네트워크 기반 첨단 무기체계의 Fault Tolerant Ethernet (FTE) 기술

김세목 | Anh Pham Hoang | 이종명
명지대학교

요 약

최근 대부분의 첨단 (Mission Critical) 무기체계가 컴퓨터 노드 연결에 의한 Ethernet 네트워크 기반으로 구성되고 있어, 이에 따른 시스템의 신뢰성 측면이 더욱 중요한 요소로 대두되고 있다. 본고에서는 첨단 무기체계의 신뢰도를 높이기 위한 Fault Tolerant Ethernet (FTE)의 기본인 이중화 방식을 비교 소개하고자 한다. 기존의 소프트웨어 접근방식과 하드웨어 접근방식은 물론, 최근 본 연구팀이 제안한 복합 (Hybrid) 방식도 비교 대상으로 하였다. 또한 FTE 구현 시 필수적 고려사항인 고장감시/자동 복구 Metric 및 관련 Simulation 결과도 함께 제시한다.

I. 서 론

첨단 (Mission Critical) 무기체계에서는 대부분의 컴퓨터 시스템이 통상적으로 요구하는 고성능 및 저 전력 특성은 물론, 높은 신뢰도가 절대적으로 요구된다. 신뢰도는 고장률에 역 비례하므로 고장률을 줄임으로써 신뢰도 증가가 가능하다. 현실적으로 고장이 없는 시스템은 불가능하므로 해결책으로 고장 시에도 시스템이 동작을 유지하도록 하는 방안이 필요하며, 이는 고장극복 (Fault Tolerant) 기능의 확보를 의미한다. 특히 다수의 컴퓨터 노드 (이하 노드), 스위치 및 관련 장치들이 연결된 최근의 네트워크 기반 첨단 무기체계에서의 고장 극복 기능 확보는 더욱 중요하게 된다.

Ethernet은 네트워크 기반 체계의 De facto Standard로 자리매김하고 있으나 첨단 무기체계에 적용을 하기위해서는 Fault Tolerant 기능이 추가 되어 FTE로 구현되어야 한다. 그간 첨단 무기체계를 위한 FTE의 다양한 이중화 구현 방안이 연구 제시되었다[1-5]. 기본적으로는 하드웨어 접근방식과 소프트웨어 접근방식 (통상 Layer_2 기반)이 주로 연구되었고, 최근 이러한 방식의 장점을 최대한 보유한 복합 (hybrid) 방식이 당 연구팀에 의해 최근 제안된 바 있다[6]. 본고에서는 FTE 구현을 위한 이중화 방식을 비교 제시하고자 한다.

세 방식 중에서 소프트웨어 접근방식은 하트비트 패킷에 의한 네트워크 Topology 인식에 의한 고장 판단이다. 따라서 고장 시 복구 시간이 수 초가 걸리는 경우도 있어 실시간성이 요구되는 무기체계들에는 사용이 매우 제한된다. 실시간성에 적합한 방식으로는 하드웨어적 접근방식과 복합 방식을 들 수 있는데, 두 방식 모두 포트에 입력되는 데이터 패킷의 존재 여부만으로 고장 판단을 수행하므로 고장 판단 기준 시간 등의 Metrics 설정은 네트워크 및 해당 무기체계 성능 발휘에 큰 영향을 주게 된다. 그러므로 포트 고장감시 및 고장 시 여분 포트로의 전환 등에 필요한 Metrics를 살펴보고 이에 대한 Simulation도 함께 제시한다.

II. FTE 이중화 방식 비교

여기서는 FTE 구현을 위한 세 가지 이중화 방식 즉 소프트웨어 접근방식, 하드웨어 접근방식, 그리고 복합방식을 비

교 제시한다.

1. 소프트웨어 접근방식

1.1 개념

소프트웨어 접근방식은 각 노드가 자신을 제외한 모든 노드에 자신의 존재를 알리는 하트비트 (Heart Beat) 신호를 주기적으로 보내고, 각 노드에서 하트비트 신호를 받지 못하는 포트를 링크 고장으로 간주하여 하트비트 신호가 지속되는 포트를 채택 사용하여 네트워크 고장을 감지하고 복구하는 방식이다.

노드 개수를 N으로 가정하면, 각각의 포트는 자신을 제외한 N-1 포트로 하트비트 패킷을 주기적으로 보낸다. 주기를 T_h 라 할 때 각 노드는 T_h초 마다 특정한 하트비트 패킷과 Acknowledgement 패킷을 기대하게 된다. 이러한 방식은 스위치는 초당 $2*N(N-1)/T_h$ 개의 부가적 패킷을 처리하는 것이고, 개별 링크에는 $2*N/T_h$의 부가적 패킷이 전송 되므로 네트워크에 부담을 주게 된다. 예를 들어 50개 노드가 100msec 주기(T_h)로 하트비트를 보내는 경우 링크에는 초당 1×10^3개의 패킷, 스위치에는 4.9×10^4개의 패킷이 추가적으로 부여된다. 이 경우 Ping 기능을 64 Bytes ICMP 패킷으로 가정할 때, 링크 당 0.512Mbps의 추가 부담을 주게 된다.

1.2 고장감시 및 복구 방법

각 노드는 두 개 포트 중에서 주 포트(Primary Port)와 보조 포트(Secondary Port)를 정한다. 평상시에는 주 포트 입력 시그널만 사용을 하고 보조 포트의 입력은 참조하지 않는다. 고장감시는 각 노드의 포트들이 하트비트 주기 T_h초 마다 정의된 하트비트 패킷을 모니터하는 것으로 이루어진다. 만약 특정 노드에서 주 포트가 하트비트 패킷을 수신하지 못한 경우, 해당 링크의 이상을 선언하고 보조 포트의 하트비트 패킷 수신 여부를 확인한다. 만약 보조 포트도 하트비트 패킷을 수신하지 못하였을 경우 해당 노드와의 네트워크 단절을 선언한다. 이 경우 물리적 복구방법 (off-line maintenance)이 강구되어야 한다.

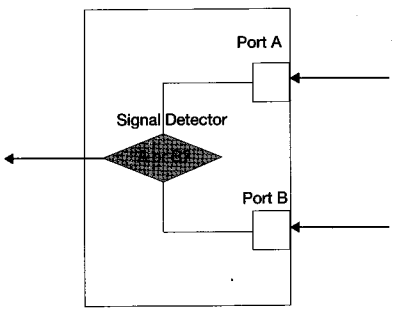
1.3 특징

소프트웨어 접근방식은 특별한 하드웨어를 사용하지 않고 기존의 COTS (Commercial-Off-The-Shelf) Ethernet 카드를 이용하기 때문에 구성이 간단한 장점이 있다. 그러나 네트워크 부하가 노드 수 제곱($O(N^2)$)과 복구 최소시간인 하트비트 주기의 곱에 비례하기 때문에 다중 노드 시스템 경우 복구 최소시간을 최소화 할 수 없다는 것이 이 방식의 중대한 단점이다. 또한, 각 노드끼리 하트비트 패킷을 송수신하며 네트워크 고장을 감시하고 복구하는 방법이므로 노드 CPU에 추가적인 부하를 주며, 링크 고장이 여러 노드에서 나타나기 때문에 고장 부분의 위치추적을 위해서는 모든 링크 상태를 취합하는 소프트웨어적 계산을 필요로 한다.

2. 하드웨어 접근방식

2.1 개념

하드웨어 접근방식은 (그림 1)과 같이 각 노드에서 두 개의 포트신호를 비교하여 물리적 신호 존재여부를 이용하는 방식이다. 즉 각 노드는 물리적 신호가 없는 포트를 링크 고장으로 간주하고 물리적 신호가 지속되는 포트를 사용하여 네트워크 고장을 감지/복구하는 방식이다.



(그림 1) 포트 시그널 감지

2.2 고장 감시 및 복구 방법

하드웨어 접근방식의 어려운 점은 물리적 신호 부재를 정의하는 방법이다. Ethernet은 Physical Layer 관점에서 보면, 패킷을 보낼 때만 전기적 시그널이 발생되는 통신망이므로 일반적으로 물리적 신호 부재 시간은 일정한 규칙이 없는 랜덤한 특성을 지닌다. 그러므로 물리적 신호의 부재를 선언하기 위하여 다음 두 가지 방법을 고려할 수 있다.

첫째, 네트워크 특성에 따라 정의되는 확률적인 최소 시간 단위 즉 임의의 한 노드가 한 패킷을 받은 후 다음 패킷을 받

을 확률적 최소 시간단위 이상을 기다린 후 물리적 시그널 부재라고 선언하는 방법과, 둘째, 송신노드에서 두 개의 포트가 같은 양의 패킷을 시간차가 적게 보낸다는 가정 하에 수신노드의 포트사이의 부재시간 차이가 허용 시간을 넘을 경우 신호 부재로 선언하는 방법이 있다.

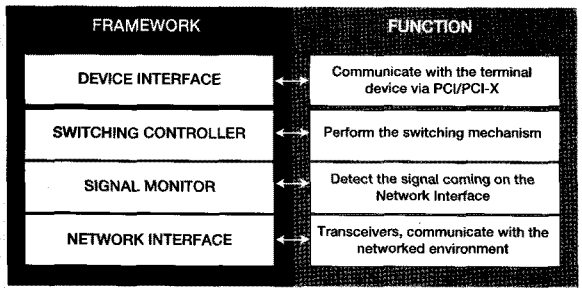
2.3 특징

하드웨어 접근방식은 소프트웨어 접근방식과는 다르게, 네트워크에 추가적 부하를 주지 않고 어느 링크에 고장이 있는지를 판단하기위해 상위 레벨로의 접근이 필요 없다는 장점이 있다. 그러나 제작사 고유의 하드웨어가 필요하므로 무기체계의 현장 임무 특성에 맞는 패러미터 (Parameter) 최적화가 제작 업체에 의존되는 단점이 있다.

3. 복합 (Hybrid) 방식

3.1 개념

복합방식[6]이란 하드웨어 접근방식을 소프트웨어적으로 해결하는 방법이다. 이 경우 하드웨어 접근방식의 포트 선택 알고리즘을 소프트웨어적으로 구현할 때 고려되어야 할 점은 어느 레벨에서 알고리즘을 구현할 것인가 하는 점이다. 하드웨어에 밀접한 driver 레벨로 내려갈수록 고장 감시와 복구 알고리즘의 반응시간을 단축할 수 있는 장점이 있으나 Network Interface Card (NIC) 하드웨어에 종속적이 되어 범용성을 갖추기 어려운 문제가 있다. 따라서 본 방식의 고장 판단은 네트워크 인터페이스에서 수신 데이터를 감시하여 실행된다. 즉 고장판단이 네트워크 인터페이스의 Layer_2에서 수행되므로 전체 네트워크의 상태를 파악하여 고장을 판단하는 소프트웨어 접근방식 보다 고장 판단 시간이 작다.

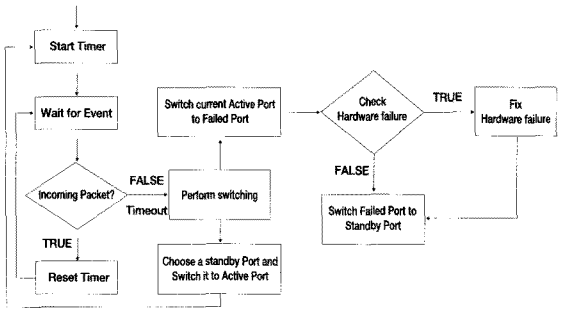


(그림 2) 소프트웨어 구조와 기능

이러한 복합방식의 소프트웨어 구조와 기능은 (그림 2)와 같다.

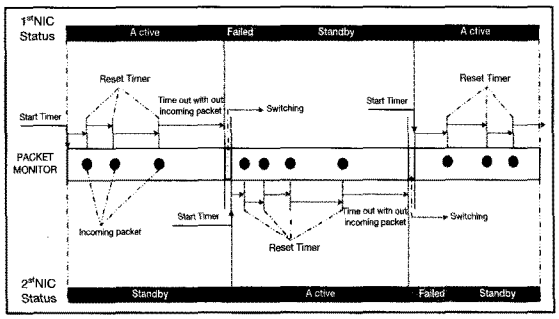
3.2 고장 감시 및 복구방법

데이터는 전송이 진행되는 동안 목적지 노드의 모든 트랜시버로 보내진다. 여기서는 네트워크 지연을 무시하여, 전송된 데이터가 목적지 노드의 각 포트로 동시에 도착한다고 가정한다. 이러한 가정 하에 데이터는 고장 감시 및 해당되는 스위칭 Mechanism에 의해 제어된다. (그림 3)은 고장 감시 및 복구 흐름도이다. 패킷의 유무 판단을 위한 입력 패킷의 유무 감시를 위한 타이머 (Timeout Interval)와 고장 극복을 위한 스위칭 과정이 이 흐름도의 주요한 요소이다.



(그림 3) 고장 감시 및 복구 흐름

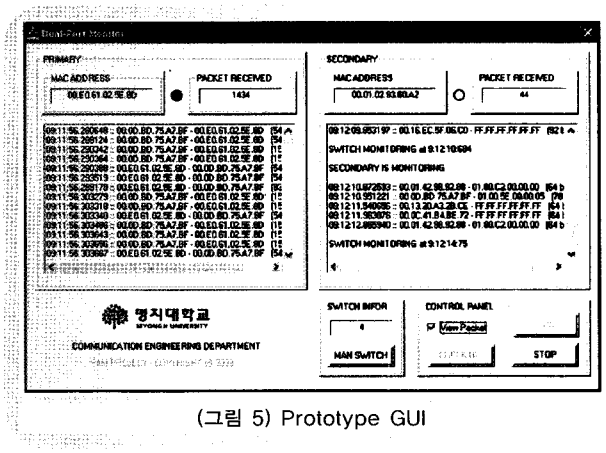
(그림 4)는 포트 전환 개념으로서, 정해진 시간 내에 패킷 수신이 확인되면 해당 Timer가 Reset되고, 그 시간 내에 패킷 수신이 없는 경우에는 Timeout이 선언되어 포트 전환 (Switching)이 수행됨을 보인다.



(그림 4) 포트 전환 Mechanism

이러한 복합 방식의 Prototype GUI 예를 (그림 5)에 나타내었다. 이 Prototype은 듀얼포트 NIC 또는 두 개의 Single-port NIC에 들어오는 패킷들을 감시한다. 감시 과정은 WinPcap과 Microsoft SDK library를 사용하여 구현하였다.

각각 포트의 상태는 다른 색깔로써 표현된다. 그림에서 파란색은 해당 포트가 주 (Primary) 모드로 동작 중임을 나타내고, 초록색은 해당 보조 (Secondary) 포트가 예비모드로 동작하고 있음을 나타낸다. 그림에는 제시되지 않았으나 해당 포트가 고장 시 적색으로 표시된다.



(그림 5) Prototype GUI

3.3 특징

복합방식은 하드웨어 접근방식의 소프트웨어 구현 방식이므로 고장 시 전환 소요시간이 소프트웨어 접근방식보다 매우 단축된다. 또한 소프트웨어 접근방식과 마찬가지로 특정 하드웨어가 필요하지 않으므로 COTS 기반의 NIC 사용이 가능하다.

4. 방식별 비교

세 가지의 이중화 방식을 요약하면 〈표 1〉과 같다.

〈표 1〉 FTE 이중화 방식 비교

방식 / 특성	하드웨어 접근방식	소프트웨어 접근방식	복합방식
고장 시 전환 (실시간성)	매우 빠름	느림	빠름
COTS 사용	불가	가능	가능

III. 고장 감시 / 자동 복구 Metric 및 Simulation

1. FTE 구현 시 주요 Metrics

앞의 (그림 3)은 포트 감시 및 고장 판단 시의 포트 전환 개념으로서, 정해진 시간 내에 패킷 수신이 확인되면 해당 Timer가 Reset되고, 그 시간 내에 패킷 수신이 없는 경우에는 Timeout이 선언되어 포트 전환(switching)이 수행됨을 보인다. 즉 두 개의 연속된 패킷에 의해 Timeout 시간이 결정된다. 따라서 첫 번째 Metric은 최적의 Timeout 시간 (Timeout Interval)이다. Timeout 시간은 너무 짧게 설정하면, 해당 포트가 잘 작동됨에도 불구하고 스위칭이 너무 자주 일어나게 된다. 반면에 Timeout 시간이 너무 길게 설정되면, 하드웨어 고장을 적시에 감지하지 못한다. Timeout 시간의 최적화는 하드웨어 접근방식에도 그대로 적용되는 사항이다. 또한 스위칭이 실행되는 동안에 데이터는 손실된다. 그러므로 두 번째 Metric은 스위칭 시의 데이터 손실 최소화이다.

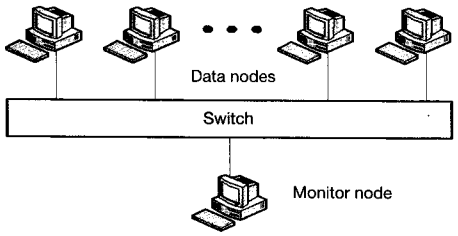
첫 번째 Metric의 해답은 포트에 입력되는 패킷 데이터의 형태에 의해 결정된다는 것이다. 그래서 최선의 해결책은 다양한 데이터 패턴을 모델링하여 이를 분석하고 실험하는 것이다. 두 번째 Metric인 스위칭 시의 데이터 손실 최소화의 해결은 소프트웨어 구현의 최적화에 의한 스위칭 타임 최소화로 해결할 수 있을 것이다. 이 경우 손실된 데이터를 복구하기 위한 Mechanism이 고려되는 경우에는 부가적 이득을 얻을 수 있다.

2. Simulation 및 결과

FTE 구현에서 주요 Metrics의 하나인 Timeout 시간 설정을 위해 본 연구에서 개발된 툴과 관련 모델링, 그리고 Simulation의 예 및 그 결과를 소개한다. 이러한 기법은 기존의 하드웨어 접근방식에도 적용가능하다.

2.1 Simulation

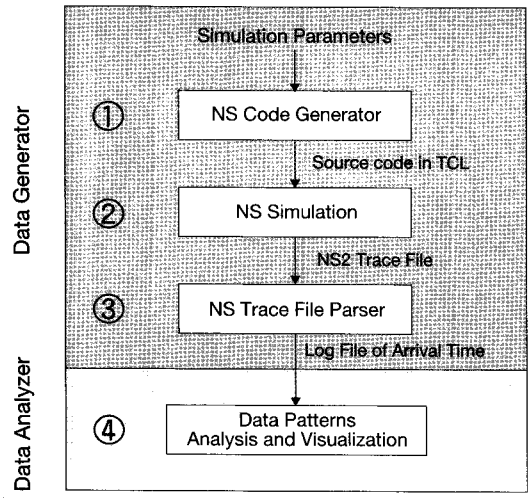
(그림 6)과 같이 테스트 모델을 생성한다. 그림에는 두 가지 형태의 노드 즉 다수의 데이터 노드와 한 개의 Monitor 노드가 존재한다. 모든 데이터 노드는 랜덤 데이터를 생성



(그림 6) 테스트 모델

하여 Monitor 노드로 보내고 Monitor 노드는 데이터 노드로부터 모든 패킷들의 도착 시각 등의 관련 정보를 기록한다. Monitor 노드에 입력되는 데이터는 연속되는 패킷이다. 여기서 데이터 노드가 생성하는 패킷의 유형은 가장 일반화된 Poisson 모델을 사용한다. 또한 Switch 부분을 단순화하여 데이터 노드와 Monitor 노드를 직접 연결하여 구현하였다.

(그림 7)은 테스트 모델 등을 시험하기 위한 네 개의 프로세스로써 구성된 Simulation Tool의 구조이다.



(그림 7) Simulation Tool 구조

〈표 2〉는 고장 시나리오와 파라미터이며, Timeout 시간의 최적 값을 찾기 위해 설정되었다.

〈표 2〉 Simulation Parameter 설정

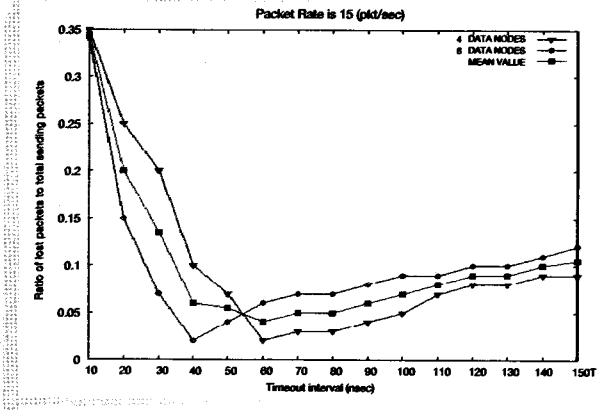
	Parameter	Values
Network Configuration	Number of data nodes	4, 8
	Link speed (Mbps)	100
	Link delay (msec)	5
Poisson Model	Rate (Packet/sec)	10, 20, 25
	Packet Size (Bytes)	1500
Switching	Timeout interval	[10, 150]
	Switching Process (msec)	5
Failure	Scenario	Uniform

Simulation은 데이터 노드의 수와 패킷 발생의 비율을 변화시키며 수행하였다. 예상한 바와 같이 데이터 노드의 수나 패킷 발생의 비율을 증가함에 따라 트래픽 부하도 증가되었다. 여기서는 8개 노드와 25 (packets/sec) 패킷율을 최대값으로 설정하였다. 즉 Monitor 노드에 최대 200개 패킷이 1초 동안에 입력되는데, 이는 FTE 무기체계의 일반적 설계 기준인 busy 하고 homogeneous 기준으로 일반화 된 180 packets/sec의 Traffic 요건을 상회하는 값이다.

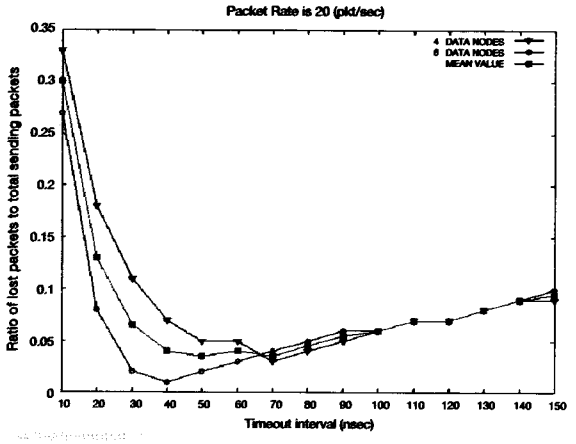
2.2 Simulation 결과

(그림 8-10)은 패킷율과 노드수를 변화시켜 Timeout 시간과 패킷 손실율 간의 관계를 나타낸 결과이다. 아울러 (그림 11)과 (그림 12)는 동일한 내용을 노드 수를 기준으로 나타낸 결과이다. (그림 8)에서 패킷 손실이 적은 Timeout 시간은 대략 40 msec ~ 120 msec의 하나의 값이다. 이 범위는 (그림 9) 및 (그림 10)에서와 같이 트래픽 부하가 증가함에 따라 x 축의 왼쪽으로 이동하여, 30 msec ~ 80 msec 구간에서 패킷 손실이 최소화된다. 따라서 이 경우 Timeout 시간 값은 공통의 구간인 40 msec ~ 80msec 내의 하나의 값을 설정하면 된다. 패킷 손실에 대한 기준을 다소 완화하면 30 msec ~ 120 msec로 확대할 수도 있다. 경험적으로 현장에서 사용 중인 Layer_1 기반의 제품이 50 msec로 설정된다고 알려지고 있어 이러한 단순 Simulation 으로도 일부 검증이 가능하다. 여기서는 하드웨어 고장이 주기적으로 발생하는 것을 가정한 결과이다.

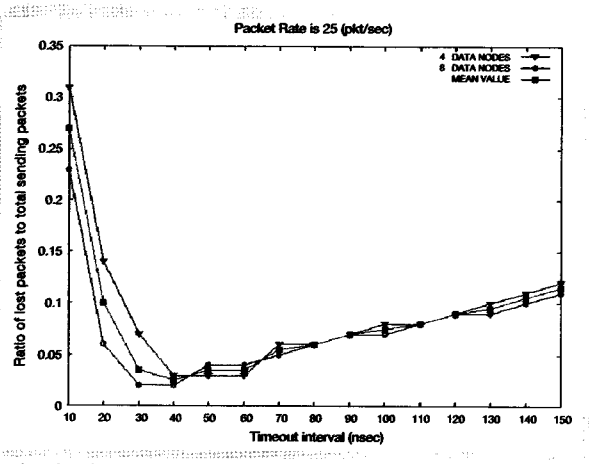
데이터 패턴이 본 실험처럼 Poisson 단일 모델로 구성된 단순한 경우가 아닌 경우에는 다양한 추가 실험이 수반되어 적절한 Timeout Interval 값을 도출하여야 한다.



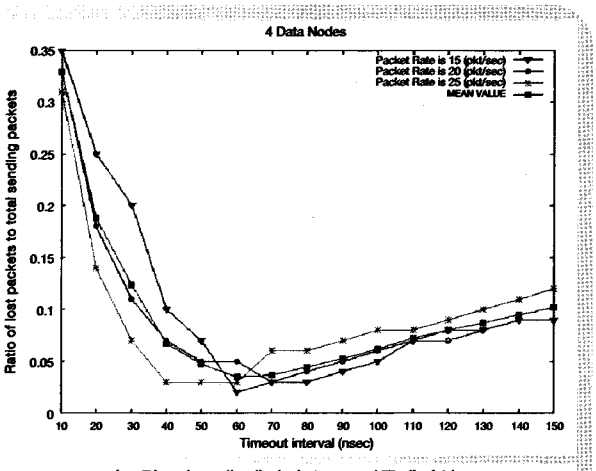
(그림 8) 15 packets/sec에서의 Timeout에 따른 패킷 손실율



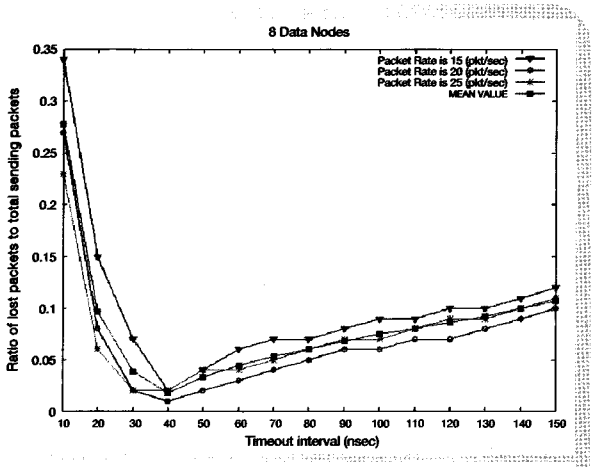
(그림 9) 20 packets/sec에서의 Timeout에 따른 패킷 손실율



(그림 10) 25 packet/sec에서의 Timeout에 따른 패킷 손실율



(그림 11) 4개 데이터 노드 시뮬레이션



(그림 12) 8개 데이터 노드 시뮬레이션

IV. 결 론

본고에서는 네트워크 기반 첨단 (Mission Critical) 무기체계의 신뢰성을 증대시키기 위한 Fault Tolerant Ethernet (FTE) 이중화 구현 기법을 비교 제시하였다. 즉 소프트웨어 접근방식, 하드웨어 접근방식, 그리고 [6]에서 제안된 복합방식의 세 가지 방식을 검토하였다.

이중에서 하드웨어 접근방식과 복합방식은 실시간적 임무에 특히 유리한 방식이다. 그러나 이러한 방식의 실전 적용을 위해서는 전송 데이터 유형에 따른 Metrics 정의와 최적값이 확률적 기반 하에서 도출되어야 한다. 따라서 본고에

서는 FTE 네트워크를 Poisson 분포 데이터 패턴으로 단순 모델화하고, Metric으로서 고장판단시간인 Timeout 기간을 설정하여 Simulation 하고 그 결과를 제시하였다. 하드웨어 고장을 포함하여 Simulation을 수행함으로써 단순 모델임에도 결과 값이 현장 운용 값에 유사하여 그 방법론의 타당성을 보여준다.

데이터 패턴이 본 실험처럼 Poisson 단일 모델로 구성된 단순한 경우가 아닌 일반적 상황을 위해, 다양한 데이터 패턴을 고려한 Metrics 분석 및 실험이 수행 되어야 하므로 연구 진행 중이다. 또한 본고에서 다룬 이중화 이외에 다중화로의 발전도 해결해야 할 요소이다. 노드수가 많은 대형 네트워크 기반의 무기체계의 경우 (예를 들면 이지스 함정전투체계), Switch나 노드의 포트 구성이 이중화가 아니라 다중화 되고 있다. 다중화 구현을 위해서는 동작중인 링크에 고장이 발생한 경우, 최적 전환을 위한 예비 포트 선정 기법 및 Mesh 구조 스위치 네트워크의 최적 경로를 판단하는 알고리즘 등의 연구가 필요하다.

감사의 글

본고는 국방과학연구소와 방위사업청 (DAPA, 과제 ADD-07-06-02)의 지원으로 수행되었습니다.

참 고 문 헌

[1] J. Huang, S. Song, L. Li, P. Kappler, R. Freimark, J. Gustin, and T. Kozlik, "An Open Solution to Fault-Tolerant Ethernet : Design , Prototyping and Evaluation, Invited Paper" , Proceeding of the 18th IEEE International Performance, Computing, and Communications Conference, Phoenix, February 1999.

[2] S. Song, J. Huang, P. Kappler, R. Freimark, J. Gustin, and T. Kozlik, "Fault-Tolerant Ethernet for IP-Based Process Control Networks", LCN, p. 116, 25th Annual IEEE International Conference on Local Computer Networks (LCN' 00), 2000.

[3] C. Edmonds, "Programmable Controller Networking-Dual Cable, Redundancy, Multiple Networks and Application", ISA' 92 Advances in Instrumentation and Control, Vol.47, Part2, October 1992.

[4] Automation Research Corporation, "Ethernet-Based Control Network Strategies, Automation Strategies Report", Automation ARC, Three Allied Drive, Dedham, MA 02026, October 1997.

[5] Jan-Erik Sarparanta, "Fault-Tolerance Requirements and Solutions", Department of Computer Science Helsinki University of Technology, May 1998.

[6] Anh Pham Hoang, Jong Myung Rhee, Se Mog Kim, and Dong Ho Lee, "A Novel Approach for Fault-Tolerant Ethernet Implementation", Communication Engineering Department, Myongji University, Rep. of Korea, International conference, NCM 2008(IEEE CS proceeding), September 2008.

약 력

김 세 목

1998년 부경대학교 학사
2007년 ~ 현재 명지대학교 통신공학과 석박사과정
2001년 (주)네오디지틀
2008년 (주)성원정보통신
2008년 ~ 현재 (주)엘피네트워크 근무
관심분야 : Fault Tolerant System, HFC Network

Anh Pham Hoang

2005년 Ho Chi Minh University of Technology, Computer Science and Engineering
2008년 ~ 현재 명지대학교 통신공학과 석사과정
관심분야 : Data Link, Fault Tolerant System

이 종 명

1976년 서울대학교 학사
1978년 서울대학교 석사
1987년 North Carolina State University, ECE Dept.(Ph. D.)
1997년 국방과학연구소 책임연구원
1999년 데이콤 연구소 부소장
2005년 하나로텔레콤 CTO (부사장)
2006년 ~ 현재 명지대학교 통신공학과 교수
관심분야 : Military Communications, Fault Tolerant System, Ad-hoc, Data Link, Convergence