OCBT 멀티캐스트 프로토콜에서 core 노드의 분산 계층 위치 결정

준회원 황 경 호*, 정회원 조 동 호*

Distributed Hierarchical Location Placement of Core Nodes in the OCBT Multicast Protocol

Gyung-Ho Hwang* *Associate Member* Dong-Ho Cho* *Regular Members*

요 약

Ordered Core Based Tree (OCBT) 프로토콜에서 core 스위치의 위치는 성능에 영향을 끼치는 가장 중요한 요소이다. 본 논문에서는 여러 level의 core 스위치를 어디에 둘 것인가에 대한 방법을 연구한다. 제안된 알고리즘은 전체 네트워크는 3개의 논리적 계층-Small, Medium, Large-으로 나누고, 네트워크의 각 노드(라우터)들은 자기 이외의 다른 노드들로부터 자신의 노드까지의 최단경로 비용의 합을 계산한다. S지역에서 최소의 비용을 가지는 노드를 core노드로 만들고, 같은 M지역에 속한 S지역의 core 노드들은 다른 core 노드들로부터의 최단 경로 합을 계산해서 그 값이 가장 작은 노드가 레벨이 하나 높은 core 노드가 된다. 그리고 M지역에서의 core노드에서 같은 방법으로 가장 높은 레벨을 가지는 core노드를 정한다. 제안한 방법을 네트워크에서 core 노드를 정하는 두 가지 일반적인 방법과 비교한다. 첫번째 방법은 random 방법으로 같은 수의 core 노드를 random하게 선정하는 방법이다. 두번째 방법은 center 방법으로 각각의 S지역에서 중심에 가장 가까운 노드를 core 노드로 만들고, M지역이나 L 지역에서의 core는 하위 레벨 core 중에서 해당 지역의 중심에 가까운 core로 결정한다. 시뮬레이션을 통하여 제안한 방법이 mean tree cost와 join latency관점에서 비교된 다른 방법들 보다 우월한 성능을 가지는 것을 확인할 수 있었다.

ABSTRACT

In the Ordered Core Based Tree(OCBT) protocol, a core location is the most important feature to affect the performance. In this paper, the location placement of multiple level cores is studied. The proposed algorithm is that each node in the network evaluates a sum of shortest path costs from all the other nodes and the entire network is divided into a hierarchy region to have 3-logical level(Small, Medium, Large). The node to have the lowest cost in each S-Region is decided to be a core node. Then, the core nodes in the each S-Region evaluate a sum of shortest path costs from all the other core nodes in the same M-Region. The core node to have the lowest cost is decided to be the upper level core node. Similarly the highest level core node is decided in the L-Region. The proposed algorithm is compared with conventional two methods to put the core nodes in the network. One is the random method to put the core nodes randomly. The other is the center method to locate the core node at the nearest node from the center of each S-Region and then to locate the highest level core node at the nearest core node from the center of the entire network. Extensive simulations are performed in the view of mean tree cost and join latency. Simulation results show that the proposed algorithm has better performance than random method or center method.

* 한국과학기술원 전기 및 전자공학과 통신정보 시스템 연구실 (gabriel@comis.kaist.ac.kr, dhcho@ee.kaist.ac.kr)
논문번호 : 99140-0413 접수일자 : 1999년 4월 13일

Ⅰ. 서 론

현재 인터넷에 대한 수요는 폭발적으로 증가하고 있으며 이로 인해 한정된 네트워크 자원을 효율적으로 이용하고자 하는 연구가 활발하다. 특히 화상회의나 주문형 비디오, 원격강의 등과 같이 전달하는 트래픽양이 큰 경우에는 하나의 송신자에서 여러 명의 수신자로 정보를 전달하기 위한 멀티캐스트 방식이 필요하다.

인터넷에서 멀티캐스트를 위한 프로토콜[10]은 여러 가지가 있는데, Distance Vector Multicast Routing Protocol (DVMRP)이나 Protocol Independent Multicast(PIM)방식들과 같이 정보를 내보내는 source 에 기반하여 최단 경로 트리를 이용하거나 최단 경로와 단방향 공유 트리의 조합을 이용하는 방식과 Core Based Tree (CBT) 프로토콜처럼 하나의 공유 트리를 이용하는 방식이 있다. CBT에서는 source 에 따라서 tree를 다시 만들지 않고 하나의 tree만을 이용하게 된다. 따라서, 멀티캐스트 그룹마다 하나의 공유 트리를 가지게 되므로 라우터에서 tree를 유지하기 위한 정보의 양이 줄어들고 bandwidth를 효율적으로 사용할 수 있다. CBT의 단점으로는 각 라우터들의 degree가 큰 경우 모든 송신자가 같은 공유 트리를 사용하게 되면 공유 트리의 link에 트래픽이 가중되는 현상이 발생하게 되고, 송신자와 수신자가 최단 경로로 연결되어 있지 않기 때문에 전달 delay가 source에 기반을 둔 tree보다 커지게 된다. CBT에서는 core 노드라고 불리는 노드들이 있는데, 멀티캐스트 그룹에 속하는 노드(송신자와 수신자 모두)는 가장 가까운 core 노드에 join request를 보내고 join- acknowledge를 받게 된다.

본 논문에서는 CBT 프로토콜을 수정한 OCBT프로토콜에서 core 노드를 두는 방법에 대해서 논의하고 제안된 방식에 대한 성능을 평가하기 위해 몇 가지 다른 방법들과 함께 시뮬레이션을 수행해서 mean tree cost와 join latency를 알아보았다.

서론에 이어 2절에서는 OCBT 프로토콜에 대해 간략히 설명하고, 3절에서는 OCBT 프로토콜에서 core 노드의 위치를 결정하는 제안 알고리즘에 대해서 기술하며, 4절에서는 core 노드를 두는 방식들의 성능을 비교하기 위한 네트워크 모델을 5절에서는 시뮬레이션 모델과 시뮬레이션에 대한 결과를 보여주며 마지막으로 제 6절에서 결론을 맺는다.

Ⅱ. OCBT(Ordered Core Based Tree) 프로토콜

멀티캐스트 경로설정에 대한 여러 프로토콜 중에서 OCBT 프로토콜[7]은 CBT 프로토콜의 문제점으로 지적된 루프의 형성에 대한 가능성과 멀티캐스트 tree를 형성하지 못 할 가능성을 배제하였다. 여러 개의core 노드를 가지는 CBT 프로토콜에서 멀티캐스트 그룹에 들어가기를 원하는 수신자는 가장 가까운 primary core 노드 또는 secondary core 노드에 join request (leaf-initiated join)를 보내고 그 core 노드가 secondary core 노드라면 다시 primary core 노드에 join request(core-initiated join)를 보내게 된다. 그러나 CBT에서는 이 두 가지 경우의 join request를 구분하지 않는다. OCBT의 가장 중요한 아이디어는 leaf-initiated join과 core-initiated join을 구분할 수 있는 mechanism을 제공하는 것이다.

OCBT에서는 각각의 core 노드가 core 노드의 hierarchy에서 자신의 지위를 나타내는 level을 가지게 되며 core 노드에서 나오는 control message에는 core 노드 자신의 level+1의 stamp가 찍힌다. OCBT에서 멀티캐스트에 참여하고자 하는 라우터들은 level-0의 join request를 가장 가까운 core 노드에 보내게 되고 acknowledge를 받는다. 만일 그 core 노드의 level이 (N)이라면 (N+1) level의 core 노드로 join request를 보내고 다시 acknowledge를 받게 된다. 이렇게 해서 가장 높은 level의 core 노드까지 전달된다. 따라서 OCBT에서는 core 노드를 어떻게 두는가에 따라서 많은 성능차이를 나타낼 수 있다

Ⅲ. Core 노드의 분산 계층적 위치 결정 알고리즘

본 논문의 목적은 멀티캐스트 그룹에 참여하고자 하는 구성원이 미리 정해져 있고, 각 구성원들은 멀티캐스트 그룹에 join과 leave를 dynamic하게 할 때 주어진 성능 척도에 대해서 최적의 성능을 낼 수 있도록 core 노드의 위치를 결정하는 것이다. 이 때의 성능 척도로는 mean tree cost와 join latency를 사용했는데, 이는 멀티캐스트 경로설정에 대한 성능을 나타내는 일반적 척도이다.

본 논문에서는 전체 네트워크를 [그림 1]과 같이

논리적인 hierarchy 지역으로 나눈다.

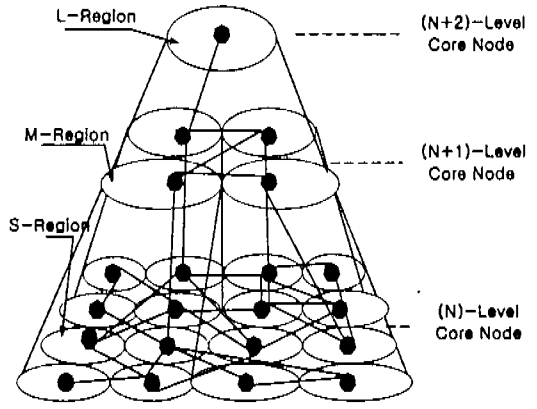

그림 1. hierarchical 네트워크 지역

M 지역은 몇 개의 S 지역을 포함하고, L 지역은 전체 네트워크를 나타낸다. 우리는 전체 네트워크를 n 개의 S 지역으로 나누고, core 노드들은 그것이 관리하는 지역에 따라서 3개의 레벨(N, N+1, N+2)을 가진다.

Core 의 위치를 결정하는 방법은 다음과 같다.

단계 1. 각각의 S 지역에서 N 레벨의 core 노드를 찾는다.

1-1. S 지역의 각 노드들은 네트워크의 모든 다른 노드들로부터의 최단 경로 비용의 합을 계산한다.

1-2. 가장 작은 비용을 가지는 노드가 해당 S 지역에서의 core 노드가 된다.

1-3. 각 core 노드는 N의 레벨을 가진다.

단계 2. N 레벨의 core 노드들에서 (N+1) 레벨의 core 노드를 선택한다.

2-1. M 지역에 속하는 모든 N 레벨 core 노드들은 같은 M 지역에 있는 다른 core 노드들로부터의 최단 경로 비용의 합을 계산한다.

2-2. 가장 작은 비용을 가지는 N 레벨 core 노드를 (N+1) 레벨의 core 노드로 선택한다.

단계 3. (N+1)레벨의 core 노드들 중에서 (N+2)레벨의 core 노드를 선택한다.

3-1. 같은 L지역에 속하는 (N+1) 레벨의 core 노드들은 다른 (N+1) 레벨의 core 노드들로부터의 최단 경로 비용의 합을 계산한다.

3-2. 최소의 비용을 가지는 (N+1) 레벨의 core 노드를 (N+2) 레벨의 core 노드로 선택한다.

제안하는 알고리즘은 네트워크의 크기와 core 레벨의 수에 대한 관점에서 볼 때 상당히 scalable 하다.

제안하는 방법과 비교가 되는 두 가지 일반적인 방법은 random 방식과 center 방식이다. Random 방식은 가장 단순한 방식으로 전체 노드 중에서 임의로 core 노드의 위치와 레벨을 선택하는 것이다. 이때 제안하는 방법과 core 노드와 레벨의 수는 같게 한다. Center 방법은 각 S 지역의 중심에 가장 가까운 노드를 N 레벨의 core 노드로 결정하고, 이들 중 같은 M 지역에 있는 core 노드들 중에서 M 지역의 중심에 가장 가까운 노드를 (N+1) 레벨의 core 노드로 선택한다. 그리고, (N+1) 레벨의 core 노드 중에서 전체 네트워크의 중심에 가장 가까운 노드를 (N+2) 레벨의 core 노드로 선택한다.

제안하고자 하는 방법의 주요 아이디어는 멀티캐스트 그룹에 참여하고자 하는 모든 노드들은 우선 가장 가까운 core 노드에게 join request를 보내기 때문에 이에 대한 비용을 줄이기 위해서 각 노드들로부터의 shortest path에 대한 cost가 가장 작은 노드를 core 노드로 선택하는 것이다.

Ⅳ. 네트워크 모델

각 시뮬레이션에서는 random graph를 사용해서 네트워크를 모델링한다. 각 노드들은 정수의 좌표를 가지고 uniform 확률로서 분포되어 있고, 두 개의 노드 (u,v) 사이에 link가 형성될 확률은 식(1)과 같다.[5]

$$P_e(u,v) = \beta \exp \frac{-d(u,v)}{\alpha L} \qquad (1)$$

여기서, d(u,v)는 두 노드 사이의 Euclidean 거리이고, L은 두 노드 사이에서 가능한 가장 긴 거리이다. 파라미터 α ($0<\alpha<1$)는 거리에 따른 연결확률을 조절한다. 즉 α가 높으면 같은 거리일 때 link 연결 확률이 높아진다. 파라미터 β ($0<\beta<1$)는 각 노드의 degree를 조절한다. β가 높을수록 degree가 증가한다. Random 그래프가 만들어 진 후에는 모든 노드가 서로 연결되어 있는지 spanning tree를 만들어서 검사한다. 그렇지 않은 경우에는 그래프를 다시 만든다. 만들어진 네트워크 모델은

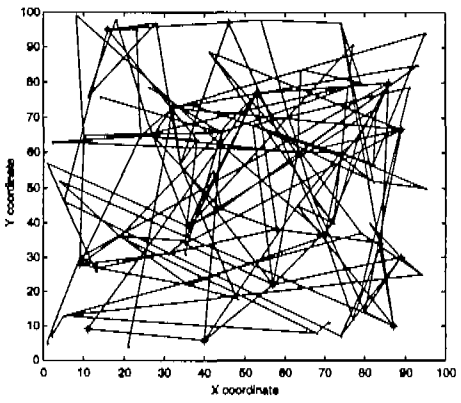

그림 2. 노드 80개의 네트워크 모델

[그림 2]와 같다. 이 방법으로 만든 그래프는 실제 인터넷 네트워크와 비슷하다고 볼 수 있다[4]. 본 논문에서는 α와 β값으로 각각 0.25, 0.2를 사용하고, L값은 100을 사용하였다.

V. 시뮬레이션 모델 및 결과

Core 노드를 두는 방식들에 대해 성능을 비교하기 위해서 시뮬레이션을 수행한다. 본 논문에서 사용된 시뮬레이션 모델은 참고문헌 [6]에서 사용된 방법과 유사하다. 시뮬레이션에서 하나의 노드를 멀티캐스트 그룹에 add 또는 remove 하는 것을 1 modification이라 하자. 이때 하나의 노드를 add 할 확률은 아래 식(2)와 같다.

$$Pc(G) = \frac{\gamma(G-M)}{\gamma(G-M)+(1-\gamma)M} \qquad (2)$$

여기서, G는 현재 멀티캐스트 그룹에 속한 노드들의 수이고, M은 전체 네트워크의 노드수이다. γ는 0에서 1사이의 값을 가지며 전체에 대한 멀티캐스트 그룹에 속하는 노드들의 수의 비율이다. 이때 확률은 G값이 γM보다 작으면 0.5보다 크고 γM보다 크면 0.5보다 작다. 따라서 이 확률식은 G값을 γM값에 수렴하도록 만든다.

Core 노드를 두는 각각의 방법에 대한 성능척도로서 mean tree cost와 join latency를 사용했다. 그리고 계산을 간단히 하기 위해서 각 link에 대한 cost는 1로 하고, 노드와 노드 사이의 delay도 1로 두었다. Mean tree cost는 한 번의 modification 후의 멀티캐스트 tree에 대한 cost를 모두 더한 후 modification 회수로 평균한 값이며, join latency는 한 개의 노드가 멀티캐스트 그룹에 join 할 때 서비스를 받기 위한 시간으로서 그 node로부터 가장 가까운 core 노드까지의 delay이다. 이때 그 core 노드가 아직 멀티캐스트 트리에 연결되어 있지 않다면, 멀티캐스트 트리에 연결된 더 높은 레벨의 core 노드까지의 delay가 join latency가 된다.

시뮬레이션에서는 random 그래프(네트워크 그래프)를 50개 만들고, 각각에 대해서 500번의 modification을 수행한다. 그리고 전체 네트워크를 16개의 S 지역으로 나누고, M 지역은 4개의 S 지역을 포함한다. 따라서, 하나의 (N+2) 레벨 core 노드와, 3개의 (N+1) 레벨 core 노드, 그리고, 12개의 (N) 레벨 core 노드를 가진다. 제안된 core 노드 결정 방법과 비교대상으로 삼은 두 가지 방법들에 대한 시뮬레이션 결과는 다음 [그림 3] 에서 [그림 6] 까지 나타내었다.

[그림 3]와 [그림 4]는 γ을 0.1로 일정하게 하고 전체 네트워크의 노드 수를 변화 시키면서 각 방법에 대한 mean tree cost 와 join latency를 나타낸 것이다.

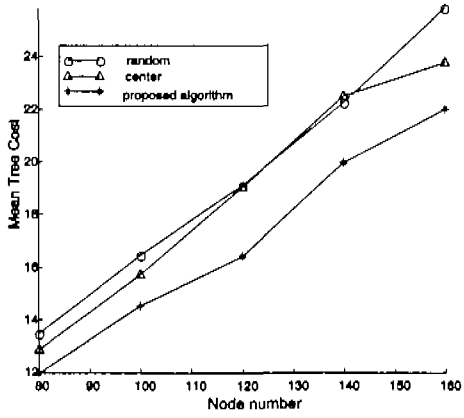

그림 3. 노드 수에 따른 mean tree cost 변화

[그림 3]에서 볼 수 있듯이 전체 노드 수가 늘어감에 따라서 mean tree cost가 증가하는 것은 당연한 결과이며 제안된 방법은 random 방식이나 center 방식에 비해서 mean tree cost가 확실히 줄어든 것을 볼 수 있다. 이는 제안된 방법이 각 노드들로부터 경로 비용을 최소로 하는 노드를 core노드로 결정했기 때문이다. 그리고 [그림 4]에서는 제안된 방법의 join latency가 다른 방법에서 보다 더 좋은 특성을 보이는 것을 알 수 있다. 또한 노드 수가 증가함에 따라 join latency가 증가하는 것을 볼 수 있다.

그리고 random 방식은 core 노드 결정을 임의로 하기 때문에 전체적으로 가장 나쁜 성능을 보였다.

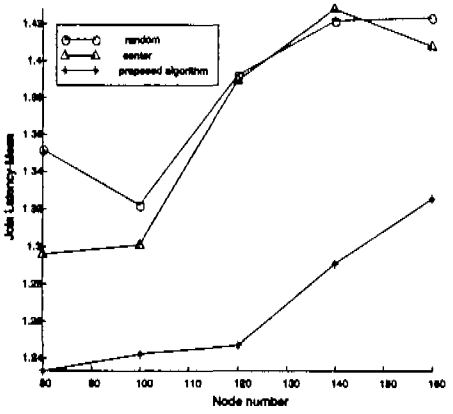

그림 4. 노드 수에 따른 join latency 변화

[그림 5]와 [그림 6]는 전체 네트워크의 노드 수를 100개로 일정하게 두고 γ을 변화 시키면서 각 방법에 대한 mean tree cost와 join latency를 나타낸 것이다. [그림5] 에서 전체 노드 수에 대한 멀티캐스트 그룹에 참여하는 노드 수의 비율이 증가할수록 mean tree cost가 증가하는 것을 볼 수 있다. 이때, 비율이 작을 때는 제안한 방법이 다른 방법보다 mean tree cost가 작았지만, 비율이 증가할수록 큰 차이가 없어진다. 이는 비율이 커지면, 거의 모든 노드들이 멀티캐스트 그룹에 참여하므로 core의 위치가 전체 cost에는 영향을 크게 주지 않기 때문이다. [그림 6] 에서 제안된 방법에서의 join latency는 다른 방법들보다 작은 것을 볼 수 있다. 전체 네트워크의 노드수가 100개로 일정하므로 멀티캐스트에 참여하고자 하는 노드로부터 core노드까지의 join latency는 비율에 따라서 비슷한 값을 나타낸다.

시뮬레이션 결과에서 볼 수 있듯이 제안된 방법은 모든 경우에 대해서 기존의 다른 방법들보다 우월한 성능을 나타내는 것을 알 수 있다.

VI. 결 론

본 논문은 인터넷 멀티캐스팅에서 OCBT 멀티캐스트 프로토콜을 이용할 때 성능에 가장 중요한 요소인 core 노드를 결정하는 방식을 제안하였다. 여기서 제안한 방법은 멀티캐스트 그룹에 참여하고자 하는 노드들이 core 노드에 join request를 보낼 때 외 경로에 대한 cost를 줄이기 위해서 각 노드들에

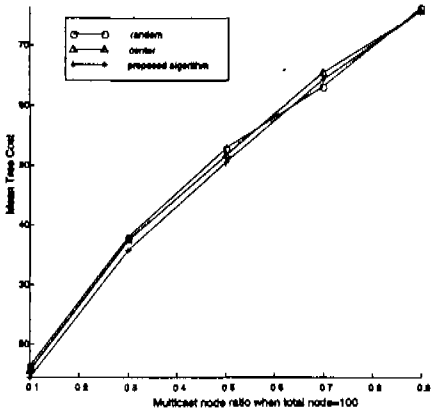

그림 5. 전체 노드 수에 대한 멀티캐스트 그룹 참여 노드수의 비율에 따른 mean tree cost 변화

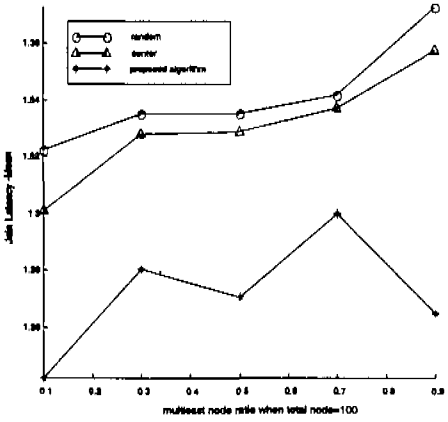

그림 6. 전체 노드 수에 대한 멀티캐스트 그룹 참여 노드수의 비율에 따른 join latency 변화

서의 shortest path cost의 합이 가장 작은 노드를 core 노드로 결정하고 계층적으로 나눈 네트워크의 지역 범위에 따라서 core 외 레벨을 결정하는 방법이다. 본 논문에서는 제안한 방법과 random하게 core 노드를 결정하는 방법과 지역적으로 중심에 가장 근접한 노드를 core 노드로 결정하는 방법에 대해서 시뮬레이션을 통해서 성능을 비교하였다. 시뮬레이션 결과를 볼 때 제안한 알고리즘은 total cost와 mean delay 면에서 다른 방법보다 월등한 성능을 나타내는 것을 알 수 있었다.

참고 문헌

[1] A.Ballardie, "Core Based Trees(CBT) Multicast

Routing Architecture," *Internet RFC 2201*, Sep. 1996.

[2] B.M. Waxman, "Performance Evaluation of Multipoint Routing Algorithm," *IEEE INFOCOM93*, pp.980-986, 1993.

[3] E.N. Gilbert, "Random Graphs," *The Annals of Mathematical Statistics*, vol.30, pp.1141-1444, 1959.

[4] Tode H., Sakai Y., Yamamoto M., Okada H. and Tezuka Y. "Multicast Routing Algorithm for Nodal Loda Balancing," *IEEE INFOCOM92*, pp.2086-2095, 1992.

[5] James Kadirire and Graham Knight, "Comparison of Dynamic Multicast Routing Algorithm for Wide-Area Packet Switched (Asynchronous Transfer Mode) Networks," *IEEE INFOCOM95*, pp. 212-219, 1995.

[6] Hwa-Chun Lin and Shou-Chuan Lai, "Core Placement for The Core Based Tree Multicast Routing Architecture," *IEEE GLOBECOM98*, pp.1049-1053, 1998.

[7] Clay Shields and J.J. Garcia-Luna-Aceves, "The Oredered Core Based Tree Protocol," *IEEE INFOCOM97*, pp.885- 892, 1997.

[8] H.C.Lin and S.C.Lai, "VTDM-A Dynamic Multicast Routing Algorithm," *IEEE INFOCOM98*, pp.1426 -1432, 1998.

[9] B.M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Area in Communications*, vol. 6, no.9, pp. 1617-1622, December 1988.

[10] Sanjoy Paul, *Multicasting on the Internet and Its Applications, Kluwer Academic Pulblishers*,1998.

황 경 호(Gyung-Ho Hwang) 준회원

1998년 2월 : 한국과학기술원 전기 및 전자공학과 졸업 (공학사)

1998년 3월~현재 : 한국과학기술원 전기 및 전자공학과 석사과정

<주관심 분야> 멀티캐스트, CDMA 무선망의 핸드오버

조 동 호(Dong-Ho Cho) 정회원

1979년 2월 : 서울대학교 전자공학과 (공학사)

1981년 2월 : 한국과학기술원 전기 및 전자공학과 (공학석사)

1985년 2월 : 한국과학기술원 전기 및 전자공학과 (공학박사)

1985년 3월~1987년 2월 : 한국과학기술원 통신공학 연구실 선임연구원

1987년 3월~1989년 12월 : 한국과학기술원 통신공학연구실 위촉연구원

1987년 3월~1998년 1월 : 경희대학교 전자계산공학과 조교수, 부교수, 교수

1989년 9월~1995년 7월 : 경희대학교 전자계산소 소장

1998. 2~현재 : 한국과학기술원 전기 및 전자공학과 부교수

<주관심 분야> 유무선 통신망, 유무선 멀티미디어 통신 서비스, 유무선 통신 프로토콜