

# SDDC BAS에 적합한 라우팅 알고리즘의 설계

## Design of Routing Algorithm for SDDC BAS

김정욱, 부창진, 이용호, 최윤기, 정진우, 김호찬\*  
(Jeong-Uk Kim<sup>1</sup>, Chang-Jin Boo<sup>2</sup>, Yongho Lee<sup>1</sup>, Yunki Choi<sup>1</sup>, Jinoo Joung<sup>1</sup>, and Ho-Chan Kim<sup>2</sup>)

<sup>1</sup>Sangmyung University

<sup>2</sup>Jeju National University

**Abstract:** In this paper, we propose an effective routing protocol that can be particularly applied to SDDC (Sangmyung Direct Digital Controller) BAS. We have used NS (Network Simulator)-2 to simulate the real building for SDDC BAS. The static, session, DV (Distance Vector), and LS (Link State) algorithms are used and the protocols for performance comparison and evaluation are compared with respect to end to end delay, throughput, and routing overhead. The simulation results show that the DV algorithm is effective for SDDC BAS.

**Keywords:** BAS, NS-2, BACnet, routing algorithm

### I. 서론

최근 신축되고 있는 많은 대형 건물들은 쾌적한 사무 환경을 제공하는 동시에 업무 능력을 향상시키고 효과적으로 건물의 운용을 도모할 목적으로 빌딩 자동화 시스템 BAS (Building Automation Systems)를 도입하고 있다. 빌딩 자동화 시스템은 고도의 기술이 집적된 제어기술과 다양한 기능의 수행을 위하여 분산화 및 모듈화가 이루어지고 있으므로, 이에 따라 네트워크의 역할이 중요해지고 있다. 고도의 안전성과 실시간성을 요구하는 빌딩 자동화 시스템에서 네트워크의 실시간성은 전체 제어시스템의 성과와 직결되며, 네트워크 프로토콜의 성능분석은 빌딩 내 네트워크 설계에 필수적인 요소이다.

BAS의 목적은 빌딩에서 활동하는 사람에게 쾌적하고 안전한 환경을 조성하고 빌딩의 운전경비를 최소화하려는 상반된 목적을 실현하고자 발생한 시스템이다[1]. 최근에는 전 세계 전력 중 23% 가량이 상업용 건물에서 이루어지고 있기 때문에, 빌딩 자동화 시스템은 에너지 관리를 위한 중요 분야로 주목받고 있다. 빌딩 자동화 시스템은 빌딩의 안정성과 효율성을 책임지는 핵심기술로 자리잡아 왔고 수요 또한 가파르게 상승하고 있는 추세다[2].

초기의 BAS는 DDC (Direct Digital Controller) 기술을 도입하여 건물 내의 다양한 장비를 한 회사가 제어하였으나

현재는 다양한 제조업체들이 자체적으로 보유한 독자적인 프로토콜을 상호 연계하는 경우가 많이 발생하고 있다. 이는 BAS의 확장과 기술발전 및 개선을 가로막는 중요한 장애물이 되어 왔다. 이를 해결하고자 개발된 통신 프로토콜이 BACnet (Building Automation and Control Network)이다. BACnet은 Ethernet, ARCNet, MS/TP (Master Slave/Token Passing), PTP (Point to Point), LonTalk 등의 LAN 표준에 기반을 두고 있으므로 제공업체 및 사용자 모두에게 이점을 준다[3,4].

본 논문에서는 클라우드 기반의 구조를 갖는 새로운 BAS인 SDDC (Sangmyung Direct Digital Controller)를 제안한다. NS-2 (Network Simulator version 2) 시뮬레이터를 이용하여 새로운 BAS 시스템에 기존의 IP 라우팅 프로토콜을 적용하여 구성하고, 다양한 라우팅 프로토콜의 성능 비교 및 평가를 통해 구현기술의 유효성을 검증하고자 한다.

### II. 관련 연구

#### 1. BAS 시스템 설계

BAS는 건물의 냉난방 계통 운전을 담당하는 기계설비, 건물의 전력 및 조명을 담당하는 전기설비, 건물의 출입통제 및 방범 계통 운전을 담당하는 방범설비, 건물의 화재 및 재난 계통 운전을 담당하는 방재설비, 그리고 건물의 자동차 출입통제 및 운영 계통 운전을 담당하는 주차관제 등의 다양한 시스템으로 구성된다. 건축물을 제외한 전기 기계 설비를 효율적으로 운영하여 재실자에게는 최적의 근무 환경을 제공하고, 관리자에게는 체계적인 운영으로 시간절감, 장비수명연장, 에너지절감 기능을 제공하게 된다.

IT 분야의 발전에 따라 여러 가지 형태의 컴퓨터 시스템들이 등장하였고, 최근에는 강력한 처리 능력을 바탕으로 중앙 집중화 형태의 클라우드 컴퓨팅 하드웨어 및 소프트웨어 기술이 발전하고 있지만 BAS 시스템은 1980년대부터 지금까지 동일한 구조를 유지하고 있다. BAS의 Automation

\* 책임저자(Corresponding Author)

논문접수: 2012. 8. 27., 수정: 2012. 9. 14., 채택확정: 2012. 9. 25.

김정욱: 상명대학교 에너지그리드학과(jukim@smu.ac.kr)

부창진, 김호찬: 제주대학교 전기공학과

(boo1004@jeju.ac.kr/hckim@jeju.ac.kr)

이용호, 최윤기, 정진우: 상명대학교 컴퓨터공학과

(yhlee512@gmail.com/filterk7@gmail.com/jjoung@smu.ac.kr)

※ 본 논문은 지식경제부 산업융합원천기술개발사업으로 지원된 연구결과임[10041740, 빌딩내 기기들을 웹을 통해 연동하여 사용자 맞춤형 실시간 최적제어 모니터링 서비스를 제공하는 소프트웨어 개발].

Layer와 Field Layer는 배관·배선 비용을 절감하기 위하여 멀티드롭 방식의 RS-485 방식을 채택하였으나, 최근에는 이더넷 방식도 등장하고 있다. RS-485는 속도가 낮아 관제점 수량이 많아지면 급격한 성능 저하가 발생한다[5].

그림 1이 BAS에서 RS-485 배선을 사용하는 대표적인 사례로서, 시스템 내 관제점 수가 많아지면 급격한 성능 저하가 발생하는 확장성(scalability)의 문제가 발생한다. 또한, 건물설비의 변경이 발생하면 저속의 RS-485를 이용하여 변경된 관제점 정보를 활용하는 모든 DDC에 관제점 변경 정보를 다운로드해야 하는 문제점이 있다.

그림 2의 SDDC BAS는 기존의 빌딩 자동화 시스템의 문제점을 개선하기 위하여 제안된 시스템이다. 기존 RS-485 기반 빌딩 자동화 시스템의 문제점을 극복하기 위하여 SDDC BAS에서는 각 현장 제어장치는 메쉬 네트워크(mesh network) 기반 구조를 가지며, RS-485가 아닌 이더넷 배선 사용을 제안한다. 또한 임베디드 컨트롤러(embedded controller)로 구현된 현장 제어장치는 이더넷 포트(ethernet-port)를 4개 장착하여 RS-485의 멀티드롭(multi-drop) 방식과 동일한 효과를 가질 수 있게 하였다. 이렇게 제안된 빌딩 자동화 시스템의 구조는 기존 시스템보다 배관 및 배선을 30% 이상 절감할 수 있으며 확장성 또한 용이하다.

본 논문에서는 유선 메시 기반 아키텍처에서 경량 RTOS 및 웹서비스(web service)를 지원하는 기계장치와 전기장비,

조명장비를 통합 운영하는 BAS이며, 임베디드 컨트롤러에 이더넷 포트를 4개 장착하여 RS-485의 멀티드롭과 동일한 효과를 가질 수 있다. 또한 웹서비스 기술 및 무선기술을 채택함으로써 BAS 시스템의 운영상 편의성을 높이고자 한다.

2. 라우팅 알고리즘

라우팅 알고리즘(routing algorithm)의 목표는 주어진 네트워크 토폴로지에서 모든 목적지로의 가장 좋은 경로를 찾는 것이다. 여기서 가장 좋은 경로란 네트워크상의 모든 노드 페어(node pairs)에서 데이터 통신망 링크를 통과하는 비용의 합이 가장 작은 경로를 의미한다. 여기서 링크 비용(metric)은 요구하는 서비스 유형에 따라 다양하게 설정할 수 있다. 다수의 노드가 포함된 네트워크에서의 라우팅 알고리즘은 가장 좋은 경로를 찾는 것 뿐만 아니라 편리하고 신속한 라우팅 테이블의 설정 및 갱신과 라우팅을 위한 트래픽(라우팅 오버헤드)을 최소화시키는 점 또한 매우 중요하다.

라우팅 방식은 크게 정적 라우팅(static routing) 방식과 동적 라우팅(dynamic routing) 방식으로 나뉜다. 정적 라우팅 방식은 네트워크 관리자가 라우팅 테이블을 직접 설정하고 관리, 갱신하는 방식으로 별도의 라우팅 프로토콜을 사용하지 않아 라우팅 오버헤드가 발생하지 않는 장점이 있지만 네트워크의 추가적인 확장 및 노드와 링크의 오작동에 적절히 대응할 수 없다.

동적 라우팅 방식은 노드들이 라우팅 프로토콜을 사용하여 주기적으로 네트워크의 상태를 체크하여 관리자의 개입 없이 네트워크 상황 변화에 따라 라우팅 테이블이 관리된다. 동적 라우팅 방식은 글로벌 라우팅(global routing) 방식과 분산 라우팅(distributed routing) 방식으로 나뉜다. 글로벌 라우팅 방식은 각 라우터가 네트워크 전체에 대한 정보를 가지고 라우팅 테이블을 갱신하는 방식이다. 각 라우터가 새롭게 인지한 네트워크 상태정보를 다른 라우터에 전파함으로써 모든 라우터가 전체 네트워크의 상태정보를 이용해 라우팅 테이블을 갱신한다. 분산 라우팅 방식은 각 라우터가 자신과 인접한 라우터들의 부분적 네트워크 정보를 이용하여 라우팅 테이블을 갱신하고 갱신된 라우팅 테이블을 인접 라우터에게 전달하여 전체적인 라우팅 테이블을 구성한다. NS-2에서 제공하는 라우팅 알고리즘의 이름과 방식은 다음과 같다[6].

1) Static 알고리즘: NS-2의 디폴트 라우팅 알고리즘으로써 다익스트라의 ‘All-Pair SPF (Shortest Path First) Algorithm’을 이용한다. 이 알고리즘은 한 노드에서 다른 모든 노드로의 최단경로를 구하는 알고리즘이다. 각 노드는 토폴로지에 있는 모든 링크에 대한 인접 매트릭스(adjacent matrix)와 링크코스트(link cost)를 이용하여 경로를 계산한다. NS-2에서 Static 알고리즘의 경로 계산은 시뮬레이션이 시작할 때 오직 한번만 계산한다.

2) Session 알고리즘: Session 알고리즘은 Static 알고리즘과 유사하게 시뮬레이션 시작 시, 다익스트라의 ‘All-Pair SPF Algorithm’을 이용하여 경로를 계산한다. 다른점이 있다면 Session 알고리즘은 시뮬레이션 동안 링크나 노드의

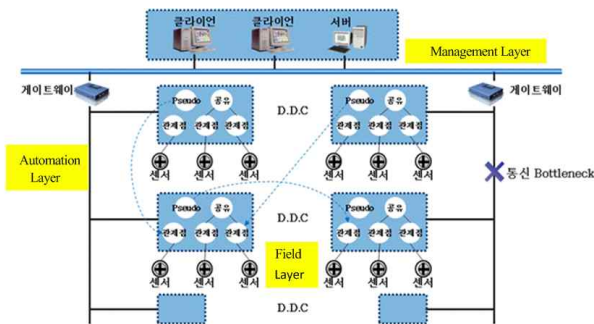


그림 1. 기존 BAS 시스템 구조.  
Fig. 1. Architecture of building automation systems.

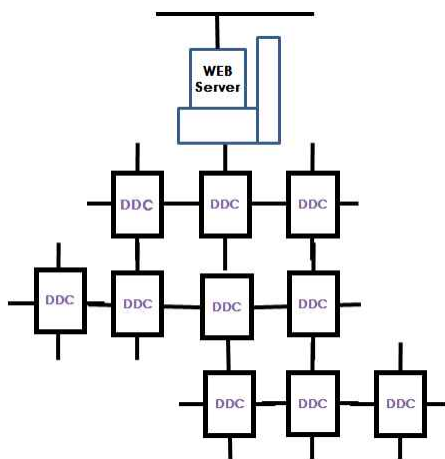


그림 2. SDDC BAS 시스템 구조.  
Fig. 2. Architecture of SDDC building automation systems.

단절 또는 토폴로지의 변경이 생기면 경로 재계산 및 복구 가 수행된다. 즉 Session 알고리즘은 어떤 노드가 전체 토폴로지에 연결되어 있다면 중단 간 연결성(end-to-end connectivity)을 보장받을 수 있다.

3) DV (Distance Vector) 알고리즘: NS-2의 DV 알고리즘은 ‘Distributed Bellman-Ford’ 라우팅 혹은 ‘Distance Vector’ 라우팅이라 부른다. DV 알고리즘은 토폴로지의 모든 노드에 대하여 가장 좋은 경로와 그 목적지에 도달하기 위해 사용되는 경로를 담고 있는 거리 벡터 테이블(Distance Vector Table)을 가진다. 이 테이블은 각 목적지 노드에 대하여 가장 좋은 하나의 경로만을 가진다. 각 노드는 하나 혹은 그 이상의 인접한 이웃끼리 정보를 주고 받아 분산된 정보를 사용하여 라우팅을 실시한다.

4) LS (Link State) 알고리즘: LS 알고리즘에서 네트워크 내의 모든 노드들은 전체 네트워크 토폴로지와 모든 링크 코스트를 알고 있다. 모든 노드들은 주기적으로 링크 상태를 네트워크 내의 모든 노드들로 방송하게 되므로 모든 노드가 동일한 네트워크 토폴로지와 링크 코스트의 정보를 공통으로 가지고 있고, 이를 바탕으로 경로를 계산한다. 경로 계산 알고리즘은 Static, Session 알고리즘과 마찬가지로 다익스트라의 ‘All-Pair SPF Algorithm’을 사용한다.

III. 컴퓨터 시뮬레이션

시뮬레이션은 NS-2를 사용하여 실제 시스템에 적용시킬 수 있는 토폴로지를 구성 후 테스트를 진행하였다.

NS-2는 UC Berkeley에서 제작된 네트워크 시뮬레이션 툴로서 C++기반에 Tcl/tk로 개발되었다. 네트워크 구성은 Tcl/tk로 작성하여 사용자로 하여금 용이한 코드의 작성을 가능하게 하고, 라우팅 프로토콜 실제구현은 C++로 함으로써 강력한 성능을 발휘하게 된다[6].

본 논문에서 사용한 네트워크 토폴로지는 그림 3과 같이 상명대학교 소프트웨어대학 5층 건물을 기반으로 설계하였고, 각 현장 제어장치는 건물 내 방 하나에 한 개가 있다고 가정하였다. 한 개의 현장 제어장치는 총 4가지의 센서 데이터를 주기적으로 전송하도록 설계하였다. 또한 각 현장 제어장치는 유선 연결 구조를 가진다. 각 센서 및 현장 제어장치의 통합 제어 및 모니터링을 위한 서버 노드는 지하 1층에 한 개 있다고 가정하고 토폴로지를 구성하였다. 각 관제점에서 발생하는 모든 데이터는 서버 노드로 전송된다.

그림 4는 현장 제어장치의 구조이며, 토폴로지 구조 상 하나의 노드가 된다. 한 개의 현장 제어장치에서 사용하는 센서 및 관제점은 총 4개이므로, 시뮬레이션에서의 현장 제어장치는 CBR (Constant Bit Rate) 트래픽 4개를 주기적(0.5s 마다)으로 발생하도록 설정하였다. 네트워크 대역은 1초에 10메가의 비트를 처리할 수 있는 10Mbps로 설정하였으며 물리적인 통신신호를 통하여 데이터 신호가 송신자로부터 수신자까지 도달하는 전파지연은 100ms로 설정하였다. 또한 각 노드는 큐가 꽉 차게 되면 패킷을 드랍시키는 DropTail 큐 관리 기법을 사용하였다. 시뮬레이션에 사용한 파라미터들을 정리하면 표 1과 같다. 관제점 패킷 발생 시간과 시뮬레이션 시간을 제외한 토폴로지와 파라미터들을

실제 SDDC BAS에 맞게 구성하였으며, 관제점에서 발생하는 패킷의 주기는 응용에 따라 다를 수 있다. 이에 시뮬레이션을 원활히 진행하기 위하여 관제점의 패킷 발생주기와 시뮬레이션 시간을 같은 비율로 축소하여 진행하였다. 4개의 라우팅 알고리즘을 비교하기 위해 동일한 파라미터에서 라우팅 알고리즘 파라미터만을 변환하며 시뮬레이션을 진행하였다.

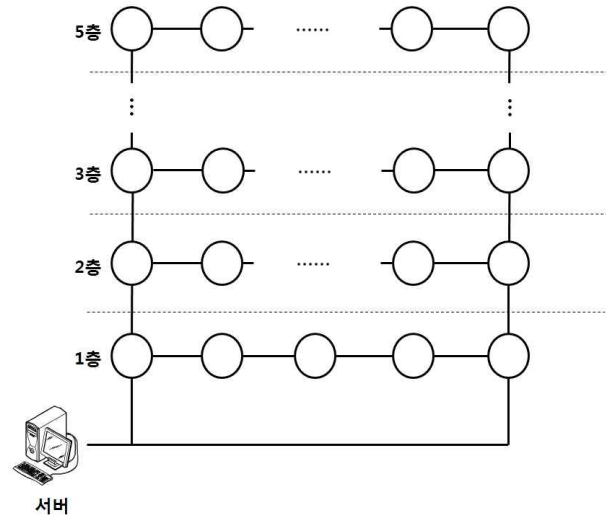


그림 3. 시뮬레이션에서 사용한 네트워크 토폴로지.  
Fig. 3. Network topology for simulation.

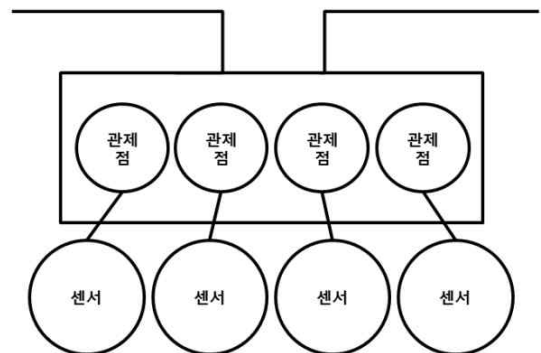


그림 4. 현장 제어장치의 구조.  
Fig. 4. Construction of control device in field.

표 1. 파라미터 설정.

Table 1. Configuration of parameters.

	Set value and method
네트워크 대역폭	10[Mbps]
노드 간 전파지연	100[ms]
전체 노드 수	69
큐 관리 기법	DropTail
라우팅 알고리즘	Static, Session, DV, LS
전송 프로토콜	TCP, UDP
관제점 패킷 발생 주기	0.5[s]
패킷 크기	각 센서 당 4[bytes]
시뮬레이션 시간	1[min]

IV. 성능 평가

본 장에서는 NS-2로 구현한 시뮬레이션 환경을 통해 각 라우팅 알고리즘의 성능을 평가하였다. 성능 평가 요소로 단대단 지연(end to end delay), 처리율(throughput) 그리고 라우팅 오버헤드(routing overhead)를 사용하였다. 그리고 데이터 전송 프로토콜을 TCP 프로토콜과 UDP 프로토콜을 적용시켜, 비교를 진행하였다.

1. 단대단 지연 평가

표 2는 각 라우팅 알고리즘의 평균 단대단 지연을 보여주고 있다. Session 알고리즘은 전송 도중 node failure가 발생해서 라우팅 경로가 바뀌지 않으면 Static과 동일하다. 본 논문에서 언급하는 평균 단대단 지연은 각 관계점에서 발생한 패킷이 서버 노드로 전송되어 처리된 시간의 평균이다. 데이터 전송 프로토콜은 TCP로 설정하여 실험을 진행하였다. 실험 결과 LS 알고리즘이 0.9705초로 가장 적었으며, DV 알고리즘 적용 시 평균 단대단 지연 시간은 0.9708초로 LS 알고리즘과 큰 차이가 없음을 보여주고 있다.

그림 5는 홉 카운트 당 발생한 최대 단대단 지연을 보여주는 그래프이다. 하지만 각 알고리즘 당 차이가 거의 발생하지 않음을 확인할 수 있다. 측정 결과로 각 알고리즘 당 최대 홉 수는 18 홉으로 모두 동일하였다. 하지만 DV 알고리즘과 LS 알고리즘을 적용 시 15홉 이상부터 최대 홉 카운트인 18 홉을 통과하는 패킷들의 수가 총 1,177개에서 1,111개로 감소하는 것을 확인할 수 있었다.

2. 처리율 평가

그림 6-9는 각 TCP 전송 프로토콜을 적용했을 때 라우팅 알고리즘에 따른 처리율의 변화를 보여주는 그래프이다. 처리율은 단위 시간에 서버로 들어오는 바이트 수의 총 합을 나타낸다. 1~30초까지는 4개의 알고리즘 모두 비슷한

표 2. 알고리즘 비교.

Table 2. Comparison of algorithms.

라우팅 알고리즘	평균 단대단 지연[s]
Static	1.0471
DV	0.9708
LS	0.9705
Session	1.0471

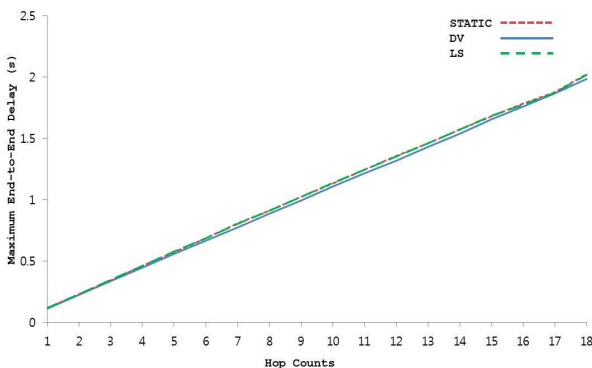


그림 5. 라우팅 알고리즘에 따른 홉 카운트 당 최대 단대단 지연.

Fig. 5. Routing algorithm for end-to-end delays.

성능을 보이지만 30-40초에서 특정 링크(서버와 1층 최좌측 노드간 링크)를 단절 시켰을 때 Static 알고리즘은 이에 적절히 대응할 수 없는 결과를 그림 6에서 보여주고 있다. Static 알고리즘이 건물의 확장 및 변경에 적절히 대응할 수 없음을 알 수 있다. 그림 7과 8에서 DV와 LS 알고리즘은 주기적으로 네트워크의 변화를 체크하기 때문에 성능저하가 크게 나타나진 않았다. 그림 9에서 Session 알고리즘의 경우 노드의 변화 시 경로를 재계산하는 과정을 거치기 때문에 성능저하가 나타난 후 다시 원래와 같은 처리율을 유지하지만 링크가 단절되는 순간 재계산하는 과정이 있기 때문에 처리율이 LS나 DV보다 큰 폭으로 저하된 것을 볼

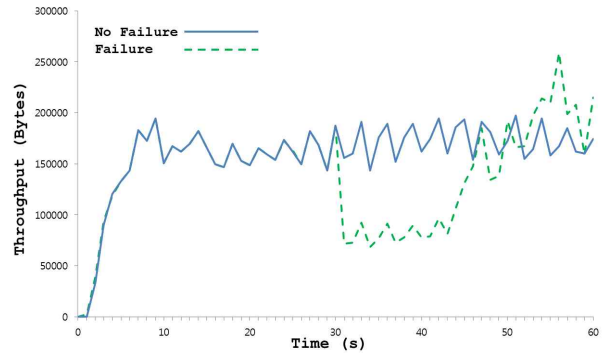


그림 6. Static 처리율(TCP).

Fig. 6. Throughput of Static algorithm(TCP).

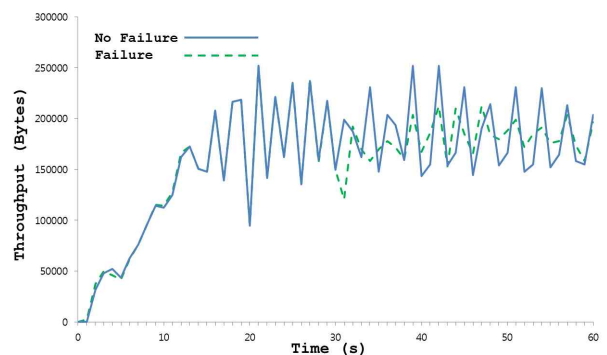


그림 7. DV 알고리즘 처리율(TCP).

Fig. 7. Throughput of DV algorithm(TCP).

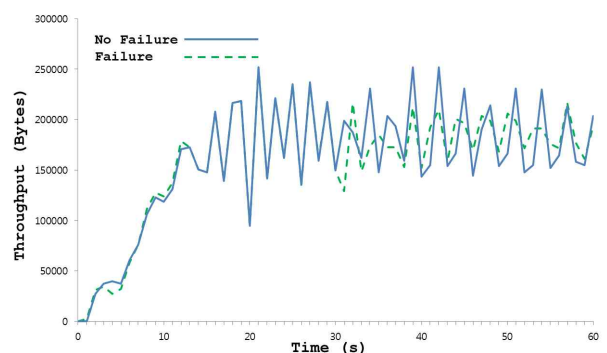


그림 8. LS 알고리즘 처리율(TCP).

Fig. 8. Throughput of LS algorithm(TCP).

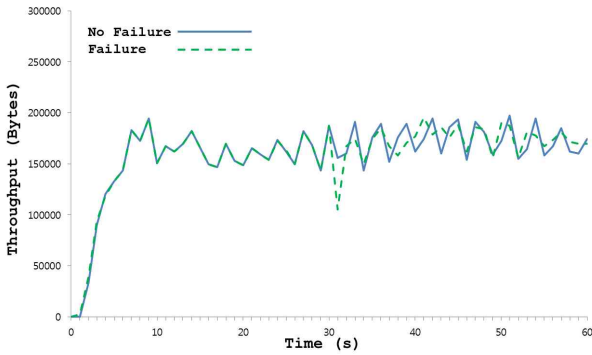


그림 9. Session 알고리즘 처리율(TCP).

Fig. 9. Throughput of Session algorithm(TCP).

수 있다. 이를 통해 DV 알고리즘과 LS 알고리즘을 적용하였을 때, 전체적인 처리율이 상승함을 확인할 수 있다. 하지만 Static 알고리즘을 제외한 나머지 알고리즘들은 네트워크 상태변화를 체크하기 위한 부하인 라우팅 오버헤드가 생기게 된다.

3. 라우팅 오버헤드 평가

그림 10은 각 라우팅 알고리즘에 따라 발생하는 라우팅 오버헤드를 보여주는 그래프이다. 라우팅 오버헤드는 전송되는 모든 패킷 바이트에 대한 제어 패킷(control packet) 바이트의 비율로 나타내었다. Static 알고리즘은 제어 패킷 발생이 없으므로 라우팅 오버헤드는 발생하지 않고 Session 알고리즘 역시 거의 Static 알고리즘과 유사하기 때문에 DV 알고리즘과 LS 알고리즘만을 비교하였다.

실험 결과로서 측정 시간이 증가 할수록 DV 알고리즘의 경우 라우팅 오버헤드가 LS 알고리즘 보다 오버헤드가 줄어드는 비율이 큰 것을 확인할 수 있었다. 이는 DV 알고리즘이 새로운 상태변화 정보가 없을 때 까지, 같은 네트워크 안에 있는 노드와 링크의 상태정보를 인접한 노드끼리 교환하기 때문이다. 그림 10에서 DV 곡선이 일정시간 후에 오버헤드가 급격히 감소되는 것은 이와 같은 DV 알고리즘의 특징을 잘 나타내 준다. 반면 LS 알고리즘은 모든 노드들이 상태변화에 관계없이 주기적으로 상태정보를 브로드

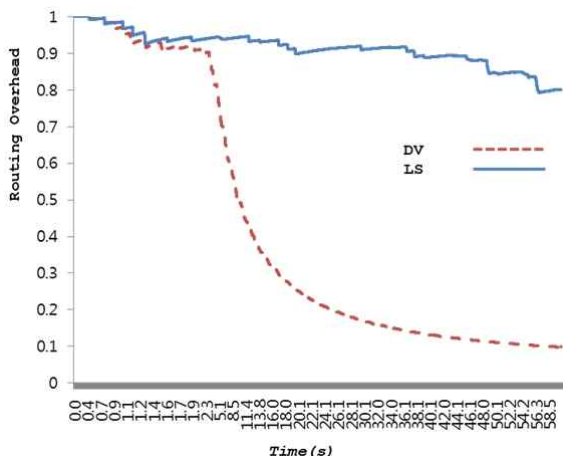


그림 10. 라우팅 오버헤드.

Fig. 10. Routing overhead.

캐스팅하기 때문에 시간이 지나도 오버헤드가 감소하지 않고 꾸준히 유지하는 것을 보여주고 있다.

4. 전송 프로토콜 평가

그림 11~14는 같은 실험환경에 UDP 전송 프로토콜을 사용했을 시 라우팅 알고리즘 별로 처리율을 나타낸다. 그림 11에서 Static 알고리즘은 TCP 전송 프로토콜을 사용했을 때와 마찬가지로 네트워크의 상태변화에 적절히 대응하지 못하는 현상이 나타났다. 또한 최선형(best effort) 서비스를 제공하는 UDP 프로토콜의 특성 상 노드가 끊긴 직후 패킷이 드롭된 것을 관찰할 수 있다. 이와 반대로 나머지 알고리즘에서는 대역폭에 비해 관제점들에서 전송되는 데이터

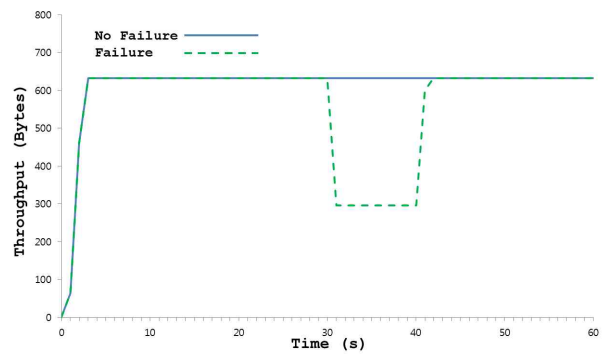


그림 11. Static 처리율(UDP).

Fig. 11. Throughput of Static algorithm(UDP).

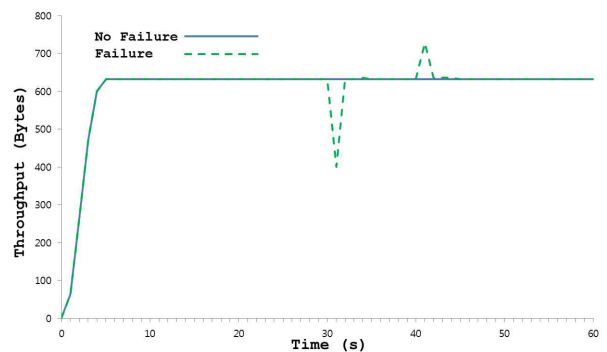


그림 12. DV 알고리즘 처리율(UDP).

Fig. 12. Throughput of DV algorithm(UDP).

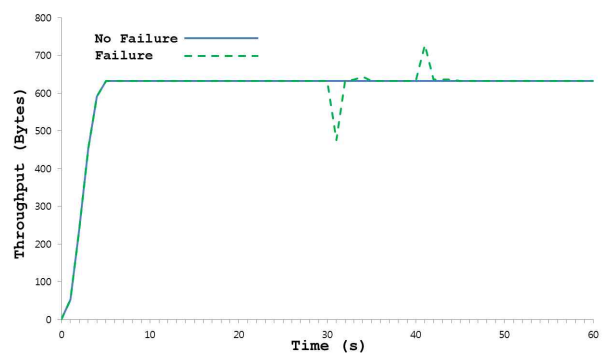


그림 13. LS 알고리즘 처리율(UDP).

Fig. 13. Throughput of LS algorithm(UDP).



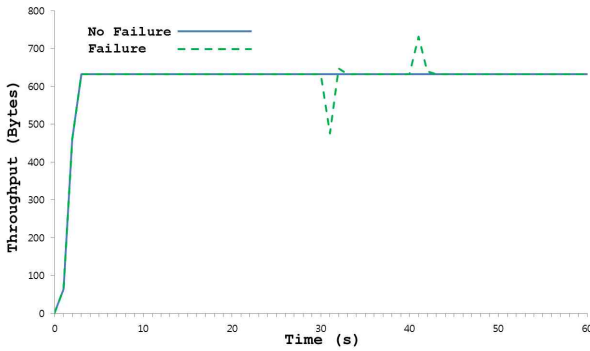


그림 14. Session 알고리즘 처리율(UDP).

Fig. 14. Throughput of Session algorithm(UDP).

의 크기가 작기 때문에 링크의 손실이 없으면 패킷이 드롭(drop)되지 않는다. 서버로 들어오는 바이트의 수가 약 200,000 바이트에 달하는 TCP 프로토콜에 비하여 0.3 %의 비율로 패킷이 발생되는 것을 확인할 수 있다.

### V. 결론

본 논문에서는 SDDC BAS에 적합한 라우팅 알고리즘을 찾아내기 위하여, NS-2 시뮬레이터 기반으로 실제 환경과 유사한 실험 환경을 구축하고 각 라우팅 알고리즘을 적용시켜 성능 비교 및 평가를 진행하였다.

실험 결과로서 라우팅 알고리즘에 따른 단대단 지연은 각 알고리즘 간 큰 차이가 발생하지 않는다는 점을 확인할 수 있었다. 하지만 처리율 결과에서 노드 고장이 발생하였을 때, Distance Vector 알고리즘과 Link State 알고리즘을 적용 시에 처리율이 전체적으로 상승하는 것을 확인할 수 있었다. 또한 라우팅 오버헤드는 Distance Vector 알고리즘이 Link State 알고리즘보다 매우 적게 발생함을 확인하였다. 따라서 빌딩 자동화 시스템이 소규모 네트워크라는 것을 고려하였을 때 Distance Vector 알고리즘이 효율적임을 확인할 수 있었다.

### 참고문헌

- [1] S. K. Park and K. H. Kim, "Technology trends of building automation system," *Magazine of the SAREK (in Korean)*, vol. 33, no. 12, pp. 43-49, 2004.
- [2] <http://www.pikeresearch.com/research/building-energy-management-systems>
- [3] S. H. Hong and S. H. Kim, "Technology trends of building automation communication network: BACnet," *Magazine of Architectural Environment Systems (in Korean)*, vol. 5, no. 3, pp. 43-49, 2011.
- [4] J. U. Kim and U. J. Kang, "Implementation of standard communication network for building automation in honeywell," *SAREK Workshop for Automatic Control Society (in Korean)*, pp. 247-254, 1999.
- [5] W. P. Hong and S. H. Lee, "Implementation of open & distributed intelligent control network for BAS," *KIEE Summer Conference 2000 (in Korean)*, vol. D, pp.

2446-2451, 2000.

- [6] The VINT Project, The NS Manual, 2002.



김정욱

1987년 서울대 제어계측공학과 졸업. 1989년 한국과학기술원 전기전자공학과 대학원 석사. 1993년 동 대학 박사. 2010년~현재 상명대학교 에너지그리드학과 교수. 관심분야는 스마트그리드, 신재생에너지, u-city 건설.



부창진

2001년 제주대학교 전기공학과 졸업. 2003년 동 대학원 석사. 2007년 동 대학 박사. 2011년~현재 제주대학교 전기공학과 시간강사. 관심분야는 건물 에너지 관리 및 제어시스템 설계.



이용호

2011년 상명대학교 컴퓨터과학전공 졸업. 2011년~현재 동 대학원 석사과정. 관심분야는 컴퓨터 네트워크, 트래픽 측정 및 분류.



최윤기

2011년 상명대학교 컴퓨터과학전공 졸업. 2011년~현재 동 대학원 석사과정. 관심분야는 컴퓨터 네트워크, 트래픽 측정 및 분류.



정진우

1992년 한국과학기술원 전기전자공학과 졸업. 1997년 미국 Polytechnic 대학 박사. 2005년~현재 상명대학교 컴퓨터 과학부 부교수. 관심분야는 컴퓨터 네트워크, 유무선 통신, 임베디드 시스템



김호찬

1987년 서울대 제어계측공학과 졸업. 1989년 동 대학원 석사. 1994년 동 대학 박사. 1995년~현재 제주대학교 전기공학과 교수. 관심분야는 신재생에너지, 에너지효율, 건물에너지 관리.