

원격 다자간 협업 환경을 위한 확장된 멀티캐스트 연결성 솔루션

An Extended Multicast Connectivity Solution for Remote Multi-party Collaborative Environment

김남곤, Namgon Kim*, 김종원, JongWon Kim*
*광주과학기술원 정보기공공학부 네트워크미디어연구실

요약 다자간 분산형 협업 시스템인 Access Grid(AG)는 IP 멀티캐스트 네트워크상에서 여러 사용자들 간에 영상, 음성 및 다양한 데이터의 공유를 통해 상호 의사소통이 가능한 공동 작업환경을 제공하기 위해 개발되었다. 멀티캐스트를 사용함으로써 AG는 대역폭 효율적으로 다수 사용자 사이의 의사소통 환경을 제공하고 있다. 그러나 IP 멀티캐스트의 설정 및 관리상 복잡성으로 인해 이를 지원하지 않는 네트워크가 다수 존재한다. 이는 AG를 이용한 협업 서비스를 이용하는 데에 큰 장애물이 되고 있다. 본 논문에서는 이러한 멀티캐스트 연결성 문제에 대한 해결책으로 응용 계층의 멀티캐스트 터널링 프로토콜인 UMTP (UDP multicast tunneling protocol) 를 확장한 멀티캐스트 연결성 솔루션인 AG Connector를 제안한다.

핵심어: Access Grid, advanced collaboration environment, overlay multicast, multicast connectivity solution

1. 서론

다자간 분산형 협업 시스템인 Access Grid(AG) [1]는 IP 멀티캐스트 네트워크 상에서 여러 사용자들 간에 영상, 음성 및 자료를 공유하고 상호 의사소통이 가능한 공동 작업환경을 제공하기 위해 개발되었다. IP 멀티캐스트는 대역폭 효율적인 반면, 설정 및 관리상의 복잡성으로 인해 지원되지 않는 네트워크가 다수 존재한다.

AG의 다자간 협업 환경을 활용하는 데 있어 IP 멀티캐스트의 지원 여부는 매우 중요한 문제이다. AG 커뮤니티에서는 이러한 멀티캐스트 연결성 문제를 해결하기 위한 방안으로 QuickBridge [3]를 제공하고 있다. QuickBridge는 멀티캐스트 네트워크 내에 QuickBridge 서버를 두고, 멀티캐스트가 불가능한 AG 노드들이 이를 통해 멀티캐스트 데이터를 UDP 유니캐스트 형태로 전달받는 방식으로 동작한다. QuickBridge의 동작은 단순하여, 안정적인 반면, 각 수신자에게 유니캐스트로 멀티캐스트 데이터를 전달하는 멀티-유니캐스트(multi-unicast) 방식으로 인해 확장성에 문제가 있고, 각 멀티캐스트 주소마다 하나의 포트번호를 할당함으로써 방화벽 환경에서 적용하기 어렵다는 단점이 있다.

본 논문에서는 이에 대한 대안으로 QuickBridge를 대체할 수 있는 새로운 멀티캐스트 연결성 솔루션인 AG Connector를 제안한다. AG Connector는 응용 계층에서의 멀티캐스트 터널링 프로토콜인 UMTP (UDP multicast tunneling protocol) [2]에 기반하여 동작한다. UMTP는 멀티캐스트 네트워크에 존재하는 한 터널 종단과 이 터널 종단과 멀티

캐스트 연결성을 갖지 않는 다른 네트워크에 존재하는 다른 터널 종단 사이에 UDP 터널을 생성하고, 이를 통해 두 터널 종단이 속하는 네트워크들 사이의 멀티캐스트 연결성을 제공한다. 이 때 두 터널 종단이 존재하는 네트워크는 모두 로컬 네트워크에서 IP 멀티캐스트 라우팅을 존재한다고 가정한다. 각 터널 종단은 자신이 속한 네트워크에서 발생하는 멀티캐스트 데이터를 수신한 후, 이를 미리 약속된 형식으로 캡슐화해 상대방 터널 종단에게 UDP 터널을 통해 전달한다. 데이터 캡슐화와 멀티캐스트를 이용한 데이터 전송은 멀티캐스트 연결성을 제공함에 있어, 네트워크 대역폭을 효율적으로 이용할 수 있고, 시스템 자원 측면에서도 효율적이다. AG Connector는 UMTP의 이러한 특징들을 기반으로 실제 인터넷 환경에서 동작 가능한 멀티캐스트 연결성 솔루션으로 설계되었다.

본 논문에서는 기존의 멀티캐스트 브리지들을 이용할 때의 한계점과 적용 불가능한 네트워크들을 예시한 후에, AG Connector를 통해 이러한 한계점들이 극복될 수 있음을 보일 것이다.

2. AG와 멀티캐스트 브리지

AG는 IP 멀티캐스트 기반의 원격 다자간 협업 환경으로 원격지의 사용자들이 비디오와 오디오 뿐 아니라, 데이터, 서비스, 그리고 응용 프로그램을 서로 공유할 수 있는 환경을 제공하여, 보다 실제감 있는 협업 환경을 제공한다. AG

사용자들은 베뉴(Venue)라고 하는 가상의 협업 공간에 참여함으로써, AG에서 제공하는 이러한 협업 환경을 활용하는 것이 가능해진다. AG 사용자들은 베뉴 클라이언트 툴을 이용해 베뉴 서버에 접속하여, 베뉴 서버로부터 AG 노드를 구성하기 위한 정보와 미디어 생산자/소비자 서비스와 같은 서비스를 실행하기 위해 필요한 정보들을 제공받게 된다. 특히 AG 사용자들 간에 오디오/비디오와 같은 미디어를 주고받기 위해 필요한 멀티캐스트 주소를 이 과정을 통해 얻게 된다.

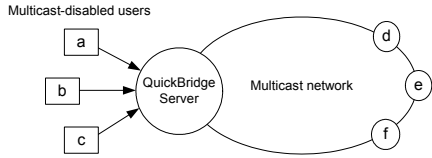


그림 1. QuickBridge의 동작.

AG는 기본적으로 참여자들 사이에 미디어 정보를 주고받기 위해 IP 멀티캐스트를 이용하기 때문에, 사용자 네트워크에서의 IP 멀티캐스트 지원 여부는 AG 활용에 있어 매우 중요한 문제이다. AG 커뮤니티에서는 멀티캐스트 연결성 문제를 해결하기 위한 솔루션으로 QuickBridge를 제공하고 있다. 그림 1은 QuickBridge가 어떻게 사용자에게 멀티캐스트 연결성을 제공하는지를 보여준다. QuickBridge는 IP 멀티캐스트 네트워크에 QuickBridge 서버를 두고, 이 서버가 멀티캐스트가 불가능한 노드들 (a, b, c)에게 멀티캐스트 네트워크에서 발생하는 멀티캐스트 데이터를 UDP 유니캐스트 형태로 전달하는 방식으로 멀티캐스트 연결성을 제공한다.

UDP 유니캐스트를 통한 QuickBridge의 멀티캐스트 연결성 제공 방식은 몇 가지 단점을 가지고 있다. 먼저 QuickBridge 서버는 각각의 유니캐스트 사용자에게 동일한 패킷을 전송하게 된다. 이는 QuickBridge 서버 자체의 시스템 성능을 요구할 뿐만 아니라, IP 멀티캐스트 방식에 비해 네트워크 대역폭을 많이 소모한다. 특히, 현재 QuickBridge가 오직 하나의 멀티캐스트 세션에 대해서만 동작하고 있는 상황에서도 많은 수의 유니캐스트 사용자들이 접속하게 되면, 사용자 수에 따라 계속해서 추가적인 대역폭을 소모해야 한다.

또한, 멀티캐스트가 불가능한 네트워크의 사용자들은 QuickBridge 서버와 연결을 설정할 때 AG를 활용하기 위해 필요한 멀티캐스트 주소의 수만큼의 UDP 연결을 생성하게 된다. 이는 사용자 시스템 자체에서 그 수만큼의 포트번호를 할당해야 함을 의미한다. AG는 오디오/비디오 데이터를 주고 받는데 RTP (Real-time Transport Protocol)를 이용하는데, RTP는 RTCP와 짝을 이루어 동작하기 때문에, 사용자는 기본적으로 오디오 2개, 비디오 2개, 총 4개의 포트번호를 할당하게 된다. 이러한 많은 수의 포트 번호 할당은 보안

상의 문제를 일으킬 수가 있는데, 특히 기업이나 관공서 등의 방화벽을 활용하는 환경에서는 활용하기 쉽지 않은 구조이다.

이러한 QuickBridge 구조의 문제점들은 AG의 구조에 적합한 더 나은 멀티캐스트 연결성 솔루션을 요구하게 되고, 또한 새로운 멀티캐스트 연결성 솔루션이 가져야 하는 구조를 말해준다. 먼저, 새로운 멀티캐스트 연결성 솔루션은 멀티캐스트 연결성 솔루션을 구동하고 있는 시스템 자체 뿐 아니라, 네트워크에도 추가적인 부담을 주지 않는 형태이어야 한다. 또한, NAT나 방화벽 뒤에 존재하는 사실 네트워크 환경에서도 쉽게 활용할 수 있도록, 적은 수의 포트만으로도 이용 가능해야 한다. 본 논문에서는 이러한 QuickBridge의 단점을 해결하고, AG에서의 멀티캐스트 연결성 문제에 대한 대안으로서 UMTP를 확장한 형태의 멀티캐스트 연결성 솔루션인 AG Connector를 소개한다.

3. UMTP 기반 멀티캐스트 연결성 솔루션

3.1 UMTP (UDP multicast tunneling protocol)

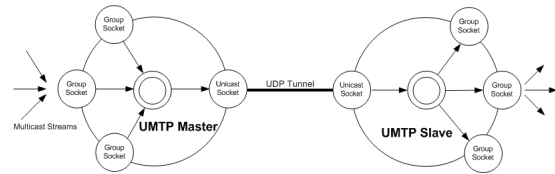


그림 2. UMTP의 멀티캐스트 터널링 동작.

본 논문에서 제안하는 멀티캐스트 연결성 솔루션은 UMTP에 기반하고 있다. UMTP는 멀티캐스트 연결성을 제공하는 응용 계층의 터널링 프로토콜이다. 그림 2는 UMTP의 멀티캐스트 터널링 동작을 보여주는 그림으로, UMTP는 터널 master와 터널 slave의, 한 쌍의 노드들 사이에 UDP 유니캐스트 데이터그램을 전달하는 UDP 터널을 이용해 동작한다. UMTP에 참여하는 각 노드는 터널 종단으로서의 역할을 하게 되고, 터널을 통해 전송되는 모든 UDP 데이터그램은 '명령어' 나 '데이터' 로써의 의미를 갖는다. 이 중, 명령어를 통해, 한 터널 종단은 다른 터널 종단과 UDP 터널을 생성하거나 생성된 터널을 중지 할 수 있다. 명령어를 통한 UDP 터널이 구성된 절차 이후에 두 터널 종단은 이를 통해서 서로 간에 데이터를 주고받음으로써, 상호 간에 멀티캐스트 연결성을 제공하게 된다.

터널을 생성하고, 이를 통해 데이터를 주고받는데 있어, 각 터널 종단은 'master' 나 'slave'로 동작하게 된다. UMTP에서는 터널 종단 중 멀티캐스트 네트워크에 접근을 시도하는 쪽을 터널 master로 정의하고, 이미 멀티캐스트 네트워크

에 존재하는 쪽은 터널 slave로 정의하고 있다. 터널 master는 터널 slave에게 주기적으로 특정 멀티캐스트 그룹 주소 (group, port)에 대한 'join' 명령어를 보낸다. 이를 통해 터널 slave는 해당 (group, port)에 대한 멀티캐스트 터널을 터널 master에게 제공해야 함을 알게 된다. 터널 slave는 터널 master에게서 명시적으로 해당 (group, port)에 대한 'leave' 명령을 받거나, 미리 약속된 시간동안 join 명령어를 받지 못하면, 해당 (group, port)에 대한 멀티캐스트 터널 제공을 중단한다.

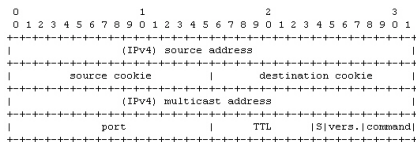


그림 3. 16비트 UMTP 디스크립터.

각 터널 종단은 UDP 데이터그램을 주고받는데 있어, 그림 3의 UMTP 디스크립터의 형태를 이용한다. UMTP 디스크립터에서 'command' 필드는 터널 종단의 행동을 결정한다. UMTP는 몇 가지 명령어를 정의하고, 이 명령어에 따라 정보를 주고받는다. 'multicast address' 필드와 'port' 필드는 멀티캐스트 터널을 요청하거나 현재 터널을 제공 중인 멀티캐스트 주소의 (group, port) 정보를 담고 있다. 'data'를 제외한 모든 명령어는 오로지 UMTP 디스크립터로만 구성된다. 'data'의 경우에는 해당 (group, port)로부터 전송된 멀티캐스트 데이터의 뒤에 UMTP 디스크립터가 추가되어 상대방 터널 종단에게 전송된다.

UMTP 디스크립터를 통해, UMTP는 터널 종단들 사이에 멀티캐스트 데이터를 전달함에 있어 데이터 캡슐화를 가능하게 했다. 각 터널 종단들은 현재 멀티캐스트 터널링을 제공하고 있는 멀티캐스트 주소인 (group, port)로 전송되는 멀티캐스트 데이터를 수신한다. 수신된 멀티캐스트 데이터를 상대방 터널 종단에게 전송하기 전에, 수신된 멀티캐스트 데이터에 UMTP 디스크립터를 붙여 UMTP 페이로드 (payload)를 구성한다. 터널 종단들은 이 UMTP 페이로드를 터널을 통해 상대방에게 전송한다. UMTP 페이로드를 수신한 터널 종단은 UMTP 페이로드 내부의 UMTP 디스크립터를 통해, 멀티캐스트 데이터가 전송되어야 하는 멀티캐스트 그룹 주소 정보를 알게 된다. 각 터널 종단은 상대 터널 종단으로부터 수신한 UMTP 페이로드로부터 멀티캐스트 데이터를 추출한다. 이렇게 추출된 멀티캐스트 데이터를 각 터널 종단은 자신이 속한 로컬 네트워크에 다시 멀티캐스트로 전송한다. 이러한 일련의 과정을 통해 두 멀티캐스트 연결을 갖지 않는 네트워크 사이에 멀티캐스트 연결성이 제공되게 되고, 또한 다수의 (group, port)에 대한 멀티캐스트 데이터에 대해서도 오직 하나의 UDP 포트만을 통해 전송하는

것이 가능해진다. 멀티캐스트를 통한 데이터 전송으로 오직 한 번의 송신으로 동일 네트워크에 존재하는 모든 참가자들에게 데이터를 전송할 수 있게 되어, 각 참가자들에게 반복적으로 데이터를 전송하는 문제를 해결하고, 네트워크 대역폭과 시스템 자원의 낭비를 막았다. 또한, UMTP를 이용하면, IP 멀티캐스트 인프라를 활용하는 데에도 도움이 된다. 사용자들 사이에 멀티캐스트 경로를 설정하는 시간이 길어져서, 사용자들이 일정 시간 동안 상호 간에 멀티캐스트 연결성을 갖지 못할 때, UMTP를 활용하게 되면, 문제 시간 동안의 멀티캐스트 연결성이 제공됨과 동시에 멀티캐스트 연결이 복구되었을 때 IP 멀티캐스트를 통한 전송까지 자연스럽게 활용할 수 있게 된다.

3.2 AG Connector

UMTP는 멀티캐스트 연결성을 갖지 않는 네트워크들 간에 멀티캐스트 연결성을 제공하기 위해 둘 사이에 UDP 터널을 생성하고 데이터를 주고받는 방법을 정의한 응용계층의 멀티캐스트 터널링 프로토콜이다. 하지만 UMTP는 실제 멀티캐스트 연결성 문제에 대한 솔루션이 아닌 네트워크 사이의 멀티캐스트 연결성을 지원하는 프로토콜이기 때문에, 실제 솔루션으로 동작하기 위해서는 추가적인 시스템 설계가 필요하다. 본 절에서는 UMTP를 확장하여, 실제 솔루션으로 동작하도록 하기 위한 추가적인 프로토콜들을 정의한다.

AG는 비디오와 오디오의 미디어를 지원하는데, AG의 비디오 서비스는 VIC (Video conferencing tool)을 이용하고, 오디오 서비스는 RAT (Robust audio tool)을 이용한다. VIC과 RAT은 모두 Mbone (multicast backbone)에서 시작된 톨로서 전송 프로토콜로 UDP 상에서 RTP를 이용한다.

첫째로, UMTP는 하나의 네트워크 내에서 멀티캐스트를 이용한 데이터 전송이 이루어지기 때문에 동일 네트워크 내에 여러 개의 UMTP master나 UMTP slave가 동작할 경우 멀티캐스트 루프가 발생할 수 있다. 이를 위해 MPROBE라는 프로토콜을 정의하였다. 모든 UMTP master와 UMTP slave는 공유된 멀티캐스트 주소로 자신의 존재를 알리는 패킷을 전송하고, 이를 통해 상호 간에 멀티캐스트 루프가 존재하는지의 여부를 파악할 수 있다. 둘째로, UMTP는 멀티캐스트 세션을 위해 UDP 터널을 생성할 멀티캐스트 주소를 공유하는 방법을 정의하지 않고 있다. UGMP (group management extension for UMTP)는 멀티캐스트 세션을 생성하기 위해 필요한 멀티캐스트 주소 정보와 'join'과 'leave' 명령어를 정의하고 있다.

3.2.1 MPROBE 프로토콜

멀티캐스트 루프 문제에 대해 UMTP는 다음과 같은 해결책을 제시하고 있다. 첫째로, 데이터 패킷으로부터 추출한 페이로드를 멀티캐스트로 전송할 때, UMTP를 구현한 시스템이 적절한 TTL (time to live) 값을 선택하는 것이다. 또한, 해당 패킷이 재수신된 것은 아닌지 검증하여야 한다. 두 번째로, UMTP 구현이 현재 터널을 설정 중인 터널 종단이 소스인 멀티캐스트 패킷을 받게 되면, 해당 터널 종단과의 터널을 즉시 해제한다. 세 번째로, 현재의 터널링 환경에서 멀티캐스트 루프가 발생할 가능성이 있다면, 각 터널 종단이 주기적으로 공유된 멀티캐스트 주소를 통해 각자의 상태 정보를 전송한다. 각 터널 종단은 이를 수신 받음으로써 가능한 멀티캐스트 루프를 파악 할 수 있다.

그러나, 이러한 해결책들은 몇 가지 단점을 가지고 있다. 먼저 고려해야 할 것은 구현 가능하다는 것이다. 멀티캐스트 루프를 발생시키는 패킷은 TTL값이 0이 되었을 때 네트워크에서 제거된다. 이를 위해서는 UMTP를 구현한 시스템에서 수신된 멀티캐스트 패킷의 TTL 값을 확인한 후 다른 터널 종단에게 전송하면서 TTL 값을 줄인 후 데이터를 전달해야 한다. 그러나, 유닉스나 윈도우즈 환경에서 활용되고 있는 표준 규격의 소켓 인터페이스에서는 사용자 응용 프로그램에서 수신된 패킷의 TTL 값을 알 수 있는 방법이 없다. 따라서 UMTP 명세에서 제안한 첫 번째 해결책은 활용하는 것에 문제가 있다. 두 번째 해결책에서의 문제점은 각 터널 종단이 모든 패킷의 소스 주소를 체크해야 한다는 점이다. 이러한 작업은 계속 진행되는 것은 터널 종단들에게 시스템 자원 소모적인 일이다. 터널 종단 들 사이에 공통의 멀티캐스트 주소를 공유하는 마지막 해결책이 가장 멀티캐스트 루프를 파악하는 데 있어 적절하다. 그러나, 현재의 UMTP 명세에서는 이에 대한 구체적인 방법을 언급하고 있지 않다. 본 절에서는 이러한 아이디어를 확장하여 MPROBE라는 멀티캐스트 루프 탐지 프로토콜을 소개한다.

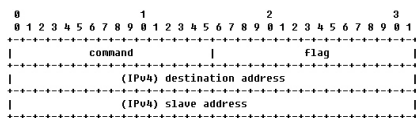


그림 4. MPROBE 패킷 형식.

MPROBE 프로토콜은 동일 멀티캐스트 네트워크에 존재하는 터널 종단들을 발견하여, 이들 사이에 발생 가능한 멀티캐스트 루프를 제거하도록 설계되었다. 각 터널 종단은 자신의 네트워크로 자신의 존재를 알리기 위해 그림 4의 MPROBE 패킷을 형성해 이를 멀티캐스트로 전송한다. 터널 종단들은 자신과 터널을 형성하고 있는 다른 터널 종단 뿐 아니라, 동일 네트워크 내에 존재하는 또 다른 터널을 형성하는 터널 종단들의 존재 또한 알 수 있다.

몇 가지 상황에서 UMTP는 멀티캐스트 루프를 발생시킬

수 있다. 하나의 터널을 구성하는 두 터널 종단이 하나의 네트워크 내에 존재할 수도 있고, 또한, 서로 다른 터널에 속하는 터널 종단들이 동일 네트워크에 존재할 수도 있다. 이러한 멀티캐스트 루프 발생 가능성들을 우리는 터널 종단이 'master' 인가 'slave'인가에 따라 다음과 같이 나눌 수 있다.

첫 번째로, master인 터널 종단이 터널의 slave로부터 MPROBE를 받았을 경우가 있다. 이 상황은 두 터널 종단이 상호 멀티캐스트 연결성을 갖는 상황이라는 것이다. 이는 터널 slave가 터널 master로부터 MPROBE를 받는 것과 동일한 상황이다. 두 가지 경우 모두에 있어서, 터널 master는 터널 slave와의 연결을 종료하고, 해당 멀티캐스트 그룹 주소로부터 탈퇴해야 한다.

두 번째로, 터널 slave가 또 다른 터널 slave로부터 MPROBE를 받았을 경우이다. 이 때, 각 터널 slave에 연결되어 있는 터널 master 들의 관계에 따라 멀티캐스트 루프의 발생 여부가 결정된다. 만약 두 터널 slave와 연결되어 있는 터널 master 들이 서로 동일한 네트워크에 존재하고 있다면, 멀티캐스트 루프가 발생하게 될 것이다. 하지만, 만약 두 터널 slave에 연결되어 있는 터널 master 들이 서로 다른 네트워크에 존재한다면, 멀티캐스트 루프를 발생시키지 않고, 더 많은 네트워크들 사이에 멀티캐스트 연결성을 제공하게 된다.

세 번째로, 터널 master가 다른 터널 master로부터의 MPROBE 패킷을 받는 경우이다. 이 때, 두 터널 master가 동일한 (group, port)에 대한 멀티캐스트 터널링 동작을 진행 중이라면, 이 들은 멀티캐스트 루프를 발생시키게 된다. 두 터널 master가 동일한 터널 slave에 연결되어 있다면, 이 들은 당연히 멀티캐스트 루프를 형성 시키게 될 것이고, 그렇지 않은 경우에는 터널 slave들의 관계에 따라 멀티캐스트 루프의 발생 여부가 결정된다. 두 터널 slave가 동일 멀티캐스트 네트워크에 존재한다면, 이는 멀티캐스트 루프를 형성 시키게 된다. 따라서 두 터널 master들은 서로 같은 (group, port)에 대한 터널을 공유해서는 안된다.

따라서, 멀티캐스트 루프 가능성을 파악하고, 멀티캐스트 루프에 적절히 대응하기 위해서는 각 네트워크에 존재하는 터널 종단이 master인지 slave인지를 아는 것이 필요하다. 그림 4에서 MPROBE 패킷 형식은 이를 위해 필요한 정보들을 담고 있다. MPROBE 패킷은 'command' 필드에 따라, MPROBE, MPROBE_ACK, MLEAVE, MTEAR_DOWN로 구분되고, flag 필드는 해당 터널 종단이 master인지 slave인지를 나타낸다.

- MPROBE : 각 터널 종단이 자신의 존재를 네트워크에 존재하는 다른 터널 종단들에게 알리기 위해 전송.
- MPROBE_ACK : MPROBE 패킷을 수신한 터널 종단들

이 이를 MPROBE 패킷의 소스 터널 종단에게 알리기 위한. 현재 멀티캐스트 터널링을 진행 중인 (group, port) 목록을 덧붙여 전송.

- MLEAVE : MPROBE_ACK를 보낸 터널 종단과 충돌이 발생할 경우, 이를 해당 터널 종단에 알리고, 해당 (group, port)에 대한 탈퇴를 요청.
- MTEAR_DOWN : 터널 master가 터널 slave와 동일한 네트워크에 존재할 경우, 터널 slave가 터널 master의 종료 를 요청.

MPROBE 프로토콜의 구체적인 동작에 대해서는 [8]에서 확인할 수 있다.

3.2.2 UGMP (Group management extension for UMTF)

UMTF 명세에서는 터널 master와 터널 slave가 상호 간에 어떻게 터널링이 필요한 멀티캐스트 그룹 주소를 공유하는가에 대한 설명이 존재하지 않는다. UMTF 명세에서는 멀티캐스트 연결을 필요로 하는 응용이 필요한 멀티캐스트 그룹 주소 (group, port)로의 터널 설정을 요구할 수 있다고 되어 있다. 본 절에서는 UMTF를 확장하여 터널링이 필요한 그룹 주소를 관리하기 위한 프로토콜인 UGMP (Group management extension for UMTF)를 소개한다. 그림 5는 UGMP의 패킷 형식을 보여준다.

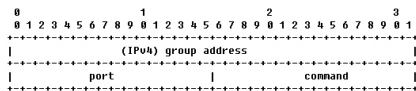


그림 5. UGMP 패킷 형식.

UGMP는 터널링이 필요한 멀티캐스트 그룹 주소 (group, port)에 대한 'join'과 'leave'의 두 가지 명령어로 구성되어 있다. UGMP 패킷 형식의 group address 필드와 port 필드를 통해 필요한 멀티캐스트 그룹 주소 정보를 전달하고, 'command' 필드의 값으로 필요한 동작을 요청하게 된다. 응용 프로그램은 이 UGMP 패킷 형식을 구현하여, 주기적으로 필요한 그룹 주소에 대한 정보를 터널 master에게 보냄으로써, 멀티캐스트 연결성을 제공받을 수 있다. 응용 프로그램이 'leave' 명령어를 전송하거나, 터널 master가 미리 약속된 시간 동안 'join' 명령어를 수신하지 못한다면, 터널 master는 해당 (group, port)에 대한 멀티캐스트 터널링을 중단하고, 멀티캐스트 그룹 주소로부터 탈퇴하게 된다.

4. 구현 및 검증

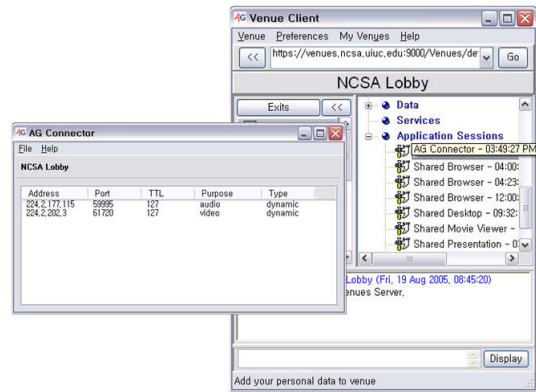


그림 6 . 구현된 AG Connector의 스크린 샷.

제한된 AG Connector는 다자간 협업 환경인 AG에서 동작할 수 있도록 구현되었다. AG 사용자들이 AG 툴킷을 통해 쉽게 활용할 수 있도록, AG에서 제공되는 공유 응용 (shared application) [4]으로 개발되었다. 공유 응용은 AG 사용자들이 응용 프로그램을 공유하여 활용할 수 있는 인터페이스로, 메뉴 서버로부터 필요한 정보를 얻을 수 있는 인터페이스를 제공한다. 그림 6은 구현된 AG Connector의 스크린 샷이다. 오른쪽의 메뉴 클라이언트 내에 보이는 공유 응용 세션 목록에서 AG Connector를 실행하면, 왼쪽처럼 현재 메뉴에서 활용 중인 멀티캐스트 그룹 주소를 보여주는 사용자 인터페이스가 제공되며 멀티캐스트 터널링 기능이 실행된다. 또한, AG Connector는 UGMP를 구현하여, 멀티캐스트가 불가능한 노드가 자신의 네트워크를 대표하는 UMTF agent에게 멀티캐스트 터널링을 요구할 수 있도록 하였다. AG Connector의 구현 결과와 최신 상태는 [9]에서 확인할 수 있다.

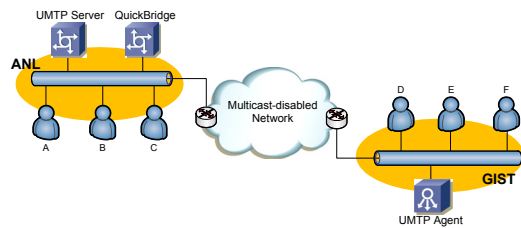
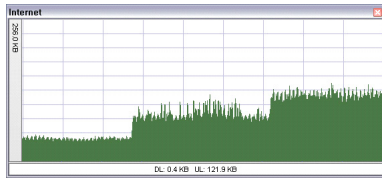
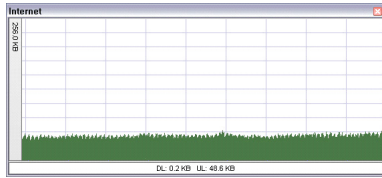


그림 7. 실험 환경.

AG Connector는 기존에 QuickBridge 에서 제공되지 못했던 연결성 문제를 해결하기 개발되었다. UMTF 기반의 멀티캐스트 연결성 솔루션을 구현함으로써, QuickBridge보다 더 작은 네트워크 대역폭과 시스템 성능을 소모하는 시스템을 개발하였다. 이러한 AG Connector의 특성을 검증하기 위해 그림 7과 같은 실제 인터넷 환경에서 동작하는 실험 환경을 구축하였다. AG Connector와 QuickBridge의 네트워크 대역



(a) QuickBridge



(b) AG Connector

그림 8 . 네트워크 대역폭 사용량 비교.

폭 소모량을 비교하기 위해미국, ANL (Argonne National Laboratory)와 한국, GIST (Gwangju Institute of Science and Technology)를 연결하였다. 이 두 네트워크들은 상호간에 멀티캐스트 연결성을 갖지 않는다. 각 네트워크에는 각각 3개의 노드들이 동작한다. ANL 측의 노드들은 오디오, 비디오 스트림을 전송하는 역할만을 하고, GIST 측의 노드들은 ANL 측에서 전송되어 오는 네트워크 스트림을 수신하는 역할만을 한다. 결과를 쉽게 보이기 위해 5개의 비디오 스트림을 보내 총 스트림의 양을키웠다 (A: 비디오 3개, B: 비디오 1개, C: 비디오 1개). 두 번의 실험을 통해, QuickBridge와 AG Connector의 대역폭 소모량을 비교하였다. 첫 번째 실험에서는 ANL 측에서 QuickBridge 서버를 실행하고, GIST측의 노드들은 이 QuickBridge 서버에 접속해 비디오 스트림들을 수신한다. GIST 측의 노드들을 한 번에 한 노드씩, 3분 간격으로 접속 시켜 네트워크 대역폭 소모량을 명확히 보일수 있도록 하였다. 그림 8(a) 는 QuickBridge 서버의 실험 결과이다.

두 번째 실험으로 동일한 네트워크 환경에서 AG Connector를 이용한 네트워크 대역폭 사용량을 체크하였다. 첫 번째 실험에서 QuickBridge 서버를 실행했던 ANL 측 머신에서 UMTF server를 실행하였다. GIST측에서는 노드 E가 UMTF agent를 실행하고, UMTF server와의 연결을 설정하였다. 첫 번째 실험과 마찬가지로 3분 간격으로 각 노드를 접속시켰다. 그림 8(b) 는 AG Connector를 이용한 실험 결과이다. 그림 8(a) 와 그림 8(b)를 비교해 보면, 그림 8(a) 에서는 새로운 사용자가 추가 될 때마다, QuickBridge 서버에서 소모되는 네트워크 대역폭이 증가하는 것을 알 수 있다. 반면에, 그림 8(b) 에서 AG Connector에서는 새로운 사용자가 추가되어도 추가적인 네트워크 대역폭 소모가 발생하지 않는다는 것을 알 수 있다. 이 실험 결과를 통해, AG Connector가 QuickBridge에 비해 네트워크 대역폭 효율적이라는 점을 보였다.

5. 결론

본 논문에서 우리는 AG의 다자간 협업 환경을 위한 멀티캐스트 연결성 솔루션으로 UMTF를 확장한 형태의 AG Connector 제안하였다. AG Connector는 UMTF를 기반으로 구현되어, 멀티캐스트 전송을 통한 네트워크 대역폭의 효율적으로 활용과, 데이터 캡슐화를 통한 하나의 포트 번호 이용으로 방화벽 환경에서도 활용이 가능하다. 우리는 실험을 통해 AG Connector가 QuickBridge에 비해 효율적임을 보였다. 앞으로 현재의 AG Connector를 보완하여, 메뉴에 참여하는 노드들 사이의 멀티캐스트 가능성과 상호 간의 멀티캐스트 연결성을 검사하는 부분을 추가할 계획이다. 이를 통해 AG 사용자들이 자신의 네트워크 환경에 투명하게 AG를 활용할 수 있는 멀티캐스트 연결성 솔루션으로 발전시킬 계획이다.

감사의 글

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업의 지원에 의한 것임.

참고문헌

- [1] ANL Futures Laboratory, Access Grid Toolkit (version 3.0.2), <http://www.accessgrid.org/>
- [2] R. Finlayson, "The UDP multicast tunneling protocol," IETF Internet Draft, Nov. 2003.
- [3] M. Daw and J. T. von Hoffman, "Guide to network bridging on the Access Grid," Access Grid Documentation Project, Mar. 2004.
- [4] MCS Futures Lab., Argonne National Laboratory, "Programmer's manual - shared applications," Jan. 2004.
- [5] liveGate, <http://www.live.com/liveGate/>.
- [6] P. Parnes, K. Synnes, and D. Schefstrom, "mTunnel: A multicast tunneling system with a user based quality-of-service mode," Computer Communication, vol. 21, no. 15, Apr. 1998.
- [7] RCBridge, <http://if.anu.edu.au/SW/rcbridge.html>.
- [8] N. Kim and J. Kim, "UDP-tunneling based multicast connectivity solution for multi-party collaborative environments," in Proc. SPIE, Oct. 2005.
- [9] Networked Media Lab., "AG connector," <http://ace.netmedia.gist.ac.kr/AG-Connector/>.