

## 실시간 운영체제의 효율적인 다중송수신을 위한 멀티캐스트 설계 및 구현

이정원<sup>o</sup>, 손필창, 김용희, 이철훈  
 충남대학교 컴퓨터 공학과  
 {jwlee<sup>o</sup>, pccson, yhkim, chlee}@ce.cnu.ac.kr

### A Design and Implementation of Multicast for Efficient Multiple Transfer in Real-time Operating System

Jungwon Lee<sup>o</sup> Philchang Son, Yonghee Kim, Cheolhoon Lee  
 Dept. of Computer Engineering, Chungnam National Univ.

#### 요 약

정보가전기기의 발전에 따라 인터넷과 퍼스널 컴퓨터 중심의 정보화는 이제 어느 장소, 어느 기기에서나 네트워크와 연결되어 사용자에게 서비스를 제공하는 유비쿼터스 시대로 접어들고 있다. 이러한 정보기기들은 기기에 특성화된 실시간 운영체제를 탑재하고 통신하게 되며, 한정된 자원으로 통신 대역폭을 효율적으로 사용하기 위한 기술을 적용해야 할 필요성이 있다. 본 논문에서는 실시간 운영체제에서 정보기기간의 효율적인 다중송수신을 위해서 멀티캐스트 전송 모듈을 설계 및 구현하였다.

#### 1. 서론

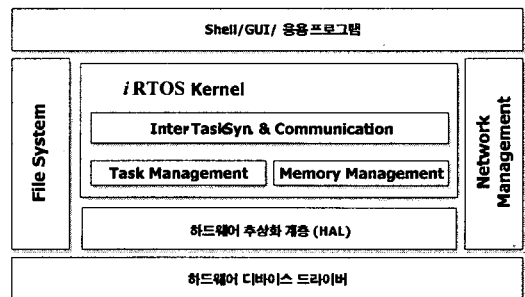
데이터 처리 속도와 전송속도의 눈부신 향상으로 많은 정보를 네트워크를 통해 제공받게 되었지만 네트워크는 한정되어 데이터 전송의 효율을 높이는 방안이 많이 고안되고 있다. 네트워크로 모든 기기들을 연결하는 유비쿼터스 환경에서 이러한 효율적인 연은 매우 중요한 문제가 되고 있다. 동일한 데이터를 여러 호스트에 전송시 기존의 브로드캐스트 방식은 같은 데이터를 여러 번 전송하는 방식인데 반해, 한번 전송으로 같은 그룹에 가입한 다수의 호스트가 전송 받는 멀티 캐스트 송수신 기법은 네트워크의 효율성을 높일 수 있어 스트리밍 서비스 같은 실시간 전송에서 많이 사용되고 있다. 본 논문에서는 실시간 운영체제 RTOS<sup>TM</sup>에 탑재된 TCP/IP 프로토콜 스택에 멀티캐스트 송수신을 위한 IGMP 프로토콜을 설계 및 구현하였다[1][2][3]. 본 논문의 구성은 2 장에서 실시간 운영체제 RTOS<sup>TM</sup>와 TCP/IP 를 소개하고, 3 장에서 멀티캐스트 송수신을 위한 IGMP 프로토콜과 Socket API 설계 및 구현을, 4 장에서 테스트 환경 및 결과에 대하여 기술한다. 마지막으로, 5 장에서는 결론 및 향후 연구 계획에 대해 기술한다.

#### 2. 관련 연구

##### 2.1 실시간 운영체제 RTOS<sup>TM</sup>

RTOS<sup>TM</sup>는 충남대학교 시스템 소프트웨어 연구실에서 개발한 선점형 우선순위 기반의 실시간 운영체제 (Real-time Operating System)이다. 태스크는 중요도에 따라 우선순위가 부여되고 가장 높은 우선순위의 태스크가 CPU 를 선점하여 수행된다. 실시간 운영체제 RTOS<sup>TM</sup>의 기본 기능은 다음과 같다.

- 우선순위 기반 멀티 태스킹
- 동적 메모리 관리
- 태스크간 통신



[ 그림 1 ] RTOS<sup>TM</sup> 의 구조

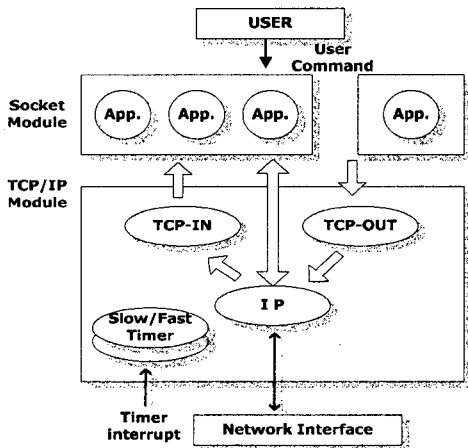
\* 이 논문은 산업자원부 중기거점과제 연구비 지원에 의한 것임

### 2.2 TCP / IP Protocol Stack

퍼스널 컴퓨터의 TCP/IP 프로토콜에 비해 임베디드 시스템 정보기기에서는 경량의 TCP/IP 스택을 필요로 한다. 본 논문에서는 RTOS™에 현재 구현되어 있는 TCP/IP 프로토콜 스택을 사용하였다.

기존의 RTOS™에 구현된 TCP/IP 의 구성내용은 다음과 같다.

- Ethernet
- IP (Internet Protocol)
- TCP (Transfer Control Protocol)
- UDP (User Datagram Protocol)
- ARP (Address Resolution Protocol)
- ICMP (Internet Control Management Protocol)



[ 그림 2 ] TCP/IP in RTOS™

### 2.3 멀티캐스트 (IGMP)

VOD 서비스와 같이 동시에 다수의 클라이언트에게 대용량 멀티미디어 데이터를 전송하는 경우 네트워크 자원의 효율적인 이용을 위해 멀티캐스팅에 기반한 데이터 전송이 필요하다[4][5].

멀티캐스트는 일-대-다 또는 다-대-다 전송을 지원한다. 이를 위해서 아래의 기본요소를 만족해야 한다.

- 멀티캐스트 호스트 그룹의 정의가 있어야 한다. 현재 인터넷 주소 체계에서는 'class D'로 지원을 하고 있다.
- 해당 세션에 대한 그룹의 가입/탈퇴 기능을 수행하는 그룹관리 메커니즘이 필요하며, IGMP 프로토콜이 해당기능을 수행하고 있다.
- 멀티캐스트 데이터를 전달하는 경로를 설정해야 한다.

### 3. 멀티캐스트 설계 및 구현

#### 3.1 IGMP

멀티캐스트 전송을 위해서 호스트그룹을 관리하는 IGMP (Internet Group Management Protocol) 프로토콜이 필요하다. IGMP 프로토콜은 호스트그룹에 가입/탈퇴를 처리하는 부분과 독립된 태스크로 동작하여 주기적으로 그룹을 관리하는 부분으로 구성된다.

##### 3.1.1 Host Group 가입

특정한 멀티캐스트 채널에 가입하여 데이터를 송수신하고자 할 경우 ARP 테이블에 해당 그룹의 MAC 주소를 등록한 후, IGMP Packet 을 전송한다. 이 그룹 가입 패킷을 라우터가 받아 이후에 수신된 가입된 그룹의 패킷을 해당 서브넷으로 포워딩해준다

##### 3.1.2 Host Group 탈퇴

멀티캐스트 그룹에서 탈퇴 할 경우 라우팅 테이블과 ARP 테이블에서 탈퇴하고자 하는 그룹의 엔트리를 삭제한다. 라우터는 주기적으로 회원 질의(Membership Query)를 서브넷으로 보내 해당하는 그룹에 가입된 호스트가 있는지 확인한다. 그런데 이때 회원 보고(Host Membership Report)가 오지 않으므로, 해당 그룹의 패킷을 서브넷으로 포워딩 시키지 않는다.

IGMPv2 의 경우, 탈퇴 메시지가 있으나 본 논문에서는 IGMPv1 으로 구현하여 탈퇴 메시지를 고려하지 않았다.

### 3.2 멀티캐스트 API

사용자가 멀티캐스트 송수신을 위해 그룹에 가입하기 위한 소켓 응용프로그램 인터페이스(Socket API)가 필요하다.

```

MK_GetSockOpt(int fd, int level, int optname, char
*optval, int *optlen);
MK_SetSockOpt(int fd, int level, int optname, char
*optval, int optlen);
    
```

[ 그림 3 ] 멀티캐스트 API 의 원형

MK\_GetSockOpt()는 현재 소켓의 설정을 보여주는 함수이며, MK\_SetSockOpt() 함수는 소켓의 설정을 수정하는 함수이다. 첫번째 인자 fd 는 소켓의 식별자이고 level 은 수정하고자 하는 소켓이 처리되어지는 Layer 를 의미한다. optname 은 수정하거나 출력할 설정값이 어떤 값인지를 나타낸다.

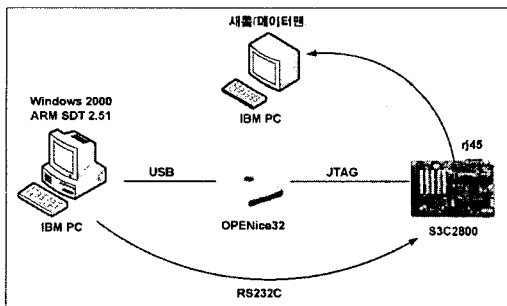
Optname	설명
IP_MULTICAST_LOOP	데이터를 호스트로 루프백할지 결정.
IP_MULTICAST_TTL	서브넷 내부 혹은 외부로의 전송 여부 결정.
IP_MULTICAST_IF	멀티캐스트 패킷이 처리되는 인터페이스 설정.
IP_ADD_MEMBERSHIP	멀티캐스트 그룹 등록.
IP_DROP_MEMBERSHIP	멀티캐스트 그룹 탈퇴.

[ 표 1 ] 멀티캐스트 관련 소켓 옵션

#### 4. 테스트 환경 및 결과

##### 4.1 테스트 환경

본 논문은 ARM920T 기반의 S3C2800 32-Bit RISC MicroProcessor 에 RTOS™를 탑재하여 테스트 하였다. 컴파일러는 ARM SDT 2.51 을 사용하였고 디버거는 Aiji System 의 OPENice32 A900 을 사용하였다. 멀티캐스트 송수신 테스트를 위해 2 대의 리눅스 서버를 추가로 사용하였다.

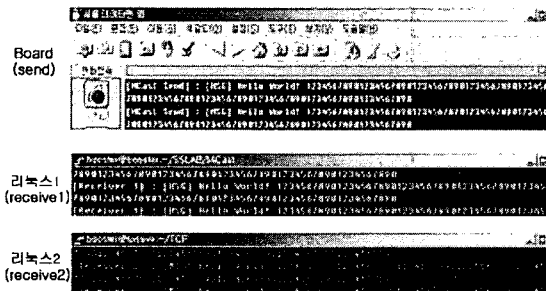


[ 그림 4 ] 테스트 환경

#### 4.2 결과

##### 4.2.1 멀티캐스트 송신

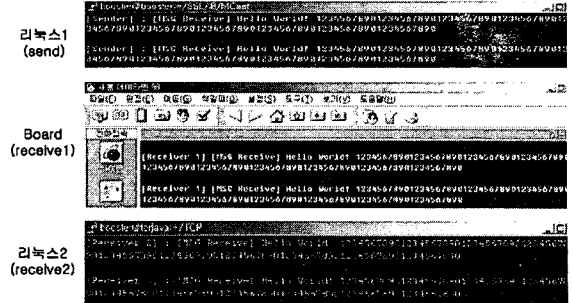
개발보드를 송신으로 두 대의 리눅스 서버를 수신 모드로 하여 테스트 하였다.



[ 그림 5 ] 멀티캐스트 송신 테스트

##### 4.2.2 멀티캐스트 수신

리눅스 서버 1 을 송신 모드로 리눅스 서버 2 와 개발보드를 수신 모드로 하여 테스트 하였다.



[ 그림 6 ] 멀티캐스트 수신 테스트

#### 5. 결론 및 향후 연구 과제

본 논문에서는 다중송수신시 동일한 데이터의 중복 전송으로 인한 네트워크 과부하를 해결할 수 있는 멀티캐스트 프로토콜을 설계 및 구현하였다. 향후 연구 과제로는 향상된 IGMPv2 프로토콜을 지원을 위해 기능을 추가하는 것과 송수신시 발생하는 손실률을 줄이기 위한 여러 처리부분을 구현하는 것이다.

#### 6. 참고 문헌

- [1] <http://www.aijssystem.com>
- [2] David Stegner 외 2 명, "Embedded Application Design Using a Real-Time OS", IEEE, 1999
- [3] C.M.Krishna, Kang G.Shin, "Real-Time Systems", The McGraw-Hill Companies, Inc.1997
- [4] S.Deering, "Host Extensions for IP Multicasting", RFC 1112, 1989
- [5] W.Fenner, "Internet Group Management Protocol, Version 2", RFC 2236, 1997