

사용자 레벨 가상화에서 가상화 영역 성능 비교

정찬주* · 강태근**

Performance Comparison of Virtualization Domain in User Level Virtualization

Chan-Joo Jeong* · Tae-Geun Kang**

요 약

본 논문은 가상화 기술에서 클라이언트 기반의 데스크톱 가상화에 필요한 기술 요소를 찾아내고 로컬 컴퓨팅 환경에서 보다 편리하고 안정된 클라이언트 가상화 기술을 제안하였다. 유저 레벨 가상화와 VMWare를 로딩하여 처리 익스플로러 유틸리티를 실행시킨 후 각 프로세스의 Private 바이트를 비교해보면 VMWare에서는 30.1MB의 메모리 용량이 사용되었고, 유저 레벨 가상화 프로세스에서는 16.6MB의 메모리 용량이 사용되어짐을 알 수 있었다. 클라이언트 가상화를 이용한 가상화 사용자 영역과 로컬 컴퓨팅 환경에서 응용 프로그램을 실행시킨 후 CPU 사용율을 비교한 결과 큰 차이가 없음을 알 수 있었다. 이러한 결과는 가상화된 사용자 영역에서 로컬 컴퓨팅 환경의 성능 저하를 최소화 하여 사용자가 필요한 기능을 효율적으로 사용할 수 있음을 알 수 있었다.

ABSTRACT

In this paper, we proposed new virtualization technology that is more convenient and stable in local computing environment, then found technique elements need to desktop virtualization which is based on clients in various virtualization technologies. After running execution of process explorer utility in user level virtualization and VMWare, we found memory capacity that is used 30.1MB in VMWare and 16.6MB in user level virtualization respectively to compare private bytes each of process. We found no significant difference of CPU utilization which is executed application program in local computing environment and user domain with user level virtualization. In this result, proposed virtualization technology is able to minimize performance degradation of local computing environment.

키워드

Virtualization, Local Computing, Kernel Level Virtualization, VMWare
가상화, 로컬 컴퓨팅, 커널 레벨 가상화, 가상머신웨어

1. 서론

가상화 기술은 하나 혹은 여러 개의 물리적 시스템에서 실제 물리적 시스템과 독립적으로 복수 개의 논

리적 시스템을 제공하는 기술로서 초기에 대규모 시스템을 개별 논리적 독립 시스템으로 보이게 하여 시스템의 개별화를 목적으로 사용된 이후 최근에는 시스템의 신뢰성, 안정성, 확장성을 보장하기 위해 가상

* 동강대학 전기에너지과(myforks@naver.com)

** 교신저자(corresponding author) : 한국전자부품연구원 광주지역본부(tgkang@keti.re.kr)

접수일자 : 2013. 08. 26

심사(수정)일자 : 2013. 11. 11

게재 확정일자 : 2013. 11. 15

화 기술이 활용되기 시작하였다[1][2].

로컬 컴퓨팅 환경에서 가상화는 시스템을 운영하는 과정에서 자원의 고갈이나 성능저하, 업그레이드 등과 같은 문제를 해결하고, 가상머신의 독립된 환경을 구성하여 편리한 개발 환경을 구축하여 내부 결함으로 인해 가상 머신 자체 문제가 발생하거나 서비스 장애가 발생하더라도 신속하게 대응하고 복구할 수 있어야 한다[3][4].

본 논문에서는 이러한 요구 사항들을 반영함과 동시에 클라이언트 가상화 기술을 기반으로 로컬 컴퓨팅 환경에 독립적으로 가상화된 사용자 영역을 구현할 수 있는 방법을 제시하고자 한다.

II. 로컬 컴퓨팅 가상화

그림 1에서와 같이 로컬 컴퓨팅 환경의 하드웨어와 호스트 운영체제 안에 각각의 가상 하드웨어가 존재한다. 가상 하드웨어에는 가상 환경에 따라 운영 체제와 응용 프로그램 설치가 필요하여 운영체제와 응용 프로그램의 라이선스 비용과 설치가 필요하다.

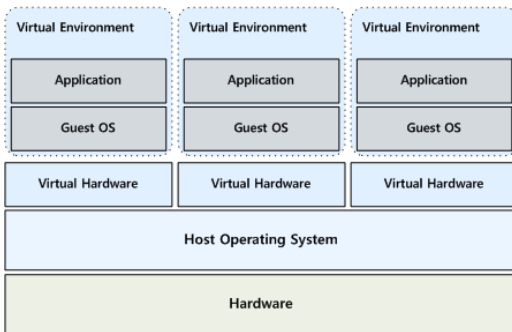


그림 1. 하드웨어 가상화
Fig. 1 Hardware virtualization

커널 레벨 가상화는 각각의 가상 환경이 생성하는 시점에 커널을 구성하고 응용 프로그램을 설치해야 한다. 그림 2와 같이 커널 레벨 가상화를 통해 생성된 가상 환경에서는 커널을 공유하지 않으므로 로컬 컴퓨팅 환경에 설치되어 있는 응용 프로그램 뿐만 아니라, 서로 다른 가상 환경의 응용 프로그램을 공유하는 것이 어렵다. 커널 레벨 가상화는 운영체제 내부 구조

에 의존적이어서 운영체제의 버전에 따라서 구조적인 변경이 반드시 필요하다.

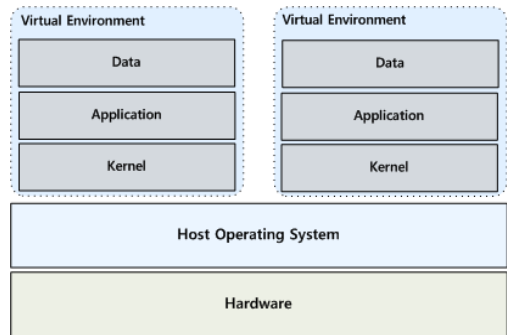


그림 2. 커널 레벨 가상화
Fig. 2 Kernel level virtualization

III. 커널모드와 유저 모드의 구조

로컬 컴퓨팅 환경에서 동작하는 잘못된 응용 프로그램으로부터 시스템 보호를 위해 운영체제는 운영체제의 코드를 실행하는 커널 모드와 사용자 응용 프로그램을 실행하는 유저 모드를 사용한다. 유저 모드에서 실행되는 응용 프로그램은 제한된 인터페이스의 집합만이 가능하고 시스템 데이터에 대한 액세스를 제한하고 하드웨어에 대한 직접적인 접근이 불가능하다. 유저 모드 프로그램이 시스템 서비스를 호출할 때 프로세서는 호출을 가로채서 쓰레드를 커널 모드로 전환하도록 호출한다. 시스템 서비스가 완료될 때 운영체제는 쓰레드 컨텍스트를 다시 유저 모드로 전환하고 호출자가 계속해서 실행하도록 한다.

윈도우는 OS/2, POSIX, Win32의 환경 서브시스템을 가지고 있다. 윈도우가 운영체제로 동작하기 위해 모든 코드를 가진다면 많은 양의 중복 함수를 가지기 때문에 시스템 크기와 성능에 대한 부정적인 영향을 고려하여 Win32가 주 서브시스템으로 설계되었다. 다른 두 서브시스템들은 요구에 의해 구성되는 반면 Win32 서브시스템은 항상 실행해야 한다. 윈도우의 환경 서브시스템 프로세스에서는 콘솔 윈도우들, 프로세스와 쓰레드의 생성과 종료, 16비트 가상 도스 머신 프로세스에 대한 지원, 내부 언어 지원함수들을 지원한다[5][6].

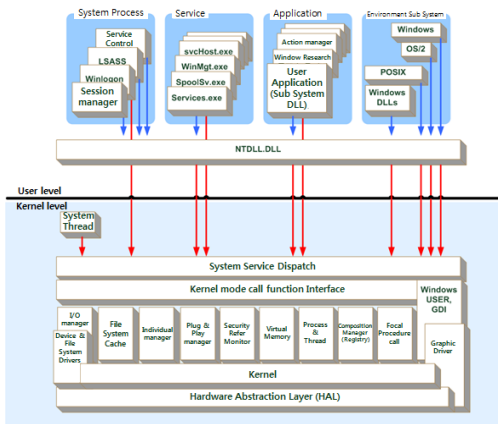


그림 3. 윈도우 아키텍처
Fig. 3 Window architecture

커널 모드 장치 드라이버는 윈도우 디스플레이 제어, 화면 출력 관리, 키보드와 마우스를 비롯한 다른 장치들로부터의 입력 수집, 응용 프로그램에 사용자 메시지 전달을 지원하고 GDI(Graphics Device Interface) 함수를 포함한다[7][8][9].

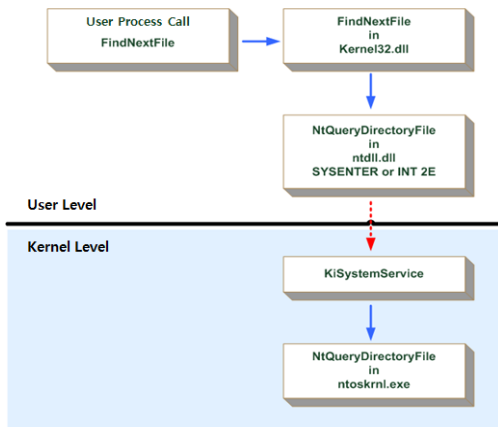


그림 4. FindNextFile의 실행 흐름
Fig. 4 FindNextFile execution flow

그래픽스 장치 드라이버들은 하드웨어 종속적인 그래픽스 디스플레이 드라이버, 프린터 드라이버, 비디오 미니포트 드라이버 등이다. 로컬 PC에서 실행되는 응용 프로그램은 직접 Win32 주 서비스를 호출하지 않고, 하나 이상의 서비스 시스템 DLL들을 통해서 호출한다. 서비스 시스템 DLL들은 문서화된 윈도우 API

함수들을 문서화 되지 않은 커널 모드 시스템 서비스를 호출로 변환한다.

Kernel32.dll, User32.dll, Advapi32.dll, Gdi32.dll과 같은 윈도우 서브시스템 DLL들은 윈도우 API 함수들을 구현해 놓은 것이다. 그림 4는 유저 레벨에서 윈도우 서브시스템 DLL내의 함수를 호출하여 커널 레벨로 진입하여 실행하는 과정을 나타낸다. 로컬 PC에서 동작하는 응용 프로그램은 윈도우 서브시스템 DLL을 로드 하고 그것의 임포트 어드레스 테이블(IAT)에서 각 함수들의 메모리 주소를 복사한다. 디렉토리 내의 파일을 나열하기 위해 응용 프로그램이 Kernel32.dll에서 제공하는 FindFirstFile API를 호출하면 IAT에 있는 FindNextFile 함수의 주소로 이동하여 Ntdll.dll을 호출한다. Ntdll.dll은 윈도우 서브시스템 DLL을 사용하기 위한 특별한 주요 시스템 지원 라이브러리이며 윈도우 실행부 시스템 서비스에 대한 시스템 서비스 디스패치 스택과 서브시스템, 서브시스템 DLL, 다른 네이티브 이미지에 의해 사용되는 내부 지원 함수들을 포함한다. Ntdll.dll은 FindNextFile에 대응하는 커널 레벨 함수인 NtQueryDirectoryFile의 번호를 EAX 레지스터에 로드하고 EDX 레지스터에는 FindNextFile 인자의 유저 레벨 주소를 로드한다. SYSETER와 INT 2E 명령을 이용하여 Ntdll.dll은 커널 레벨로 진입한다. 윈도우 서브시스템 DLL이 응용 프로그램 메모리 주소 공간에 로드되기 때문에 응용 프로그램의 메모리 주소 공간에 접근하게 되면 API를 변경시키거나 응용 프로그램 임포트 테이블 변경이 가능한데 이것이 API 후킹이다. API 후킹을 이용하면 프로세스나 네트워크 포트를 숨길 수 있고 파일에 대한 쓰기가 다른 파일에서 이뤄지도록 할 수 있으며 응용 프로그램이 특정 프로세스의 핸들을 구하지 못하게 할 수 있다. 또한, API 호출을 로그로 남긴다면, 응용 프로그램이 어떤 식으로 동작하는지 알 수 있다.

IV. 가상화 영역의 성능비교

가상화 연구를 통해 시장을 점유하고 있는 대표적인 벤더는 VMWare와 Microsoft 그리고 Xen 등이 있다. 본 절에서는 이 세 개의 가상화 벤더들 중에서

현재 가상화 시장의 70% 이상을 점유하고 있는 VMWare와 본 논문에서 구현된 유저 레벨의 가상화의 성능을 평가하고 비교 분석한다. 성능을 평가하기 위하여 동일한 Windows XP(SP 3) 시스템 테스트 환경에서 구현된 유저 레벨 가상화와 Guest OS로 Windows XP를 설치한 VMWare를 구동시키고 테스트 환경의 CPU와 메모리의 변화를 측정하고 그 결과를 비교 분석하였다.

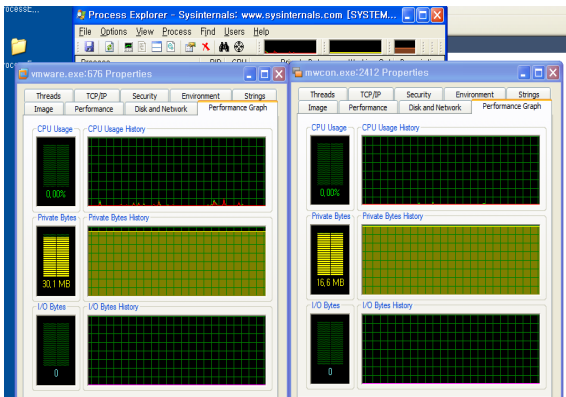


그림 5. Process explorer 실행
Fig. 5 Process explorer execution

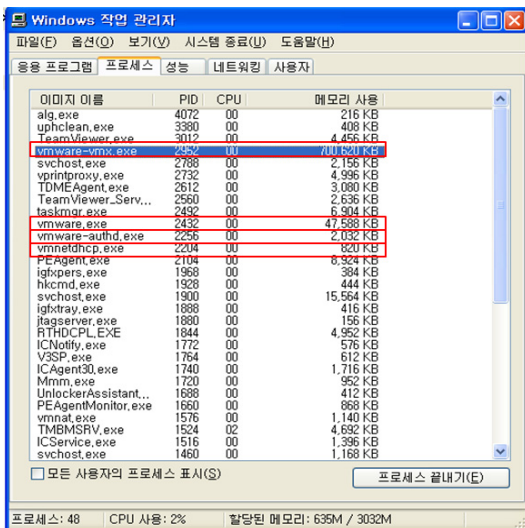


그림 6. VMWare에서 guest OS 구동에 따른 메모리 변화
Fig. 6 Memory change of guest OS running on the VMWare

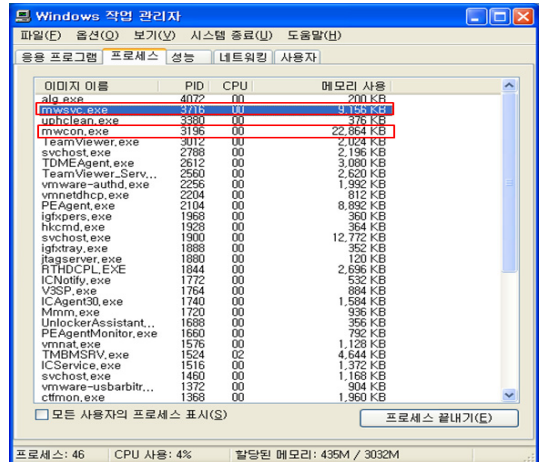


그림 7. 유저 레벨 가상화(u-level) 구동에 따른 메모리 변화
Fig. 7 Memory change of guest OS running on the VMWare

그림 5와 같이 프로세스의 동작을 확인하는 프로그램인 Process Explorer 유틸리티를 실행시키고 유저 레벨 가상화와 VMWare를 로딩하였다. 각 프로세스의 Private Bytes를 비교해보면 VMWare는 30.1MB를 유저 레벨 가상화 프로세스는 16.6MB를 유지하고 있다. 그림 6은 VMWare에서 Guest OS를 구동시켰을 때의 프로세스 메모리 변화량을 나타낸 것이고 그림 7은 유저 레벨 가상화(u-level) 프로세스의 메모리 변화량을 나타낸 것이다. Guest OS(Windows XP) 부팅으로 VMWare는 vmware-wmx.exe 프로세스를 생성하게 되며 해당 메모리의 사용량은 700,620KB로 VMWare의 가상 머신에서 Guest OS를 구동하기 위해 필요하다.

반면, 유저 레벨 가상화 프로세스는 테스트 환경에서 애플리케이션 형태로 동작하기 때문에 VMWare와 같이 Guest OS를 구동하기 위한 메모리와 같은 코일 자원을 필요로 하지 않는다.

그림 8과 그림 9는 VMWare 프로세스의 메모리 사용량과 유저 레벨 가상화(u-level) 프로세스의 메모리 사용량을 막대 그래프로 나타낸 것이다. 막대 그래프에서도 나타나듯이 VMWare 프로세스 메모리 사용량과 유저 레벨 가상화(u-level) 프로세스의 메모리 사용량의 확연한 차이를 확인할 수 있다.

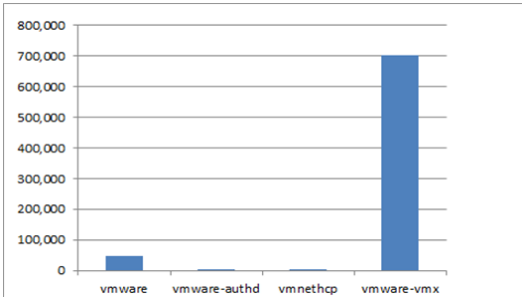


그림 8. MWare 프로세스의 메모리 사용량
Fig. 8 Memory usage of VMWare process

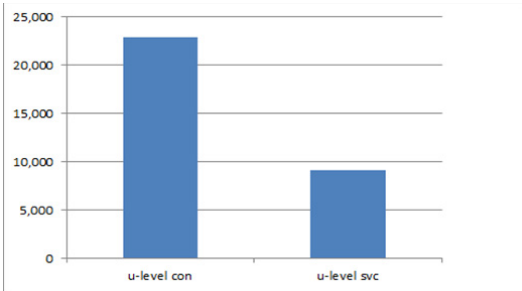


그림 9. 유저 레벨 가상화(u-level) 프로세스의 메모리 사용량
Fig. 9 Memory usage of user-level virtualization (u-level) process

표 1. VMWare와 유저 레벨 가상화(u-level) 프로세스의 메모리 사용량 비교

Table 1 VMWare and user-level virtualization(u-level) compared to the process's memory usage

Virtualization domain Classification	Process	Usage Memory (KB)
VMWare	vmware	47,588
	vmware-authd	2,032
	vmnethcp	820
	vmware-vmx	700,620
User-level Virtualization (u-level)	u-level con	22,864
	u-level svc	9,156

표 1은 VMWare와 유저 레벨 가상화(u-level) 프로세스가 동작할 때의 사용 메모리를 나타낸 것이고, 그림 10은 VMWare의 Guest OS를 부팅시 VMWare 프로세스의 CPU 변화를 나타내고 있다.

로컬 컴퓨팅 환경에서 가상화된 사용자 영역의 테스트를 위하여 가상 데스크톱으로 사용하기 위한 가상화된 사용자 영역의 생성이 필요하다. 먼저 가상화 사용자 영역을 확보하기 위해서는 프로그램을 실행하여 가상화 공간을 설정해야 한다. 또한 이렇게 생성되는 가상화된 사용자 영역을 설정함에 있어 로컬 컴퓨팅 환경에서 사용 가능한 공간의 크기를 초과해서 생성할 수 없다.

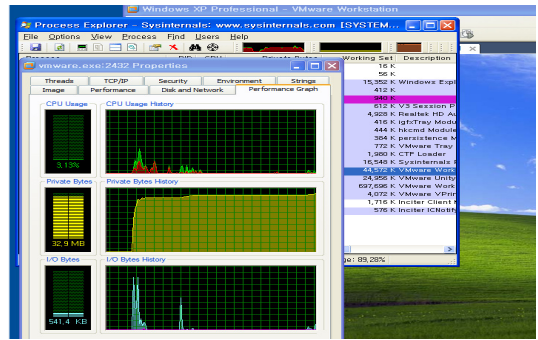


그림 10. VMWare에서 Guest OS 구동에 따른 CPU 변화
Fig. 10 CPU change of guest OS running on the VMWare

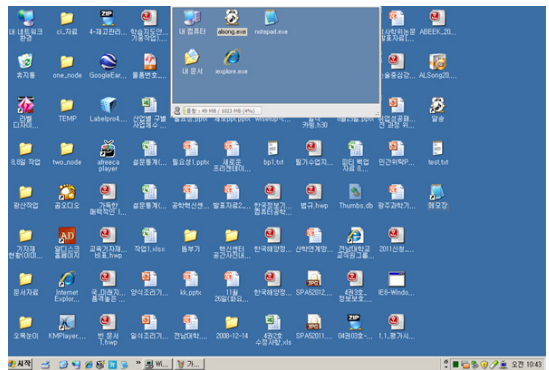


그림 11. 가상화 사용자 영역 실행
Fig. 11 Execution of virtualization user domain

가상화된 사용자 영역에서의 사용 공간은 생성 시간을 고려하여 1G Byte의 크기로 생성하도록 하였다. 이렇게 생성된 가상화된 사용자 영역은 사용자의 삭제 명령을 통해 실시간 삭제 및 재생성도 가능하지만, 중복 생성은 허용하지 않는다. 이것은 로컬 컴퓨팅 환경에서 가상화된 사용자 영역을 오직 하나만 존재할

- of Informaion Security and Crypthology, Vol. 19, No. 2, pp. 26-34, 2009.
- [2] Sejung Lim, Gwangjun Kim and Taegeun Kang, "Important Information Protection using Client Virtualization", The Journal of the Korea Institute of Electronic Communication Sciences, Vol. 6, No. 1, pp. 111-117, 2011.
- [3] Daeyoung Hong, Wonseok Ko and Sungsoo Lim, "Virtualization Techniques for Secure and Reliable Computing", The Journal of Korea Institute of Information Scientists and Engineering, Vol. 26, No. 10, pp. 50-57, 2008.
- [4] Sejung Lim, Gwangjun Kim and Taegeun Kang, "Application Program Virtualization based on Desktop Virtualization", The Journal of the Korea Institute of Electronic Communication Sciences, Vol. 5, No. 6, pp. 595-601, 2010.
- [5] Heiser Gernot, Elphinstone, Kevin Kuz, Ihor Klein, Gerwin, Petters, and Stefan M. "Towards Trustworthy Computing Systems: Taking Microkernels to the Next Level", Operating Systems Review Vol. 41, No. 3, pp. 3-11, 2007.
- [6] Jiyeon Kim, Hyungjong Kim, Choosik Park and Myungjoo Kim, "A Study on the Analysis of Virtualization Techniques Vulnerability in Cloud Computing Environment", The Journal of Korea Institute of Informaion Security and Crypthology, Vol. 19, No. 4, pp. 72-77, 2009.
- [7] L. Cherkasova, D. Gupta, and A. Vahdat, "Comparison of the three cpu schedulers in xen," SIGMETRICS Perform. Eval. Rev., Vol. 35, No. 2, pp. 42-51, 2007.
- [8] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization", in SOSP'03: Proceedings of the nineteenth ACM symposium on Operating systems principles. New York, NY, USA : ACM, pp. 164-177, 2003.
- [9] D. Ongaro, A. L. Cox, and S. Rixner, "Scheduling I/o in virtual machine monitors," in VEE'08 : Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments. New York, NY, USA : ACM, pp. 1-10, 2008.
- [10] Chiyeon Kim, "A Study for Transaction Processing Supporting Scalability in the Cloud", The Journal of the Korea Institute of Electronic Communication Sciences, Vol. 7, No. 4, pp. 873-879, 2012.

저자 소개



정찬주(Chan-Joo Jeong)

1985년 조선대학교 전산기공학과 졸업(공학사)

1988년 조선대학교 대학원 전산기공학과 졸업(공학석사)

1998년 조선대학교 대학원 전기공학과 졸업(공학박사)

1998년~현재: 동강대학 전기에너지과 부교수 부교수

※ 관심분야 : 가상화, 모바일 네트워크, 영상통신



강태근(Tae-Guen Kang)

2009년 2월 전남대학교 컴퓨터공학과(공학사)

2011년 8월 전남대학교 대학원 컴퓨터공학과(공학석사)

2011년~현재 한국전자부품연구원 광주지역본부 연구원

※ 관심분야 : 가상화, ATM망, 컴퓨터 네트워크, 실시간 데이터 통신 등

