멀티미디어 무선 패킷망에서 지연시간을 보장하는 공정큐잉

양 현 호*

Delay Guaranteed Fair Queueing (DGFQ) in Multimedia Wireless Packet Networks

Hyunho Yang*

요 약

공정큐잉은 자원이 노드들 간에 공유되는 유선 및 무선 멀티미디어망에서 주요한 주제중의 하나이다. 대부분의 공정큐잉 알고리즘은 GPS 알고리즘에 근거하고 있으며 공정성을 강조하는 반면 망에서 멀티미디어를 서비스를 지원하기 위해서는 필수적인 제한된 지연시간 보장 측면은 간과하고 있다. 이 논문에서는 새로운 공정큐잉 방안인 지연 보장 공정큐잉 (DGFQ, Delay Guaranteed Fair Queueing)을 제안한다. 이 방식은 무선 패킷망에서 멀티미디어 서비스를 제공하기 위하여 각 flow 별 지연시간 요구에 맞추어 제한된 지연시간을 보장한다.

ABSTRACT

Fair queueing has been an important issue in the multimedia networks where resources are shared among nodes both wired and wireless. In most fair queuing algorithms, based on the generalized processor sharing(GPS), emphasizes fairness guarantee while overlooking bounded delay guarantee which is critical to support multimedia services in the networks. In this paper, we propose a new fair queueing scheme, delay guaranteed fair queueing (DGFQ), which guaranteeing bounded delay of flows according to their individual delay requirements for multimedia services in the wireless packet networks.

키워드

Fair queueing, Bounded delay, Quality of Service (QoS), Multimedia network.

Ⅰ. Introduction

The wireless technologies are envisioned to support multimedia services, both error-sensitive and delay-sensitive applications, over the bandwidth-constrained wireless medium. With this vision in mind, the issue of providing fair and bounded delay channel access among multiple contending hosts over a scarce and shared wireless channel has come to the fore. Fair queueing has been a popular paradigm to achieve this goal in both wireline and packet cellular networking environments [1].

Among them a series of algorithms are based on generalized processor sharing (GPS) [2] approach. WFQ [3], also called pack-by packet GPS (PGPS), self clocked fair queueing (SCFQ) [4], start time fair queueing (SFQ) [5]. These

---

*순천청암대학 컴퓨터정보과학과

접수일자 : 2003. 6. 20

algorithms guarantee the fairness according to the relative weight of flows. Many of wireless fair queueing algorithms have been proposed as a variation of these algorithms.

In [5], the authors reviewed typical GPS based fair queueing algorithms and proposed a new simple and deadline guaranteeing fair queueing algorithm, i.e., start-time fair queueing (SFQ). Their proposed scheme implemented virtual time function less complex by adopting start tag of currently transmitting packet as virtual time value. In addition, SFQ allocates bandwidth fairly regardless of admission control as well as variation in server rate. Even though this scheme improves delay guarantee and fairness bound, the authors overlooked the service differentiation according to the flow characteristics, e.g., guaranteed bandwidth service or best effort service, which is critical to support multiple heterogeneous sessions both realtime and non realtime traffics.

A delay and data decoupled fair queueing (D-FQ) was presented in [6], to mitigate the inherent drawback of GPS based fair queueing algorithm i.e., the inability to deliver independent control over delay and bandwidth guarantees, which results in inflexible bandwidth management and inefficient link utilization [7]. Their approach for independent control for delay and data rate is to separate scheduling algorithms for real time and non-real time flows. Although, this scheme not only guarantees the delay deadline of real time flows but also improves the performance of non-real time flows in terms of packet delay, it should be pointed out that the computational complexity increased due to the dual scheduling architecture specifically for the explicit delay guarantee (EDG) algorithm which support real time flows.

In this paper, we propose a new fair queueing scheme i.e., delay guaranteed fair queueing (DGFQ), guaranteeing bounded delay of multimedia services. Our model is basically a GPS based fair queueing model with some modifications to guarantee bounded delay. In detail, we classified the flows into two categories, i.e., delay guaranteed (DG) class and non-delay guaranteed (NG) class by introducing additional weight factor to apply differentiated tagging operation for each classes, i.e., set a little earlier start tag to DG class flows than NG class flows. With this policy we can get better delay performance for DG class at the same fairness guarantee without serious increase in computational complexity.

The rest of this paper is organized as follows. Section 2 is devoted to describe the network model being considered. In Section 3, we describe detailed architecture and algorithm of delay guaranteed fair queueing (DFGQ). Then in Section 4, the performance of proposed scheme is evaluated. Finally, we concluded our work in Section 5.

Ⅱ. Network Model

For the purpose of this work, we basically consider wireless networks supporting multimedia services with guaranteed delay performance. We also expect that this scheme performs similarly for the wired network.

In detail, we consider packet-cellular network with a high-speed wired backbone and small, partially overlapping, shared channel wireless cells. Each cell is served by a base station, which performs the scheduling of packet transmissions for the cell. Neighboring cells are assumed to transmit on different logical channels. Every mobile host in a cell can communicate with the base station, though it is not required for any two mobile hosts to be within range of each other. Each flow of packets are identified by a certain flow index. To concentrate our work on the enhancement of the delay guarantee performance, we excluded channel error to leave as a future work. Thus, we do not make any explicit mathematical assumptions about the error model in our framework. It should be noted that in our model, we do not consider the scenario where a wireless channel is shared across several neighboring cells, which is more complicated and introduces the hidden/exposed station problems [8].

III. Delay Guaranteed Fair Queueing(DGFQ)

In this section, we propose a new fair queueing algorithm, delay guaranteed fair queueing (DGFQ), which provides differentiated bounded delay for each traffic classes having different service requirements. This proposal includes algorithm description, analytical properties and implementation in error free channel.

3.1. Algorithm Descriptions

Our proposed model, delay guaranteed fair queueing (DGFQ) basically adopts start-time fair queueing (SFQ) algorithm proposed in [5]. In DGFQ, as is in the SFQ, two tags i.e., a start tag and a finish tag, are associated with each packet. However, unlike WFQ and SCFQ, packets are scheduled in the increasing order of the start tags of the packets. Furthermore, $v(t)$ is defined as the start tag of the packet in service at time t . Finally, we assume that, in SFQ, WFQ or DGFQ scheme, there is a certain interval of time in which all flows are scheduled at least once, we call it scheduling interval.

All flows are classified in to a number of classes according to their delay bound requirements. The simplest and basic classification is to make two classes, one for delay guaranteed (DG) flows and the rest for non delay guaranteed (NG) flows. In our scheme, we introduce the service differentiation coefficient, α $(0<\alpha\leq 1)$, to handle each flow classes differently. When $\alpha = 1$, which is the case for NG class, our proposed scheme is identical to SFQ. By varying α, we can customize delay bound for individual flows i.e., adjust the relative service order of each flows in a scheduling interval.

The complete algorithm is defined as follows.

i) On arrival, k^{th} packet of flow f, p_f^k is stamped with start tag $S(p_f^k)$, computed as

$$S(p_f^k) = \max[A(p_f^k)], F(p_f^{k-1}), \quad k \geq 1 \qquad (1)$$

where $F(p_f^k)$, the finish tag of packet, p_f^k is defined as

$$F(p_f^k) = S(p_f^k) + \alpha_f \frac{l_f^k}{\phi_f} \qquad (2)$$

where $F(p_f^0) = 0$ and ϕ_f is the weight of flow f and α $(0<\alpha_f\leq 1)$ is the service differentiation coefficient for flow f. The value of α is 1 for NG class, or appropriate value for DG class.

ii) Initially the system virtual time is 0. During a busy period, the system virtual time at time t, $v(t)$, is defined to be equal to the start tag of the packet in service at time t. At the end of a busy period, $v(t)$ is set to the maximum of finish tag assigned to any packets that have been serviced by then.

iii) Packets are serviced in the increasing order of the start tags; ties are broken arbitrarily.

As shown in Fig. 1, in DGFQ, the next packet is scheduled after $\alpha_f \frac{l_f^k}{\phi_f}$, while in SFQ scheme, the next packets of each on going flow scheduled after $\frac{l_f^k}{\phi_f}$. As mentioned before, the range of α is $0 < \alpha \le 1$, the scheduling time difference i.e., $(1-\alpha_f)\frac{l_f^k}{\phi_f}$, is limited to one packet transmission duration. It is evident from (1) and (2) that the differentiated service time (virtual start time of a packet) can not go before system virtual time of the packet arrival instant. Due to this property, DGFQ maintain the fairness between each flows. On the other hand, it is still another issue to compensate lagged service in the error-prone channel environment, thus exclude from our considerations. Finally, for the computational complexity of computing virtual time $v(t)$, two schemes are identical to the complexity of $O(\log N)$ per packet, where N is the number of flows in the system.

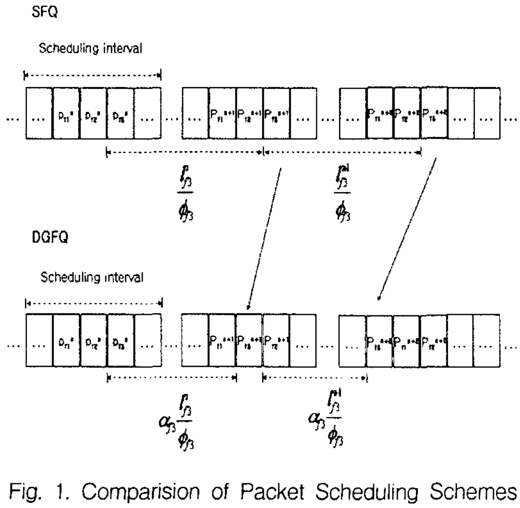

Fig. 1. Comparision of Packet Scheduling Schemes (SFQ v.s. DGFQ)

3.2. Analytical Properties of DGFQ

3.2.1. Fairness Guarantee

To derive fairness guarantee of DGFQ, we need to prove a bound on

$$\left| \frac{W_{f(t_1,t_2)}}{\phi_f} - \frac{W_{g(t_1,t_2)}}{\phi_g} \right|$$

for any interval in which both flows f and g are backlogged. We achieve this objective by establishing a lower and an upper bound on $W_{f(t_1,t_2)}$ and $W_{g(t_1,t_2)}$ in Lemmas 1 and 2, respectively.

Lemma 1. *If flow f is backlogged throughout the interval $[t_1, t_2]$, then in an DGFQ server*

$$\phi_f(v_2-v_1) - l_f^{\max} \le W_f(t_1, t_2) \tag{3}$$

where $v_1 = v(t_1)$ and $v_2 = v(t_2)$.

Lemma 2. *In an DGFQ server, during any interval $[t_1, t_2]$*

$$W_{g(t_1,t_2)} \le \phi_{g(v_2-v_1)} + l_g^{\max} \tag{4}$$

where $v_1 = v(t_1)$ and $v_2 = v(t_2)$.

Since unfairness between two flows in any interval is maximum when one flow receives maximum possible service and the other minimum service, Theorem 1 follows directly from Lemmas 1 and 2.

Theorem 1. For any interval $[t_1, t_2]$ in which flows f and g are backlogged during the entire interval, the difference in the service received by two flows at an DGFQ server is given as

$$\left| \frac{W_{f(t_1,t_2)}}{\phi_f} - \frac{W_{g(t_1,t_2)}}{\phi_g} \right| \leq \frac{l_f^{\max}}{\phi_f} + \frac{l_g^{\max}}{\phi_g} \qquad (5)$$

Theorem 1 demonstrates that DGFQ guarantees the same fairness as in SFQ [5]. This is intuitively evident that DGFQ only adjusts the start tag, hence the scheduling order, in a set of backlogged flows to be serviced in a given time interval $[t_1, t_2]$, to insure earlier service for delay guaranteed flow.

3.2.2. Delay Guarantee

On the contrary to the fairness guarantee, DGFQ shows better performance than SFQ, for the delay guaranteed (DG) flows such as real time videos. Let r_f be the rate assigned to packet p_f^k length of l_f^k. Then finish tag of packet p_f^k, $F(p_f^k)$ is defined as

$$F(p_f^k) = S(p_f^k) + \alpha_f \frac{l_f^k}{r_f^k} \qquad k \geq 1 \qquad (6)$$

Start tag of a packet and the system virtual time are defined as before.

As in SFQ, DGFQ provides deadline guarantees when the server capacity is not exceeded. To drive the deadline guarantee, let us formalize the meaning of the term "capacity is not exceed."' Let rate function for flow f at virtual time v, denoted by $R_{f(v)}$, be defined as the rate assigned to the packet that has start tag less than v and finish tag greater than v. Formally

$$R_{f(v)} = \begin{cases} r_f^k \text{ if } \exists k \ni [S(p_f^k) \leq v < F(p_f^k)] \\ 0 \quad otherwise \end{cases}.$$

Let B be the set of flows served by the server. Then the capacity of a server with average rate C is *not exceeded* if

$$\sum_{n \in B} R_n(v) \leq C \qquad v \geq 0 \qquad (7)$$

To derive the delay guarantee of DGFQ servers, we first derive a bound on the work done by an DGFQ server in the virtual time interval $[v_1, v_2]$ in Lemma 3.

Lemma 3. *If the server with parameter C is not exceeded, then the aggregate length of packets that have start tag at least* v_1 *and at most* v_2, *and are served in the same busy period, denoted by* $\widehat{\mathcal{W}}(v_1, v_2)$, *is given by*

$$\widehat{\mathcal{W}}(v_1, v_2) \leq C \sum_{n=0}^{n=k-m-1} \frac{l_f^{m+n}}{r_f^{m+n}} + \sum_{n \in B \wedge n \neq f} l_n^{\max} + l_f^k \qquad (8)$$

whenever $v_1 = S(p_f^m, r_f^m)$, $v_2 = S(p_f^k, r_f^k)$, and

$$v_2 - v_1 = \sum_{n=0}^{n=k-m-1} \frac{l_f^{m+n}}{r_f^{m+n}}.$$

For brevity, we will denote

$$\theta_f^k = \sum_{n \in B \wedge n \neq f} \frac{l_n^{\max}}{C} + \frac{l_f^k}{C}.$$

Further, we can define $A_e(p_f^k, r_f^k)$, the expected arrival time of packet p_f^k that has been assigned rate r_f^k as

$$A_e(p_f^k, r_f^k) = \max\left\{A(p_f^k),\ A_e(p_f^{k-1}, r_f^{k-1}) + \alpha_f \frac{l_f^{k-1}}{r_f^{k-1}}\right\} \quad (9)$$

where $A_e(p_f^0, r_f^0) = -\infty$.

Theorem 2 defines the delay guarantee of DGFQ servers.

Theorem 2. *If the capacity of DGFQ server with parameter C is not exceed, then the departure time of packet p^j_f, D_{DGFQ} satisfies the following inequality.*

$$D_{DGFQ}(p_f^k) \le A_e(p_f^k, r_f^k) + \theta_f^k \quad (10)$$

To compare this with SFQ, we can rewrite (10) as follows

$$D_{DGFQ}(p_f^k) \le A_e(p_f^{k-1}, r_f^{k-1}) + \alpha_f \frac{l_f^{k-1}}{r_f^{k-1}} + \theta_f^k \quad (11)$$

and the departure time of a packet at a SFQ server, given in [5] is

$$D_{SFQ}(p_f^k) \le A_e(p_f^{k-1}, r_f^{k-1}) + \frac{l_f^{k-1}}{r_f^{k-1}} + \theta_f^k \quad (12)$$

The difference of (11) and (12) is

$$(1-\alpha_f)\frac{l_f^{k-1}}{r_f^{k-1}} \quad (13)$$

where $0 < \alpha_f \le 1$. It is clear from the comparison above that, in DGFQ, the departure time of packet p_f^k is earlier by (13) than in SFQ. Hence, the delay bound is more tightly provided than in SFQ.

Ⅳ. Performance Evaluation

4.1. Simulation Environments

We used simulations method to evaluate our proposed algorithm. For the simulation, we assumed 2Mbps wireless channel which is typical capacity of current wireless mobile networks. We implemented DGFQ scheme in error free channel model to concentrate our evaluation work on the key features of proposed scheme, i.e., delay guaranteed scheduling. In the simulation, we selected average delay, maximum delay and throughput as the performance measures.

Detail definitions of these measure are explained in the following Section 4.3. Moreover, we compared these measures for the following three fair queueing schemes; SFQ, DGFQ, D-FQ[6]. SFQ also regarded as a special case of DFGQ where the *service differentiation coefficient* $\alpha = 1.0$. For the case of D-FQ is simulated to get the reference value of the delay performance because it is explicitly guarantee delay deadline with somewhat costly complex system architecture with computational complexity of $O(N_{AW} N_{RT})$ where N_{AW} is the number of allocation windows and N_{RT} is the number of realtime flows [6], and it also meaningful to compare the performance of our simple scheme with computational complexity of $O(\log N)$ where N is the number of flows in the system. Simulation performed for 1000 seconds, further, repeated 20 times to get more refined results.

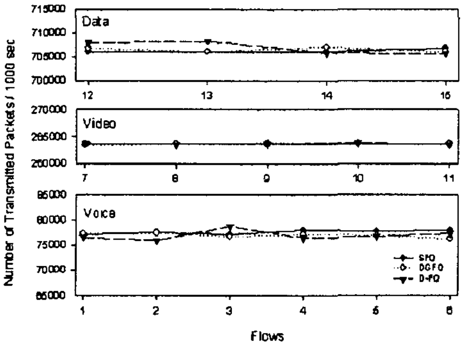

Fig. 2. Total Transmitted Packets

4.2 Traffic Source Models

In this work we choose the same traffic model used in [6] to make the results be comparable. In the simulation, a voice flow is modelled as ON-OFF signal with ON and OFF duration having exponential distributions. A video flow is modelled by modified MPEG source, where there are three types of frame, i.e., I, B and P frames. Each frame size is determined by a Lognormal distribution with a specified mean and standard deviation. A video source generates 24 frames per second. Data flow is modelled by Poisson arrival with truncated Pareto distributed burst size. Table 1 shows the traffic model parameters used in this simulation, where voice and video traffic are assumed to be real time flow while data traffic is assumed to be non real time flow.

Table 1. Summary of Traffic Sources.

	Voice	Video	Data
Model	ON/OFF	MPEG	Poission
Delay Deadline	20 ms	41.7 ms	-
Data Rate	32kbps	114kbps	300kbps
Relative Weight	0.0016	0.057	0.15

4.3. Results and Discussions

To evaluate the performance of DGFQ, we choose throughput, average delay and maximum delay as performance measures. The definitions and simulation results are given in the following.

4.3.1. Throughput

We used *throughput* as a fairness measure, which is total transmitted packets during the whole simulation duration, say, 1000 seconds. As described in previous Section Ⅲ, basically DGFQ does rearrange the service order for the packet from a set of flows according to the individual flow characteristics.

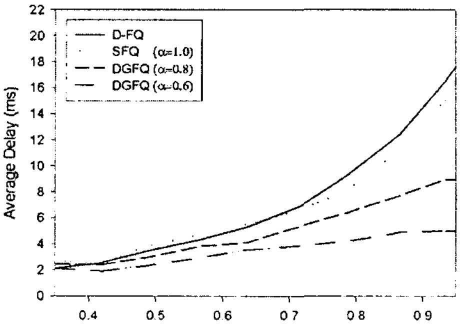

Fig.3. Average Delay

This is the only difference from SFQ. Thus conceptually, there is no changes from the results of SFQ, as far as throughput is concerned. Again, fairness among flows is guaranteed analytically by (5). Therefore, as shown in the Figure 1, it is natural to conclude that there is no difference in throughput for respective flow classes, i.e., voice, video and data.

4.3.2. Average Delay

In our work average delay is defined as the average time interval between the arrival and departure of a packet for a certain time duration. To compare the average delay of three

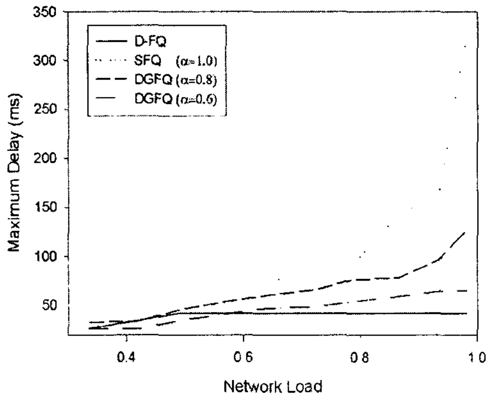

Fig. 4. Maximum Delay.

schemes, we choose 3 classes of 6 realtime video flows, then assign value for each classes, 1.0 (for SFQ), 0.8 and 0.6 respectively.

As shown in the Figure 3, the *service differentiation coefficient* α is the key parameter to control delay performance. When $\alpha=1$, our proposed scheme is identical to the start time fair queueing(SFQ) scheme. With varying α, we can handle classes of flows, having different delay bound requirements, maintaining the same fairness guarantees of flows. From the figure, our new scheme outperforms D-FQ and SFQ, independent of network load conditions. Specifically, with lower value of α, we can precisely control average delay requirement. Naturally, average delay increases for all scheme with increasing network load.

4.3.3. Maximum Delay

The *maximum delay* is another critical performance measure for real time multimedia flows. We define maximum delay as the maximum time interval between the arrival and departure of a packet in the system in a certain duration of time, say, simulation duration. We can get the results simultaneously with average delay from the same simulation. From Figure 4, we can conclude that maximum delay could be also controllable with our new scheme, though the performance is not as good as for the average delay case. Specifically, as far as for the low network load conditions are concerned, say below 60 % of full network load, maximum delay performance is comparable to that of D-FQ. On the contrary, for relative high load conditions, say above 80 % of full network load, the maximum delay performance of DGFQ get worse with increasing network load. However, in either case, DGFQ controls the maximum delay more tightly than SFQ algorithm does.

For the numerical details, the target performance measure, i.e., maximum delay bound of 50ms, typical value for the realtime video traffic, is satisfied by DGFQ for the network load up to 60 %, when $\alpha=0.6$, while for SFQ, which is $\alpha=1.0$ case, meets the target only for the low load condition, say, below 30 % of full network load. D-FQ remains in the satisfied condition all the way by virtue of complex explicit delay guarantee (EDG) algorithm. It is noteworthy that our proposed scheme could manage maximum delay guarantee by simpler mechanism than D FQ, except for the extremely high load conditions.

V. Conclusions

We proposed a new delay guaranteed fair queueing scheme, DGFQ, guaranteeing bounded delay of multimedia services with simpler algorithm than other delay guaranteed queueing algorithm SFQ [5] and D-FQ [6]. From the simulated results, our proposed scheme outperforms for average delay than other schemes in comparison. Proposed scheme performs well for maximum delay bounds except for the high network load conditions with no degradation of fairness guarantees. With some modifications and performance tuning, our proposed DGFQ scheme will be a good alternatives for the fair queueing algorithm guaranteeing QoS, in the context of delay, for the multimedia wireless packet networks.

Finally, we just consider about error-free wireless channel which is too idealistic to apply our work in the practical systems. So, much more work should be done for the error-prone wireless channel case as a future work.

References

[1] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," IEEE Wireless Communications, Vol.9 No.5, pp. 76-83, Oct. 2002.

[2] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," IEEE/ACM Transactions on Networking, Vol.1 No.3 pp. 344-357, Jun. 1993.

[3] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," Proc. ACM SIGCOMM, pp. 1-12, Sep. 1989.

[4] S. J. Golestani, "A self-clocked fair queueing scheme for high speed applications," Proc. INFOCOM'94, pp. 636-646, Apr. 1994.

[5] P. Goyal, H. M. Vin and H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," IEEE/ACM Transactions on Networking, Vol.5 No.5, pp. 690-704, Oct. 1997.

[6] S. Lee, K. Kim, A. Ahmad, "Delay and data rate decoupled fair queueing for wireless multimedia networks," Proc. GLOBECOM '02, vol. 1, pp.946-950, 2002.

[7] H. Zang, "Service disciplines for guaranteed performance service in packet switched networks," Proc. of the IEEE, vol. 83, no. 10, pp. 1374-1399, 1995.

[8] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, " MACAW: A Medium Access Protocol for Indoor Wireless LAN's," ACM Comput. Commun. Rev., vol. 24, pp. 215-225, Oct. 1995.

양현호(Hyunho Yang)

1986년 광운대학교 전자공학과 (공학사)
1990년 광운대학교 대학원 전자공학과 (공학석사)
2003년 광주과학기술원 정보통신공학과 (공학박사)
1989년 ~ 1990년 삼성 SDS 주식회사 Network Engineer
1991년 ~ 1997년 포스데이타 주식회사
POSCO 광양제철소 Network Operation Manager 1997년 ~ 현재 순천청암대학 컴퓨터정보과학과 조교수
※관심분야: 무선멀티미디어 통신, Ad Hoc 네트워크