

지연 허용 네트워크에서 접촉 노드의 능력 값에 기반한 동적 TTL 기법

강민욱*, 정운원**

A Dynamic TTL Scheme based on Capacity of Contact Nodes in Delay-Tolerant Networking

Min Wook Kang*, Yun Won Chung**

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임
(No.2017-0-00613, MEC 환경에서의 콘텐츠 기반 지연 감내 네트워킹 기술 개발)
"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음"
(IITP-2020-2017-0-01633)

요 약

낮은 노드의 밀도로 인해 노드 간 안정적인 라우팅 경로의 제공이 어려운 환경에서 동작하는 지연 허용 네트워크에서 노드는 주위 노드와의 기회적 접촉을 통해 메시지를 전달한다. 지연 허용 네트워크에서는 제한된 노드의 배터리 전력 및 버퍼 공간 자원으로 인해 메시지의 TTL(time-to-live)을 효과적으로 설정해야 할 필요가 있다. 본 논문에서는 서로 다른 자원 특성을 가지는 일반 노드와 데이터 물로 구성된 지연 허용 네트워크 환경에서 노드의 버퍼 잔여량, 배터리 전력 잔여량, 목적지 노드로의 전달 예측률의 가중합으로 정의된 노드의 능력을 정의하고 두 접촉 노드의 능력 값에 기반하여 메시지의 TTL을 동적으로 설정하는 접촉 노드의 능력 값에 기반한 동적 TTL 기법을 제안한다. 성능 분석 결과 제안 기법은 정적 TTL 기법에 비해 전달지연이 증가하지만 대부분의 고려하는 파라미터 영역에서 우수한 전달률을 가지고 최대 19%의 전달률 향상을 확인하였다.

Abstract

In Delay Tolerant Network (DTN), where stable routing path between nodes may not be available due to the sparse node density, messages are delivered through opportunistic contacts with neighbor nodes. Therefore, it is necessary to set the value of time-to-live (TTL) effectively due to the restricted resources of battery power and buffer storage. In this paper, a dynamic TTL scheme based on the capacity of contact nodes is proposed, where TTL value is dynamically determined at message forwarding, based on the capacity value of two contact nodes, which is defined as a weighted sum of residual buffer storage, residual power, and delivery predictability to a destination node in a DTN network with general nodes and data mules with different resource characteristics. Performance analysis results showed that the proposed scheme has better delivery ratio in most of the considered parameter value ranges, with a maximum of 19% of improvement, although delivery latency is higher than a static TTL scheme.

Keywords

delay tolerant network, residual buffer storage, residual power, delivery predictability, delivery ratio, delivery latency

* 송실대학교 정보통신공학과 박사과정
- ORCID: <http://orcid.org/0000-0002-7475-8143>
** 송실대학교 정보통신공학부 교수(교신저자)
- ORCID: <http://orcid.org/0000-0002-2964-0330>

· Received: Nov. 12, 2020, Revised: Dec. 11, 2020, Accepted: Dec. 14, 2020
· Corresponding Author: Yun Won Chung
School of Electronic Engineering, Soongsil University, 369, Sangdo-Ro,
Dongjak-gu, Seoul, 06978, Korea
Tel.: +82-2-820-0908, Email: ywchung@ssu.ac.kr

1. 서 론

지연 허용 네트워크인 DTN(Delay Tolerant Network)은 낮은 노드 밀도로 인해 메시지가 발생된 소스 노드와 메시지의 목적지 노드 간 안정적인 라우팅 경로의 제공이 어려운 열악한 환경에서도 주위 노드와의 기회적(Opportunistic) 접촉을 통해 메시지를 효과적으로 전달하기 위해 제안된 네트워크이다[1][2]. 이를 위해 DTN에서는 전달 계층 위에 번들(Bundle) 계층을 정의하여 목적지 노드로의 라우팅 경로가 존재하지 않는 경우에도 메시지를 번들 계층에 저장하고 추후 주위 노드와 무선 통신 범위 안에 들어오는 기회적 접촉이 발생하면 접촉 노드에게 메시지를 전달하고, 이러한 휴 간 기회적 메시지 전송을 반복하여 수행함을 통해 최종 목적지 노드로 메시지를 전달하고자 한다.

가장 기본적인 DTN 라우팅 프로토콜은 Epidemic 프로토콜로 ‘전염병’이라는 뜻의 이름에서도 알 수 있듯이 메시지를 가진 노드는 접촉하는 모든 노드에게 메시지를 전달함으로써 최종 목적지 노드로 메시지 전달을 도모하는 플러딩(Flooding) 방식의 프로토콜이다[3]. Epidemic 프로토콜은 동작이 매우 간단한 장점이 있지만 많은 메시지 복사로 인해 트래픽 부하가 증가하는 단점이 있다. Spray & Wait 프로토콜은 새로운 메시지 발생 시 네트워크에 존재할 수 있는 총 메시지 복사본의 수를 L 개로 제한하는 프로토콜로 L 개의 메시지 복사본을 메시지 생성 노드를 포함하여 총 L 개의 노드에게 1개씩 확산하는 Spray 단계와 L 개의 노드가 목적지 노드를 만났을 때만 메시지를 전달하는 Wait 단계로 구성된다[4]. Spray & Wait 프로토콜은 메시지 복사본의 수를 제한하여 트래픽 부하를 줄일 수 있으나 Spray 단계에서 메시지 확산 시 무작위(Blind) 방식으로 분배하는 단점이 있다. PROPHET 프로토콜에서는 노드 간 접촉 이력을 이용해 두 노드 간 전달 예측률(Delivery predictability)을 계산하고 두 노드의 접촉 시 자신이 가지고 있는 메시지의 목적지 노드에 대해 상대 노드가 전달 예측률이 더 큰 경우 메시지를 전달한다[5]. PROPHET 프로토콜에서는 전달 예측률을 이용하여 메시지를 전달할 확률이 더 높

은 노드로 메시지를 전달함으로써 최종 목적지 노드로 메시지를 효과적으로 전달하고자 한다. PROPHET 프로토콜은 IRTF(Internet Research Task Force) RFC(Request for Comments) 6693 표준으로 제정된 프로토콜로 PROPHET 프로토콜의 성능 향상을 위한 다양한 연구가 진행되어 왔다[6]-[9].

DTN에서 이동 노드는 배터리 전력, 버퍼 크기 등의 자원이 제한적이다. 따라서 각 메시지는 TTL(Time-To-Live) 종료 시까지 최종 목적지 노드로 전달되지 않은 경우 버퍼에서 이를 삭제한다. 전술한 Epidemic, Spray & Wait, PROPHET 프로토콜에서 TTL은 통상 고정된 값을 가지는 것으로 가정하는데 TTL이 큰 경우 버퍼의 크기가 충분하지 않다면 버퍼 오버플로우(Overflow)가 발생하여 메시지가 삭제되고 전달률 또한 감소된다. 반면, TTL이 작은 경우 버퍼의 크기가 충분하더라도 TTL 종료로 인한 메시지 삭제가 자주 발생하여 전달률 또한 감소된다. 따라서, 효과적인 성능을 위해 TTL을 적절히 설정하는 것이 중요하다.

DTN에서 TTL 관련 연구를 간략히 살펴보면 [5]의 연구에서는 제한된 버퍼 자원을 효율적으로 사용하기 위해 버퍼에서 메시지를 삭제해야 하는 경우 잔여 TTL이 작은 메시지부터 우선적으로 삭제하는 기법을 제안하였다. [10]의 연구에서는 TTL 및 버퍼 크기를 변화시키면서 DTN 프로토콜의 전달률, 부하율, 전달지연에 미치는 영향을 분석하였다. [11]의 연구에서는 복사된 메시지의 분포 정도를 활용하여 메시지가 네트워크에 많이 분포되어 있을 경우 상대적으로 TTL을 더 감소시키는 기법을 제안하였고 성능 분석 결과 PROPHET 프로토콜에 제안 기법을 적용한 경우 전달률은 동일하고 버퍼 사용률은 약 0.03% 정도만 감소하는 것을 확인하였다.

본 논문에서는 전달 노드의 버퍼 사용량, 배터리 전력 사용량, 목적지 노드로의 전달 예측률과 같은 노드의 특성이 제대로 반영되지 않은 기존 연구의 문제점을 해결하기 위해 노드의 특성을 고려한 동적 TTL 기법을 제안한다. 특히, 일반 이동 노드보다 더 우수한 자원을 가지며 넓은 영역을 커버 가능한 차량 혹은 UAV(Unmanned Aerial Vehicle)와 같은 데이터 물(Data mule)[12]을 고려하고 데이터

물의 특성을 효과적으로 활용하여 TTL을 동적으로 설정함으로써 전달률을 향상시키고자 한다.

본 논문의 2장에서는 제안 기법에 대해 상세히 설명하고 3장에서는 제안 기법의 성능을 전달률 및 전달 지연 측면에서 분석한다. 마지막으로 4장에서는 본 논문의 결론 및 추후 연구를 제시한다.

II. 동적 TTL 기법

본 논문에서는 그림 1과 같이 보행자와 같은 일반 노드(*general node*) 및 트램과 같은 데이터 물로 구성된 DTN 환경을 가정한다. 데이터 물은 버퍼의 크기 및 배터리 전력이 충분한 노드이며, 반면, 일반 노드는 버퍼의 크기 및 배터리 전력이 제한된 특성을 가진다. 그림 1의 빨간색 화살표에서와 같이 노드는 메시지를 버퍼에 저장하고 이동하며 파란색 화살표로 표시된 노드 간 기회적 접촉 시 전달 기준을 만족하면 메시지를 접촉 노드에게 전달한다. 제안하는 동적 TTL 기법에서는 자원이 충분한 데이터 물에서는 일반 노드에 비해 메시지의 TTL을 크게 설정하여 목적지 노드로의 메시지 전달률을 높이고자 한다. 반면, 일반 노드에서는 버퍼 잔여량, 배터리 전력 잔여량 및 목적지 노드로의 전달 예측률 등을 적절히 고려하여 TTL을 설정함으로써 메시지 전달률을 높이고자 한다.

본 논문에서는 기본 DTN 프로토콜로 PROPHET 프로토콜을 가정하여 두 노드의 접촉 시 목적지 노드로 전달 예측률이 더 큰 노드로 메시지가 전달된다[5].

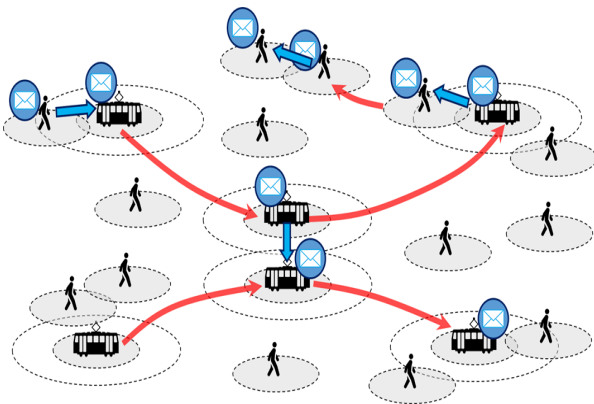


그림 1. 고려하는 DTN 환경
Fig. 1. A considered DTN environment

제안 기법에서 메시지는 일반 노드에서만 생성되는 것으로 가정하며 메시지 최초 생성 시 초기 TTL은 $TTL_{default}$ 로 설정하고 버퍼의 크기 및 배터리 전력이 제한된 특성을 가지는 일반 노드에서 너무 큰 TTL로 인한 버퍼 오버플로우 등의 문제 발생을 줄이기 위해 TTL 최대값은 $TTL_{default}$ 를 가정한다. 반면, 버퍼의 크기 및 배터리 전력이 충분한 데이터 물에서의 TTL 최대값은 TTL_{max} 를 가정한다. 노드 A는 자신의 잔여 버퍼 비율, 잔여 배터리 전력 비율, 목적지 노드 D로의 전달 예측률에 기반하여 노드 A의 목적지 노드 D로의 능력인 $Capacity(A,D)$ 를 식 (1)과 같이 계산한다.

$$Capacity(A,D) = W_B \times B(A) + W_E \times E(A) + W_P \times P(A,D) \quad (1)$$

$$W_B + W_E + W_P = 1 \quad (2)$$

상기 식에서 $B(A)$ 는 노드 A의 잔여 버퍼 비율, $E(A)$ 는 노드 A의 잔여 배터리 전력 비율, $P(A,D)$ 는 PROPHET 프로토콜에서 정의된 노드 A의 목적지 노드 D로의 전달 예측률[5], W_B 는 잔여 버퍼 비율의 가중치, W_E 는 잔여 배터리 전력 비율의 가중치, W_P 는 전달 예측률의 가중치에 해당한다. 이 때, 각 가중치는 0에서 1까지의 값을 가지고 가중치의 합은 1로 가정한다. 따라서 $Capacity(A,D)$ 는 0에서 1까지의 값을 가지게 된다. 두 노드는 접촉 시 자신이 가지고 있는 메시지에 대해 목적지 노드로의 전달 예측률을 비교하여 상대 노드가 전달 예측률이 더 큰 경우 메시지를 전달한다. 이 때, 수신되는 메시지의 TTL은 능력 값에 기반하여 동적으로 설정되는데 표 1과 같이 송신 노드 및 수신 노드의 종류에 따라 총 4가지 경우에 대해 동적으로 설정된다.

일반 노드 B가 일반 노드 A로부터 목적지 노드 D인 메시지 m 을 수신하는 경우 일반 노드 A의 현재 TTL인 $TTL(A,D)_m$ 에 노드 A의 목적지 노드 D에 대한 능력에 대한 노드 B의 목적지 노드 D에 대한 능력의 비를 곱한 값으로 $TTL(B,D)_m$ 를 설정한다. 단, 수신된 메시지의 TTL 최대값은 일반 노드의 최대 TTL 값인 $TTL_{default}$ 를 넘지 않도록 하기 위해 상기 결정된 $TTL(B,D)_m$ 와 $TTL_{default}$ 중 더 작은 값을 최종 $TTL(B,D)_m$ 로 설정한다.

표 1. 동적 TTL 설정

Table 1. Setting of dynamic TTL

Transmitting node A	Receiving node B	TTL of receiving node B
General node	General node	$TTL(B,D)_m = \min \left(TTL_{default}, TTL(A,D)_m \times \frac{Capacity(B,D)}{Capacity(A,D)} \right)$ (3)
General node	Data mule	$TTL(B,D)_m = \min \left(TTL_{max}, TTL(A,D)_m \times \frac{TTL_{max}}{TTL_{default}} \right)$ (4)
Data mule	General node	$TTL(B,D)_m = \min \left(TTL_{default}, TTL(A,D)_m \times \frac{TTL_{default}}{TTL_{max}} \times \frac{Capacity(B,D)}{Capacity(A,D)} \right)$ (5)
Data mule	Data mule	$TTL(B,D)_m = \min \left(TTL_{max}, TTL(A,D)_m \times \frac{Capacity(B,D)}{Capacity(A,D)} \right)$ (6)

데이터 물 B가 일반 노드 A로부터 목적지 노드 D인 메시지 m을 수신하는 경우 $TTL_{default}$ 에 대한 현재 일반 노드 A에서의 메시지 TTL인 $TTL(A,D)_m$ 의 비를 데이터 물에서 정의 가능한 최대 TTL 값인 TTL_{max} 와 곱한 값으로 $TTL(B,D)_m$ 를 설정한다. 이 경우 수신된 메시지의 TTL 최대값은 TTL_{max} 를 넘지 않도록 하기 위해 상기 결정된 $TTL(B,D)_m$ 와 TTL_{max} 중에서 더 작은 값을 $TTL(B,D)_m$ 로 설정한다.

일반 노드 B가 데이터 물 A로부터 목적지 노드 D인 메시지 m을 수신한 경우 현재 데이터 물 A에서의 메시지 TTL인 $TTL(A,D)_m$ 의 TTL_{max} 에 대한 비와 $TTL_{default}$ 를 곱하고 이후 노드 A의 목적지 노드 D에 대한 능력에 대한 노드 B의 목적지 노드 D에 대한 능력의 비를 곱하여 $TTL(B,D)_m$ 를 설정한다. 단, 수신된 메시지의 TTL 최대값은 일반 노드의 최대 TTL 값인 $TTL_{default}$ 를 넘지 않도록 하기 위해 상기 결정된 $TTL(B,D)_m$ 와 $TTL_{default}$ 중 더 작은 값을 $TTL(B,D)_m$ 로 설정한다.

데이터 물 B가 데이터 물 A로부터 목적지 노드 D인 메시지를 수신한 경우 현재 데이터 물 A에서의 메시지 TTL인 $TTL(A,D)_m$ 에 노드 A의 목적지 노드 D에 대한 능력에 대한 노드 B의 목적지 노드 D에 대한 능력의 비를 곱한 값으로 $TTL(B,D)_m$ 를 설정한다. 이 경우 수신된 메시지의 TTL 최대값이 TTL_{max} 를 넘지 않도록 하기 위해 상기 결정된 $TTL(B,D)_m$ 와 TTL_{max} 중 더 작은 값을 $TTL(B,D)_m$ 로 설정한다.

표 2. 실험 환경

Table 2. Simulation environment

Parameter	Value
Simulation time (s)	43,200
Router	PRoPHET router
Type of interfaces	General interface high speed and long range interface
Transmission range of interfaces (m)	General interface: 10 High speed and long range interface: 100
Packet transmission speed of interfaces (Mbytes/s)	General interface: 0.25 High speed and long range interface: 10
Node interface	Pedestrians: general interface Tram: general interface, high speed and long range interface
Movement model	Pedestrians: Shortest path map based m movement Tram: Map route movement
Number of hosts	Pedestrian: 150 Tram: 6
Initial battery energy (KJ)	Pedestrian: 1~10 (default: 5) Tram: 50
Scanning and scanning response energy (J/s)	0.1
Transmission energy (J/s)	0.5
Basic consumption energy (J/s)	0.01
Buffer size (Bytes)	Pedestrian: 5M~50M (default: 10M) Tram: 100M
$TTL_{default}$ (min)	30~300 (default: 120)
TTL_{max} (min)	$2 * TTL_{default}$
Message size (MBytes)	$U[0.25,0.75] \sim U[2.5,3.0]$ (default: $U[0.5,1.0]$)
Message interval (s)	$U[5,15] \sim U[50,60]$ (default: $U[25,35]$)

III. 성능 분석

본 논문에서는 표 2의 파라미터 환경에서 제안 프로토콜 및 정적 TTL을 가정한 기존 PProPHET 프로토콜의 성능을 ONE(Opportunistic Network Environment) 시뮬레이터[13],[14]를 사용하여 식 (7), (8)에서 정의된 전달률 및 전달지연의 측면에서 비교하였다.

$$\text{전달률} = \frac{\text{전달 메시지 수}}{\text{생성 메시지 수}} \quad (7)$$

$$\text{전달지연} = \frac{\text{전달 메시지의 전달지연 시간 총합}}{\text{전달된 메시지 수}} \quad (8)$$

그림 2는 일반 노드의 TTL_{default}를 변화시키면서 전달률을 비교한 그래프이다. 두 기법 모두 TTL_{default} 값이 30분에서 120분까지 증가하는 경우 전달률이 증가하고 TTL_{default}가 120분 이상에서 증가하는 경우 전달률이 감소하는 것을 볼 수 있다. TTL_{default}가 30분에서 120분까지 증가하는 경우 노드의 버퍼에서 TTL 종료로 인해 삭제되는 메시지가 감소하고 메시지가 버퍼에서 보존되는 시간이 증가하여 전달률이 증가한다. 그러나 TTL_{default}가 120분 이상 증가하는 경우 노드 버퍼의 수용 한계로 인해 메시지 오버플로우가 발생하여 메시지 전달률은 감소하게 된다.

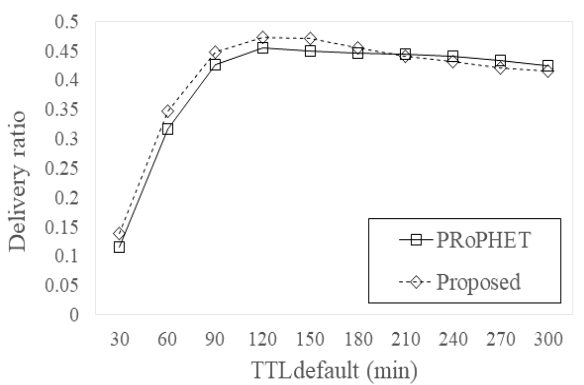


그림 2. TTL_{default} 변화에 따른 전달률
Fig. 2. Delivery ratio for varying TTL_{default}

제안 기법은 효과적인 TTL 설정을 통해 노드의 특성에 따라 TTL을 적절히 설정함으로써 TTL_{default}가 매우 큰 경우를 제외한 대부분의 TTL_{default}에서 기존 기법보다 우수한 전달률을 보이며 최대 19%의 성능 향상을 가지는 것을 확인할 수 있다.

그림 3은 그림 2의 환경에서 전달지연을 비교한 그래프로 두 기법 모두 TTL_{default}가 증가함에 따라 전달지연이 증가하는 것을 볼 수 있다. 이는 TTL_{default} 값이 증가함에 따라 버퍼에 저장되는 메시지의 수가 증가하고 이로 인해 두 노드의 접촉 시 임의의 메시지가 전달될 확률이 감소하여 결과적으로 최종 목적지로 전달되는데 걸리는 시간이 증가하게 되기 때문이다. 제안 기법은 기존 기법에 비해 데이터 물에서 메시지의 TTL 값이 증가하게 되어 기존 기법에 비해 전달 지연이 증가하는 것을 볼 수 있다.

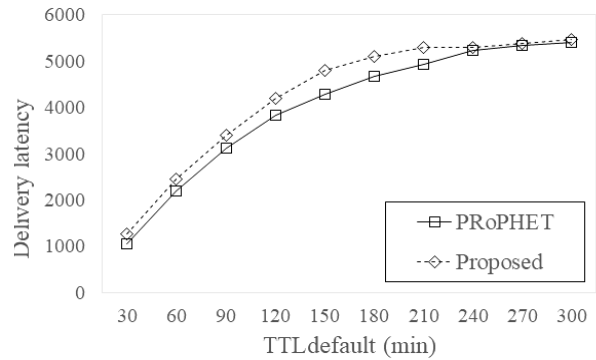


그림 3. TTL_{default} 변화에 따른 전달지연
Fig. 3. Delivery latency for varying TTL_{default}

그림 4는 일반 노드의 버퍼 크기를 변화시키면서 전달률을 비교한 그래프이다. 두 기법 모두 버퍼의 크기가 작은 영역에서는 버퍼의 크기가 증가함에 따라 더 많은 메시지를 버퍼에서 수용 가능하여 전달률이 증가하게 된다. 반면, 버퍼의 크기가 일정 값 이상인 경우에는 버퍼에서 수용 가능한 메시지가 증가하더라도 두 노드의 접촉 시 전달 가능한 메시지의 수는 인터페이스의 전송 속도에 의해 제한되기 때문에 전달률은 더 이상 증가하지 않고 수렴하는 것을 볼 수 있다. 제안 기법은 고려하는 파라미터 환경에서 기존 기법에 비해 모든 버퍼 크기에서 더 우수한 전달률을 가지며 최대 7%의 성능 향상을 가지는 것을 알 수 있다.

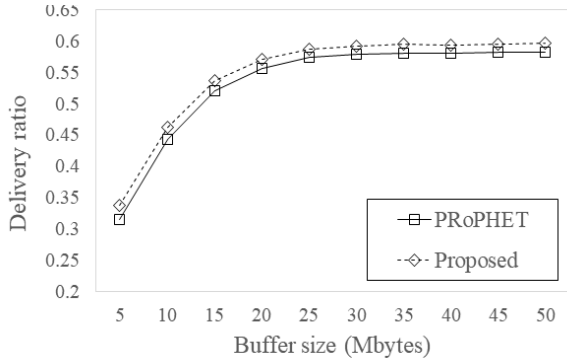


그림 4. 버퍼 크기 변화에 따른 전달률
Fig. 4. Delivery ratio for varying buffer size

그림 5는 그림 4의 환경에서 전달지연을 비교한 그래프로 두 기법 모두 일반 노드의 버퍼 크기가 작은 값에서 증가함에 따라 전달지연이 약간 증가하다가 버퍼의 크기가 일정 값 이상에서는 전달지연이 수렴하는 것을 볼 수 있다. 버퍼의 크기가 작은 영역에서 버퍼의 크기가 증가하는 경우 버퍼에서 수용 가능한 메시지의 수가 증가하여 제한된 무선 인터페이스 전송 환경에서 임의의 메시지를 성공적으로 목적지 노드까지 전달하는데 걸리는 시간은 증가하게 된다. 버퍼의 크기가 충분히 큰 경우에는 버퍼의 크기가 증가하더라도 버퍼에 저장되는 메시지 수는 큰 변화가 없고 따라서, 전달 지연 또한 변화가 적은 것을 알 수 있다. 제안 기법은 기존 기법에 비해 좀 더 큰 전달 지연을 가지는 것을 알 수 있다.

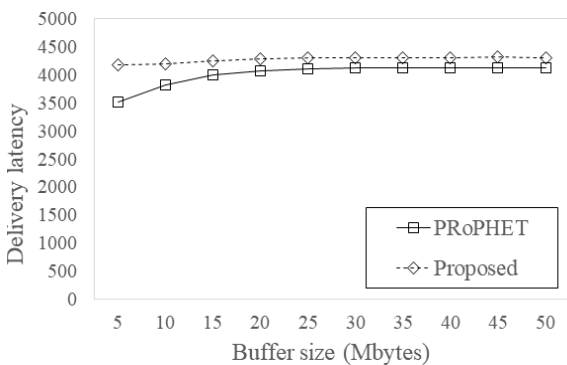


그림 5. 버퍼 크기 변화에 따른 전달지연
Fig. 5. Delivery latency for varying buffer size

그림 6은 메시지 크기를 변화시키면서 전달률을 비교한 그래프이다. 두 기법 모두 메시지 크기가 증가함에 따라 전달률이 감소하는데 이는 버퍼의 크기 및 메시지 발생 간격이 고정된 환경에서 메시지

크기가 증가하는 경우 버퍼 오버플로우가 증가하여 삭제되는 메시지가 증가하고 버퍼에 저장될 수 있는 메시지의 수 또한 감소하기 때문이다. 제안 기법은 고려하는 파라미터 환경에서 기존 기법에 비해 모든 메시지 크기에서 더 우수한 전달률을 가지며 최대 8%의 성능 향상을 가지는 것을 알 수 있다.

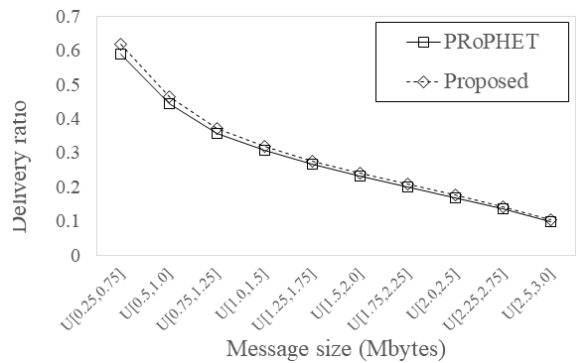


그림 6. 메시지 크기 변화에 따른 전달률
Fig. 6. Delivery ratio for varying message size

그림 7은 그림 6의 환경에서 전달지연을 비교한 그래프로 두 기법 모두 대부분의 고려하는 메시지 크기에서 메시지의 크기가 증가함에 따라 전달 지연은 다소 감소하는 것을 알 수 있으며 제안 기법은 기존 기법에 비해 좀 더 큰 전달지연을 가지는 것을 알 수 있다.

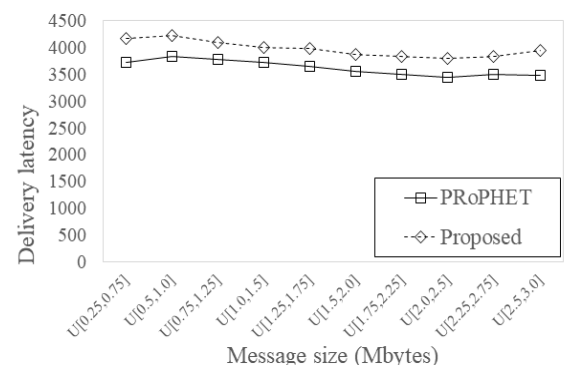


그림 7. 메시지 크기 변화에 따른 전달지연
Fig. 7. Delivery latency for varying message size

그림 8은 메시지 발생 간격을 변화시키면서 전달률을 비교한 그래프이다. 두 기법 모두 메시지 발생 간격이 증가함에 따라 전달률이 증가하는데 이는 제한된 버퍼 크기 환경에서 메시지 발생 간격이 증가함에 따라 버퍼 오버플로우가 감소하고 각 메시지가 전달될 수 있는 기회는 증가하기 때문이다. 제

안 기법은 고려하는 파라미터 환경에서 기존 기법에 비해 모든 메시지 발생 간격에서 더 우수한 전달률을 가지며 최대 5%의 성능 향상을 가지는 것을 알 수 있다.

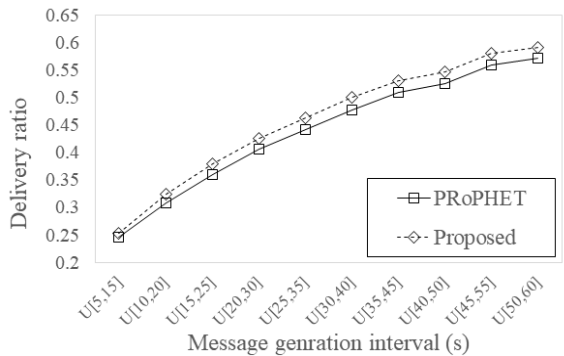


그림 8. 메시지 발생 간격 변화에 따른 전달률
Fig. 8. Delivery ratio for varying message generation interval

그림 9는 그림 8의 환경에서 전달지연을 비교한 그래프로 두 기법 모두 대부분의 고려하는 메시지 발생 간격에서 메시지 발생 간격이 증가함에 따라 전달지연이 다소 증가하는 것을 알 수 있으며 제안 기법은 기존 기법에 비해 좀 더 큰 전달지연을 가지는 것을 알 수 있다.

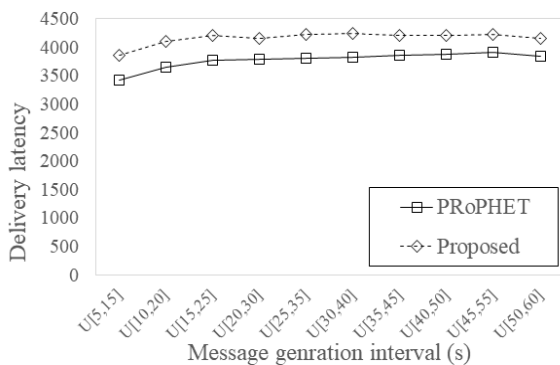


그림 9. 메시지 발생 간격 변화에 따른 전달지연
Fig. 9. Delivery latency for varying message generation interval

그림 10은 일반 노드의 초기 배터리 에너지를 변화시키면서 전달률을 비교한 그래프이다. 두 기법 모두 초기 배터리 에너지가 1KJ에서 5KJ까지 증가하는 경우 배터리 에너지가 모두 소비되어 동작하지 못하는 노드의 수가 감소하게 되어 전달률은 증

가하는 반면, 초기 배터리 에너지가 5KJ 이상인 경우 고려하는 시뮬레이션 시간 동안 대부분의 노드가 생존하여 전달률 유사한 값을 가지는 것을 알 수 있다. 제안 기법은 고려하는 파라미터 환경에서 기존 기법에 비해 모든 초기 배터리 에너지 크기에 더 우수한 전달률을 가지며 기존 기법에 비해 최대 5%의 성능 향상을 가지는 것을 알 수 있다.

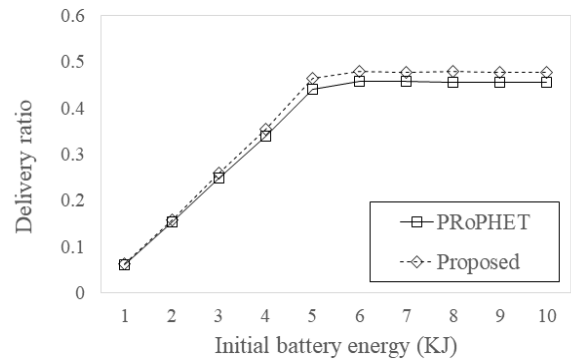


그림 10. 초기 배터리 에너지 변화에 따른 전달률
Fig. 10. Delivery ratio for varying initial battery energy

그림 11은 그림 10의 환경에서 전달지연을 비교한 그래프로 두 기법 모두 초기 배터리 에너지가 작은 구간에서 초기 배터리 에너지가 증가하는 경우 전달지연이 증가하고 이후 구간에서 유사한 전달지연을 가지는 것을 알 수 있다. 초기 배터리 에너지가 작은 구간에서 초기 배터리 에너지가 증가하면 메시지 전달에 참여하는 노드의 수가 증가하여 버퍼에 저장되는 메시지의 수가 증가하고 메시지 전달지연은 증가하게 된다. 제안 기법은 기존 기법에 비해 좀 더 큰 전달지연을 가지는 것을 알 수 있다.

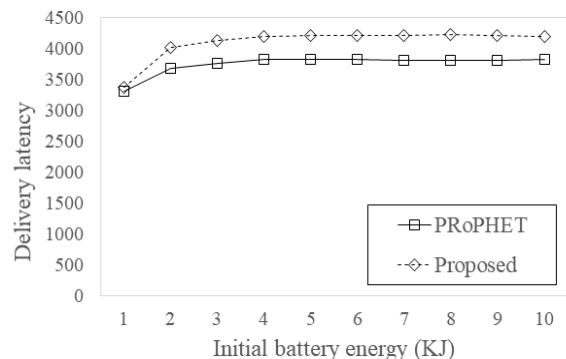


그림 11. 초기 배터리 에너지 변화에 따른 전달지연
Fig. 11. Delivery latency for varying initial battery energy

IV. 결론 및 추후 연구

본 논문에서는 서로 다른 자원 특성을 가지는 일반 노드와 데이터 플로 구성된 네트워크 환경에서 버퍼 잔여량, 배터리 전력 잔여량, 목적지 노드로의 전달 예측률의 가중합으로 정의된 임의의 노드와 노드의 목적지 노드 간에 정의된 능력을 정의하고 두 노드의 접촉 시 노드의 능력 값에 따라 수신된 메시지의 TTL을 동적으로 설정하는 접촉 노드의 능력 값에 기반한 동적 TTL 기법을 제안하였다. 일반 노드의 $TTL_{default}$, 버퍼 크기, 메시지 크기, 메시지 발생 간격, 초기 배터리 에너지를 변화시키면서 수행한 성능 분석 결과 제안 기법은 정적 TTL 기법에 비해 높은 전달지연을 보이지만 대부분의 고려하는 파라미터 영역에서 높은 전달률을 가지고, 최대 19%의 전달률 향상을 확인하였다.

DTN은 기본적으로 지연을 허용하는 서비스를 대상으로 제안된 네트워크로 전달지연의 허용도가 일반 네트워크보다 더 높은 특성을 가지며, 따라서 전달지연이 기존 기법에 비해 다소 높더라도 우수한 전달률을 가지는 제안 기법은 DTN 환경에 보다 더 적합한 기법이라 할 수 있다. 추후 노드의 자원 특성 뿐 아니라 변화하는 네트워크 트래픽에 따라 메시지의 TTL을 동적으로 설정하는 기법을 추가로 제안하여 본 논문에서 제안하는 동적 TTL 기법의 성능을 보다 향상시키는 연구를 진행하고자 한다.

References

- [1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, and K. Scott, "Delay-Tolerant Networking: An Approach to Interplanetary Internet", *IEEE Communications Magazine*, Vol. 41, No. 6, pp. 128-136, Jun. 2003.
- [2] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges", *IEEE Communications Survey and Tutorial*, Vol. 8, No. 1, pp. 24-37. Jan. 2006.
- [3] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance Modeling of Epidemic Routing", *Computer Networks*, Vol. 51, No. 10, pp. 2867-2891, Jul. 2007.
- [4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks", *Proceedings of the ACM SIGCOMM Workshop on Delay-tolerant Networking*, Philadelphia Pennsylvania USA, pp. 252-259, Aug. 2005.
- [5] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks", *IETF RFC 6693*, Aug. 2012.
- [6] H. H. Lim, D. Y. Seo, and Y. W. Chung, "An Improved PRoPHET Routing Protocol through the Restriction of Message Duplication", *Journal of KIIT*, Vol. 14, No. 6, pp. 81-86, Jun. 2016.
- [7] S. K. Pandey and A. K. Singh, "Efficient PRoPHET with Buffer Management for Multicasting in DTN", *Proceedings of International Conference on Inventive Research in Computing Applications*, Coimbatore, India, pp. 1200-1205, Jul. 2018.
- [8] Z. Du, C. Wu, T. Yoshinaga, and Y. Ji, "A PRoPHET-based DTN Protocol for VANETs", *Proceedings of Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, Guangzhou, China, pp. 1876-1879, Oct. 2018.
- [9] Y. Mao, C. Zhou, Y. Ling, and J. Lloret, "An Optimized Probabilistic Delay Tolerant Networks (DTN) Routing Protocol based on Scheduling Mechanism for Internet of Things (IoT)", *MDPI Sensors*, Vol. 19, No. 2, 2019. <https://doi.org/10.3390/s19020243>
- [10] S. Hossen, M. Billah, and S. Yasmin, "Impact of Buffer Size and TTL on DTN Routing Protocols

in Intermittently Connected Mobile Networks", International Journal of Engineering & Technology, Vol. 7, No. 3, pp. 1735-1739, 2018. <https://doi.org/10.14419/ijet.v7i3.14122>

- [11] J. Kim, D. Chang, T. T. Kwon, and Y. Choi, "A Study on Manipulating TTL of DTN for Efficient Buffer Management", KICS Annual Fall Conference, 2010.
- [12] D. Palma, A. Zolich, Y. Jiang, and T. A. Johansen, "Unmanned Aerial Vehicles as Data Mules: An Experimental Assessment", IEEE Access, Vol. 5, pp. 24716-24726, Nov. 2017.
- [13] The Opportunistic Network Environment Simulator, <http://www.netlab.tkk.fi/tutkimus/dtn/theoone/>, [accessed: Nov, 11, 2020]
- [14] A. Keranen, J. Ott, and T. Karkkainen, "The ONE Simulator for DTN Protocol Evaluation", Proceedings of International Conference on Simulation Tools and Techniques, pp. 55:1-10, Mar. 2009. <https://doi.org/10.4108/ICST.SIMUTOOLS2009.5674>

정 윤 원 (Yun Won Chung)



1995년 2월 : 한국과학기술원
전기및전자공학과(공학사)
1997년 2월 : 한국과학기술원 전기
및 전자공학과(공학석사)
2001년 8월 : 한국과학기술원 전기
및 전자공학과(공학박사)
2001년 10월 ~ 2002년 12월 :

King's College London Visiting Post-doctoral
Research Fellow

2003년 1월 ~ 2005년 8월 : 한국전자통신연구소 연구원
2005년 9월 ~ 현재 : 숭실대학교 전자정보공학부 교수
관심분야 : 이동통신 네트워크, 성능 분석, 이동성 관리,
DTN, ICN, SDN/NFV

저자소개

강 민 옥 (Min Wook Kang)



2015년 2월 : 숭실대학교
정보통신전자공학부(공학사)
2017년 2월 : 숭실대학교
정보통신공학과(공학석사)
2017년 3월 ~ 현재 : 숭실대학교
정보통신공학과(박사과정)
관심분야 : DTN, ICN, SDN/NFV