안드로이드 환경에서 SPICE 프로토콜의 활용에 관한 연구

정준권*, 정성민*, 김태경**, 정태명***
*성균관대학교 전자전기컴퓨터공학과
**서울신학대학교 교양학부
***성균관대학교 정보통신대학
e-mail:*{jkjung, smjung}@imtl.skku.ac.kr, **tkkim@stu.ac.kr, ***tmchung@ece.skku.ac.kr

A Study on the Use of the SPICE Protocol on Android Environment

Jun-Kwon Jung*, Sung-Min Jung*, Tae-Kyung Kim**, Tai-Myoung Chung***
*Department of Electrical and Computer Engineering, Sungkyunkwan University
**Department of Liberal Art, Seoul Theological University
***College of Information and Communication Engineering, Sungkyunkwan University

요 약

모바일 클라우드 서비스에 대한 요구에 따라 다양한 관련 기술들이 소개되고 있다. 이 중에서 모바일 원격접속 프로토콜은 RFB와 RDP, 두 개의 표준 프로토콜로 나뉜다. RFB 프로토콜은 구조의 단순함이, RDP 프로토콜은 빠른 응답속도가 상대적인 장점이다. 이 중, RDP 프로토콜은 윈도우즈 환경에 종속적인 제약이 있는데, SPICE 프로토콜은 이러한 RDP 프로토콜과는 달리 가상화 환경에 최적화된 오픈소스 프로토콜이다. 이러한 SPICE 프로토콜의 특징은 모바일 클라우드 환경에 상당한 이점을 가질 수 있다. 본 논문은 PC환경에서 RFB와 RDP, SPICE 프로토콜을 비교해 보고 모바일 환경에서의 활용성을 가늠해 본다. 그리고 모바일 환경에 가장 적합한 SPICE 프로토콜이 안드로이드에서 동작가능한지를 살펴보고 발생 가능한 문제점과 해결방안을 살펴보고자 한다.

1. 서론

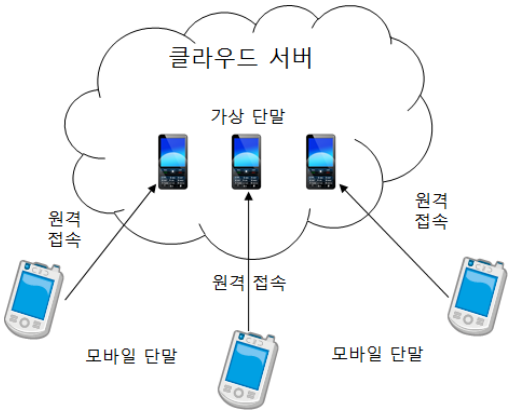
모바일 클라우드 서비스에 대한 요구가 증가하면서 이를 위한 다양한 기술들도 함께 발전하고 있다. 이 중 원격접속 프로토콜 또한 다양한 형태로 제공되고 있다. 원격접속 프로토콜은 물리적으로 떨어져 있는 사용자 컴퓨터를 원격으로 관리할 수 있게 해 준다. 사용자들은 이런 장점을 통해 수많은 시간적, 경제적 이익을 얻게 된다. 이러한 원격접속 프로토콜은 크게 두 가지의 국제 표준으로 나뉘게 된다[1]. Remote Frame Buffer(RFB) 프로토콜은 주로 Virtual Network Computing(VNC)라는 프로그램이 사용하는 프로토콜로써 서버의 화면을 캡쳐, 인코딩하여 클라이언트에 보여주는 동작을 한다. Remote Desktop Protocol(RDP)은 윈도우 원격접속 프로그램에서 사용하며, 서버의 동작 메시지를 수신한 클라이언트가 서버의 화면을 구성하여 보여주는 동작을 한다. RFB 프로토콜은 소스가 공개되어 있으며 다양한 무료 프로그램들이 제공되고 있다. 반면, RDP 프로토콜은 마이크로소프트사의 라이선스 정책 때문에 윈도우즈 환경에 종속적인 문제가 있다. 이런 RDP 프로토콜과는 다르게 가상화 환경에서 동작하는 원격접속 프로토콜인 SPICE 프로토콜이 오픈소스 형태로 존재한다. SPICE 프로토콜은 QEMU라는 가상화 에뮬레이터와 연계되어 동작한다. 이러한 특징을 가진 SPICE 프로토콜은 모바일 클라우드의 모바일 가상화 시스템과 연계되어 빠른 반응속도를 지닌 원격접속 프로토콜로 기대된다. 본 논문은 RFB, RDP, SPICE, 이 세 가지 원격접속 프로토콜의 특징과 장단점을 비교하고, 모바일 환경에서 어느 프로토콜이 유리한지 살펴본다. 또한 SPICE 프로토콜이 안드로이드 환경에서 어떠한 결점을 가지고 있는지와 앞으로 이를 어떻게 개선할 지를 제시하고자 한다. 2장에서는 RFB, RDP, SPICE, 세 가지 프로토콜의 특징과 장단점을 살펴보고, 3장에서는 SPICE 프로토콜을 안드로이드 환경에서 사용하였을 때 발생가능한 문제와 이를 해결할 방안을 찾는다. 마지막으로 4장에서는 결론을 맺는다.

2. 관련연구

2.1 모바일 클라우드

모바일 클라우드는 모바일 기기들에게 제공되는 클라우드 서비스이다. 클라우드 서버는 모바일 기기의 낮은 성능을 극복한 고성능 컴퓨팅 서비스를 제공할 수 있다[2]. 모바일 기기들은 씬 클라이언트처럼 최소한의 성능만 있으면 클라우드 서버를 통해 고성능 컴퓨팅 서비스를 받을 수 있다. 이를 위해서 클라우드 서버는 모바일 가상화를 활용한다[3]. 모바일 가상화를 통해 클라우드 서버가 여러 개의 모바일 가상머신을 생성하고, 모바일 단말이 해당 가상머신에 원격접속하여 서비스를 제공받는다. 즉, 모바일

클라우드 서비스에서 원격접속 프로토콜이 클라이언트와 서버를 이어주는 역할을 하는 것이다. (그림 1)은 모바일 클라우스 서비스의 개략적인 모습을 보여준다.



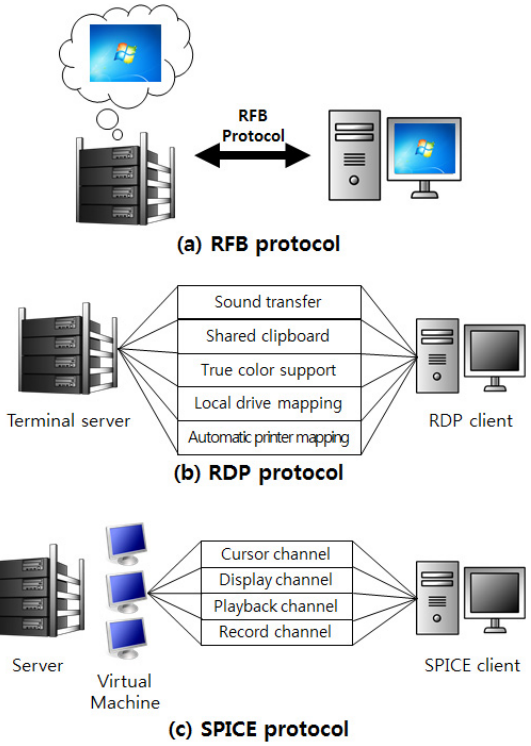
(그림 1) 모바일 클라우드 서비스

2.2 RFB와 RDP, SPICE

다양한 프로토콜 중에서 표준 원격접속 프로토콜로 채택된 것은 RFB 프로토콜과 RDP 프로토콜이다. RFB 프로토콜은 프레임 버퍼를 활용하여 원격 컴퓨터의 화면 그대로를 클라이언트에 출력하여 보여는 방식으로 구현되어 있다[4]. RFB 프로토콜은 구현하기 간단한 구조로 되어 있다. 게다가 소스코드까지 공개가 되어 있어, 사용자는 다양한 공개 프로그램을 활용 할 수 있다. RFB 프로토콜의 단점은 화면 이미지를 지속적으로 전달해야 하는 특징상 어느 정도 네트워크 트래픽을 감당할 수 없으면 원격 제어에 상당한 불편함이 생길 수 있다는 것이다.

RDP 프로토콜은 윈도우즈에서 볼 수 있는 원격접속 프로토콜이다[5]. 원격서버에서 발생한 이벤트 메시지를 클라이언트가 전달받아 클라이언트의 화면에 나타내는 방식이다. RDP 프로토콜은 채널을 활용하여 각각의 메시지를 구분한다. RDP 프로토콜은 이미지가 아닌 이벤트 메시지를 전달하기 때문에 RFB 프로토콜에 비해 더 적은양의 패킷을 주고받게 되고 응답속도도 빠르다. 다만, 마이크로소프트사의 라이선스 정책 때문에 RDP 원격서버는 윈도우즈만 가능하다[6]. 하지만, 클라이언트는 라이선스에 제약이 없어 다양한 플랫폼으로 제공된다.

SPICE 프로토콜은 RDP 프로토콜의 구조와 유사한 오픈소스 프로토콜이다. SPICE 프로토콜은 RDP 프로토콜처럼 여러 개의 채널을 만들어 각각의 데이터를 개별적으로 전송, 수신하는 구조를 가진다. SPICE 프로토콜은 QEMU라는 가상머신 에뮬레이터와 연동되어 동작한다. 따라서 QEMU가 지원하는 모든 가상환경 운영체제에 접속이 가능한 이점이 있다. 현재는 맥OS와 리눅스, 윈도우즈 버전의 클라이언트가 제공되지만 모바일 클라이언트는 지원하지 않고 있다. <표 1>은 세 가지 원격접속 프로토콜의 장단점을 비교하였고, (그림 2)는 세 가지 원격접속 프로토콜의 간단한 동작 구조를 보여준다.



(그림 2) 원격접속 프로토콜

3. SPICE for Android

3.1 SPICE 프로토콜의 안드로이드 적용가능성

SPICE 프로토콜은 RDP 프로토콜과 유사한 구조를 가지고 있다[7]. 이 때문에 SPICE는 빠른 응답속도와 적은 트래픽을 가질 수 있다. 또한 SPICE는 RDP와는 달리 다양한 서버 플랫폼을 지원하며 소스도 공개되어 있기 때문에 다양한 서버/클라이언트 플랫폼을 지원할 수 있다. SPICE 프로토콜은 대부분의 PC환경을 지원하지만 아직 모바일 플랫폼에 맞는 버전은 존재하지 않는다.

다양한 모바일 플랫폼 중 안드로이드는 리눅스커널을 기반으로 하고 있다. 그러므로 리눅스에서 동작하는 SPICE 프로토콜을 안드로이드 환경으로 포팅하여 사용할 수 있는 가능성이 있다. 리눅스용 SPICE 프로토콜은 C로 구현되어 있다. C 소스를 안드로이드 환경에 맞춰 포팅한다면 NDK 쓰듯이 안드로이드에서도 SPICE를 활용할 수

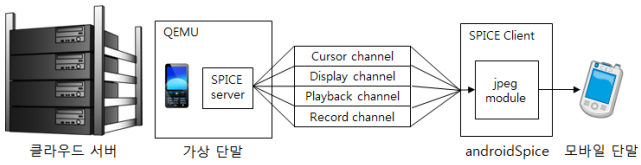
<표 1> RFB, RDP, SPICE 프로토콜의 비교

	RFB 프로토콜	RDP 프로토콜	SPICE 프로토콜
장점	간단한 구조 다양한 공개 프로그램	빠른 응답 속도 다양한 클라이언트 플랫폼 제공	빠른 응답 속도 다양한 서버/클라이언트 플랫폼 지원
단점	많은 트래픽 발생 느린 응답 속도	원격서버는 윈도우즈 한정	모바일 클라이언트 미지원

있을 것이다. 다만, 리눅스용 SPICE 프로토콜을 포팅하기 이전에 SPICE 프로토콜이 활용하는 리눅스 모듈 또한 포팅되어야 할 것이다. 현재 구글코드 사이트에 SPICE 프로토콜을 안드로이드 환경에 맞도록 수정한 오픈소스가 androidSpice라는 파일명으로 공개되어 있다[8].

3.2 안드로이드용 SPICE 프로토콜의 문제점과 해결방안

앞장에서 언급했듯이 SPICE 프로토콜은 RDP 프로토콜과 동작과정이 유사하다. 따라서 클라이언트의 조작에 대한 응답시간이 짧을 것으로 예상된다. 하지만, 실제 androidSpice의 소스를 분석해보면 RDP만큼의 속도가 나오지 않으리라 예상된다. androidSpice 소스를 분석해 본 결과, 서버에서 수신한 메시지를 안드로이드가 직접 화면에 출력해 주지 않는 것을 발견했다. spicy.c의 JNI 함수에서 스레드를 생성함과 동시에 jpeg encoder를 생성한다. 생성된 encoder는 원격 화면이 갱신될 때마다 raw2jpg함수를 호출하여 서버로부터 수신한 메시지를 통해 생성한 화면 정보를 jpeg포맷으로 전환하여 java영역으로 전달하게 된다. 타겟 프로그램인 androidSpice는 리눅스용 SPICE를 수정하여 안드로이드 환경에서 동작하도록 만든 것이다. 그렇기 때문에 클라이언트의 동작은 모두 C언어 영역에서 이루어지게 된다. 그러나 안드로이드가 사용자 인터페이스를 제공하는 영역은 java이다. 따라서 안드로이드가 서버로부터 수신한 메시지를 직접 처리하지 않고 ndk 호출을 통해서 간접적으로 원격접속을 처리하게 된다. 현재 androidSpice는 ndk 라이브러리가 원격서버의 화면을 구성하면 그 정보를 이미지로 인코딩 시켜 안드로이드 User Interface로 전달하는 방식으로 구현되어 있다. androidSpice에서 가장 많은 부하가 발생하는 부분은 이 영상처리 부분이다. 이러한 부분 때문에 androidSpice가 메시지 처리방식의 이점을 제대로 살리지 못하고 있다. (그림 3)은 안드로이드 SPICE 프로토콜의 동작 구조를 보여준다.



(그림 3) 안드로이드용 SPICE 프로토콜

앞 문단에서 설명한 문제점을 해결하기 위해선 원격서버에서 전달하는 메시지를 C영역이나 java영역 한군데에서 영상을 완벽하게 처리할 수 있어야 한다. 그러므로 ndk 라이브러리에서 액티비티의 뷰를 구성할 수 있거나, 리눅스 SPICE 프로토콜의 화면 메시지를 java에서 처리할 수 있도록 변환을 시킬 수 있어야 한다. 다만, 첫 번째 방식은 현재 ndk 버전에서는 불가능 하다. ndk에서 뷰의 제어를 지원하지 않기 때문이다. 두 번째 방식으로 문제를 해결하기 위해서는 SPICE 서버가 전달하는 메시지와 안드로이드가 지원하는 그래픽 명령에 대한 완벽한 이해가 필요하다.

다른 방법으로는 SPICE 클라이언트에 출력하는 프레임 수를 제어하는 것이 될 것이다. 기본적으로 android Spice는 동작하는 모든 행동에 대해 화면을 구성하고 클라이언트 화면에 출력을 한다. 마우스의 이동같이 사용자가 세세하게 인식하지 않는 행동들을 찾아내어 이 부분의 화면 구성 및 출력을 차단하면 훨씬 쾌적한 동작환경을 구성할 수 있을 것이다. 이 방식으로 문제를 해결하기 위해선 각각의 동작들 중 사용자의 인식에 영향이 없거나 적은 부분을 구분해 내는 것이 매우 중요하다.

4. 결론

클라우드 서비스가 모바일 환경까지 퍼지면서 모바일 가상화 기술과 더불어 모바일 원격접속 서비스에 대한 관심도 늘어났다. 원격접속에 대한 표준은 크게 RFB와 RDP로 나누어지는데 이 중 RFB 프로토콜은 속도의 문제가 RDP 프로토콜은 플랫폼 제한의 문제를 가지고 있다. SPICE라는 원격접속 프로토콜은 RDP 프로토콜과 같은 구조를 가지면서 가상화 플랫폼에 적합하도록 개발되어 있다. 가상화 시스템과 연동이 되기 때문에 운영체제에 큰 영향을 받지 않는 SPICE는 모바일 환경에 가장 적합한 원격접속 프로토콜로 기대된다. 본 논문에서는 SPICE 프로토콜의 모바일 플랫폼에서의 활용가능성을 알아보았다. 또한 SPICE가 안드로이드 환경에서 원래의 성능을 제대로 발휘하지 못하는 원인을 알아내고 이를 극복하기 위해선 어떤 해결책이 나와야 하는가에 대해 기술하고 있다. SPICE 프로토콜이 모바일 환경에서도 폭넓게 사용할 수 있으려면 안드로이드와 리눅스 환경에 대한 깊은 이해가 필요하다. 안드로이드 버전의 SPICE 프로토콜의 결함을 해결할 수 있다면 모바일 가상화 시대에 가장 적합한 원격접속 프로토콜이 될 수 있을 것이다.

참고문헌

[1] 장승주, “윈도 운영체제에서 원격 시스템 관리 프로그램 기술 동향”, 2010. 08

[2] ETRI, “차세대 컴퓨팅을 위한 가상화 기술”, 전자통신동향분석 23권 4호, 2008

[3] 송미숙 외 3명, “모바일클라우드 가상단말 협업기술 및 프로비저닝”, 보안공학연구논문지 제 9권 제 1호, 2012

[4] Tristan Richardson, "The RFB Protocol", RealVNC Ltd, 2005

[5] MSDN, http://msdn.microsoft.com/en-us/library/aa383015(VS.85).aspx, 2012. 09

[6] MSDN, http://msdn.microsoft.com/en-us/library/cc241880(PROT.10).aspx, 2012. 09

[7] SPICE, http://spice-space.org, 2012. 09

[8] spice-client-android, http://code.google.com/p/spice-client-android/, 2012.10