무선통신기반 온라인 게임을 위한 네트워크 엔진 연구개발

손성옥[0] , 아미루자만, 노재춘
세종대학교 컴퓨터공학과

The Network Engine Research and Development for Wireless based on Online Game

Sung-Ok Son[0], Amiruzzaman Md., Jaechun No
College of Electronics & Information Engineering
School of Computer Engineering Sejong University, Seoul, Korea
sonsuok@sejong.ac.kr, m_amiruzzaman@email.com

요 약

본 논문에서는 802.11b의 네트워크 성능 측정을 통해 도출된 결과를 토대로 하여 802.11b기반의 온라인 게임을 위한 네트워크 엔진을 개발하였다. 네트워크 엔진 서버는 I/O Multiplexing 기법을 사용하였으며, 클라이언트는 .Net Compact Framework 로 개발하였다.

1. 서 론

본 논문에서는 무선 통신(802.11b)을 기반으로 하는 온라인 게임을 위한 네트워크 엔진을 개발을 하였다. 네트워크 엔진을 개발하기 위하여 TCP/IP 기반의 ICMP를 사용하여 802.11b의 네트워크 성능을 측정하였다. 측정은 2가지 방향(Server ↔ PDA, PDA ↔ PDA)으로 나눠서 실시하였고, 본 논문에서 실시 한 측정 결과를 토대로 하여 네트워크 엔진을 개발하였다. 네트워크 엔진이 구동될 서버의 운영체제는 Linux 이며, 네트워크 엔진은 BSD Socket 기반으로 한 I/O Multiplexing 기법으로 구현 하였다. 그리고 게임부분 해당되는 클라이언트는 802.11b를 지원하는 PDA를 사용하였다. PDA의 운영체제는 Widows Mobile 5.0 이며, .NET Compact Framework 개발환경에서 WinSocket으로 네트워크 엔진을 구현하였다.

2. 관련 연구

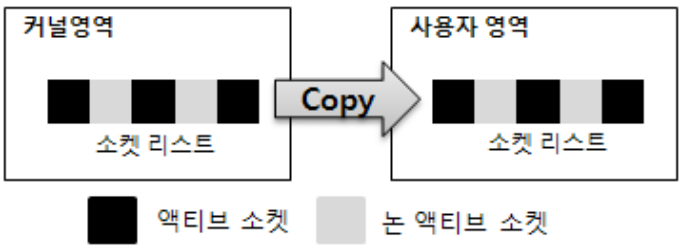
802.11b의 성능측정을 위하여 ICMP를 이용하였고, 네트워크 엔진 구현 방식은 I/O Multiplexing 방식을 사용하였다. 클라이언트는 .Net Compact Framework 개발환경에서 MicroSoft사에서 제공하는 Windows Socket을 사용하였다.

2.1 ICMP(Internet Control Message Protocol)

일반적으로 ICMP(인터넷 제어 메시지 프로토콜)[1]라고 하면 데이터 통신 중 패킷이 여러 요인으로 인해 버려지거나 훼손될 경우 패킷 훼손에 대한 오류 대처방법으로 흐름제어 기법이나 재전송 기법 등의 여러 가지 진단기능을 수행할 수 있도록 구현되어있는 프로토콜을 ICMP프로토콜(RFC. 792)이라고 하며, OSI 기본 참조 모델의 망 계층에 해당한다. ICMP 프로토콜의 대표적인 예로 ping이 있다.

2.2 I/O Multiplexing

온라인 게임서버가 되기 위한 최소한의 요구사항이라고 할 수 있는 다중접속 서버로[2][3]구현되어야하는데 Linux 에서 가장 기본이 되는 다중 접속 서버 구현 방법은 I/O Multiplexing(입출력 다중화) 방법이며, BSD Socket의 Select()와 Poll()이 대표적이다.



[그림 1] Select()

커널 영역에 등록된 소켓 리스트는 사용자 영역으로

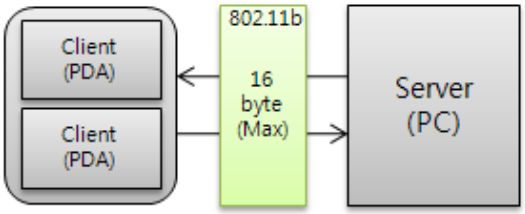
● 본 논문은 Seoul R&BD Program(10557) 지원을 받아 수행 되었음

복사된 뒤 Select()에 의해 사용자 영역 소켓 리스트 목록을 일정주기로 순회하며 소켓을 비교감시 하는 방식이다[1][2].

2.3 .NET Compact Framework

.NET Compact Framework는[5] 스마트 장치(PDA,휴대폰 등)용 응용 프로그램을 개발할 때 MS사에서 제공되어지는 개발환경이다. 따라서 .NET Compact Framework는 스마트 장치용 응용프로그램을 개발할 때 필요한 클래스 라이브러리를 제공 한다.

3. 네트워크 엔진 개발



[그림 2] 전체 구조

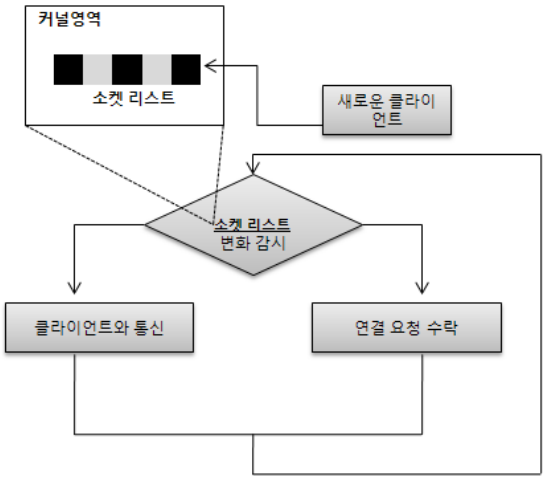
본 논문에서 802.11b의 성능 측정 결과 16byte가 가장 적합 하였다. 802.11b의 자세한 성능측정 사항은 4.1성능측정에 나타나 있다. 802.11b 성능 분석을 토대로 네트워크 엔진 서버의 패킷 최대크기는 16byte하였고, 서버엔진 구조는 PDA의 제한된 리소스자원 때문에 Sever로의 중앙 집중식 방법으로 구현하였다[3]. 개발환경은 [표 1]과 같다.

[표 1] 개발환경

	Server(PC)	Client(PDA)
OS	Linux Kernel 2.6.11	Windows Mobile 5.0
컴파일러	gcc 4.0.0	Visual Studio 2005
Socket	BSD Socket	Winsock

3.1 Server(PC) 구현

서버는 I/O Multiplexing 사용하였으며, [그림 3]에 서버 구조가 나타나 있다. 커널영역 로부터 복사된 사용자 영역의 소켓리스트는 Select()가 감시 하다가 Socket의 변화가 생길 때 새로운 클라이언트의 연결요청인지 클라이언트와 통신인지 판단하도록 구현하였다.



[그림 3] 서버 구조

클라이언트(PDA)로 부터 받은 패킷은 서버에서 필요한 정보만 처리 하여 다른 클라이언트로 전송하여 클라이언트들 간의 동기화를 맞춘다. 이모든 과정에서 사용된 패킷크기는 16byte를 넘지 않도록 구현 하였다.

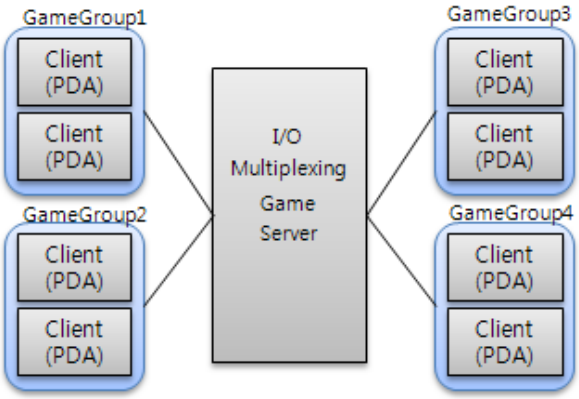
[표 2] I/O Multiplexing

```
while(1)
 {
   time.tv_sec = 3;
   select(sock_list+1, &temp, 0, 0, &time)

   for(sock=0; sock<sock_list+1; sock++)
   {
     if(FD_ISSET(sock, &temp))
     {
       if(sock==serv_sock) { /*클라이언트 연결*/
         client_len = sizeof(client_addr);
         client_sock = accept(serv_sock,
         (struct sockaddr *)&client_addr, &client_len);
         /*Socket list 변경*/
         FD_SET(client_sock, &reads);
          if(sock_list<client_sock)
              sock_list=client_sock;
         /*플레이어 접속 처리*/
          player.count++;
          player.fde[player.count]=client_sock;

       }
     }
 }
```

3.2 Client(PDA) 구현

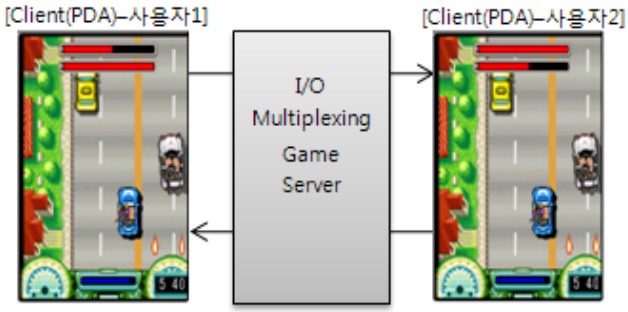
I/O Multiplexing 방법으로 개발한 게임은 Racing 게임

으로 클라이언트(PDA)는 두 명씩 그룹을 이루어 게임을 한다. 게임하는 동안 클라이언트들 간에 실시간으로 클라이언트들 간에 정보를 주고받는다.



[그림 4] 클라이언트 구조

클라이언트의 구현된 화면은 [그림 5]아래와 같다.



[그림 5] 클라이언트

사용자1과 사용자2는 같은 게임그룹에서 실시간으로 주고받은 데이터 정보중 상대방의 적 정보를 실시간으로 화면에 나타내도록 구현하였다.

4. WIFI(802.11b) Performance 측정

모바일 네트워크 기반의 게임서버 만들기 위해서 네트워크 성능측정을 하였다.

4.1 성능측정

성능측정을 위하여 802.11b기반 PDA를 통해 일정크기의 패킷신호를 PDA로 송신하여 패킷이 되 돌아오는 시간을 측정하였다. 평균응답 시간은 각 패킷이 전송 후 되돌아오기까지 걸린 시간을 말하며 측정 단위는 ms(milli-second) 로 표시하였다. 측정방법은 ICMP를 이용하였고 성능측정은 양방향으로 수행하였다.

1) Server(PC) to Client(PDA)

2) Client(PDA) to Client(PDA)

[표 3]성능측정 환경

	PC	PDA
제품종류	Desktop PC	HP Rx1950
제품사양	Intel Pentium 4	Samsung 300Mhz
네트워크	Ethernet 10/100MB	802.11b (11Mbps max)

4.1.1 Server(PC) to Client(PDA)

Server(PC) to Client(PDA)의 측정을 위하여 1000개의 패킷을 16byte부터 순차적으로 증가시켜 65,500byte (64k)까지 Server에서 Client로 전송하였다. 측정 결과 패킷의 크기가 아래 [표 4]와 같이 16byte일 때 2ms부터 순차적으로 증가하여 65,500byte(64k)일 때는189ms로 측정되었다. 온라인 게임 서버는 응답속도가 250ms를 넘지 않도록 하여야 함에 따라 Server(PC) to Client(PDA 측정에서 안정적인 범위의 패킷 크기는 16byte부터 65,500byte(64k)로 나타났다[4].

[표 4] Server(PC) to Client(PDA) 측정

패킷크기 (byte)	Packets Sent	Packet Lost	Avg (ms)
16	1000	0	2
128	1000	0	2
256	1000	0	3
512	1000	0	4
1,024(1k)	1000	2	5
2,048(2k)	1000	1	8
4,096(4k)	1000	2	16
8,192(8k)	1000	1	34
16,384(16k)	1000	1	49
32,764(32k)	1000	2	96
65,500(64k)	1000	2	189

4.1.2 Client(PDA) to Client(PDA)

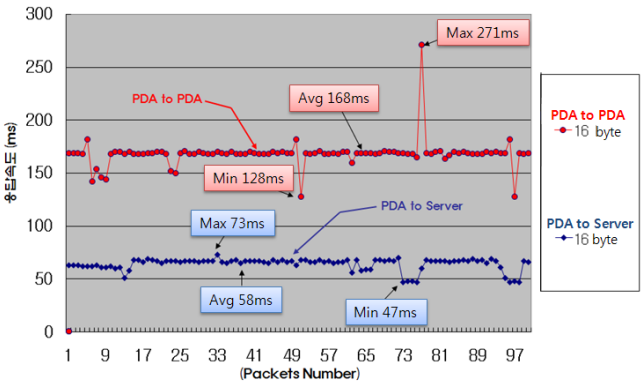
Client(PDA) to Client(PDA)의 측정을 위하여 100개의 패킷신호를 16byte크기로 Client(PDA)에서 Client(PDA)로 전송하였다. 측정결과 [표 5]와 같이 100개의 패킷신호를 16byte크기로 전송하였을 때 응답속도는 Min 128ms 이며, Max 271ms로 측정되었다. 이 측정결과에

따라 평균은 168ms로 나타났다. 그러므로 온라인 게임 서버가 되기 위한 응답속도 250ms 내에 들게 되어 안정적인 범위가 된다.

[표 5] Client(PDA) to Client(PDA) 측정

패킷크기(byte)	Packets Sent	응답 속도(ms)
16	100	Min 128
		Avg 168
		Max 271

5. 결 론



[그림 6] 성능측정 종합

성능측정을 하기위해서 패킷크기를 달리하여 1000개의 패킷을 Server to PDA 전송 측정하였고 100개의 패킷을 PDA to PDA와 PDA to Server로 양방향으로 전송한 뒤 되돌아오는 응답속도를 측정하였다. 패킷크기 중 16byte크기의 패킷을 전송하였을 때 PDA to PDA와 PDA to Server의 응답속도를 비교하였다. PDA to PDA에서는 Max 271ms, Min 128ms로 측정되어 Avg 168ms 로 나타났고, PDA to Server는 Max 73ms, Min 47ms로 측정되어 Avg 58ms로 나타났다. 위 측정 결과를 종합해 볼 때 16byte 인 경우에 온라인 게임서버가 되기 위한 안정적인 응답속도 250ms 미만의 측정값을 얻을 수 있었다. 또한 각 패킷간의 전송간격이 일정하면서 짧은 주기로 통신 하였을 경우에 좀 더 신뢰성 있는 무선 네트워크 통신을 할 수 있었다.

성능측정 결과 본 논문에서는 802.11b 기반에서 온라인 게임서버가 되기 위한 패킷 최대크기는 16byte라는 결과를 얻었다. 그리고 이를 토대로 온라인 게임을 위한 네트워크 엔진을 I/O multiplexing방법으로 개발 하였다.

본 논문에서는 네트웍 엔진을 I/O multiplexing방법으로 개발 하였지만, I/O multiplexing은 소켓리스트가 길어지면 Socket을 순회하는 주기 시간이 길어지는 등의 보안해야하는 사항이 있어 향후에는 본 논문에서 측정한 결과를 토대로 좀 더 안정적이고 효율적인 IOCP 나 Realtime signal 방법으로 네트워크 엔진 구현할 예정이다.

참 고 문 헌

[1]STEVENS, FENNER, RUDOFF “Unix Network Programming", AddisonWesley

[2] 이홍기 “Linux RealTime Signal 기반의 온라인 게임 서버 엔진의 구현”, 2003

[3] 신동원 “A Study on Online Game Server Architecture" KGDA 네트워크

[4] 홍은실, 한미라, 최양희 “Traffic Analysis of Online Game Traffic", 2002

[5]http://msdn2.microsoft.com/ko-kr/library/9s7k7ce5(VS.80).apx .NET Compac Framework