

# DTN에서 Hybrid Spray and Wait 라우팅 프로토콜<sup>☆</sup>

## Hybrid Spray and Wait Routing Protocol in DTN

현 성 수<sup>1</sup>      정 현 진<sup>1</sup>      최 승 식<sup>1\*</sup>  
Sung-su Hyun      Hyeon-jin Jeong      Seung-sik Choi

### 요 약

DTN(Delay/Disruption Tolerant Network)은 행성과 행성 간의 통신, 행성과 인공위성의 통신과 같이 중단 간 연결이 보장되지 않고, 빈번한 연결 부재가 발생하고 기존 인터넷 인프라가 충분히 갖춰지지 않는 네트워크에 적합한 차세대 네트워크이다. 본 논문에서는 DTN 환경에서의 노드 간에 접촉한 기록 데이터를 이용하여 주기적으로 만나는 것을 확인하고, 향후 만나는 시간을 예측하여, 예측한 시간을 토대로 메시지를 누구에게 보내야 효율적인지를 선별하여, 메시지를 보내도록 한다. 또한 기존 라우팅 기법인 Spray and Wait 라우팅을 선별된 노드에게 적용하는 Hybrid Spray and Wait 알고리즘을 제안한다. 제안 알고리즘을 검증하기 위해 헬싱키 대학의 ONE(Opportunistic Network Environment) Simulator를 이용하여 이를 실험하였다. 제안하는 알고리즘의 전달성공률을 Binary Spray and Wait 라우팅의 성능과 비교하였고, 10% 적은 오버헤드를 보임을 확인하였다. 또한 불필요한 메시지의 복사를 줄일 수 있음을 확인하였다.

☞ 주제어 : DTN 프로토콜, Spray and Wait 프로토콜, One simulator

### ABSTRACT

DTN is the next generation network that is used in not guaranteed end-to-end connection such as communication between planet and satellite, frequent connection severance, and not enough for qualified network infrastructure. In this paper, we propose the hybrid Spray-and-Wait algorithm to predict the node contact time by monitoring the periodic contacts information between the nodes. Based on this method, we select one node on the basis of prediction time and copy a message for spray and wait algorithm. In order to verify the the hybrid Spray and Wait algorithm, we use the ONE(Opportunistic Network Environment) Simulator of Helsinki University. The delivery probability of the proposed algorithm is compared to the Binary Spray and Wait algorithm, it is showed that it has 10% less overhead than Binary Spray and Wait routing. It has also shown that it reduces unnecessary copying of this message.

☞ keyword : DTN Protocol, Spray and Wait Protocol, One simulator

## 1. 서 론

서로 다른 시스템을 가진 컴퓨터를 연결시켜주는 TCP/IP(Transmission Control Protocol / Internet Protocol) 및 ad-hoc 네트워크 등과 같은 프로토콜이 활발히 연구되어, 현재 거대한 유·무선 인터넷 통신망 구축이 가능하게 되었다. TCP/IP 및 ad-hoc 네트워크 등의 공통적인 특징은 이질적으로 연결된 망들을 이용하여, 미리 라우팅 경로를 설정하고 이를 이용하여 메시지를 전송하게 된다. 또한

중단 간 연결된 상태에서 안정적인 데이터 전송을 보장해 준다. 하지만 기존의 인프라를 상실한 재난 상황, 전쟁 환경, 위성 또는 행성 간 통신과 같이 네트워크 단절이 빈번히 일어나며, 메시지가 중단 간 라우팅 경로를 미리 설정하지 못할 경우, 기존의 프로토콜을 이용하는 통신은 적용하기 어렵다. 이런 특수한 환경에서 적용할 수 있는 프로토콜로 DTN(Delay/Disruption Tolerant Network)[1]이 제안되었다. DTN은 중단간 연결이 보장되지 않으며, 네트워크의 단절이 빈번히 일어나는 특수한 환경에 적용할 수 있도록 제안된 네트워크이다.

DTN에서의 각 노드는 네트워크 단절이 빈번히 일어나므로 일시적으로 메시지를 저장할 수 있는 Bundle layer을 가지고 있다. 이는 전송할 메시지가 있을 경우, 먼저 Bundle layer에서 일정한 크기의 버퍼에 메시지를 일시적으로 저장하고, 노드와 접촉시 메시지를 포워딩하여 전달

<sup>1</sup> Computer Science Engineering, Incheon National Univ., 406-772, Korea.

\* Corresponding author (sschoi@incheon.ac.kr)

[Received 18 November 2013, Reviewed 19 November 2013(R2 24 February 2014), Accepted 21 April 2014]

☆ 본 연구는 2013년 인천대학교 교내연구비 지원으로 수행되었음.

하게 된다. 즉 각 노드들은 저장 및 전달(store-and-forward) 기반의 메시지 전달을 사용하여 종단 간 연결이 끊어지더라도 메시지를 보존할 수 있다. 또한 각 노드는 메시지를 운반해주는 릴레이 노드 역할로 네트워크에 참여한다. 통신하고자 하는 두 노드가 있으면, 네트워크 안에 속해져 있는 모든 다른 노드들은 잠재적인 릴레이 노드가 된다. 전송할 메시지를 가진 소스 노드는 특별한 조건 없이 접촉하는 모든 노드를 릴레이 노드로 선택하여 메시지를 포워딩하거나 또는 일정한 조건을 만족하는 몇 개의 릴레이 노드만을 선택하여 메시지를 포워딩하게 되는데 이는 여러 라우팅 기법들에 의해 결정된다.

DTN에서의 각 노드는 미리 라우트 경로를 선정할 수 없기에, 릴레이 노드를 선별하는 과정이 중요하다. 기존의 라우팅 기법은 릴레이 노드 선별 과정이 존재하지 않아 더 높은 확률로 메시지를 전달할 기회를 잃을 경우가 발생 할 수 있다. 본 논문에서는 두 노드 간의 만남 기록을 기반으로 릴레이 노드를 선별하는 알고리즘을 제안한다. 또한 메시지 포워딩 과정을 기존 라우팅 기법을 활용하여 메시지를 보내는 Hybrid Spray and Wait 라우팅 기법을 제안한다. 기존의 라우팅 기법을 사용하지만, 릴레이 노드 선별 과정이 존재하지 않아 더 높은 확률로 메시지를 전달할 기회를 잃을 경우가 발생 할 수 있다. 여러 라우팅 기법 중 히스토리 데이터를 이용해 릴레이 노드에 따라 적절한 라우팅 기법을 선택할 수 있다는 것에 대해 독창성을 가지고 있다. 이를 헬싱키 대학의 ONE(Opportunistic Network Environment) Simulator[2]를 이용하여 제안방법의 유효성을 평가했다.

본 논문의 구성은 2,3장에서는 기존 DTN 라우팅 프로토콜 종류와 히스토리 데이터를 접목시킨 라우팅 프로토콜에 대하여 설명한다. 4장에서는 제안하는 라우팅 기법의 히스토리 데이터를 작성하는 방법과 라우팅 기법을 결정하는 알고리즘을 제안한다. 5장에서는 제안방법에 대한 성능평가를 실시하며, 6장에서 논문의 결론을 맺는다.

## 2. 관련 연구

DTN 환경에서의 라우팅 프로토콜은 크게 Deterministic 라우팅 프로토콜과 Stochastic 라우팅 프로토콜로 나뉜다 [3]. Deterministic 라우팅 프로토콜은 노드가 향후 이동하는 이동 정보나 위치 정보를 미리 알고 있는 상황을 가정한다. 이는 노드의 이동성 정보와 해당 노드별 사용 가능한 정보의 양을 이용하여 포워딩 경로를 설계하게 하는

알고리즘을 사용한다[4]. 대표적으로 oracle 정보를 이용하여 전송을 수행하는 Oracle-based 기법[5]과 노드가 움직이는 경로를 미리 예측한 Space-time graph를 이용한 라우팅 기법[6]이 있다. 본 논문에서는 네트워크에 있는 모든 노드의 위치 및 이동성 정보를 알지 못하는 상황을 가정하므로, Deterministic 프로토콜의 라우팅기법과 추구하는 바가 다르다.

Stochastic 라우팅 프로토콜에서는 네트워크 변화가 언제 바뀔지 알 수 없는 환경으로, 메시지를 언제 누구에게 포워딩 할지를 고려해야 한다. 잘 알려진 라우팅 기법으로는 Epidemic 라우팅[7], Direct Delivery 라우팅[8], Prophet 라우팅[9], MaxProphet 라우팅[10], 그리고 Spray and Wait 라우팅[11] 기법들이 있다.

먼저 Epidemic 라우팅은 자신의 전송 범위 안에 다른 노드가 들어오면 자신이 갖고 있는 메시지들의 정보를 서로 교환한다. 상대 노드가 갖고 있지 않은 메시지가 있을 경우, 해당 메시지를 복사하여 전송한다. 이런 구조를 토대로 메시지를 운반하여 목적지에게 메시지를 포워딩하는 방법이다. 메시지를 전염시키는 것과 같은 이 방법은 가장 효과적인 방법이라고 알려져 있으나, DTN에서 각 노드는 제한된 크기의 버퍼를 가지고 있기 때문에 버퍼의 부하가 발생하게 된다. 이는 이론적으로는 메시지를 전달하는데 가장 효율적인 방법으로 보이나, 버퍼의 한계로 메시지의 복사는 한정적이게 된다. 그로인해 네트워크의 메시지 과부하가 발생하고, 이는 오히려 전달 성공률이 낮아지고 높은 오버헤드를 발생시키는 결과를 낳는다.

Direct Delivery 라우팅은 Epidemic 라우팅과 반대로 메시지를 복사하는 과정이 없이, 목적지에게 메시지를 직접 포워딩을 해주는 방식이다. 메시지의 릴레이 과정을 생략하여 직접 전송을 해주기 때문에 Epidemic 라우팅의 버퍼 한계 문제점을 해결하였으나, 목적지를 직접 접촉해야 하기 때문에 긴 지연시간이 요구된다.

Spray and Wait 라우팅은 Epidemic 라우팅의 기본적인 메시지 복사과정이 같다. 하지만 만나는 모든 노드에게 메시지를 복사하는 Epidemic 라우팅과 다르게, Spray and Wait 라우팅은 메시지를 복사할 수 있는 횟수를  $L$ 로 제한한다. 또한 이를 위해 spray phase 상태와 wait phase 상태를 이용한다. spray phase 상태는 최대 복사할 수 있는  $L$ 만큼 메시지를 복사하고, 횟수  $L$ 만큼 복사과정을 했을 경우 wait phase 상태가 되어 더 이상 복사본을 만들지 않고 메시지가 목적지에 전달되기를 대기한다. 무분별한 메시지 복사로 네트워크 전체에 불필요한 전송을 증가시키는 Epidemic 라우팅과 반대로 복사할 수 있는 수를  $L$ 만큼

제한함으로써 네트워크 전체의 버퍼의 부하와 오버헤드를 감소시킨다. 이는 가장 이상적인 메시지 전달 과정을 가지는 Epidemic 라우팅에 버퍼 부하 문제를 해결한 이론적인 성능에 가장 근접한 높은 전송확률을 나타낸다.

싱글 카피 방법인 Spray and Wait 라우팅과 반대로, 최대 복사 횟수의 절반을 넘겨주는 Binary Spray and Wait 라우팅[11]기법이 있다. Binary Spray and Wait 라우팅도 spray phase 상태와 wait phase 상태를 가지며, Spray and Wait와는 다르게 수신 노드에게  $\frac{L}{2}$  만큼의 메시지 복사 권한을 부여한다. 다시 수신노드는 메시지가 만들어진 후로  $i$ 번 만나는 노드에게  $\frac{L}{2^i}$  만큼의 메시지 복사 권한을 부여하여 제한된 L만큼을 Spray and Wait 라우팅에 비해 빠른 메시지 복사를 할 수 있도록 한다.

또한 Spray and Wait 라우팅의 최대 복사본의 수를 최소한으로 하기 위해, 최대 복사본의 수를 단계에 따라 나눠 메시지를 복사하도록 하는 방법이 있다[12]. 최소한의 복사를 하고, 지정된 deadline 동안 메시지 전달이 이뤄지지 못하면 최대복사본의 수를 단계적으로 늘려 메시지의 복사를 최소화 시키도록 하는 데에 목적을 두었다. 이는 네트워크에 최소한의 메시지 복사를 유도할 수 있음을 입증하였다.

Prophet 라우팅은 노드들의 과거 접촉 정보를 이용하여 노드 간의 메시지의 전달 확률을 추정한다. 이를 이용해 목적지에 전달될 확률이 높은 노드에게 메시지를 전달한다. 메시지 복사를 전혀 하지 않고 메시지 전달만을 하므로 Epidemic 라우팅의 무분별한 메시지 복사를 줄이고, Direct Delivery 라우팅에 비해 높은 전달성공률을 보인다. 하지만 이 방법은 단일 호스트와 호스트 사이에서 라우팅 성능이 떨어지는 단점을 가지고 있다. 노드의 에너지 소모량을 이용하여 메시지 전달 확률을 재 정의하고, 메시지의 replications density, length, remaining life time을 적용한 mechanism이 연구되었다.[13]

MaxProp 라우팅은 Prophet 라우팅처럼 과거 접촉 정보를 기록하는 방법을 기초로 하되, 전송을 먼저 할 메시지와 폐기될 메시지의 순위를 결정하는 우선순위 방식을 사용한다. 우선순위는 알고리즘을 이용하여 동적으로 정의된 threshold 값을 기준으로 폐기 유무를 결정하고, 메시지의 hop count를 비교하여 먼저 보내질 메시지를 선택한다. 메시지를 전송할 경우는 Dijkstra 알고리즘을 이용해서 경로 비용(path cost)을 추정하고, 경로 비용이 적은 경로를 탐색하여 메시지를 전송하는 방법을 사용한다.

### 3. 히스토리 데이터에 대한 연구

Epidemic 라우팅의 무분별한 메시지 확산을 줄인 Spray and Wait 라우팅은 제한적인 메시지 복사가 효과적인 메시지 확산임은 이미 알려져 있다. 하지만 Spray and Wait 라우팅과 Binary Spray and Wait 라우팅은 Epidemic과 마찬가지로 만나는 모든 노드에게 메시지를 전달하도록 되어있고, 어느 노드에게 메시지를 복사해야 효율적으로 메시지 확산을 할지는 관심되지 않는다. 이는 이미 최대 복사본의 수를 모두 사용하여 wait phase 상태일 경우, 앞으로 가까운 시간에 만날 노드에게 더 빠르게 메시지 전달을 할 수 있는 기회를 포기하게 된다. 네트워크 전체에 릴레이 되고 있는 메시지의 수가 많아진다면, 노드들의 버퍼의 부하로 메시지를 운반할 수 있는 릴레이 노드의 수는 감소하게 될 것이다.

Prophet 라우팅과 MaxProp 라우팅은 히스토리 데이터를 이용하여 효율적인 라우팅 경로를 설정하는데 기초를 두고 있다. 하지만 네트워크에 속해있는 노드의 수가 적어 접촉 기회가 적을 경우 효율적으로 메시지를 전달하지 못하게 된다. 또한 경로 비용을 통해 메시지 전달을 시도하였을 때, 빠른 메시지 전달을 목적으로 두지 못할 것이다.

Stochastic 프로토콜에서의 각 라우팅은 성능을 보완하기 위하여 히스토리 데이터와 메시지 복사 과정을 동적으로 결정하는 연구들이 활발히 이뤄지고 있다. 먼저 Epidemic 라우팅의 무분별한 메시지 복사를 막기 위해 두 노드 간 접촉 정보를 저장하여 소셜 네트워크를 형성하고, 형성된 군집 구조를 이용해 메시지 복사를 제한하는 방법이 있다.[14] 제한된 구역 내에서 노드간의 집단을 형성하고, 해당 집단에 메시지를 확산시킬 것인가를 알고리즘을 통해 허용하도록 하는 방법이다.

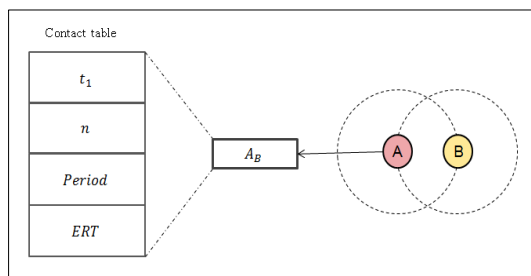
이밖에도 실제 사람의 이동을 추적한 데이터를 분석하여 각 개인이 여러 다른 종류의 중심을 기준으로 움직인다는 실험 결과가 입증되었다. 이를 개인이 움직이는 중심값을 이용하여 메시지를 전달할 노드를 선택하는 mechanism을 제안하였다[15]. 각 노드의 지역적 주기성을 이용하여 가장 적합한 라우트 방법을 결정하는 방법[16]과 노드의 속성, 움직임 그리고 노드가 속해있는 네트워크에 따라 나눠진 지역을 이용하여 적절한 라우트 방법을 결정하는 방법이 연구되었다[17].

이처럼 최근 연구들에서 노드의 주기성 및 과거 접촉 정보를 이용한 연구들이 활발히 이뤄지고 있다. 본 논문

에서는 노드 간의 접촉 데이터를 기반으로 한 Hybrid Spray and Wait 라우팅을 제안한다. 접촉 데이터를 이용하여 더 높은 확률로 메시지를 전달할 기회를 선별하며, 또한 이를 이용해 네트워크에 더 적은 메시지 복사를 유도하도록 목표를 둔다. 이를 위해 메시지 복사 전달을 기반으로 하는 Spray and Wait 라우팅의 scheme을 기반으로 하며, 높은 확률의 노드들에게 빠른 메시지 복사를 유도하기 위해 Binary Spray and Wait 라우팅 또한 기반으로 한다. 다시 말해, 라우팅 기법을 동적으로 변경하도록 하여 높은 기회로 메시지를 전달하고, 효율적인 메시지 복사를 유도하도록 하는 라우팅 프로토콜을 제안한다.

## 4. Hybrid Spray and Wait 프로토콜

### 4.1 접촉 시간 테이블 구조



(그림 1) 접촉 시간 테이블  
(Figure 1) Contact time table

노드를 선별하여 라우팅 기법을 결정하기 위해서는 먼저 노드 간에 접촉 정보를 저장하여야 한다. 임의의 두 노드가 만날 경우 서로는 상대방을 만난 시간과 만난 횟수에 대한 데이터를 저장하는 과정을 먼저 시행한다.

위 그림 1은 임의의 두 노드가 만났을 때, 접촉 시간 테이블을 저장하는 과정을 나타낸 그림이다. A 노드의 전송 범위 안에 B 노드가 들어왔을 때, B의 접촉 정보에 대한  $A_B$  인덱스를 로드하거나 생성한다.  $A_B$  인덱스를 생성하는 경우, 두 노드의 초기 만난 시간 변수  $t_1$ , 두 노드가 접촉 횟수 변수  $n$  을 생성하여 갱신한다. 다음으로 두 노드간 만나는 평균 시간 간격  $Period$ 을 갱신한다. 두 노드가  $n$ 번째 만났을 때, 접촉 횟수 및 만나는 평균 시간 간격을 이용하여 앞으로 두 노드가 만날 예측 될 시간  $ERT(Estimated Reconnected Time)$ 을 생성하고 갱신한다.

이  $ERT$ 를 이용하여 메시지를 보낼 노드를 선별한다. 이러한 과정은 A 노드의 전송 범위가 같은 B 노드에서  $B_A$  인덱스를 로드하면서 동일한 과정을 진행한다.

### 4.2 ERT 측정

$ERT$ 를 이용하여 메시지를 전송해 줄 것인지를 판단하게 된다. 두 노드가 처음으로 만난 경우가 아닐 경우, 만날 때마다  $ERT$ 를 갱신하여 메시지 전송 유무를 판단한다.  $ERT$ 를 계산하기 위해서는 먼저  $Period$ 을 구해야 한다.

$$Period_n = \frac{t_{now} - t_1}{n - 1} \quad (n > 1) \quad (1)$$

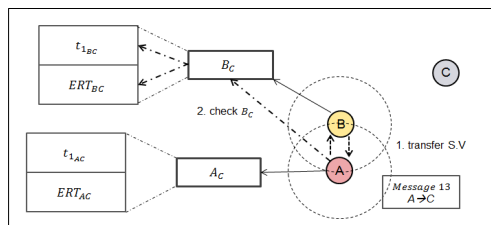
$n$ 번째까지 두 노드가 만나는 평균 시간 간격  $Period_n$ 이라고 할 때, 현재 시간  $t_{now}$ 가 두 노드가 만난 현재 시간이 되고 기준점은  $t_1$ 이 될 것이므로  $t_{now} - t_1$ 로 최초로 만난 시간부터 가장 마지막에 만난 시간 사이의 간격을 알 수 있다. 또한 두 노드가 만난 총 횟수  $n$ 으로 나눈 값을  $Period$ 에 대입한다. 이 때 만난횟수  $n$ 이 1보다 클시 적용이 된다.

$$\begin{aligned} ERT &= t_n + Period_n \quad (2) \\ &= t_n + \frac{t_{now} - t_1}{n - 1} \quad (n > 1) \end{aligned}$$

현재 상대노드와  $n$ 번째까지 만난 횟수가 존재할 경우,  $ERT$ 은 가상의 미래 예측 시간이 된다. 여기서  $ERT$ 은 가장 최근에 만난 시간  $t_n$  에서 평균적으로 만나는 시간의 주기를 가지는 시간  $Period_n$ 을 더한 값으로 예측할 수 있다. 식 (2)는  $t_1$ 을 기록한 이후, 즉  $n$ 이 1보다 클 경우 적용된다.

### 4.3 Hybrid SprayandWait 알고리즘

DTN 환경에서 각 노드는 자신이 보유하고 있는 메시지를 정의한 Summary Vector를 가지고 있다. 임의의 두 노드가 접촉하였을 경우, 두 노드는 Summary Vector를 서로 교환하여 상대방이 없는 메시지가 존재 할 경우 복사 및 릴레이 하도록 시도 할 것이다. 일반적인 Spray and Wait 라우팅에서는 상대방이 가지고 있지 않은 메시지가 있을 경우 메시지 복사를 시도 할 것이다. 하지만 Hybrid Spray and Wait 라우팅은 상대방 노드가 목적지 노드와의 접촉 시간 테이블을 먼저 읽어 오게 된다.



(그림 2) 상대방의 접촉 시간 데이터를 확인하는 과정  
(Figure 2) The process of confirming the other node's contact table

위 그림 2는 A 노드가 목적지 노드인 C 노드에게 보낼 메시지 Message 13을 소유하고 있으며, 이동 중 B 노드를 만났을 때 통신이 이뤄지는 상황을 나타낸 그림이다. 먼저 B에게서 Summary Vector를 받아와 Message 13이 존재하는지 확인한다. 만약 B 노드가 메시지를 가지고 있지 않다면, A노드는 B노드에게서 C 노드를 만났었던 기록 데이터 Bc를 로드해온다. 만약 Bc 인덱스가 존재하지 않을 경우, A 노드는 B 노드에게 메시지 Message 13을 복사하는 과정을 포기한다. 반대로 A 노드가 Bc 인덱스를 확인할 경우, ERT를 이용한 제안 알고리즘을 사용하여 라우팅을 결정한다.

**Algorithm. Hybrid SprayandWait**

```

if  $t_{a,c} \neq 0, t_{b,c} \neq 0$  then
    if  $ERT_{a,c} > ERT_{b,c}$  then
        isBinary = true
    else
        if  $P(a, b, c) > 0$  then
            isBinary = true
        else
            isBinary = false

if  $t_{a,c} = 0, t_{b,c} \neq 0$  then
    isBinary = true

if  $t_{a,c} \neq 0, t_{b,c} = 0$  then
    isBinary = false

if  $t_{a,c} = 0, t_{b,c} = 0$  then
    none
    
```

$t_{Node1, Node2}$  : met time  
 $P(a, b, c)$  :  $Period_{ac} - Period_{bc}$   
 $ERT_{Node1, Node2}$  : ERT Time

(그림 3) Hybrid Spray and Wait 알고리즘  
(Figure 3) Hybrid Spray and Wait Algorithm

위 알고리즘 그림 3은 그림 2와 같이 A 노드가 C 노드에게 전송할 메시지를 가지고 있을 때, A 노드 전송범위 안에 B 노드가 들어왔을 때 알고리즘을 적용하는 과정이다.  $t_{a,c}$ 는 두 노드 a와 c가 과거에 만난 시간을 의미한다. 만난 데이터가 없을 경우 Null을 가지게 된다.

먼저 두 노드 모두 목적지 노드 C를 만난 기록이 있을 경우, ERT 시간을 서로 비교하는 과정으로 들어간다. A 노드가 C 노드를 앞으로 만날 시간  $ERT_{AC}$ 가 B 노드가 앞으로 C 노드를 만날 시간  $ERT_{BC}$  보다 클 경우 목적지 노드를 만나러 가는 방향으로 움직일 가능성이 크다고 예상되기 때문에, Binary 기법으로 남은 복사본 수의 절반을 B 노드에게 전달해주며, 메시지를 복사시킬 수 있는 권한을 부여해준다. 반면 자신의  $ERT_{AC}$ 가 작을 경우, 서로의 만나는 주기를 비교한다.  $P(a, b, c)$ 가 양수가 나올 경우, 자신보다 늦게 만나더라도 자주 만난다고 판단하여 Binary 기법으로 메시지를 전달해주며, 반대로는 Spray 기법으로 메시지를 전달해준다.

A 노드와 C 노드가 만난 기록이 없으나 B 노드는 만난 기록이 있을 경우, A 노드는 B 노드에게 Binary 기법으로 메시지를 릴레이한다. 그리고 A 노드와 C 노드가 만난 기록이 있으나 B 노드는 만난 기록이 없을 경우, 불필요한 메시지 전송을 줄이기 위하여 Spray 기법으로 메시지를 릴레이 시킨다. 마지막으로 A, B 두 노드 모두 C 노드를 만난 기록이 없을 경우, 전달을 포기하고 그냥 지나치게 된다.

**5. 실험 및 결과**

**5.1 시뮬레이션 환경**

(표 1) 시뮬레이션 환경  
(Table 1) Simulation Setting

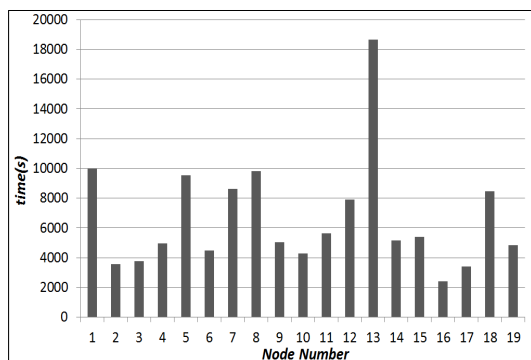
Number of nodes	20, 100, 200
Simulation time	40000s
Node Mobility Model	Random Waypoint
Number of Copies	2,4,6,8,10
Buffer Size (Mbyte)	2,5,10,15,20
Map Size(m)	400×400 1000×1000

제안 알고리즘의 검증을 위해 헬싱키 대학의 One Simulator를 사용하였으며, 1000×1000(m) 크기의 구역에

서 Random Waypoint 모델로 움직이는 노드가 20, 100, 200개 있을 경우 실험을 하였다. 먼저 노드 수가 20개일 경우, 노드간의 주기성을 확인하는 실험을 하였다. 다음으로 노드가 100개일 경우, Hybrid 기법이 Binary 기법과 Spary 기법과 메시지 복사수를 비교하였고, 노드수가 100, 200개일 경우 주요 라우팅 기법들과 전달성공률, 오버헤드 및 전달지연율의 성능 평가를 비교하였다. 이 때 Binary 기법, Spray 기법 그리고 Hybrid 기법의 최대 복사수는 10개로 고정시켰으며 네트워크의 모든 노드의 버퍼는 최소 2Mbyte에서부터 최대 20Mbyte까지 늘리며 실험하였다.

## 5.2 주기성 확인

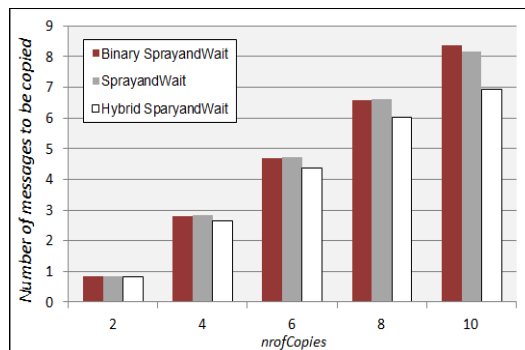
노드간의 주기성을 확인하기 위하여 노드의 수가 20개일 경우, 임의의 한 노드를 만나는 주기성을 알아보는 실험을 하였다. 실험은 400×400(m) 크기의 구역에서 노드의 버퍼를 10Mbyte로 하고, 최대 복사본의 수를 2개에서 10개까지 늘려가며 실험을 하였다.



(그림 4) 일정한 시간동안 특정 한 노드를 다시 만나는데 걸리는 평균 시간

(Figure 4) Average met again time to any particular node at a set time

그림 4는 각 노드가 번호가 0번인 노드를 다시 만나는데 걸리는 주기를 나타낸 그래프이다. 19개 노드는 평균적으로 6627.602초 간격을 가진다. 하지만 노드 16은 2391.6초를 가지며, 13번 노드는 약 8배인 18643.5초의 주기 간격을 가지는 것을 확인할 수 있다. 즉, 어느 특정 노드를 만나는데 걸리는 시간은 독립적이며 서로 다르다는 것을 확인할 수 있다.

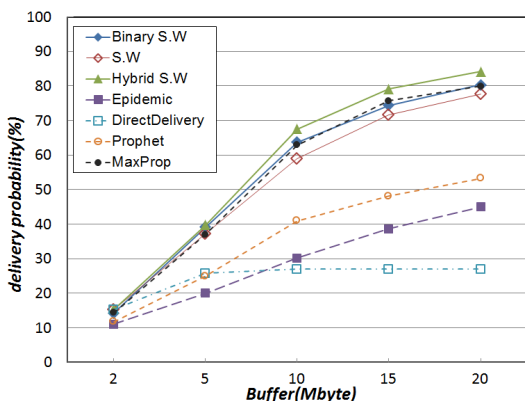


(그림 5) 한 메시지를 전송하는 데 복사되는 메시지의 수  
(Figure 5) Number of messages to be copied in order to send a message

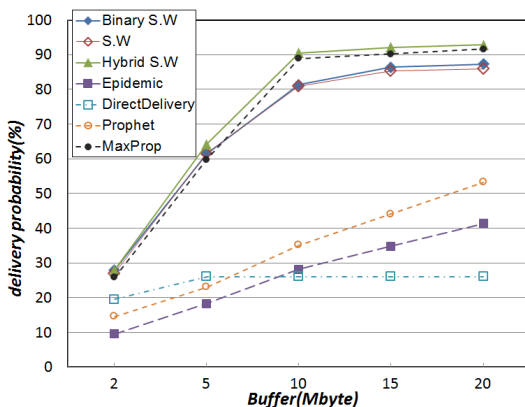
먼저 그림 5는 총 노드수가 100개일 때, 한 메시지를 전송하는 데 복사되는 메시지의 수이다. 버퍼의 크기가 10Mbyte일 때, Binary 기법, Spray 기법은 모두 약 9번의 메시지의 복사가 필요하였다. 반면에 Hybrid 기법은 노드 한 개가 줄어든 약 7개의 메시지의 복사가 필요하다. 이는 목적지에게 성공적인 메시지 전달을 위해서 Hybrid 기법이 더 적은 비용이 필요하며, 네트워크에 더 적은 부하를 일으킨다는 것을 알 수 있다.

## 5.3 성능 평가

다음으로 총 노드수가 100개, 200개일 경우 각각 주요 라우팅 기법들과 성능 비교를 하였다. 성능 비교는 버퍼의 크기에 따라 전달성공률, 오버헤드 비율, 전달 지연율을 측정하였다. 전달성공률은 메시지 전달 시도에 대한 메시지 수신 비율을 의미한다. 즉, 높은 전달성공률은 목적지에 더 많은 메시지가 도착하였다는 것을 의미한다. 오버헤드 비율은 불필요한 메시지 릴레이 과정을 나타낸 것으로, 오버헤드 비율이 적을 수록 한 메시지를 전달하는데 적은 릴레이 과정을 한다는 의미이다. 전달 지연율은 네트워크 내에서 메시지가 목적지에 전송되는 데 걸리는 지연을 의미하며, 이는 적은 지연 시간을 갖는 네트워크가 임의의 한 메시지가 전달되는데 걸리는 시간이 더 적다는 것을 의미한다.[18] 이 밖에도 각 노드들은 동일한 통신조건을 가지고 있는 환경이며, Epidemic 라우팅을 제외한 메시지 복사 과정이 있는 모든 라우팅 기법은 최대 복사본의 수를 10개로 고정하였다.



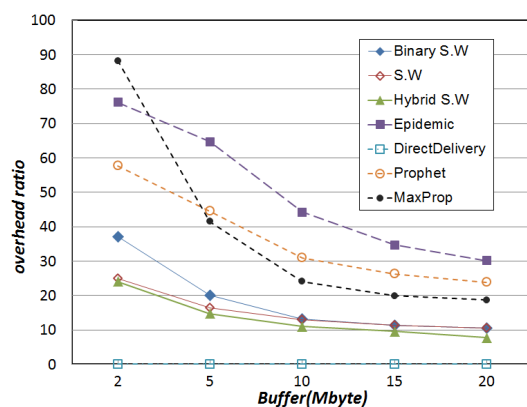
(그림 6) 노드수가 100일 경우, 전달성공률  
(Figure 6) Delivery Probability in the case that number of nodes are 100



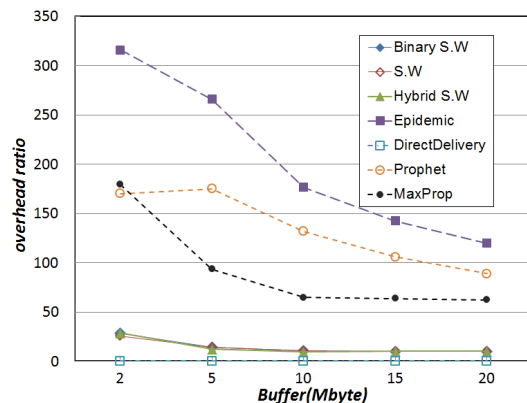
(그림 7) 노드수가 200일 경우, 전달성공률  
(Figure 7) Delivery Probability in the case that number of nodes are 200

그림 6과 7은 다양한 버퍼 크기에 따라 각 라우팅 기법의 메시지 전달 성공률을 보여준 그래프이다. Epidemic 라우팅은 만나는 모든 노드에게 메시지 복사를 시도한다. 하지만 네트워크에 속해있는 노드는 제한된 버퍼의 크기를 가지고 있으며, 무분별한 복사는 오히려 버퍼의 부하를 야기하게 되어 성공적인 메시지 전송할 기회를 잃게 만든다. 이로 인해 버퍼의 크기가 커질수록 메시지 전달 성공률도 높아지지만, 대체로 낮은 성능을 보인다. 노드수가 100개일 경우 Hybrid Spray and Wait 라우팅이 가장 좋은 성능을 보이며, 노드수가 200개일 경우 MaxProp 라

우팅과 Hybrid Spray and Wait 라우팅이 가장 좋은 성능을 내는 것을 알 수 있다. 기존의 Binary Spray and Wait 라우팅과 Spray and Wait 라우팅은 버퍼의 크기가 10Mbyte일 경우, Hybrid Spray and Wait 라우팅보다 약 10% 성능이 떨어지는 것을 확인할 수 있다. 이는 Hybrid Spray and Wait 라우팅이 앞으로 메시지의 목적지를 만날 확률이 높은 노드를 선별하는 과정과 메시지의 동적 복사 과정으로 네트워크에 효율적인 메시지 복사가 이뤄짐을 알 수 있다. 정해진 경로를 이동하는 노드들 상황에서 유리한 Prophet 라우팅은 노드의 움직임을 예측할 수 없는 환경에서 성능이 떨어지는 것을 확인할 수 있다.

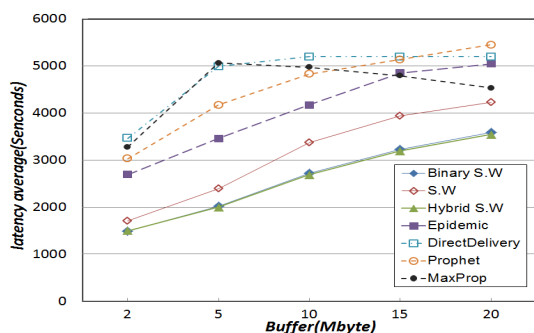


(그림 8) 노드수가 100일 경우, 오버헤드  
(Figure 8) Overhead ratio in the case that number of nodes are 100

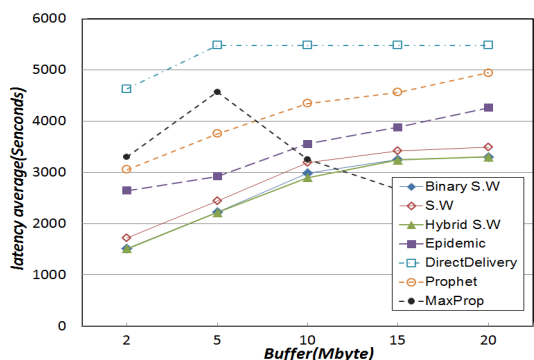


(그림 9) 노드수가 200일 경우, 오버헤드  
(Figure 9) Overhead ratio in the case that number of nodes are 200

그림 8과 9는 한 메시지를 전송하기 위해 불필요한 메시지가 얼마나 릴레이를 되는지를 나타낸 오버헤드 비율 그래프이다. 메시지 릴레이 과정없이 목적지에게 메시지를 직접 포워딩하는 **DirectDelivery** 라우팅은 오버헤드가 발생하지 않는다. **Epidemic** 라우팅은 노드수 100, 200일 때 모두 무분별한 메시지 복사로 인해 가장 높은 오버헤드 비율을 보인다. **MaxProp** 라우팅은 버퍼 크기가 커질수록 오버헤드 비율이 감소하나, **Binary** 라우팅, **Spray** 라우팅 그리고 **Hybrid** 라우팅보다 모두 높은 오버헤드 비율을 보임을 확인할 수 있다. 전달성공률에서 가장 좋은 성능을 보이는 **MaxProp** 라우팅과 **Hybrid** 라우팅은 오버헤드 측면에서 **Hybrid**가 더 적은 오버헤드로 더 높은 전달성공률을 보이는 것을 확인할 수 있다. 즉, 기존의 **Spray** 라우팅과 **Binary** 라우팅의 전달성공률을 증가시키면서 낮은 오버헤드 비율을 보이는 것을 확인할 수 있다.



(그림 10) 노드수가 100일 경우, 전달지연율  
(Figure 10) Latency Average in the case that number of nodes are 100



(그림 11) 노드수가 200일 경우, 전달지연율  
(Figure 11) Latency Average in the case that number of nodes are 200

그림 10과 11에서 버퍼의 수가 커질수록 전체적으로 메시지의 평균적인 전달지연율이 증가하는 것을 보인다. 노드가 200개일 경우, 버퍼의 크기가 15Mbyte 이상 커질 경우 **MaxProp** 라우팅이 가장 낮은 전달지연율을 보이며, **Binary** 기법, **Spray** 기법 그리고 **Hybrid** 기법은 이보다 큰 전달 지연율을 보이는 것을 확인하였다. 이는 경로 비용을 통해 빠른 메시지 전달을 계속적으로 시도하기 때문에 낮은 전달지연율을 보인다.

## 6. 결 론

**Spray and Wait** 라우팅과 **Binary Spray and Wait** 라우팅은 **Epidemic** 라우팅의 무분별한 복사본의 생성을 보완하였지만, 릴레이 노드를 선별하는 과정 없이 만나는 모든 노드를 릴레이 노드로 선택하여 메시지를 복사하는 과정만이 있었다. 이는 버퍼의 크기가 제한적일 때, 네트워크에 속해있는 모든 노드의 버퍼에 불필요한 메시지를 증가시켜 네트워크의 총 전달성공률에 잠재적으로 영향을 줄 수 있다. 이를 위해 본 논문에서는 노드 간의 히스토리 데이터를 토대로 특정한 릴레이 노드만을 선별하고, 선별된 노드에게 제안 알고리즘을 이용한 포워딩하는 방법을 제안하였다. 그 결과 모든 라우팅과 비교하였을 때, 다른 라우팅기법보다 높은 전달성공률을 보이며 **Binary Spray and Wait** 라우팅과 **Spray and Wait** 라우팅보다 전달성공률을 10% 높였음을 확인하였다. 이를 통해 히스토리 데이터를 통해 릴레이 노드를 선별하는 과정이 메시지가 전달되는 확률에 영향을 미칠 수 있음을 확인하고, 선별된 노드에게만 메시지 복사를 하는 과정이 효율적이라는 것을 확인하였다. 추후의 연구에서는 히스토리 데이터를 관리하는 방법에 대하여 연구가 이루어 질 것이다.

## 참 고 문 헌(Reference)

- [1] Delay Tolerant Networking research group  
<http://www.dtnorg.org>
- [2] A. Keranen, J.Ott, and T. Kakkainen, "The ONE Simulator for TEN Protocol Evaluation," *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Mar. 2009.
- [3] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks:



- Overview and Challenges,” *IEEE Communications Survey and Tutorial*, pp. 24-37, Jan. 2006.
- [4] Sang Ho So, Man Kyu Park, Se Chul Park, Jae Yong Lee, Byung Chul Kim, Dae Young Kim, Min Su Shin, Dae Ig Chang, and Ho Jin Lee, “Delay Tolerant Network Routing Algorithm based on the Mobility Pattern of Mobile Nodes,” *IEEK*, Vol.46 No.4, pp. 13-28, Apr. 2009.
- [5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Single-copy routing in intermittently connected mobile networks,” *In Proc. of IEEE Secon*, Apr. 2004.
- [6] Huai-En Lian, Chien Chen, Je-Wei Chang, Chien-Chung Shen, Rong-Hong Jan, “Shortest Path Routing with Reliability Requirement in Delay Tolerant Networks,” *Future Information Networks, 2009. ICFIN 2009. First International Conference on*, Oct. 2009.
- [7] T.Clausen and P.Jacquet, “Optimized Link State Routing Protocol(OLSR),” *RFC 3626*, IETF Network Working Group, Oct. 2003.
- [8] Spyropoulos, T., Psounis, K., and Raghavendra, C. S. “Single-copy routing in intermittently connected mobile networks,” *In Proc. Sensor and Ad Hoc Command Distance Vector (AODV) Routing*, RFC 3561, IETF Network Working Group, July 2000.
- [9] C.Perkins, E.Belding-Royer, and S.Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” *RFC 3561*, IETF Network Working Group, July 2000.
- [10] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks,” *In Proc. IEEE Infocom*, April 2006.
- [11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks,” *In Proc. ACM WDTN*, pages 252 - 259, Aug. 2005.
- [12] Eyuphan Bulut, Zijian Wang, Boleslaw K. Szymanski, “Cost Effective Multi-Period Spraying for Routing in Delay Tolerant Networks,” *IEEE/ACM Transactions on*, Oct. 2010.
- [13] M Hussein Mamoun, “Efficient DTN Routing Protocol”, *International Journal of Computer Applications pp. 0975-8887*, vol. 80, no. 9, Oct. 2013.
- [14] Feng Li and Jie Wu, “LocalCom: A Community-based Epidemic Forwarding Scheme in Disruption-tolerant Networks,” *IEEE Secon*, Apr. 2009.
- [15] P. Hui, J. Crowcroft, and E. Yoneki, “BUBBLE Rap: Social-Based Forwarding in Delay Tolerant Networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576 - 1589, Nov. 2011.
- [16] Etienne C. R. de Oliveira and C’ elio V. N. de Albuquerque, “NECTAR: A DTN Routing Protocol Based on Neighborhood Contact History,” *SAC ’09*. pp.8-12, Mar. 2009.
- [17] Guizhu WANG, Yingying YANG, Xin SONG, Jie WANG, Weiming LI and Dawei DING, “Research for Delay Tolerant Network Node Similarity-based on Social Network,” *JCIS*, pp.5189 - 5195, July 2013.
- [18] Rashid, S., Q. Ayub, M.S. Zahid and A.H. Abdullah, “E-DROP: An Effective Drop Buffer Management Policy for DTN Routing Protocols,” *International Journal of Computer Applications pp. 0975 - 8887*, vol. 13, no. 7, Jan. 2011.

◎ 저 자 소 개 ◎



**현 성 수**

2009년 인천대학교 컴퓨터공학과 학사과정  
관심분야 : 무선 인터넷 프로토콜, 라우팅 프로토콜  
E-mail : hyunss0331@incheon.ac.kr



**정 현 진**

2012년 인천대학교 경영학부, 컴퓨터공학과 졸업 (학사)  
2014년 인천대학교 컴퓨터공학과 졸업 (석사)  
2014년~현재 인천대학교 컴퓨터공학과 박사과정  
관심분야 : 무선 인터넷 프로토콜, 라우팅 프로토콜  
E-mail : oasishjj@incheon.ac.kr



**최 승 식**

1988년 연세대학교 전자공학과 졸업 (학사)  
1990년 KAIST 대학원 전기 및 전자공학과 졸업 (석사)  
2002년 KAIST 대학원 전기 및 전자공학과 졸업 (박사)  
1990년~2004년 KT서비스개발연구소 선임연구원  
2004년~현재 인천대학교 컴퓨터공학과 부교수  
관심분야 : 무선 엑세스제어, 무선자원관리, 무선 인터넷 프로토콜  
E-mail : sschoi@incheon.ac.kr