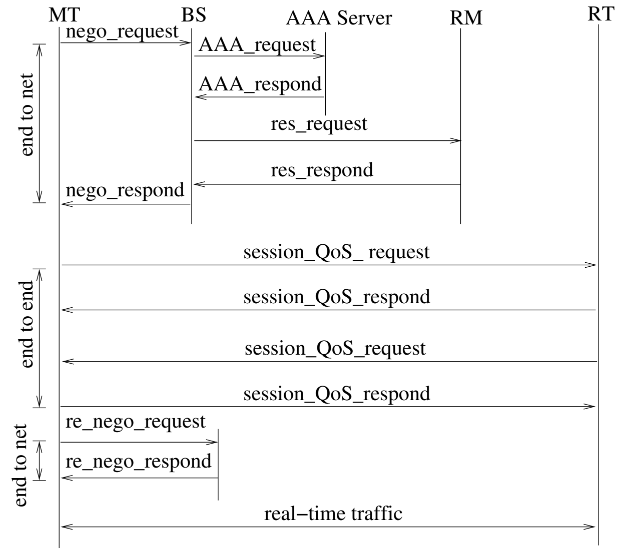

Fig. 2. Type I: Negotiation Procedures between an MT and an RT.

both the MT and the RT need to obtain the actual network profiles through that message.

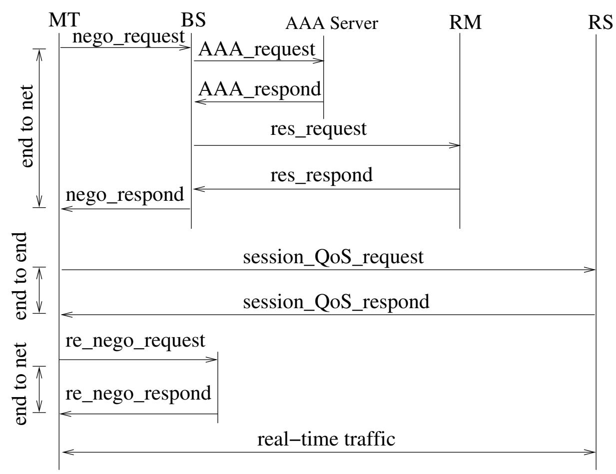

Fig. 3. Type II: Negotiation Procedures between single MT and an RS.

2) Procedures for Type II Applications: The negotiation between a single MT and an RS is much similar to the negotiation between an MT and an RT displayed in Fig. 3. The difference is that type I is a peer-to-peer communication, while type II is a client-server communication. Also there is only a one-way negotiation (i.e., from an MT to an RS), because the majority data flow is from the server to the client.

3) Procedures for Type III Applications: Fig. 5 shows that the authentication and resource initiation parts for multiple MTs to an RS is the same as type I and type II. The difference is that the RS will not send the session_QoS_response back to the MT until the RS has set up all of the connections. Then, the RS will determine the minimum QoS parameters among all the MTs. In this case, the RS will use the minimum network

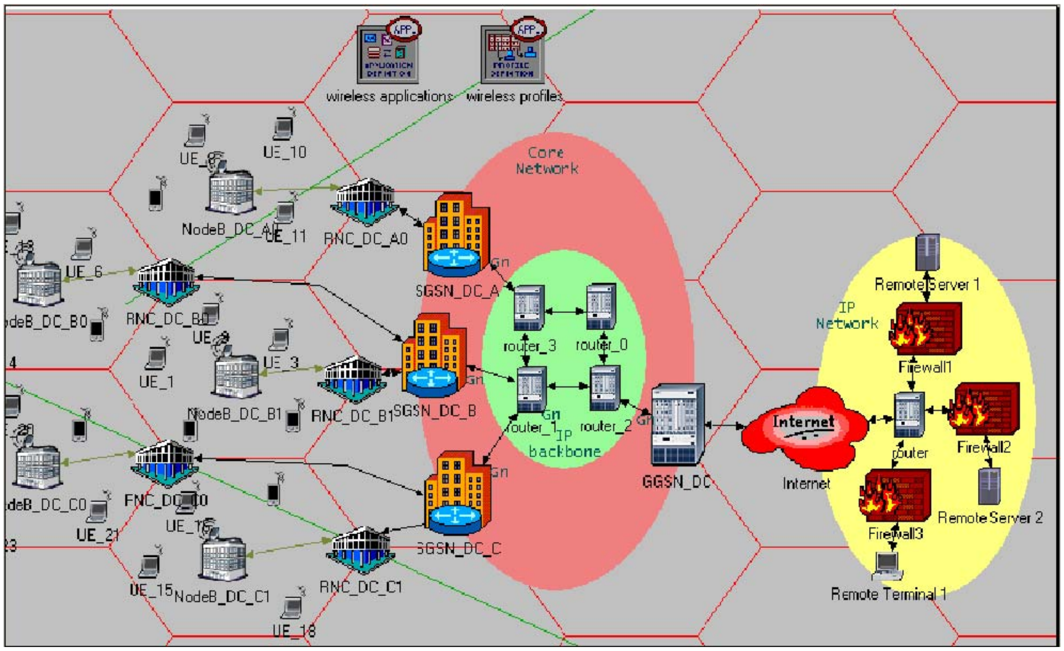

Fig. 4. Simulation Architecture.

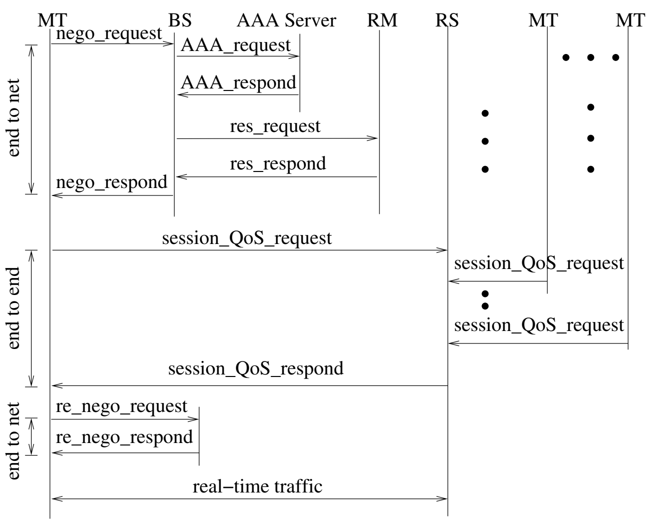

Fig. 5. Type III: Negotiation Procedures between multi MTs and an RS.

QoS parameters to communicate with all of the MTs to save transmission bandwidth among all of the participants.

IV. Performance Evaluation

We evaluate the performance of our negotiation protocol in terms of two aspects: benefit, i.e., bandwidth efficiency, and overhead, i.e., negotiation delay. Bandwidth efficiency is defined as the actual bandwidth during transmission over the bandwidth acquired by negotiation. We compare our protocol with Dynamic Service Negotiation Protocol (DSNP) [8], which is proposed for wireless Diffserv services.

A. System Design

Our simulation is based on the universal mobile telecommunication system (UMTS) module provided by OPNET. Our objective is to simulate the protocol architecture displayed in Fig. 1 and protocol procedures presented in Section III as realistically as possible. We deploy the system architecture displayed in Fig. 4. We will see that such an architecture fits our objective well.

In our system, there are three Serving GPRS Support Nodes (SGSNs), which control three domains, respectively. On one hand, each SGSN manage one or two cells representing a domain. The mobile terminals, called user equipments (UEs), are dispersed into those cells. Some UEs are fixed, and others are mobile. Within each cell, a NodeB is located in the center of the cell. Each NodeB is connected to a radio network controller (RNC). Several RNCs, then, are controlled by an SGSN, which plays the role of mobile switch center (MSC) plus visitor location register (VLR) in second generation wireless systems. On the other hand, each SGSN is attached to the routers of the core network, which is connected to the Internet (represented by the Internet cloud in Fig. 4) through a Gateway GPRS Support Node (GGSN). We can simulate various traffic load by configuring the core routers and links. On the other side of Internet, there is a simple IP network consisting of two remote servers and one remote terminal, which are under the protection of firewalls, respectively. In such a test bed, each domain belonging to one SGSN will affect the other two domains. Therefore, combined with configuring the background traffic at core network, we can simulate various network traffic load as what we want to be.

Since we deploy UMTS as underlying technology, the terminology is inherited from UMTS. The relationship between the test bed in Fig. 4 and the protocol architecture in Fig. 1 is listed as follows. The UE is corresponding to the MT, and the NodeB is corresponding to the BS. The resource manager, DHCP server and AAA server in Fig. 1 are located in the SGSN in Fig. 4.

B. Bandwidth Utilization

We evaluate bandwidth utilization of our negotiation protocol compared to DSNP [8] with regard to three types of application as described in Section III.

Our protocol yields higher bandwidth efficiency compared to DSNP in three cases displayed in Figs. 6, 7, and 8. We denote B_n as bandwidth allocated to an MT at first end-to-net negotiation phase and B_s as the service requiring bandwidth at end-to-end negotiation phase. The bandwidth allocated to the MT at second end-to-net renegotiation phase is represented by B_r and B_a is the actual bandwidth used by the MT during real-time transmission phase. Therefore, the bandwidth efficiency without renegotiation will be B_a/B_n, while the bandwidth efficiency with renegotiation will be B_a/B_r. Three groups of simulation parameters are listed in Table I.

TABLE I
SIMULATION PARAMETERS.

Parameters	Type I	Type II	Type III
B_n	100 (Kbps)	4.0 (Mbps)	2.0 (Mbps)
B_s	70 (Kbps)	4.0 (Mbps)	0.7 (Mbps)
B_r	50 (Kbps)	2.0 (Mbps)	0.7 (Mbps)

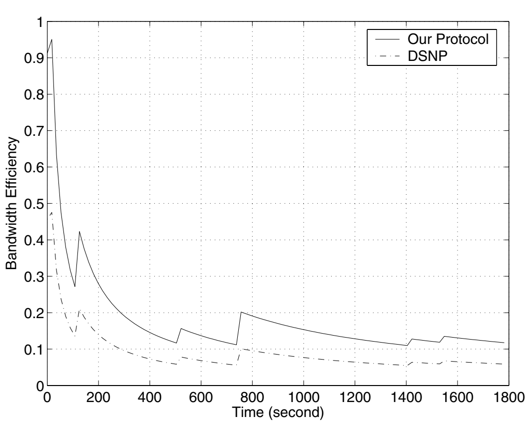

Fig. 6. Bandwidth Efficiency for Type I Application.

From the results, we can see that bandwidth efficiency is closely related to two factors: (I) the traffic pattern of each application; and (II) background traffic. In general, each type of application has its own pattern, which means the data rate is not constant and has its own characteristics. For example, in Fig 6, only within the first 50 seconds, the terminal utilizes nearly 90% bandwidth, and during the rest time of transmission, only around 20% bandwidth is used. As to the background traffic, it makes sense that if the core network is suffering in heavy traffic load, then the delay in

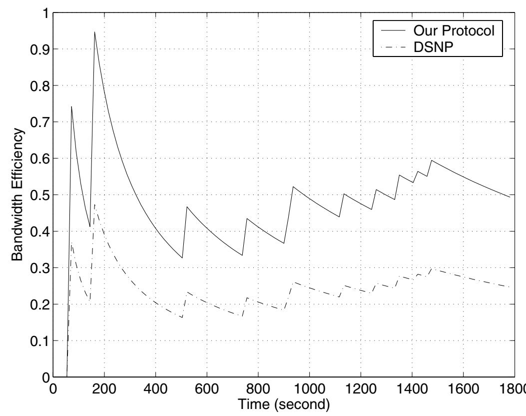

Fig. 7. Bandwidth Efficiency for Type II Application.

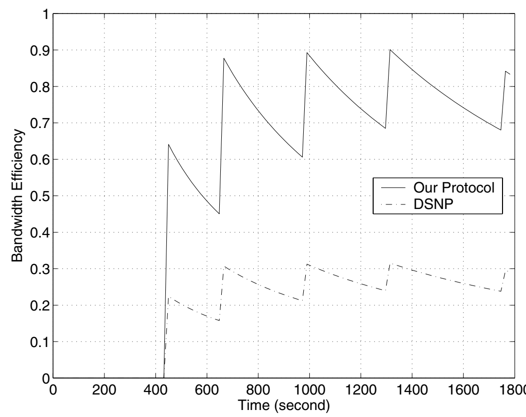

Fig. 8. Bandwidth Efficiency for Type III Application.

the queues of core routers will be increased, which results in degradation of the end-to-end transmission rate. Moreover, in order to simulate that UEs start transmission randomly, we have the UEs sleep some time through "wireless profile" in Fig. 4 before they wake up. Therefore, the start time for each UE may be different, which is revealed in Figs. 6, 7, and 8. Furthermore, the zig-zag results, especially in Figs. 7 and 8, are closely related to their traffic patterns. Sometimes, transmission rate increases and during other time, transmission rate decreases. Because for most applications, the amount of information to be transmitted are not the constant. For example, for HTTP service, different pages may contain different amount of data. Especially for multimedia service, due to the MPEG coding scheme, there are three kinds of frames: "I", "B" and "P". In general, "I" frames are much longer than "B" and "P" frames. And the distribution of these frames are not fixed as well. Therefore, in reality, the transmission rates of multimedia application cannot be constant. For different services, the traffic patterns may be different, while the benefit of our protocol over DSNP still holds true.

C. Negotiation Delay

The delay we consider in this paper consists of propagation delay, processing delay, and Media Access Control (MAC) delay. The propagation delay is determined by the distance between each entity in the protocol. The major processing delay is the AAA delay, which is dependent on the authentication algorithms. In this paper, we use AAA delay given in [15]. The MAC delay concerned in this paper is introduced by the contention of media access at IEEE 802.11 MAC layer. It is closely related to the packet arrival rate, λ, representing the number of packets per time slot [16].

TABLE II
SIMULATION PARAMETERS.

Para of ours	Values	Para of DSNP	Values
dist_MT_BS	1000 m	dist_MS_DHCP	1000 m
dist_BS_AAA	2000 m	dist_MS_QGS	10^5 m
dist_BS_RM	2000 m	dist_QGS_AAA	2×10^4 m
delay_AAA	4.934 ms	delay_AAA	4.934 ms

TABLE III
SIMULATION RESULTS.

λ	Type I (ms)	Type II (ms)	Type III (ms)	DSNP (ms)
0.1	22.386753	18.051213	22.928320	22.488670
0.2	31.599099	24.993779	29.936784	31.930454
0.3	38.387385	29.971479	34.975502	38.680413
0.4	45.453796	35.325610	40.246651	45.702509
0.5	70.436235	54.300770	59.169464	70.480976

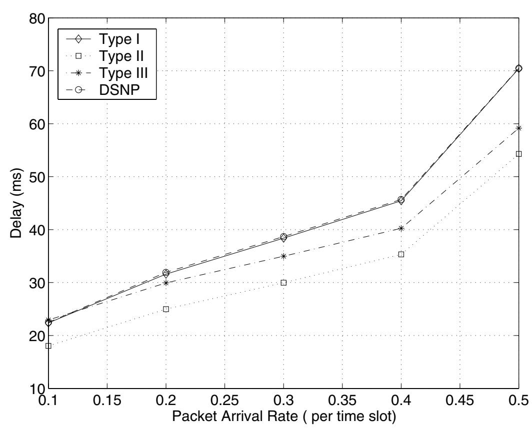

Fig. 9. Negotiation Delay.

The parameters we used in this simulation are listed in Table II. The distance between each entity is mainly defined by the template of OPNET. The simulation results are listed in Table III and Fig. 9. Since DSNP only provides end-to-net negotiation, the delays of DSNP for three types of applications are the same. The delay of our protocol for type I application is very close to that of DSNP, because the total messages of both negotiation protocol are similar in this case. A close look at Fig. 8 reveals that the delay of DSNP is even higher than that of our protocol for type I application, which is also displayed in Table III. Since DSNP deploys centralized architecture, the distances between QGS and AAA server or MS are greater than that of our protocol. Therefore, the transmission delay of DSNP is a little higher than that of our protocol. For the other two applications, the delays are smaller than that of DSNP due to the fewer messages needed for our protocol compared to that of DSNP. This means that the overhead of our negotiation protocol is smaller than that of DSNP. Therefore, our negotiation protocol can achieve higher bandwidth efficiency without the additional expense of overhead.

V. CONCLUSION

In this paper, we introduced a new negotiation protocol for real-time multimedia applications over wireless networks. Our protocol could yield higher network resources utilization compared to other existing negotiation protocols by combining the application-oriented negotiation and network-oriented negotiation. In our design, we proposed a distributed system architecture to solve the scalability problem. Based on the architecture, we specified the negotiation procedures for three types of real-time applications. Moreover, we presented simulation architecture based on UMTS module. Finally, the simulation results displayed that higher bandwidth efficiency and shorter delay were achieved by our negotiation protocol.

REFERENCES

[1] R. Braden, D. Clark, and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” in *IETF RFC1633*, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services,” in *IETF RFC2475*, Dec 1998.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP) -Version 1 Functional Specification,” in *IETF RFC2205*, Sept 1997.

[4] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture,” in *IETF RFC3031*, Jan 2001.

[5] S. Nanda, K. Balachandran, and S. Kumar, “Adaptation Techniques in Wireless Packet Data Service,” *IEEE Communications Magazine*, vol. 38, no. 1, pp. 54–64, 2000.

[6] D. Kutscher, J. Ott, and C. Bormann, “Session Description and Capability Negotiation,” in *IETF Internet Draft, draft-ietf-mmusic-sdpng-04.txt, work in progress*, March 2002.

[7] Z. Huang, S. Shen, and M. Ji, “Use Case Scenario on Negotiation Message for MPEG-21 Peer to Peer Communication,” in *International Organisation for Standardisation of MPEG*, 2002.

[8] J. Chen, A. Mcauley, V. Sarangan, S. Baba, and Y. Ohba, “Dynamic Service Negotiation Protocol (DSNP) And Wireless Diffserv,” in *IEEE International Conference on Communications*, vol. 2, pp. 1033 –1038, 2002.

[9] T. Nguyen, N. Boukhatem, Y. Doudane, and G. Pujolle, “COPS-SLS: A Service Level Negotiation Protocol for the Internet,” in *IEEE Communications Magazine*, vol. 40, pp. 158–165, May 2002.

[10] J. Rosenberg, H. Schulzrinne, and G. Camarillo, “SIP: Session Initiation Protocol,” in *IETF RFC3261*, June 2002.

[11] X. Wang and H. Schulzrinne, “RNAP: A Resource Negotiation and Pricing Protocol,” in *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pp. 77–93, June 1999.

[12] D. Durham and J. Boyle, “The COPS (Common Open Policy Service) Protocol,” in *IETF RFC2748*, Jan 2000.

[13] D. Mitton, M. St.Johns, and S. Barkley, “Authentication, Authorization, and Accounting: Protocol Evaluation,” in *IETF RFC3127*, June 2001.

[14] D. Grossman, “New Terminology for Diffserv,” in *IETF draft, draft-ietf-diffserv-new-terms-04.txt, work in progress*, Mar 2001.

[15] A. Hess and G. Schafer, “Performance Evaluation of AAA/Mobile IP Authentication,” in *http://www-tkn.ee.tuberlin.de/publications/papers/pgts2002.pdf*, 2002.

[16] M. Oliver, A. Escudero, and J. Borras, “Performance of the CSMA/CA IEEE 802.11 Protocol for Different Physical Layer Implementations,” in *Proc of WPMC’99*, Sept 1999.

