

선박 내 유선망에서 차등화 서비스 지원을 위한 패킷 폐기 기술

이성로*, 권장우*, 정민아**, 허경°

Packet Drop Technique for Differentiated Services in Wired Ship Area Networks

Seong Ro Lee*, Jang-Woo Kwon*, Min-A Jeong**, Kyeong Hur°

요약

선내 유선 네트워크에서는 선내의 각종 센서와 제어기를 자율적으로 구성관리하고 원격제어를 제공하는 기능을 수행한다. 이러한 선박 내 유선 통신망에서 차등화서비스가 지원된다면, 우선순위가 높은 실시간 플로우는 최소 전송률을 보장 받고 빠르게 전송되어, 지능적인 선박 통신망을 구현할 수 있다. 본 논문에서는 이러한 선박 내 유선 네트워크에서 차등화 서비스를 지원하기 위해, 기존 RIO (RED In and Out) 방식보다 우수한 패킷 폐기 기술을 제안한다. 제안한 패킷 폐기기술은 개별 플로우별로 관리하지 않고 특정한 기준에 따라 몇 개의 플로우 그룹으로 나누어, 그룹별 관리를 통해서 동일 클래스 내 플로우 간 공평성을 제공한다. 시뮬레이션 결과에서는 RIO 방식보다 링크 이용률은 조금 감소하나, 다중 병목 구간을 경유하는 플로우들이 RIO 방식보다 더 많은 전송률을 보장 받음을 알 수 있다.

Key Words : DiffServ, Ship Integrated Network, Packet Drop, TCP

ABSTRACT

An wired ship area network has functionality of remote control and autonomous management of various sensors and instruments embedded or boarded in a ship. For such environment, the DiffServ (Differentiated Services) realizes that the high-speed real-time flow with the higher priority has the guaranteed minimum data rate and is delivered faster. As a result of this DiffServ effect, the intelligent Ship Area Networks can be implemented. In this paper, an packet drop technique is proposed to outperform the previous RIO (RED In and Out) drop mechanism for DiffServ in ship area networks. the proposed packet drop technique does not manage the individual flows and divides them into several flow groups according to a criterion. And it guarantees the fairness between individual flows in the same QoS class through the group-based control. In simulation results of the proposed packet drop technique, the link utilization decreases than RIO. But it guarantees more data rates to DiffServ flows passing multiple bottleneck links.

* 본 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2009-0093828)과 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업(NIPA-2014-H0401-14-1009) and This work was supported by INHA UNIVERSITY Research Grant. (INHA-47274)

• First Author : Mokpo National University Department of Electronics Engineering, srlee@mokpo.ac.kr, 정회원

° Corresponding Author : Gyeongin National University of Education Department of Computer Education, khur@ginue.ac.kr, 중신회원

* Inha University Department of Computer Science & Information Technology, jwkwon@inha.ac.kr

** Mokpo National University Department of Computer Engineering, majung@mokpo.ac.kr, 정회원

논문번호 : KICS2014-10-398, Received October 8, 2014; Revised November 17, 2014; Accepted November 17, 2014

I. 서 론

최근 국내에서는 ETRI와 현대중공업이 공동으로 스마트쉽을 만들기 위해 IT기반 선박용 통합모니터링 인프라 기술인 선박 네트워크(Ship Area Network)를 개발하고 있다^{1,2)}.

선박용 부가 서비스 지원 선박 네트워크 구조기술은 선박 네트워크모델을 제시하고 선박 네트워크에 연결된 기관감시제어장치, 항해기록장치, 통합항해장치를 선박 내에서 통합적으로 모니터링을 가능하도록 한다. 그리고 육상에서는 인공위성을 통해 선박 내 IT기자재에 대한 원격제어가 가능하다. 이는 조선 산업에 첨단 IT기술을 접목해 선박장치 통합제어를 기반으로 지능화 및 자동화된 항해정보 시스템을 개발하고 광대역 유무선 네트워크 및 각종 센서연동 기술을 기반으로 선박 항해를 실시간으로 모니터링해서 선박의 각종 센서 및 장치들을 원격에서 유지보수하고 제어할 수 있는 기술을 의미한다³⁾.

또한 선박의 표준네트워크는 International Electronics Committee (IEC)에서 IEC 61162-1, 2, 3, 4로 규정하고 있다. IEC 61162-1은 NMEA 0183 (National Marine Electronics Association)으로서 1980년대부터 주로 선박의 항해장비가 공통적으로 장착한 4,800bps의 single talker, single listener 시리얼 데이터 통신이며, IEC 61162-2는 NMEA 0183 High Speed 버전으로 38.4 kbps의 single talker, multi listener, 전기적신호는 RS422인 직렬통신이다. NMEA 2000은 multi-talk, multi-listen, multi master, PnP이며 물리계층은 CAN 2.0B를 사용하고 IEC 61162-3으로 표준화되어 선박에서는 200m, 250 kbps를 사용한다. NMEA 2000은 SAE J 1939 (Society of Automotive Engineers)의 single packet와 multi packet외에 fast packet 프로토콜을 추가하여 선박에서 실시간으로 제어하여야 하는 instrument level network에 맞도록 수정하였다. 또한 IEC 61162-4는 ethernet base의 MiTS(Maritime Information Technology Standards) 프로토콜로서 노르웨이에서 시작한 것으로 shipboard control network의 표준으로 사용하고 있다⁴⁾.

Internet Engineering Task Force (IETF)는 사용자들이 요구하는 QoS를 보장하기 위한 차세대 네트워크 구조로써 Integrated Service (IntServ)와 Differentiated Service (DiffServ)를 제시하였다^{5,6)}. IntServ는 사용자 개별 플로우에 대한 end-to-end QoS를 보장하지만 플로우가 점유하는 모든 라우터들이 각

플로우에 대한 상태정보를 유지해야 하고, 접속제어 (Admission Control)에 대한 방법으로 Resource ReSerVation Protocol (RSVP)을 사용하기 때문에 확장성 (Scalability) 문제가 있다⁷⁾. 또한 IntServ는 기존 인터넷의 하부구조를 수정해야 하는 단점도 있다⁸⁾. DiffServ 방식은 각 사용자 플로우별 관리가 아닌 집합 단위의 클래스별 관리를 사용하고, DiffServ Code Point (DSCP)를 이용하여 IP 패킷에 대한 Per Hop Behavior (PHB)를 규정한다. 같은 DSCP 코드를 가진 모든 패킷들은 동일한 방식으로 처리된다. 이러한 방법은 다수의 사용자 플로우를 소수의 클래스로 분류하여 처리하는 집합 방식의 메커니즘이며 대규모의 플로우들을 전달하는 내부 네트워크에 적합한 확장성을 갖고 있다. 사용자 플로우들은 DiffServ 도메인 입구 라우터에서 계약된 Traffic Profile 준수 여부에 따라 In-Profile (In)과 Out-of-Profile (Out) 패킷으로 표기되어 네트워크 혼잡 시 각 라우터에서 RED In and Out (RIO) 버퍼 방식에 따라 Out 패킷을 우선적으로 폐기하여 In 패킷을 보호한다^{9,10)}.

선내 유선 네트워크에서는 선내의 각종 센서와 제어기를 자율적으로 구성관리하고 원격제어를 제공하는 기능을 수행한다. 이러한 선박 내 유선 통신망에서 차등화서비스가 지원된다면, 우선순위가 높은 실시간 플로는 최소 전송률을 보장 받고 빠르게 전송되어, 지능적인 선박 통신망을 구현할 수 있다. 본 논문에서는 이러한 선박 내 유선 네트워크에서 차등화 서비스를 지원하기 위해, 기존 RIO (RED In and Out) 방식보다 우수한 패킷 폐기 기술을 제안한다. 제안한 패킷 폐기기술은 개별 플로우별로 관리하지 않고 특정한 기준에 따라 몇 개의 플로우 그룹으로 나누어, 그룹별 관리를 통해서 동일 클래스 내 플로우 간 공평성을 제공한다. NS-2 시뮬레이션 결과 [11]에서는 RIO 방식보다 링크 이용률은 조금 감소하나, 다중 병목 구간을 점유하는 플로우들이 RIO 방식보다 더 많은 전송률을 보장 받음을 알 수 있다. 또한 이러한 DiffServ 통신 기술은 Layer 3의 네트워크 계층 뿐 아니라, Layer 2의 링크 계층에도 적용할 수 있는 기술이다.

본 논문의 2장에서는 제안된 패킷 폐기기술을 제안하고, 3장에서는 제안된 패킷 폐기기술을 시뮬레이션을 통하여 성능을 분석하고, 4장에서 결론을 맺는다.

II. 선박 내 유선망에서 차등화 서비스 지원을 위한 패킷 폐기 기술

기존의 선내 유선네트워크와 NMEA 네트워크의

구성은 그림 1과 같이 선내 제어 네트워크(이더넷기반의 MiTS)에 게이트웨이가 연결되어있고, 그 하부에는 각 장치들이 연결되어 있는 NMEA 장치네트워크 (instrument network)로 구성되어진다. 본 논문에서는 각 게이트웨이로 유입되는 다양한 선내 통신 트래픽에게 차등화된 QoS를 제공하여 지능적인 선내 통신망을 지원하고자 한다. 즉, 그림 1의 게이트웨이에서 RIO와 제안된 패킷 폐기 기술들을 적용하여, 플로우 그룹 단위로 차등화된 데이터 전송률을 보장하고자 한다.

하나의 도메인 내에서도 플로우(flow)들 간에 지나가는 경로는 매우 다양하기 때문에 각각 플로우들 간의 RTT(Round Trip Time)과 목적지까지 도착하는 동안 겪는 병목구간의 수가 다르게 된다. 그림 2의 시뮬레이션 결과에서도 알 수 있듯이 RTT가 길고 홉(hop) 수가 많은 플로우의 경우 그렇지 않은 플로우와 비교할 때 수율이 상대적으로 떨어지게 됨을 알 수 있다. 이 중에서 병목을 겪고 있는 홉의 수가 많은 경우에 그 영향이 더욱 확실하다.

본 논문에서 다루고 있는 네트워크는 대역폭 할당 (bandwidth provisioning)이 잘 되어 있는 상태를 가정한다. 기존의 DiffServ 네트워크에서도 충분한 대역폭을 준비하였다면 모든 플로우들이 계약한 목표 전송률(target rate)을 만족시킬 수 있다. 하지만 대역폭이 충분할 경우 지나가는 병목구간의 수가 적은 플로우는 잉여 대역폭의 대부분을 차지하며, 병목구간의 수가 많은 플로우는 간신히 목표 전송률을 맞출 정도로만 잉여 대역폭을 차지하게 된다는 문제점이 발생한다. 그림 2의 결과에서 보듯이 Packet Dropper가 없을 경우에는 모든 플로우가 목표 전송률을 만족시키기 위해서는 60%의 잉여 대역폭이 필요하다. 하지만 이 경우에 병목구간을 한 개만 통과하는 플로우의 경

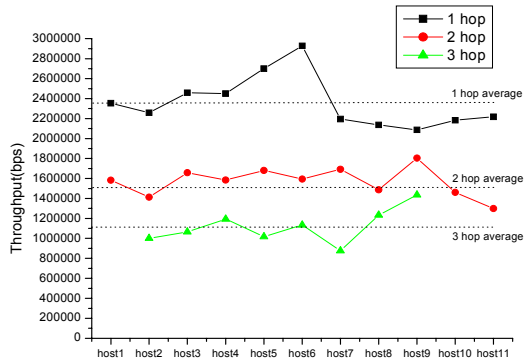


그림 2. 병목 구간의 수에 따른 수율의 차이 (**160%로 over-provisioning 되어 있는 네트워크에서의 수율 측정결과)
Fig. 2. Throughputs according to number of bottleneck links at 160% BW over-provisioning networks

우에는 목표 전송률의 2.3배의 수율을 가지고 있음을 확인할 수 있다.

이런 문제점 때문에 동일한 서비스 클래스에 계약을 했다더라도 목적지까지 도착하는 동안 겪는 상황에 따라서 실제 플로우가 받게 되는 QoS에는 차이가 발생한다. 이 문제는 DiffServ의 기본 취지인 CoS(Class of QoS) 만을 제공한다는 측면에서 발생하는 문제점이다.

본 논문에서 제안하는 Intelligent Drop Algorithm은 플로우별 관리는 하지 않고, 플로우 그룹 단위의 제어를 실시한다. 하나의 서비스 클래스에 속해 있는 플로우들을 특정한 기준에 따라 몇 개의 플로우 그룹으로 나누어 주고, 그룹간의 불공평성을 감소시켜서 플로우간의 수율 공평성을 제공하고자한다. 본 논문에서 플로우 그룹을 나누는 기준으로 각 종단 단말(Leaf Terminal)에서 발생하는 트래픽 특성에 따라 우선순위를 부여하고, 이에 따라 Label을 붙여, Multi-Protocol Label Switching (MPLS)과 같이, 네트워크 계층 또는 링크 계층에서의 스위칭을 실시하는 것이다. 이러한 방식은 플로우의 수에 영향을 받지 않기 때문에 확장성 문제는 없다고 할 수 있다.

Intelligent Drop Algorithm의 핵심은 이렇게 나뉜 플로우 그룹별로 패킷 폐기 확률을 다르게 한다는 점이다. 즉, 각 종단 단말에서 생성된 Label에 따라 플로우 그룹을 분류하고, 그룹 단위로 서로 다른 P_drop_leaf (leaf early drop probability)을 적용하여 네트워크 도메인으로 유입되는 플로우 간의 불공평을 해결하려 한다. 각 종단 단말에서는 전송하는 트래픽의 우선 순위에 따라 Label을 배정하고, 계약하는 목표 전송률의 양 만큼 In(In-profile) 패킷으로 표기하

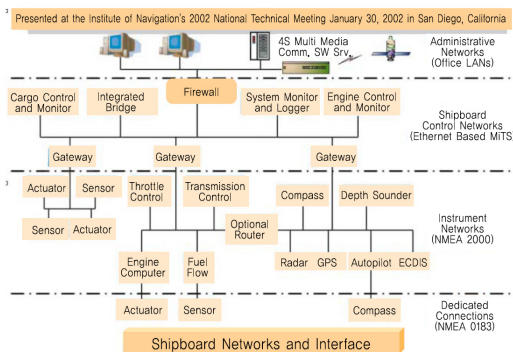


그림 1. 기존의 선내 유선네트워크(MiTS), NMEA 네트워크
Fig. 1. Wired networks in a ship(MiTS), NMEA networks

고, 초과된 패킷들을 Out(Out-of-profile) 패킷으로 표기한다. 또한, 계약된 전송률 보다 매우 높은 데이터 전송률을 갖지 못하도록, 미리 패킷 폐기를 실행한다. 무작정 목표 전송률만을 기준에 두고, 이를 초과하는 플로우 그룹의 패킷들을 폐기 시킨다면 네트워크자원을 충분히 사용하지 못하게 될 수 있지만, 본 기술에서는 남은 대역폭을 공평하게 나누어 가지기만 한다면 패킷 폐기가 거의 발생하지 않도록 하고 있다. 따라서 링크 이용률 측면에서도 큰 문제가 생기지 않게 된다.

게이트웨이에서는 Label 종류에 따라 구분된 입력 인터페이스별로 서로 다른 P_drop_max (max drop probability)를 적용하여, 게이트웨이를 통과하고 있는 플로우들간의 불공평을 해결한다. 게이트웨이에 적용하는 기존의 RIO 버퍼^[9]에서는 식 (1)에 따라 패킷 폐기를 수행하나, 본 기술에서는 입력 인터페이스별로 다른 P_max_drop을 사용하여서 각 인터페이스별로 공평하게 남은 대역폭을 사용하도록 하여 플로우 간의 공평성을 지원하려는 것이다. P_drop_leaf 와 P_max_drop를 계산하는 방식은 같으며, 적용하는 방식에 차이가 있을 뿐이며, P_drop_max을 계산하는 알고리즘의 pseudo code는 그림 3과 같다.

$$P_{drop} = P_{drop\ max} \times \frac{q_length_{avg} - th_{min}}{th_{max} - th_{min}} \quad (1)$$

그림 3에서 group_n.share_index는 group_n의 플로우들이 전송하는 패킷들 중에 하나의 In packet당 Out packet의 개수를 뜻한다. 이 식은 이 후 패킷 폐기 확률을 구하기 위한 가장 기본이 되는 식이다. share_index는 비율의 형태를 취하고 있기 때문에 여러 장점들을 갖는다. 즉, 1Mbps 트래픽 만큼 계약된 플로우와 2Mbps 만큼 계약된 트래픽 간에 잉여 대역폭을 나누어 가지는 양에는 분명 차이가 있어야 한다. 물론, 대역폭 점유 비율이 2배 정도가 되어야 한다. 이런 점을 고려해서 만들어진 인덱스 값으로 1Mbps 트래픽의 플로우가 0.5Mbps의 잉여 대역폭을 점유했다면, 2Mbps 트래픽의 플로우가 1Mbps의 잉여 대역폭을 점유하여야 같은 share_index값을 가지게 된다.

group_n.share_ratio는 group_n이 전체 잉여 대역폭 중에 현재 점유하고 있는 양을 간접적으로 나타내는 값이다. 그룹이 전체 10개(N=10)가 있고 10개의 그룹이 모두 같은 양을 점유하고 있다면, 모든 그룹의 share_ratio는 0.1이 될 것이다. 또한 많이 점유하고 있다면 0.1보다 큰 값이 되고 반대의 경우에는 0.1보

```

Assume that there are N groups.
Initially,
no_in=0, no_out=0 for all group
coeff = (P_balance-1) / (1-N)^2

When packet arrive.
if packet belongs to group n,
then check its dscp (DiffServ Code Point) {
    if IN then group_n.no_in++;
    else if OUT then group_n.no_out++;
    group_n.share_index = group_n.no_out / group_n.no_in;

    group_n.share_ratio = group_n.share_index / (group_1.share_index + group_2.share_index + ..... + group_N.share_index)
    group_n.drop_index = group_n.share_ratio * N;
    if (drop_index ≤ 1.0) then P_drop_max = P_balance * group_n.drop_index^2
    else if (drop_index > 1.0) then P_drop_max = coeff * (group_n.drop_index - N)^2+1.0
}
if at LEAF TERMINAL
check its dscp {
    if IN then interface n . P_drop_in = 0.0
    else if OUT then interface n . P_drop_out = 0.1 * P_drop_max
    IN packet is not dropped at leaf router
}
if at GATEWAY
check its dscp {
    if IN then interface n . P_max_drop_in = 0.01 * P_drop_max
    interface n . P_drop_in = P_max_drop_in * {
        (q_length_avg_in-th_min_in)/(th_max_in-th_min_in)}
    else if OUT then interface n . P_max_drop_out = P_drop_max
    interface n . P_drop_out = P_max_drop_out * {
        (q_length_avg_out-th_min_out)/(th_max_out-th_min_out)}
}

According to P_drop_in, P_drop_out, Determine whether this packet is dropped or not.

Finally if the packet is dropped,
then check its dscp{
    if IN then no_in -= 0.5;
    else if OUT then no_out -= 0.5;
}
    
```

그림 3. P_drop_max을 계산하는 알고리즘
Fig. 3. Computing Pseudo-codes for P_drop_max

다 작은 값이 될 것이다. 그렇다면 drop_index의 값은 수식에 따라 많이 점유한 경우에는 1보다 크게 되고,

반대의 경우에는 1보다 작은 값을 갖게 될 것이다. 이 값은 아래의 수식에 따라 P_{drop} 을 결정하는 함수의 입력 값이 된다.

그림 4의 특성으로는 $drop_index$ 가 1.0보다 클 때와 작을 때의 P_{drop_max} 값의 차이가 크다는 점이다. $drop_index$ 가 1인 경우는 위의 설명에서와 같이 공평하게 잉여 대역폭을 나누어 가지고 있는 그룹에 해당되며, 이 경우에는 $P_balance$ 가 패킷 폐기 확률이 된다. 하지만 $drop_index$ 가 1보다 큰 경우, 즉 잉여 대역폭을 많이 점유하고 있는 경우에는 급격히 지수적으로 패킷 폐기 확률이 증가하게 되는 것이다. 반대의 경우에는 지수적으로 급격히 패킷 폐기 확률이 감소하게 된다. 따라서 그룹별로 이런 규칙에 따라 잉여 대역폭을 많이 점유하게 되면, 높은 패킷 폐기 가능성을 가지게 되므로 수율이 떨어지게 되며 이만큼을 다른 플로우들이 가져갈 수 있게 되는 것이다.

중단 단말(Leaf Terminal)에서는 내부 게이트웨이에서 혼잡에 의한 패킷 폐기가 일어나기 전에, 미리 불공평한 부분을 제거하는 역할을 하기 때문에 In packet에 대한 패킷 폐기는 실시하지 않는다. 하지만 위의 P_{drop_max} 를 기반으로 Out packet에 대한 패킷 폐기를 실시하게 된다. 그렇지만 앞에서 설명한 것과 같이, 혼잡이 일어나기 전에 패킷을 폐기하는 것이기 때문에, 너무 큰 패킷 폐기확률이 적용되면 전체

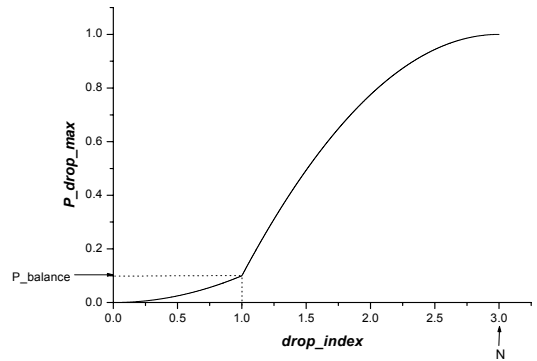


그림 4. $drop_index$ 와 P_{drop_max} 의 관계
Fig. 4. Relationship between $drop_index$ and P_{drop_max}

수율이 떨어지기 때문에 적절한 상한 값을 찾는 과정이 필요하다. 그래서 P_{drop_max} 에 0.1을 곱한 값을 사용하였으며 이는 조정될 수 있는 계수 값이다.

본 기술은 MPLS과 같이 여러 고정 경로를 제공하는 2.5 계층 프로토콜이 3 계층 프로토콜을 지원해 주는 경우에 적용된다. Label을 규정하는 데, 가장 많이 사용되는 방식이 <Ingress terminal, Egress terminal> 쌍이 같은 플로우들을 하나의 FEC (Forwarding Equivalence Class)로 정의하여, 목적지 주소에 할당된 Label을 할당하는 것이다. 이러한 환경에서 본 기술이 적용되면 FEC 단위로 그룹들이 점유하고 있는

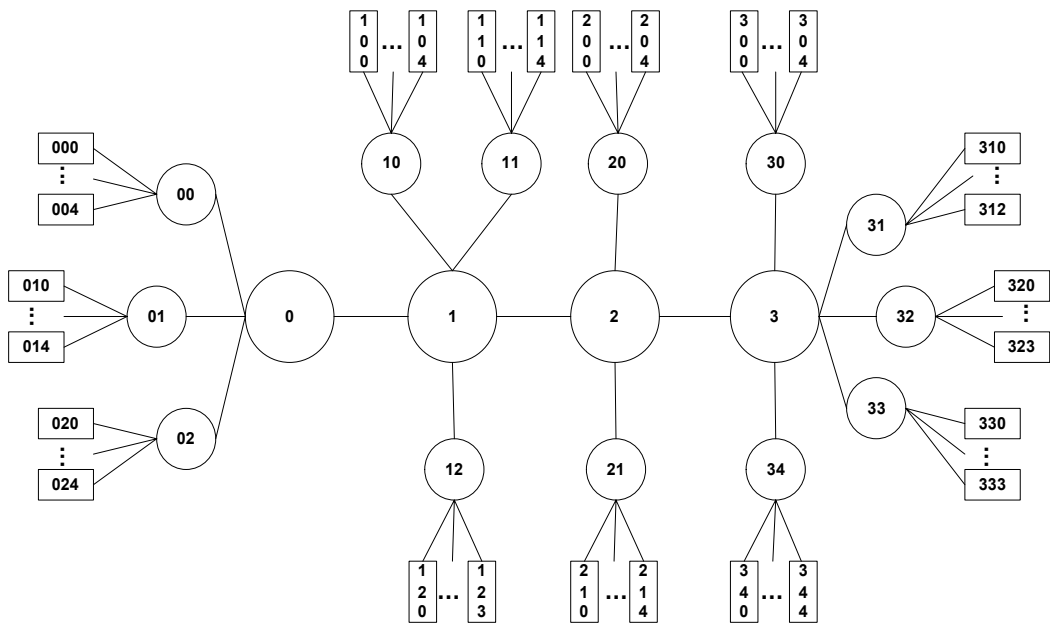


그림 5. 시뮬레이션 토폴로지
Fig. 5. Simulation Topology

잉여 대역폭을 골고루 나누어 쓰도록 하는 방향으로 트래픽을 제어를 하게 된다. 이는 플로우 단위로 제어 하는 경우와 비교할 때 복잡도가 크게 감소한다. 게이트웨이에서는 본 기술이 혼잡 제어의 성격을 가지게 된다. 현재 가장 많이 사용되고 있는 혼잡 제어 기술로는 RIO 방식이 있다. 본 기술에서도 RIO의 기술을 일부 적용해서 사용한다. 게이트웨이에서는 혼잡 상황이 더 심각해지지 않게 하면서, 되도록 적은 패킷 폐기가 발생하도록 하는 것이 주된 목적이므로, RIO 큐의 상황을 고려하여서 패킷 폐기 확률을 조절할 필요가 있다. 따라서 식 (1)의 RIO에서 사용한 P_drop 수식을 이용한다. 기존 RIO 기술과 다른 점은, 첫째로 P_max_drop의 값이 입력 인터페이스에 따라 달라진다는 것이다.

III. 시뮬레이션결과 및 분석

그림 5의 시뮬레이션 토폴로지에서 0, 1, 2, 3 번 노드는 선박 내 유선 네트워크에서 게이트웨이 역할을 수행하는 노드를 나타내고, 표 1은 시뮬레이션 시

생성된 전체 플로우들을 나타낸다. 그리고 송신-수신 노드의 쌍은 서로 다른 FEC를 나타내고, 여기서는 홑수에 따라 FEC 그룹을 분류하였다. 그림 2에서 문제점을 보여주는 상황과 동일한 조건에서 Intelligent Drop Algorithm을 적용한 경우의 결과를 그림 6에 나타내었다.

그림 2의 RIO 기반 결과와 비교하여, 1 홑, 2 홑, 3 홑 플로우들 간의 수율 격차가 상당히 줄어들었음을 알 수 있다. 이는 제안하는 기술이 타당함을 입증하고 있다. 또한, 3 홑에서도은 목표 전송률(1Mbps)보다 높은 수율을 얻을 수 있음을 확인하였다. 이 결과는 그림 2에서 가까스로 목표 전송률을 만족시켰던 것과 비교하면 상당히 향상된 결과이다.

아래의 그림 7과 그림 8은 게이트웨이 간 대역폭 여유분을 다르게 설정하면서, 각 경우 1 홑, 2 홑, 3 홑의 플로우들의 평균 수율을 기존 RIO 방식과 비교한 결과이다. 0% ~ 280% 까지 네트워크 대역폭 할당을 변경한 경우에 대한 실험 결과이다. 전체적으로 모든 경우에 대해 Intelligent Drop Algorithm을 적용한 경우의 수율 공평성이 우수함을 알 수 있다.

표 1. 그림 5에서 송/수신 터미널 연결 쌍
Table 1. Source-destination pairs in Fig. 5

source host	destination host	hops	source host	destination host	hops	source host	destination host	hops
000	120	1	020	121	1	110	224	1
001	300	3	021	220	2	111	330	2
002	301	3	022	221	2	112	331	2
003	302	3	023	311	3	113	332	2
004	222	2	024	312	3	114	333	2
010	122	1	100	223	1	200	340	1
011	123	1	101	320	2	201	341	1
012	303	3	102	321	2	202	342	1
013	304	3	103	322	2	203	343	1
014	310	3	104	323	2	204	344	1

표 2. 그림 5에서 시뮬레이션 설정 변수 값
Table 2. Parameter values in Fig. 5

RIO based Network		Network employing Intelligent Drop algorithm	
maximum window size	10000 packets	maximum window size	10000 packets
th_min_in	90 packets	th_min_in	90 packets
th_max_in	200 packets	th_max_in	200 packets
th_min_out	40 packets	th_min_out	40 packets
th_max_out	90 packets	th_max_out	90 packets
P_drop_max_in	0.02	P_drop_max_in	variable
P_drop_max_out	0.1	P_drop_max_out	variable
token depth	200 packets	token depth	200 packets
token rate	125 packet	token rate	125 packet
target rate	1 Mbps	target rate	1 Mbps
simulation time	100 seconds	simulation time	100 seconds

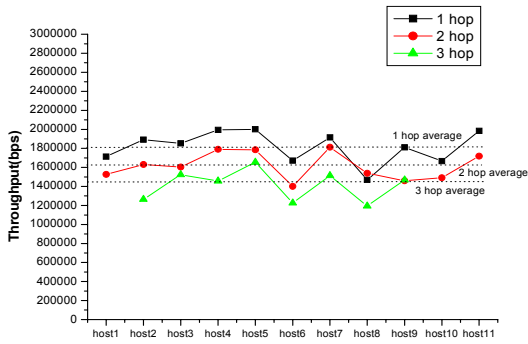


그림 6. Intelligent Drop Algorithm을 적용한 결과(**160%로 over-provisioning 되어 있는 네트워크에서의 수율 측정결과)
Fig. 6. Throughputs with Intelligent Drop Algorithm at 160% BW over-provisioning networks

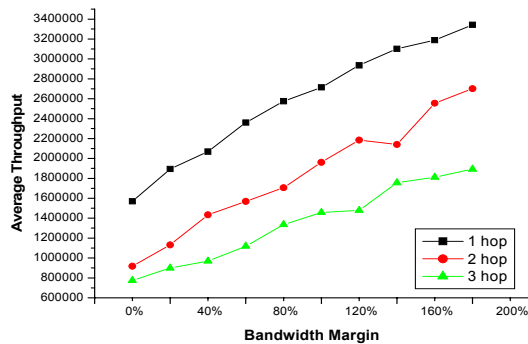


그림 7. RIO based Network에서 대역폭 여유분에 따른 각 홉의 평균 수율
Fig. 7. Throughputs with RIO based Network at each Bandwidth Margin

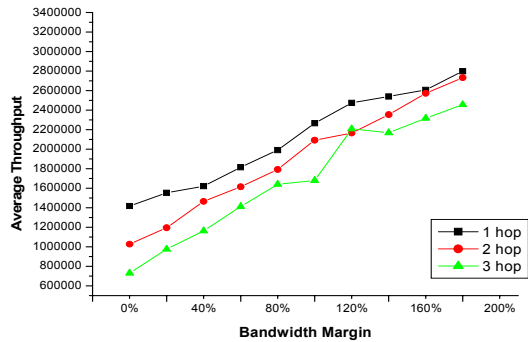


그림 8. Intelligent Drop Algorithm을 적용 시 대역폭 여유분에 따른 각 홉의 평균 수율
Fig. 8. Throughputs with Intelligent Drop Algorithm at each Bandwidth Margin

IV. 결 론

본 논문에서는 지능적인 선박 통신망을 구현하기 위해, 선박 내 유선 통신망에서 차등화서비스를 지원

하는 기술을 제안하였다. 제안한 패킷 폐기기술은 개별 플로우별로 관리하지 않고 FEC 기준에 따라 몇 개의 플로우 그룹으로 나누고, 홉 수에 따라 동일 클래스 내 플로우 간 수율 공평성을 향상시킨다. 시뮬레이션 결과는 다중 병목 구간을 경유하는 플로우들이 더 많은 전송률을 보장 받음을 나타내었다. 또한 이러한 DiffServ 통신 기술은 Layer 3의 네트워크 계층 뿐 아니라, Layer 2의 링크 계층에도 적용할 수 있는 기술이며, 제안한 기술은 그룹 단위의 CoS만을 제공하는 DiffServ에서 플로우 단위 QoS 성능을 향상시킬 수 있는 기술이라 할 수 있다. 또한, 시뮬레이션 환경과 같이, 중소 단위의 플로우들이 유입되는 선박 내 유선망에서 효과적으로 적용될 수 있는 기술로서 향후 실제 적용하여 추가적인 연구를 수행할 수 있다.

References

- [1] S. Park and S. R. Lee, "Marine disasters prediction system model using marine environment monitoring," *J. KICS*, vol. 38C, no. 3, pp. 263-270, Mar. 2013.
- [2] S.-H. Lee, J.-H. Kim, K.-D. Moon, K. Lee, and J. H. Park, "Performance analysis on integrated ship area network," *J. KICS*, vol. 38C, no. 3, pp. 247-253, Mar. 2013.
- [3] K.-I. Kim, "Adjusting transmission power for real-time communications in wireless sensor networks," *J. Inf. Commun. Convergence Eng.*, vol. 10, no. 1, pp. 21-26, Mar. 2012.
- [4] O.-S. Park and J.-M. Ahn, "Implementation algorithms and performance analysis of maritime VHF data system based on filtered multi-tone modulation," *J. KICS*, vol. 38C, no. 3, pp. 254-262, Mar. 2013.
- [5] R. Barden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: An overview," *Internet RFC 1633*, Jun. 1994.
- [6] S. Blake, D. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *Internet RFC 2475*, Dec. 1998.
- [7] J. Wroclawski, "The use of RSVP with IETF integrated services," *Internet RFC 2210*, Sept. 1997.

[8] X. Xiao and L. M. Ni, "Internet QoS: The big picture," *IEEE Network Mag.*, pp. 8-18, Mar./Apr. 1999.

[9] D. Clark and W. Fang, "Explicit allocation of best effort packet delivery service," *IEEE / ACM Trans. Netw.*, vol. 6, no. 4, pp. 362-373, Aug. 1998.

[10] W. Feng, et. al., "Understanding and improving TCP performance over networks with minimum rate guarantee," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 173-187, Apr. 1999.

[11] UCB/LBNL/VINT, "Network simulator-ns2," <http://www-mash.cs.berkeley.edu/ns>, 1998.

정민아 (Min-A Jeong)



1992년 2월: 전남대학교 이학사
 1994년 2월: 전남대학교 이학석사
 2002년 2월: 전남대학교 이학박사
 2002년 4월~2003년 2월: 광주과학기술원정보통신공학과 Post-Doc

2003년 4월~2005년 2월: 전남대학교 전자통신기술연구소 Post-Doc
 2011년 9월~2013년 2월: Cleveland Clinic Research
 2005년 3월~현재: 목포대학교 컴퓨터공학과 부교수
 <관심분야> 데이터베이스데이터마이닝, 생체인식시스템, 무선통신응용분야, 임베디드시스템

이성로 (Seong Ro Lee)



1987년 2월: 고려대학교 전자공학과 공학사
 1990년 2월: 한국과학기술원 전기 및 전자공학과 공학석사
 1996년 8월: 한국과학기술원 전기 및 전자공학과 공학박사
 1997년 9월~현재: 목포대학교

공과대학 정보전자공학과 교수

<관심분야> 디지털통신시스템, 이동 및 위성통신시스템, USN/텔레메틱스응용분야, 임베디드시스템

허경 (Kyeong Hur)



1998년 2월: 고려대학교 전자공학과 졸업
 2000년 2월: 고려대학교 전자공학과 석사
 2004년 8월: 고려대학교 전자공학과 공학박사
 2004년 8월~2005년 8월: 삼성

종합기술원(SAIT) 전문연구원

2005년 9월~현재: 경인교육대학교 컴퓨터교육과 부교수

<관심분야> 유무선통신시스템설계, 상황인지기술, 컴퓨터교육

권장우 (Jang-Woo Kwon)



1990년 2월: 인하대학교 전자공학과 공학사
 1992년 2월: 인하대학교 전자공학과 공학석사
 1996년 8월: 인하대학교 전자공학과 공학박사
 1996년~1998년 특허청 사무관

1998년~2009년 동명대학교 컴퓨터공학과 부교수

2006년~2012년 정보통신산업진흥원 인재양성단장

2012년~현재: 인하대학교 컴퓨터공학과 교수

<관심분야> HCI, 생체신호처리, 지능형 시스템