

MANET에서의 동적 주소 할당 기법에 관한 연구

김세훈 정현구^o 이은주 윤현주* 이준원

한국과학기술원, 한국정보통신대학교*

{kimsh, hkjeong^o, leeej, joon}@camars.kaist.ac.kr, juyoon@icu.ac.kr

A Study on Dynamic Address Allocation Protocol for MANET(Mobile Ad hoc NETWORK)

Sehoon Kim, Hyunku Jeong^o, Eunju Lee, Hyeon-Ju Yoon*, Joonwon Lee

Korea Advanced Institute of Science and Technology, Information and Communication University*

요 약

MANET(Mobile Ad hoc NETWORK)은 기간망(infrastructure network)에 대한 의존 없이 필요에 따라 노드(node)들이 자발적으로 임시적인 네트워크를 형성해 데이터를 주고 받는 형태이다. 지금까지 MANET에 관한 대부분의 연구는 노드들이 네트워크에 합류하기 전에 IP주소를 할당 받는다는 가정하에 이루어졌다. 기존의 네트워크에서는 IPv4주소의 부족으로 모든 노드에게 영구적으로 IP주소를 할당 할 수 없어 동적으로 IP주소를 할당할 필요가 있을 때 DHCP를 이용해 노드들에게 IP주소를 할당했다. 그러나 MANET의 경우 노드의 이동성으로 인해 중앙집중화 된 DHCP 서버로의 접근을 항상 보장할 수 없다. 그러므로 본 논문에서는 MANET의 노드들에게 유일한 IP주소를 동적으로 할당하기 위한 토큰-기반 동적 주소 할당 프로토콜을 제안하였다.

1. 서론

무선 이동 통신 기술의 빠른 성장과 무선 통신 장비의 가격 저하로 많은 사람들이 핸드폰과 같은 무선 통신 장비를 손쉽게 사용할 수 있게 되었다. 현재 무선 이동 통신망은 크게 기간망과 비기간망, 두 종류로 나눌 수 있다. MANET은 설치된 기간망을 기반으로 동작하는 기존의 셀룰러망과는 달리 이동 노드간의 통신만으로 동작하는 비기간망이다. MANET은 유선 기간망의 도움 없이 빠른 시간에 적은 비용으로 구축할 수 있어 유선망 구축이 어렵거나 짧은 시간에 네트워크를 구성해야 하는 전쟁, 인명구조 등의 상황에 사용할 수 있다.

지금까지 MANET에 관한 대부분의 연구에서는 노드들이 MANET에 합류하기 전에 IP주소를 할당 받는다는 가정하에 연구되어 노드들에게 IP주소를 동적으로 할당하는 문제에 대해 고려하지 않았다. 그러나 MANET은 필요에 따라 노드들이 자발적으로 임시 네트워크를 형성하여 데이터를 주고 받는 형태이기 때문에 노드들에게 동적으로 IP주소를 할당하는 문제는 매우 중요하다.

인터넷에서는 IPv4주소의 부족으로 모든 노드에게 영구적으로 IP주소를 할당할 수 없다. 그러므로 동적으로 IP주소를 할당할 필요가 있을 때 DHCP[1]와 같은 프로토콜을 이용했지만, MANET에서는 노드들의 이동으로 인해 중앙집중화 된 DHCP 서버로의 접근을 항상 보장할 수 없다. 그러므로 MANET의 노드들에게 동적으로 IP주소를 할당하기 위해서는 중앙 집중형이 아닌 다른 방식의 프로토콜이 필요하다. 본 논문에서는 MANET의 모든 노드에게 각각의 유일한 주소를 할당할 수 있는 토큰-기반 동적 주소 할당 프로토콜을 제안한다.

본 논문과 구성은 다음과 같다. 2장에서는 동적으로 IP주소를 할당하는 기존의 관련 연구에 대해 소개하고, 3장에서는 본 연구에서 가정하고 있는 시스템 모델과 프로토콜 설명, 그리

고 동작방식에 대해 설명한다. 마지막으로 4장에서는 본 논문의 결론과 향후 과제에 대해 기술한다.

2. 관련연구

DHCP는 기존 네트워크에서 노드들에게 동적으로 IP주소를 할당하려는 목적으로 사용된다[1]. 하지만 MANET에서는 DHCP 서버로의 접근을 항상 보장할 수 없기 때문에 DHCP는 MANET 환경에서 사용되기 적합하지 않다.

IP주소 필드의 크기가 큰 IPv6의 경우에는, MAC(Medium Access Control)계층 주소가 유일하다는 가정하에, IP주소 생성 시 MAC주소를 이용해 모든 노드에게 유일한 IP주소를 할당할 수 있다. 하지만 IPv4처럼 IP주소 필드가 MAC주소보다 크지 않은 상황에서는 MAC주소를 이용해 IP주소를 생성할 수 없다. 예를 들어 48-bit인 IEEE 802.11 MAC계층 주소는 32-bit인 IPv4 주소 필드에 더해 질 수 없다.

Perkins가 제안한 프로토콜[2]은 AREQ(Address Request)를 MANET에 플러딩(flooding)하여 중복 주소를 찾아내기 위한 시도를 반복한 후 그 결과를 바탕으로 IP주소를 할당하는 프로토콜이다. 새로 MANET에 합류하는 노드는 임의의 주소를 선택한 후 그 주소의 할당여부를 확인하기 위해 선택한 주소를 목적지 주소로 하여 MANET에 AREQ를 플러딩한다. 일정 시간 내에 AREQ에 대한 AREP(Address Reply)를 받지 못하면 위와 같은 과정을 AREQ RETRIES만큼 반복한다. 모든 시도에서 AREP를 받지 못하면 노드는 선택한 주소가 아직 MANET내에 할당되지 않았다고 결론 내리고 그 주소를 자신에게 할당한다. 하지만 위의 프로토콜은 플러딩에 기반한 프로토콜이기 때문에 유선망에 비해 제한된 대역폭을 가진 MANET에 적합하지 않다. 또한 MANET이 여러 파티션(partition)으로 분할된 경우, 분할되었던 파티션이 다시 병합된 경우, 또는 여러 노드가 동

시에 같은 임의의 주소를 선택한 후 *AREQ*를 플래딩했을 경우 등의 해결책을 제시하지 않았다.

Nesargi가 제안한 프로토콜[3]은 새로 MANET에 합류하는 노드 *n*이 MANET에 주소 할당 요청을 보내면 *n*의 이웃 노드 중 임의의 한 노드 *i*가 *n*을 대신하여 주소 할당 작업을 수행한다. MANET 내의 모든 노드는 서로에게 할당된 주소 목록을 가지고 있다. *i*는 목록에 없는 임의의 주소를 선택해 *Initiator Request*를 MANET에 플래딩해 주소 할당을 요청한다. MANET 내의 모든 노드는 그 주소가 목록에 없다면 승인 메시지를 *i*에게 보낸다. 모든 노드에게 승인 받았다면, *i*는 *n*에게 그 주소를 할당해 준다. 이 프로토콜은 플래딩에 기반하여 주소 할당 작업을 수행하고 승인 메시지를 단 하나의 노드에게서라도 받지 못하면 주소 할당이 이뤄지지 않기 때문에, 네트워크에 큰 부하를 줄 수 있으며 주소 할당에 걸리는 시간이 길어질 수 있다.

3. 토큰-기반 동적 주소 할당 프로토콜

3.1. 시스템 모델

본 논문에서 가정하고 있는 시스템 모델은 다음과 같다.

- 외부와 연결이 없는 MANET
- IPv4의 주소 필드를 이용하여 노드에게 주소를 할당
- 노드는 MANET을 떠날 때 정상적으로 종료해야 하지만, 배터리 부족, 노드의 이동 등으로 인해 MANET을 의도치 않게 떠날 수도 있다.
- MANET에서의 메시지 전송 소요 시간은 매우 가변적이며, 메시지가 전송 중에 손실될 수 있다.
- MANET은 여러 파티션으로 분할될 수 있고, 분할된 파티션은 다시 병합될 수 있다. 파티션은 파티션 아이디(PID)로 구별된다.
- MANET의 파티션당 하나의 토큰만이 유지되어야 하며, 파티션내에서 가장 큰 주소를 가진 노드가 토큰을 가진다. 이 노드를 할당기라 부른다.
- 할당기만이 새로 합류한 노드에게 주소를 할당할 수 있는 권한을 가진다.
- 두 개 이상의 노드가 동시에 주소 할당 요청을 할 수 있다.

3.2. 기본 알고리즘

3.2.1. MANET 초기화

MANET에 처음으로 합류한 노드 *n*은 초기화 과정의 일환으로 한 홉(1-hop) 떨어진 이웃 노드들에게 *Neighbor Query*를 플래딩한 후에 *Neighbor Query* 타이머를 설정한다. *Neighbor Query* 타이머 기간 내에 *n*이 이웃 노드들로부터 어떠한 응답 메시지도 받지 못한다면, 위의 과정을 *Neighbor Query Retries*만큼 반복한다. 이 기간 내에도 이웃 노드로부터 응답을 받지 못하면 *n*은 MANET에 자신 외에 아무도 없다고 결론 내리고 스스로 임의의 주소를 할당 한 후 토큰을 생성한다. 또한 MAC주소가 유일하다는 가정하에 자신의 주소와 MAC주소의 쌍으로 이루어진 파티션 아이디 PID(Address_{*n*}, MAC_{*n*})를 스스로 할당한다.

3.2.2. 새로운 노드의 MANET 합류

MANET에 새로 합류한 노드 *n*은 한 홉 떨어진 이웃 노드들에게 *Neighbor Query*를 보낸다. 만약 할당기 *a*가 *Neighbor Query*를 받았다면, 자신의 주소 Address_{*a*}에 1을 더한 주소 Address_{*a*}+1과 PID를 담은 *Address Reply*를 *n*에게 보내고 토큰을 넘겨준다. 동시에 토큰 정보를 *n*의 주소로 갱신한다. 새로 토큰을 받은 *n*은 *Update Token*을 MANET에 플래딩한다. *Update*

*Token*을 받은 노드들 또한 토큰 정보를 *n*의 주소로 갱신한다.

할당기가 아닌 노드가 *Neighbor Query*를 받았다면, 그 노드는 할당기의 주소를 담은 *Allocator Reply*를 *n*에게 보낸다. *n*은 할당기 *a*에게 *Address Request*를 보내 주소 할당을 요청한다. *Address Request*를 받은 *a*는 *n*에게 *Address Reply*를 보냄으로써 주소와 PID를 할당하고 토큰을 넘겨준다. 동시에 *a*는 토큰 정보를 갱신하고, 토큰을 새로 받은 *n*은 *Update Token*을 MANET에 플래딩한다. *Update Token*을 받은 노드들은 토큰 정보를 갱신한다.

*Update Token*의 손실이나 전송 지연으로 인해 일부 노드들이 최신의 토큰 정보를 받지 못해 잘못된 토큰 정보를 가질 수 있다. 만약 *n*이 이런 노드들로부터 잘못된 토큰 정보를 담은 *Allocator Reply*를 받는다면 *n*은 할당기가 아닌 노드에게 *Address Request*를 보내 주소 할당을 요청할 것이다. 그러나 요청을 받은 노드는 자신이 토큰을 MANET의 다른 노드에게 넘긴 사실을 알고 있기 때문에 자신의 토큰 정보를 *Allocator Reply*에 담아 *n*에게 보낸다. 이러한 과정을 거쳐 결국 *n*은 올바른 토큰 정보를 받아 할당기에게 주소 할당을 요청할 것이다.

만약 두 개 이상의 노드가 동시에 주소 할당을 요청하는 경우, 할당기는 먼저 도착한 *Address Request*에 대해 *Address Reply*로 응답하고, 동시에 자신의 토큰 정보를 갱신하기 때문에 후에 도착한 *Address Request*는 *Address Reply* 대신 새로 갱신된 토큰 정보를 담은 *Allocator Reply*를 받는다.

3.2.3. MANET으로부터 노드의 탈퇴

노드 *k*가 MANET을 떠날 때 *k*는 할당기 *a*에게 *Address Clean*을 보냄으로써 자신이 MANET을 떠난다는 사실을 알린다. *a*는 *k*의 주소를 캐쉬에 저장한다. 이후에 MANET에 새로 합류하는 노드에게 Address_{*a*}+1 대신 캐쉬에 저장된 주소를 할당한다. 이와 같이 캐쉬에 있는 주소를 할당하는 경우에는 토큰을 넘기지 않고 스스로 유지한다.

항상 하나의 토큰만이 유지되어야 하기 때문에 할당기 *a*가 MANET을 떠날 때에는 자신이 가지고 있는 토큰을 내놓고 떠나야 한다. 그러므로 *a*는 MANET의 노드중 자신을 제외하고 가장 높은 주소를 가진 노드의 주소를 *Recalculate Token*에 담아 MANET에 플래딩한다. 이 메시지를 받은 노드들은 토큰 정보를 갱신한다. MANET의 모든 노드들은 토큰 정보를 갱신하게 될 것이며, *a*를 제외하고 가장 큰 주소를 가진 노드가 토큰을 가지게 될 것이다.

3.3. 확장 알고리즘

3.2절에서 기술한 기본 알고리즘은 MANET이 여러 파티션으로 분할되지 않거나 메시지 손실, 메시지 전송 지연 등이 없는 환경이라면 올바르게 동작할 것이다. 하지만 MANET 환경에서는 노드의 이동성 등으로 인해 메시지가 손실되거나, 메시지 전송이 지연됨으로써 MANET에 토큰이 존재하지 않거나 토큰이 2개 이상 존재하는 상황이 생길 수 있다. 또한 이로 인해 MANET의 여러 노드에게 중복된 주소가 할당되는 상황이 발생할 수 있다. 이런 상황을 막기 위하여 본 절에서는 기본 알고리즘을 확장하여 MANET에 분할이 일어나거나 분할된 파티션이 병합되는 경우, 메시지가 손실되거나 메시지 전송이 지연될 때에도 프로토콜이 올바르게 동작하도록 하기 위한 확장 알고리즘을 기술하였다.

3.3.1. 메시지 손실, 메시지 전송 지연

MANET의 특성으로 인하여 유선망에 비해 메시지가 손실되거나 전송이 지연될 가능성이 높다. 그뿐만 아니라 노드가 갑작스럽게 죽거나 이동을 많이 하는 노드들로 이루어진

MANET이라면 메시지가 손실되거나 메시지의 전송이 지연될 가능성이 더 크다. 이러한 이유로 MANET에 토큰이 사라지거나 2개 이상이 존재할 수도 있다. 우선 MANET에 토큰이 존재하지 않게 되는 경우는 아래와 같은 상황에서 발생할 수 있다.

- ① 새로 MANET에 합류한 노드에게 보내진 *Address Reply*가 전송 중에 손실되는 경우
- ② 할당기가 의도치 않게 배터리 부족이나 이동 등으로 인해 비정상 종료하는 경우
- ③ 할당기가 MANET을 떠나면서 자기 자신을 제외하고 MANET내에서 가장 큰 노드에게 토큰을 넘겨줄 때 토큰을 받은 노드가 이미 의도치 않게 MANET을 떠난 경우

위와 같은 상황이 일어나게 되면 MANET에서 토큰이 사라지기 때문에 새로 합류한 노드 n 은 *Address Request*에 대한 응답을 할당기 a 로부터 받지 못한다. 노드 n 이 *Address Request* 타이머 기간 내에 a 로부터 어떠한 응답도 받지 못한다면 n 은 *Address Request Retries*만큼 위의 과정을 반복한다. 이 기간 내에도 응답을 받지 못한다면 n 은 $Address_n$ 의 할당 여부를 확인하기 위하여 주소 $Address_n$ 로 풀러딩한다. 이 메시지에 대한 응답이 있다면 $Address_n$ 로 *Address Request*를 보내 주소 할당을 다시 요청한다. 그렇지 않고 응답이 없다면 n 은 a 가 MANET을 떠났다고 생각하고 스스로 $Address_n$ 를 할당하고 토큰을 생성함으로써 이 MANET의 할당기가 된다.

하지만 MANET에 토큰을 가진 할당기 a 가 존재함에도 불구하고 MANET에 새로 합류한 노드 n 은 메시지 손실 등의 이유로 *Address Request*에 대한 응답을 받지 못할 수 있다. 이 때 n 은 a 가 MANET을 떠났다고 생각하고 스스로 주소를 할당하고 토큰을 생성하기 때문에 MANET내에 일시적으로 토큰이 두 개 이상 존재할 수 있다. MANET내에 토큰이 두 개 이상 존재하는 경우는 아래와 같은 상황에서 발생할 수 있다.

- ① *Address Request*가 전송 중에 손실되는 경우
- ② MANET에 새로 합류한 노드 n 이 최신 토큰 정보로 갱신이 안 된 이웃 노드에게 얻은 주소로 *Address Request*를 보냈을 때, 그 주소를 가진 노드가 이미 MANET을 떠났을 경우
- ③ *Address Reply*의 전송 지연으로 인하여 토큰을 넘겨 받아야 할 노드 a 가 아직 토큰을 넘겨 받지 못하고 있는 상황에서 새로운 노드 n 이 a 에게 주소 할당을 요청하는 경우

위와 같은 이유로 MANET에 새로 합류한 노드가 스스로 토큰을 생성했다면 그 후 *Update Token*을 MANET에 풀러딩한다. 만약 *Update Token*을 받은 MANET의 노드 중 유지하고 있던 토큰 값이 새로 받은 토큰 값보다 더 크다면 노드는 $(1/Token_n) * \beta$ 시간만큼 기다린 후 *Update Token*을 MANET에 풀러딩한다. $Token_n$ 은 노드가 유지하고 있는 토큰 값이고 β 는 임의의 상수 값이다. 이와 같은 노드가 MANET에 많다면 여러 노드들이 동시에 *Update Token*을 풀러딩 함으로써 MANET에 큰 부하를 줄 수 있다. 그러므로 위와 같이 임의의 지연을 줌으로써 풀러딩 메시지 수를 줄일 수 있다. 이처럼 일시적으로 MANET내에 토큰이 존재하지 않거나 토큰이 두 개 이상 존재할 수는 있으나, 위의 알고리즘을 이용하면 결국 MANET내에서 가장 큰 주소를 가진 노드가 토큰을 가지게 된다.

3.3.2. MANET 분할 및 병합

MANET은 여러 파티션으로 분할될 수 있고, 분할된 파티션들이 다시 병합될 수 있다. MANET이 여러 파티션으로 분할된 사실을 감지하기 위하여 PID내의 주소 값과 같은 주소를 가진 노드는 주기적으로 MANET에 *Advertise Pid*를 풀러딩한다. 이

메시지를 일정 기간 동안 못 받은 노드는 MANET에 파티션이 생겼다고 생각하고 새로운 PID를 설정하기 위하여 MANET에 자신의 주소와 MAC주소로 이루어진 PID를 풀러딩한다. 많은 노드가 동시에 풀러딩 메시지를 전송하는 것을 막기 위하여 노드들은 $Address_n * \beta$ 시간만큼 기다린 후 메시지를 전송한다. $Address_n$ 는 노드의 주소이고 β 는 임의의 상수 값이다. 자신보다 더 낮은 주소 값을 가진 노드에게 *Update Pid*를 받은 노드는 새로운 PID로 파티션 정보를 갱신하고 자신이 보내려던 *Update Pid*는 취소한다.

다른 파티션에 위치한 두 노드 k 와 l 이 서로의 전송 거리 안에 위치함으로써 다른 PID를 담은 메시지를 주고 받게 된다. 서로의 PID를 보고 파티션이 병합된 사실을 알게 된다. 만약 l 이 유지하고 있는 PID내의 주소 값이 k 가 유지하고 있는 값보다 더 크다면 l 은 k 의 PID 정보를 *Update Pid*에 담아 MANET에 풀러딩한다. 노드들은 *Update Pid*의 주소 값이 자신이 가지고 있는 주소보다 작다면 PID정보를 갱신한다. 또한 k 와 l 은 서로의 토큰 정보를 이용하여 중복된 주소를 가진 노드들에게 새로운 주소를 할당하도록 한다.

3.3.3. 중복 주소 탐지(Duplicate Address Detection)

위에서 기술한 알고리즘은 MANET의 노드들에게 동적으로 유일한 주소를 할당할 수 있다. 하지만 MANET에 극심한 메시지 손실이 발생하거나 메시지 전송에 걸리는 소요시간을 예측할 수 없을 때, 중복된 주소가 할당될 가능성도 있다. 그리하여 본 논문에서는 [4]에서 제안한 Weak DAD를 이용하였다. Weak DAD는 라우팅 프로토콜을 수정하여 라우팅 테이블과 제어 패킷에 주소와 MAC 주소를 키 값으로 유지한다. 중복된 주소가 할당되더라도 주소와 키 값을 동시에 이용하여 라우팅을 수행하기 때문에 주소가 같아도 의도하지 않았던 노드에게 데이터가 전송되지 않는다. 또한 중복된 주소를 가지고 있는 노드에게 새로운 주소를 할당받게 하여 중복된 주소를 갖지 않도록 한다.

4. 결론 및 향후 과제

본 논문에서는 MANET내의 노드들에게 동적으로 주소를 할당할 수 있는 기존의 프로토콜을 고찰, 분석하였으며, MANET을 위한 새로운 동적 주소 할당 프로토콜로써 토큰-기반 동적 주소 할당 프로토콜을 제안하였다. 이 프로토콜은 토큰을 이용하여 MANET에 새로 합류하는 노드들에게 동적으로 주소를 할당하며, 메시지 손실, 메시지 전송 지연, 네트워크에 분할이 생기는 등의 경우에서도 올바르게 동작할 것이다.

향후 ns-2 시뮬레이터를 이용하여 이 프로토콜의 성능을 검증할 것이다. 주소 할당에 걸리는 시간, 주소 할당을 위해 필요한 메시지 개수 등을 다른 기법들과 비교하여 제안한 프로토콜의 성능을 평가할 것이다.

참고문헌

- [1]R. Droms, "Dynamic Host Configuration Protocol," Network Working Group- RFC 2131, March 1997.
- [2]J.E. Perkins, J.T. Malinen, R. Wakikawa, E.M. Belding-Royer, and Y.Sun, "IP Address Autoconfiguration for Ad Hoc Networks,draft-ietf-manet-autoconf-01.txt," IETF MANET Working-Group, July 2000.
- [3]S. Nesargi and R. Prakash, "MANETconf:Configuration of Hosts in a Mobile Ad Hoc Network," in *Proceedings of the 21th IEEE Conference on Computer Communications*, pp. 1059-1068, June 2002.
- [4] Nitin H. Vaidya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 206-216, July 2002.