Cross-Layer-Design QoS Policy Management Framework for Wireless Heterogeneous Network

Shintaro Mori, Koji Ishii and Shigeaki Ogose

Graduate School of Engineering, Kagawa University
2217–20 Hayashi, Takamatsu, Kagawa 761-0396, Japan
Phone/FAX: +81-87-864-2210, E-mail: s.mori@ieee.org, ogose@eng.kagawa-u.ac.jp

Abstract

In future wireless and/or mobile networks, voice, high-capacity data, and multimedia contents might be converged onto a single wireless network platform, which leads to increased complexity and heterogeneity. To realize such advanced communications, it is necessary to consider users’ QoS requirements. In our related work, we proposed a novel QoS framework based on cross-layer design that is jointly optimized among all layers. In this paper, we propose a detailed protocol scheme, that is, a management scheme of users’ QoS requirements for wireless heterogeneous network. Moreover, we reveal the effectiveness of our framework.

1. Introduction

Recently, with the proliferation of various mobile devices, e.g., cellular phones, smart phones, and tablet computers, wireless services have rapidly expanded [1]. To achieve more advanced communications over wireless networks, as well as to provide various multimedia services conveniently, it is necessary to consider users’ Quality of Service (QoS) requirements in case of poor radio transmission links [2].

Moreover, in current familiar network systems, the protocol stacks is designed on the basis of layered architecture such as the seven-layer model. In other words, in terms of development, management, and maintenance, layered architecture easily promotes implementation based on a modularization scheme. In wireless networks, however, layered architecture does not have sufficient flexibility. To optimize signal processing, cross-layer design (or cross-layer optimization) has been attracting interest [3], [4]. Cross-layer design is a new paradigm in network architecture, which involves the interaction and sharing of significant information among layers.

In our related works [5], [6], we proposed a new QoS framework based on cross-layer design, that is jointly optimized over all layers. Our previously proposed technique utilizes side information (SI) and QoS policy as common information of users’ QoS requirements, and adaptive control is carried out based on them.

In this paper, we focus on the following two concerns.

• We propose a management scheme of QoS policy for the wireless heterogeneous network.

• To demonstrate the improvement of network performance upon the introduction of our cross-layer-design framework, we evaluate throughput and latency by computer simulation.

2. Cross-Layer-Design QoS Framework

Figure 1 shows block diagrams of our QoS framework based on cross-layer design at transmitter and receiver sides. In our framework, a significant difference from the conventional scheme is that the cross-layer optimizer adaptively fulfills wireless transmission control based on users’ QoS requirements. To realize these adaptive controls, the QoS converter works to exchange and share the QoS information as side information (SI) across protocol layers. For the adjustment of system parameters, the cross-layer optimizer can automatically calibrate data transmissions based on SI.

In our framework, to achieve adaptive control in wireless heterogeneous systems or several homogeneous terminals, SI
is summarized or assembled as QoS policy. As the format of QoS policy, users’ QoS requirements are expressed using the Extensible Markup Language (XML) format \[7\], \[8\]. Therefore, any users’ terminal can utilize SI flexibly and openly.

3. QoS Policy Management Scheme

3.1 Network model

In this work, as shown in Fig. 2, we assume that the wireless heterogeneous network environment is converged among macrocell, picocell, and femtocell networks. Namely, mobile network operators deploy several picocell and femtocell base stations (BSs) inside a macrocell area. In this environment, the macrocell is aimed at expanding the coverage where users can use and drive mobile services. The picocell can eliminate hotspots where network traffic is concentrated. The femtocell achieves mobile data offloading, that is, load balancing, among various wireless networks.

3.2 Procedures of handoff/handover transactions

Figure 4 shows forwarding procedures of the QoS policy, when a users’ terminal is moved from a macrocell network to a femtocell network. In our scheme, to transmit the QoS policy, we utilize Session Initiation Protocol (SIP) transactions. Specifically, a users’ terminal accesses a macrocell BS, and transmits a connection request. The macrocell BS renews and updates the users’ subscriber and position information. Then, the macrocell BS notifies with the requests of users’ registration and connection to the femtocell BS.

In the femtocell BS, the users’ request for a new connection to the femtocell network is verified. If the users’ terminal is allowable and available in the femtocell network, the QoS policy is exchanged between macrocell and femtocell BSs via the brokers. Specifically, the macrocell BS sends the request for QoS policy forwarding to the macrocell broker. Then, the macrocell broker attaches to the QoS policy management database, transmits the QoS policy to the femtocell broker, and the femtocell broker registers the QoS policy. Finally, to join and access the femtocell network, the users’ terminal
obtains connection information, such as the Care of Address (CoA) from the femtocell BS by using the ‘INVITE’ message.

4. Simulation Results

4.1 Simulation environments

To demonstrate the effectiveness of our cross-layer design, we evaluate network performance using the network simulator ns-2[12]. Table I shows simulation parameters. We set the bandwidth as 100 Mbit/s, and the latency as 5 ms in the wireline transmission section, whereas in the wireless transmission section, we set the radio bandwidth as 15 kHz (i.e., we assume a centesimal scale-down model of a cellular or WiFi system). We embed these facilities of radio transmissions into ns-2 by using the C++ programming language[13].

We introduce the DiffServ technique [14] in the wireline transmission section to achieve adaptive priority control. We assume that the DiffServ system has enough backlog and utilize the Round Robin scheduling algorithm. We generate constant-bit-rate (CBR) traffic using a traffic generator in ns-2.

4.2 Network performance

In this simulation, we compare our framework with the conventional framework, that is, without introducing cross-layer design and without adaptive control. Figures 5 and 6 show average TCP/UDP packet delay and average TCP/UDP throughput for two kinds of randomly mixed network flows (i.e., delay-sensitive and best-effort traffic), respectively. We define channel occupation ratio, $\rho$, as

$$\rho = \frac{(U_Q + U_B)}{C}$$

(1)

where $C$ is the channel capacity, and $U_Q$ and $U_B$ are the amounts of delay-sensitive and best-effort network traffic, respectively.

As a result, in the case of the $\rho < 0.6$ region, our cross-layer-design framework could not improve packet delay or throughput. This is because the bandwidth of the radio channel was allocated sufficiently, and network congestion might not have occurred. As shown in Fig. 5, for example, our framework can improve average TCP and UDP packet delays at $\rho = 0.8$ by 33.2% and 33.3%, respectively. Additionally, average packet delay maintains a constant value regardless of $\rho$ owing to adaptive cross-layer control and QoS differentiation. On the other hand, in Fig. 6, our framework can
improve average TCP and UDP throughputs at $\rho = 0.8$ by 68.7% and 55.2%, respectively. Consequently, the effectiveness of our cross-layer-design framework for the improvement of network performance is demonstrated under mixed and converged dual flows such as delay-sensitive and best-effort traffic.

4.3 Tradeoff between system complexity and performance improvement

When the cross-layer design is introduced into any network systems, it is necessary to consider the tradeoff between system complexity, which means the cost of introducing the cross-layer design, and performance improvement [4]. In our framework, as for a new overhead resulting from utilizing the cross-layer design, there is an interaction between application (or users) and the QoS converter. Namely, this interaction is to change from users’ QoS requirements to the SIs and/or the QoS policy. Additionally, this overhead occurs at every initialization of communication sessions flow by flow.

In this work, we cannot strictly evaluate the tradeoff between the overhead of cross-layer design and the improvement of network performance. This is because this tradeoff depends on the practical wireless communication environments. However, if the cross-layer design and its adaptive controls are available, there is an interaction between application (or users) and the QoS converter. Namely, this interaction is to change from users’ QoS requirements to the SIs and/or the QoS policy. Additionally, this overhead occurs at every initialization of communication sessions flow by flow.

In this work, we cannot strictly evaluate the tradeoff between the overhead of cross-layer design and the improvement of network performance. This is because this tradeoff depends on the practical wireless communication environments. However, if the cross-layer design and its adaptive controls are available, there is an interaction between application (or users) and the QoS converter. Namely, this interaction is to change from users’ QoS requirements to the SIs and/or the QoS policy. Additionally, this overhead occurs at every initialization of communication sessions flow by flow.

Generally, when $\rho$ is sufficiently large without exceeding the channel capacity, average packet delay and average throughput maintain constant values. This is because the QoS differentiation mechanism works well. Moreover, the curves between our framework (with cross-layer design) and the conventional one (without cross-layer design) are sufficiently separated.

5. Conclusions

In this paper, we proposed a novel QoS policy management scheme based on cross-layer design, and we analyzed the effectiveness of our framework. From a quantitative perspective, an adaptive control based on our framework could be useful for the improvement of network performance. Specifically, our framework could improve average TCP and UDP packet delays by 30.6% and 31.3%, and TCP and UDP throughputs by 68.7% and 55.2%, respectively. Moreover, we demonstrated that our framework could achieve effective network performance, even if we incur a system overhead cost of introducing cross-layer design.

References