

온라인 게임 QA를 위한 네트워크 에뮬레이션 기술동향 Network Emulation Technology Trend for Online Game QA

한국전자통신연구원 신광식(kwangsik@etri.re.kr)

■ 1. 서론

2000년대 들어서면서 폭발적으로 증가한 인터넷 망의 확충과 보급을 기반으로 온라인 게임은 국내 게임 산업을 대표하는 문화 아이콘으로 자리잡았다. 2008년 게임 백서에 의하면 2007년 국내 온라인 게임 시장은 2조 2,403억 원 규모로 전체 게임 산업의 43.5%를 차지하고 있다. 2007년에는 바다이야기의 여파로 국내 게임시장이 전년 대비 30.9%나 감소하였음에도 불구하고, 온라인게임은 26% 이상 고 성장하였고, 이 시기에 온라인게임 수출액은 약 7.5억 달러로 전년 대비 24.4%의 높은 성장을 기록하였다[1]. 이는 전체 게임수출의 95.5%로 온라인 게임이 해외수출에서 절대적인 위치를 차지하고 있음을 보여주고 있다. 2008년 수출 10억 달러를 달성하는 등 앞으로도 온라인 게임은 그 성장세가 지속될 것으로 기대되기 때문에, 국내 게임산업에 있어서 온라인 게임의 중요성은 보다 더 높아지고 있다.

지난 10년간의 지속적인 성장으로 국내 온라인 게임은 명실공히 세계 1위의 위상을 얻었다. 그만큼 게임 산업규모도 커지면서, 매년 수많은 온라인 게임이 출시되고 있지만, 과도한 출시경쟁으로 인해 완성도가 떨어지는 게임이 난립하는 문제가 발생하고 있다. 이는 게임을 개발한 후 게이머에게 선보이기 전에 마지막으로 점검하는 단계인 QA를 충분히 하지 않음으로 써 발생한다. 게임 개발 경력이 많은 '블리자드'나 '코나미' 등과 같은 해외 개발사의 경우 이미 QA가 체계화되어 있지만, 대부분의 한국 개발사는 전문화된 QA 인원을 충분히 확보하지 않은 상태에서 단순 '버그테스트' 정도만 수행하는 수준이다[2]. 온라인 게임은 일반 게임과 달리 온라인이라는 특수한 네트워크 환경에 영향을 받기 때문에, 일반 사용자를 테스터 모집하는 등 QA 과정에서 특히 어려움을 겪곤 한다.

온라인 게임은 다양한 네트워크 환경을 갖는 게이머에게 서비스를 제공하기 위해 저 대역의 네트워크 환경에서도 원활한 서비스가 가능하도록 통신프로토콜을 설계한다. 게임의 장르와 성격에 따라 차이는 있지만, 최근의 온라인 게임은 게임자체의 상호작용을 극대화하고 다른 네트워크 조건하에서 게임에 참여하는 게이머 사이에 비교적 균등한 환경을 제공하기 위해서 가장 좁은 대역폭을 갖는 네트워크 접속조건(Access Network) 맞춰서 서비스 환경을 제한한다. 최근 이러한 온라인 게임 서비스 환경에 대한 게임 패킷의 트래픽 특성에 대한 분석연구가 수행되었다. 분석 결과에 따르면, 온라인 게임에서 사용되는 패

킷은 하나하나의 패킷 크기는 작지만, 주기적으로 많은 수의 패킷이 폭발적(burst)으로 발생하는 특성을 보인다[3-4]. 이처럼 패킷이 불규칙하게 전달되는 이유는 게임진행과정에서 게이머의 행동변화도 있지만, 해당 패킷이 네트워크를 통해 전달되는 과정에서 발생하는 지연, 특히 네트워크 상의 혼잡으로 인한 혼잡지연 등의 영향이 더욱 크다. 그러므로, 온라인 게임에 대한 성공적인 QA를 위해서는 실제네트워크와 유사한 테스트 환경을 구축하여, 다양한 네트워크 특성에 대해 테스트하는 것이 요구된다. 본 논문에서는 대규모 사용자 서비스용 온라인 게임 서버 QA를 위한 네트워크 에뮬레이션 기술을 소개한다.

■ 2. 온라인 게임 QA 기술 동향

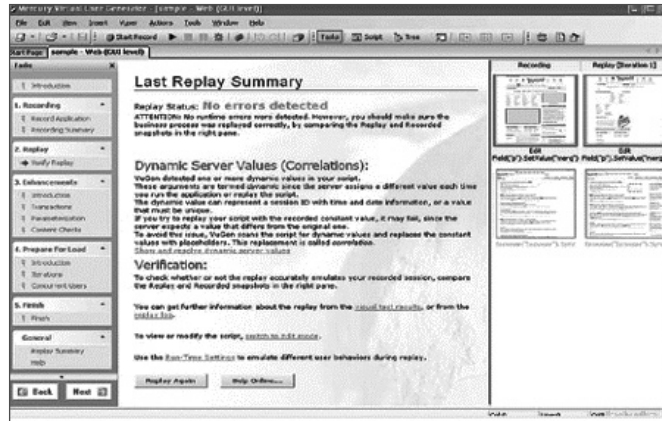
게임 QA팀에서 수행하는 테스트 영역은 크게 기획, 프로그램, 게임 플레이, 그래픽/사운드, 서버/네트워크 영역으로 구분된다. 이중 서버/네트워크 영역은 사실상 게임 개발이 완료된 이후에 온라인 게임 서비스의 안정성을 검증하는 시험 단계로 일반 게이머들에게 서비스를 시작하기 전에 마지막 수행하는 최종 점검과정으로 생각할 수 있는 가장 중요한 테스트 영역이다. 주로 게이머의 원활한 네트워크 접속과 게임 진행, 다양한 네트워크 사용환경으로 인해 발생할 수 있는 문제 등을 점검하여, 대규모 게이머가 동시 접속했을 때 발생하는 서버의 부하 및 이상 동작을 막고 접속하는 네트워크 환경에 관계없이 모든 게이머가 일관성 있는 서비스를 받을 수 있도록 해결안을 찾는 과정을 포함한다.

2.1 기존의 서버 부하테스트 기술

그 동안 인터넷의 발전과 함께 원활한 웹 기반(웹, 인트라넷 등) 서비스를 제공할 수 있는 서버구축을 위해 다양한 서버 부하테스트 기술이 발전되어 왔다. 서버 부하테스트와 성능 측정기술로서 현재 제품화되어 많이 사용되고 있는 도구로 HP의 LoadRunner, Compuware의 QALoad, 그리고 Empirix의 e-Load와 OneSight 등이 있다[5-8].

HP의 LoadRunner는 일반 응용프로그램용 서버 테스트 시장의 50% 이상을 점유하고 있는 부하테스트 및 분석 프로그램으로 웹 서버 부하 테스트를 목적으로 개발되었다. 수 천명의 가상유저를 생성하여 부하를 발생시킴으로써 웹 서버의 부하를 테스트하는 구조로 동작하며, 테스트 스크립트 작성이 용이하고, 다양한 프로토콜을 지원하여 웹 서버 외에도 다양한 응용프로그램에 대해 서버의 성능테스트가 가능하기 때문에 현재 가장 많이 사용되고 있는 부하테스트 도구이다. 최근 게임 서버 테스트 기능이 추가되어 일부 온라인 게임서버 테스트 용으로 사용되고 있지만, 서비스 되는 일련의 패킷 사이에 상관관계가 적은 웹 서비스와 달리 게이머와 게임서버간의 상호작용이 많은 온라인 게임에서는 단순한 부하 테스트 이상의 온라인 게임서버에 대한 QA 도구로 사용하기에는 부족한 점이 많다. 또한 테스트에 사용되는 가

상유저의 개수에 따라 비용을 지불해야 하기 때문에 대규모 사용자 접속 환경을 구축하고 테스트하는 온라인 게임 QA 도구로 사용하기에는 무리가 있다. 그림 1은 HP의 LoadRunner를 보여준다.



[그림 1] HP의 LoadRunner

Compuware의 QALoad는 클라이언트/서버, ERP, e비즈니스, 웹 응용프로그램에 대한 자동화된 부하 테스트 도구로서 패킷 캡처를 통해 데이터를 수집하고 통계결과를 이용하여 시스템의 운영 전 단계에서 시스템의 성능 및 안정성 관련 문제점을 식별하여 최적화 한다. Empirix의 e-Load는 박스당 1000명의 가상 유저를 생성하여 서버의 안정성을 테스트 할 수 있는 웹 서버 성능 테스트 도구로 성능 측정 도구인 OneSight와 함께 사용이 가능하다.

이처럼 기존의 서버 부하테스트 기술은 주로 웹 응용프로그램을 서비스하는 서버 테스트를 목적으로 개발되었기 때문에, 서버부하 테스트 외에 온라인 게임과 같이 서버와 클라이언트 간의 인터랙션이 많은 시스템에서 QA를 수행하기에는 한계가 있다.

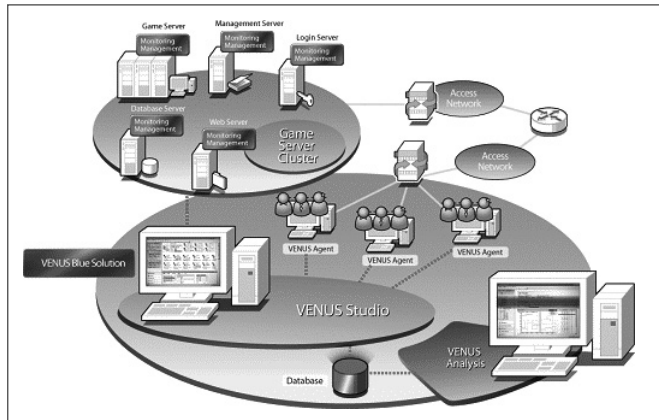
2.2 온라인 게임 서버 테스트 기술

현재 대다수의 온라인게임 업체에서 서버/네트워크의 QA를 위한 테스트 방법은 대규모의 테스터를 모집하여 최종 검증이 끝나지 않은 게임을 선발된 게이머에게 공개하여 게임에 참여토록 하여 그들의 의견을 수렴하는 방법을 사용한다. 물론 실제 사용자가 온라인을 통해 게임에 참여하는 동안 발생하는 데이터를 기준으로 QA를 수행하는 것은 가장 정확한 방법이다. 그러나, 대규모의 게임테스터를 모집하여 테스트를 수행하기 위해 서는 많은 시간과 비용이 소모된다. 뿐만 아니라 수 많은 사람들이 자유롭게 게임을 즐기는 과정에서 발생하는 문제점이나 의견을 수집하기 때문에, 특정상황에 대한 재현이 사실상 불가능하고 테스트를 원하는 시나리오에 맞춰 테스터들을 통제하는 것이 힘들다.

한국전자통신연구원에서는 이와 같은 문제를 해결하고자, 대규모의 가상유저를 생성하여 게임유저로

서 온라인 서버에 접속하여 테스트를 수행하는 비너스 블루를 개발하였다.

가상 유저는 그래픽과 사운드 출력 등이 없이 서버와의 통신 모듈만 가진 일종의 클라이언트로 하나의 호스트에 수백 명의 가상 유저가 생성 될 수 있으며, 간단한 사용자 인터페이스를 이용하여 제어된다. 가상유저를 생성하기 위한 SDK를 제공하여 이를 기반으로 가상유저 제어함으로써, 대규모의 다양한 클라이언트 움직임을 발생시킴으로써 게임 서버에 대용량의 부하를 발생시켜 게임 서버의 안정성과 성능을 테스트한다. 가상 유저가 생성되는 각 호스트는 서로 다른 네트워크 환경을 고려하여, 대역폭, 지연, 손실 등의 네트워크 속성을 각기 다르게 설정할 수 있다. 테스트 결과에 대한 분석과 모니터링을 제공하며, 특정 게임 장르에 종속되지 않는 구조를 가지고 있어, 재 사용성이 뛰어나고 기본 시뮬레이션 기능과 프로토콜의 조합을 통해 다양한 상황에 대한 시뮬레이션이 가능하다. 그림 2는 비너스 블루의 시스템 구성도를 보여준다.



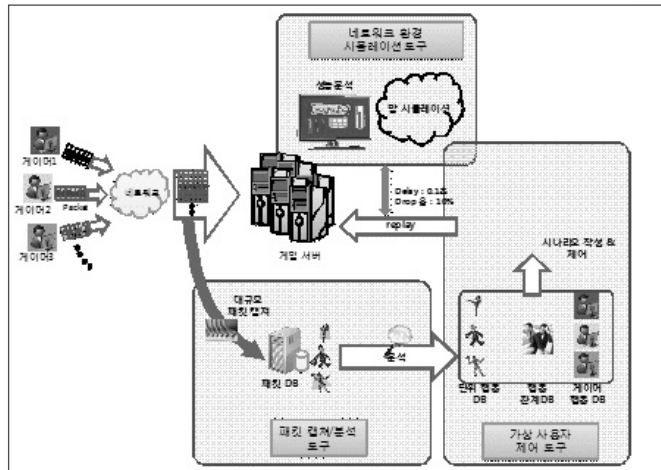
[그림 2] VENUS Blue 전체 시스템 구성도

2.3 최근의 온라인 게임 QA 기술

게임사에서 개발과 서비스를 함께하던 초기와 달리 최근에는 게임을 만드는 개발사와 이를 게이머에게 서비스하는 퍼블리셔가 분리되어 각각의 분할된 업무에 대해 독립적으로 운용되고 있다. 게임개발 이후 최종적으로 수행하는 서버/네트워크 영역에 대한 QA는 실제 서비스를 책임지고 있는 퍼블리셔에서 주로 수행한다. 그러나, 비너스 블루는 게임클라이언트로부터 최소한의 서비스가 가능한 게임 로직과 네트워크 모듈만을 추출해야만 가상유저를 생성할 수 있다. 게임 개발사 입장에서는 자사의 클라이언트 프로그램에서 핵심모듈만 추출하여 비너스 블루에서 제공하는 SDK를 이용하여 가상유저를 만드는 것이 가능하지만, 퍼블리셔는 게임클라이언트를 자유롭게 수정할 수 없으므로 게임사의 도움 없이는 사실상 비너스 블루를 사용한 서버테스트가 불가능하다. 게임 개발사와 퍼블리셔가 각기 다른 회사로 운영되는 현 상황에서 개발사에게 퍼블리셔에게 소스코드를 제공하는 경우는 거의 없다. 그러므로, 한국전자통신

연구원에서는 이러한 퍼블리셔의 입장을 반영하여 게임 클라이언트 소스코드 임베딩이 필요 없이 QA를 수행할 수 있도록 새로운 버전의 온라인 게임 QA 도구(OQA)를 개발하고 있다.

OQA는 게임에 사용되는 프로토콜을 분석하고, 분석된 게임 프로토콜을 적용한 가상 유저를 대규모로 생성하여 게임 서버의 부하 테스트를 수행한다. 이는 게임 클라이언트의 다양한 액션을 수행하는 사실적 가상 유저를 생성하고, 가상 유저 구현에 소스 코드 수정을 요구하지 않는다. 또한, 대규모로 생성되는 가상 유저의 사실적이고 효율적인 제어를 위해 가상 유저의 역할 및 목표를 미리 정의할 수 있는 시나리오 적용 기능을 제공하고, 다양한 게임 클라이언트 네트워크 환경을 적용할 수 있도록 네트워크 에뮬레이션 기능을 포함한다. 그림 3은 OQA의 전체 구성도를 보여준다.



[그림 3] OQA 전체 시스템 구성도

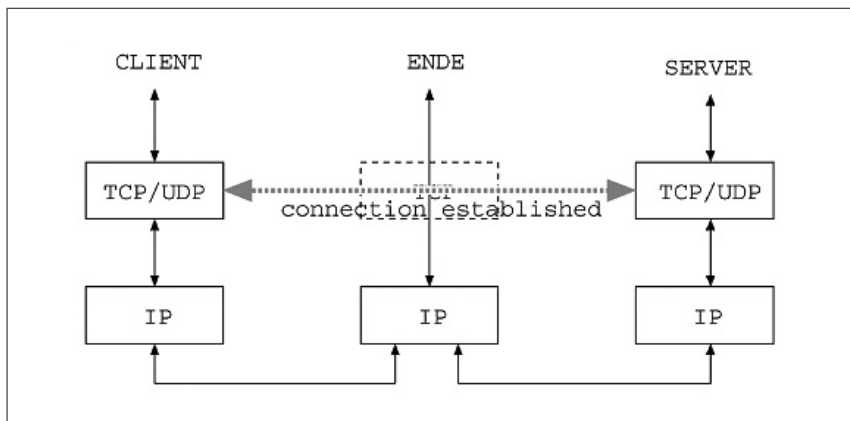
비즈니스 블루가 각 PC 호스트에 가상 유저를 자동 생성하고 제어, 관리하는 구조라면, OQA는 게임 패킷 분석 도구, 가상 유저 제어 도구, 네트워크 환경 에뮬레이션 도구가 사용자의 목적에 맞게 분리, 결합 적용되어 호스트 PC상에 설치되는 독립적인 구조를 갖는다.

■ 3. 네트워크 에뮬레이션 기술 동향

네트워크 기반의 다양한 서비스가 발전함에 따라 이를 검증하기 위한 테스트 환경 연구에 대한 관심이 높아지고 있다. 서비스 트래픽을 임의의 네트워크 환경을 통해 서비스를 함으로써 실제 네트워크의 속도와 환경을 반영하는 네트워크 에뮬레이션 기술은 온라인 서비스를 미리 테스트할 수 있는 검증기술로서 각광받고 있다. 기존의 네트워크 에뮬레이션 기술은 구조적인 측면에서 크게 단일 노드에서 중단-

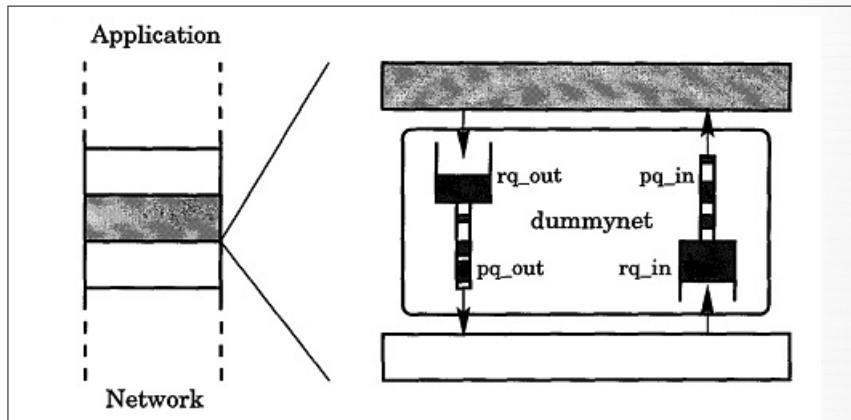
대-종단 네트워크 특성을 적용하는 단일노드 에뮬레이션 기술과 네트워크 토폴리지를 구성하여 패킷이 이동해 가면서 자연스럽게 네트워크 특성에 적용 받는 다중노드 에뮬레이션 기술로 분류된다.

단일노드 에뮬레이션 기술은 소프트웨어 모듈로 주로 송신 프로그램과 같은 호스트에서 동작하며 특정 네트워크 설정을 적용한 후 패킷을 네트워크로 전송한다. 호스트 단에서 동작하기 때문에 에뮬레이션 기술 적용이 쉽고, 호스트 수 증가를 통해 에뮬레이션 데이터 확장이 용이한 반면, 네트워크의 종단-대-종단 특성을 각 패킷에 대해 임의의 연산모델로서 적용하기 때문에, 네트워크 상태와 다른 패킷과의 상호관계에 의해 실제 라우터의 큐에서 발생하는 큐잉지연, 큐잉손실 등의 재현에 취약하고, 라우팅 프로토콜과 같은 프로토콜 환경 변화에 대한 에뮬레이션은 불가능하다. 단일노드 에뮬레이션 기술은 그 기능이 수행되는 시스템 레벨에 따라서 응용프로그램 의존적인 유저모드 방식과 응용프로그램과 독립적으로 네트워크 드라이버 단에서 동작하는 커널모드 방식으로 분류된다.



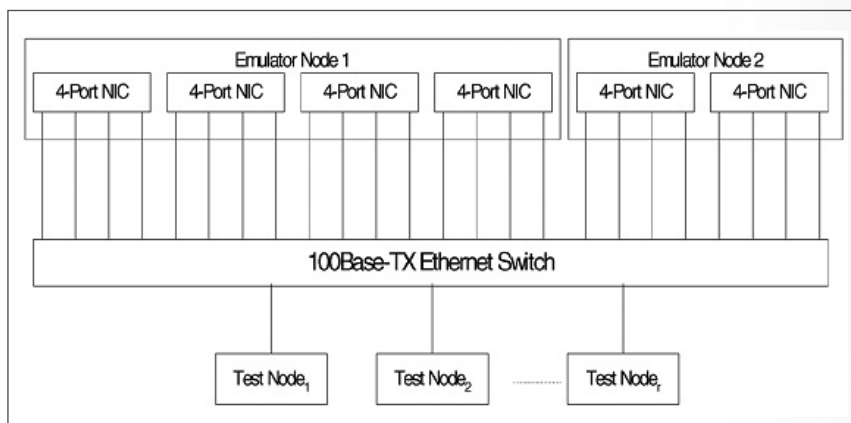
[그림 4] ENDE를 사용한 클라이언트와 서버의 연결

그림 4의 ENDE와 같이 유저모드에서 동작하는 에뮬레이션 기술은 네트워크 모델을 단순화하여 쉽게 구현할 수 있는 반면, 타겟 응용프로그램이 변경될 때마다, 에뮬레이터를 그에 맞게 변경하거나, 타겟 응용 프로그램을 에뮬레이터에 맞게 작성해야 하는 등 확장성이 부족하다[5]. 반면, 그림 5의 Dummynet과 같이 커널모드에서 동작하는 에뮬레이션 기술은 네트워크 드라이버 단에서 패킷을 가로채서 에뮬레이션을 적용하기 때문에 응용프로그램이 바뀌어도 그대로 사용할 수 있지만, 복잡한 알고리즘 적용이 어렵다[6].



[그림 5] Dummynet의 동작원리

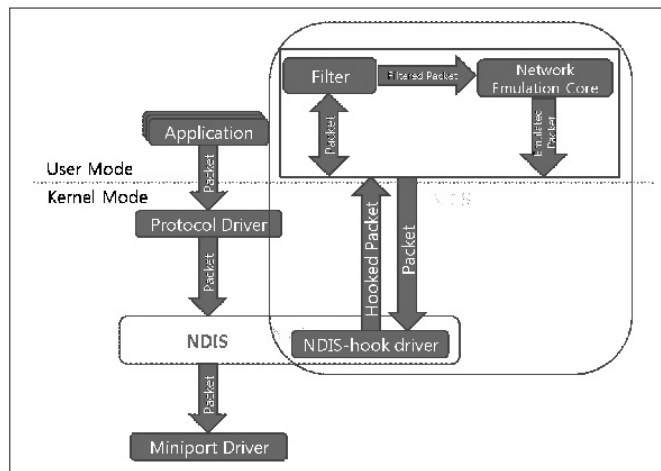
한편, 대표적인 다중노드 에뮬레이션 기술인 그림 6의 EMPOWER는 호스트 단에서 동작하는 소프트웨어 모듈로 동작하는 대신 독립된 에뮬레이션 장치로써 네트워크 포트를 통해 송/수신 노드와 연결이 되는 구조를 갖는다[7]. 시스템 내부에 가상의 네트워크 토폴로지를 생성하여 라우팅 알고리즘에 따라 패킷이 이동하면서 네트워크 특성이 적용된다. 송신노드로부터 전달받은 패킷을 가상의 네트워크를 통해 이동시키는 과정에서 에뮬레이션을 수행하고 최종 수신노드로 전달한다. 실제 네트워크 구성과 유사한 가상의 네트워크 토폴로지를 구성하고 패킷이 각 노드를 통해 이동하는 과정에서 네트워크 상태와 다른 패킷에 영향을 받기 때문에 단일노드 에뮬레이터에서 재현이 어려웠던 네트워크 특성(큐잉 손실, 큐잉 지연 등)의 재현이 용이하다.



[그림 6] EMPOWER의 물리적 구조

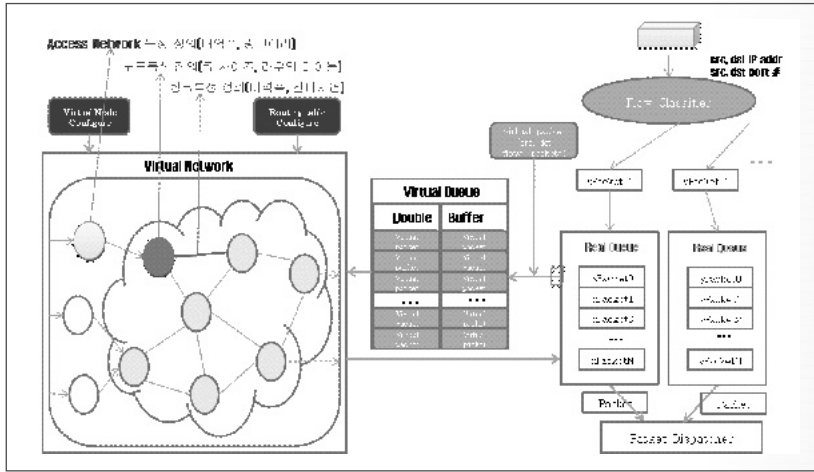
■ 4. 온라인 게임 QA를 위한 네트워크 에뮬레이션 기술 소개

본 논문은 현재 한국전자통신연구원에서 개발중인 대규모 트래픽을 처리하는 온라인 게임 서버의 QA를 위한 네트워크 에뮬레이션 기술을 소개한다. 대규모 사용자의 트래픽에 대한 에뮬레이션을 위하여 에뮬레이터는 호스트 단의 소프트웨어 모듈로써 동작하여 호스트에서 생성한 패킷을 에뮬레이션 한 후 서버로 전달한다. 호스트 단에서 동작하는 기존의 에뮬레이터는 종단-대-종단의 단일노드 에뮬레이션을 수행하였지만, 본 논문은 에뮬레이터 내에 가상의 네트워크 토폴로지를 구성하여, 가상패킷이 노드사이를 이동하면서 다양한 네트워크 특성이 재현되는 다중노드 에뮬레이션 기술이다. 응용프로그램 독립적으로 에뮬레이션을 수행하면서 복잡한 알고리즘 적용을 용이하게 하기 위하여, 그림7과 같이 응용프로그램에서 전송한 패킷을 커널모드의 네트워크 드라이버(NDIS)단에서 가로챌 후 유저모드의 에뮬레이션 모듈에 전달하여 에뮬레이션이 끝나면 다시 네트워크 드라이버를 통하여 외부로 전송하는 구조를 갖는다.



[그림 7] 호스트상에서의 패킷 흐름 구조

기존의 단일노드 에뮬레이터는 특정 알고리즘 적용 결과에 따라 개개 패킷의 전송 시간을 조절함으로써 패킷 지연, 손실, 순서 바뀜(out of order) 등을 재현하는 수준에 그쳤지만, 제안하는 기술은 그림 8과 같이 가상의 네트워크 토폴로지를 구성하여 다수의 노드를 생성한 후 각 노드의 라우팅 테이블에 따라 패킷이 이동하면서 발생하는 전파지연, 큐잉지연, 큐잉손실, 지터 등을 재현한다. 에뮬레이션은 두 단계로 나누어져 동시에 수행되는데, 첫째는 네트워크 드라이버로부터 가로챌 패킷을 플로우 별로 분류한 후 패킷 큐(Real Queue)에 저장하고 실제 패킷에 대한 참조 값으로 가상패킷(Virtual Packet)을 생성하여 가상패킷 큐(Virtual Queue)에 저장하고, 두번째 단계에서는 가상패킷 큐로부터 패킷을 읽은 후 가상 네트워크(Virtual Network)를 구성하는 가상노드의 라우팅 테이블과 링크 속성에 따라 패킷을 전달한다.

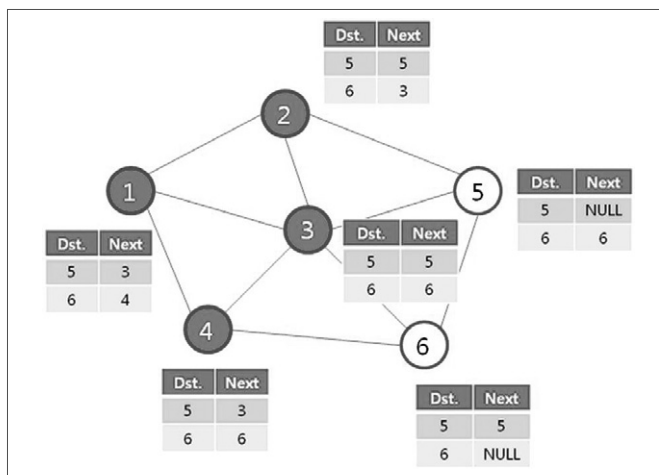


[그림 8] 에뮬레이터 코어 구조와 실제 및 가상 패킷의 흐름

그림에서 플로우 분류기(Flow Classifier)는 패킷의 IP헤더에서 실제 패킷이 전송하게 될 송신 IP, port#와 수신 IP, port#를 검사하여 사용자가 지정한 규칙에 따라 플로우를 분리한다. 각 플로우는 고유의 실제 패킷 큐(Real Queue)에 저장되고, 실제 패킷에 대한 가상패킷(패킷 데이터 대신 실제 패킷이 저장된 주소 값만 갖고 있음)을 생성하여 가상 네트워크 내부에서 사용될 src, dst 노드 주소(가상네트워크 내부에서 처음 패킷이 전달되는 가상노드와 외부로 전송 전에 마지막으로 전달될 가상노드 주소)와 플로우 아이디, 패킷의 시퀀스 번호(에뮬레이션 네트워크 모니터링 및 분석을 위해 각 패킷마다 고유의 식별 번호 부여)를 부여한 후 가상패킷 큐에 저장한다. 실제 패킷 큐는 빠른 처리를 위하여 검색과정이 필요 없는 순환 큐 구조를 갖고 각 플로우 별로 저장공간을 할당하는데, 이는 에뮬레이션이 끝날 때 까지 실제 패킷은 큐에 머물러야 하므로 지연이 길어서 큐에 오래 머물러야 하는 패킷으로 인해 실제 패킷 큐에서 낭비되는 공간이 발생하는 것을 막아준다. 가상패킷 큐는 플로우의 구분 없이 가상 패킷을 저장하며, 두 스레드가 동시에 가상패킷 큐에 접근할 수 있도록 동기화를 위해 별도의 메모리 접근제한이 필요 없는 더블 버퍼로 구성된다.

가상네트워크(Virtual Network)는 사용자가 지정한 네트워크 환경(각 노드의 대역폭, 큐 크기, 링크의 전파지연 등)과 라우팅 알고리즘(각 노드의 라우팅 테이블작성)에 따라 네트워크 토폴로지를 구성한 후 가상패킷 큐로부터 패킷을 읽어오고 가상노드 사이로 패킷을 전달하여 dst노드에 도달한 가상패킷이 참조하는 실제 패킷은 패킷 발송기(Packet Dispatcher)에 의해 외부로 전송된다. 가상네트워크에서 손실된 가상패킷을 참조하는 실제 패킷은 외부로 전송하지 않고, 제거한다. 이처럼 가상패킷이 가상 네트워크를 통과하는 동안 발생하는 지연과 손실은 실제 패킷에 그대로 적용됨으로써 실제 네트워크 에뮬레이션이 적용된다. 온라인 서버 테스트를 위해 에뮬레이션을 적용할 때는 가상네트워크 내부의 목적지노드

(dst노드)로 지정되는 가상노드는 미리 정하여, 각 가상노드는 라우팅 테이블에 가상네트워크 내의 모든 노드에 대한 정보를 기록할 필요 없이 테스트 과정에서 목적지 노드로 지정된 가상노드에 대한 라우팅 정보만을 기록함으로써 라우팅 정보를 보다 효율적으로 관리한다. 그림 9는 5, 6번 노드 만이 목적지노드로 사용될 때 각 가상노드의 라우팅 테이블의 예를 보여준다. 그림에서 5, 6번 노드의 라우팅 테이블에서 Next가 NULL인 것은 실제 패킷을 외부로 전송함을 나타낸다. 각 가상노드마다 라우팅 테이블을 가지고 있으므로, 동적 라우팅 알고리즘 적용이 가능하므로 동적인 라우팅 경로 변경으로 인하여 순서 바뀜(out of order)와 같은 네트워크 특성에 대한 에뮬레이션이 가능하다.



[그림 9] 가상 네트워크에서 가상노드의 라우팅 테이블 예

가상 네트워크 내부에서 노드와 노드 사이의 링크 속성은 각 가상노드에서의 패킷 이동 속도를 조절함으로써 정의한다. 링크 속성은 크게 전파지연과 대역폭으로 정의 하는데, 전파지연은 가상노드와 가상노드 사이의 패킷 이동 시간으로 설정하고, 대역폭은 토큰버킷 모델에 기반한 알고리즘을 구현하여 각 노드마다 지정된 버킷크기와 토큰생성률으로써 평균전송률과 최대전송률을 설정한다. 한편, 각 가상노드에 패킷이 도달하고 전송될 때마다 해당 패킷의 정보와 시간, 손실 정보 등을 기록하여 에뮬레이션 네트워크의 동작 상태를 모니터링 할 수 있다.

5. 결론

본 논문은 대규모 사용자 서비스용 온라인 게임 서버 QA를 위한 네트워크 에뮬레이션 기술을 소개하였다. 호스트 단에서 동작하는 에뮬레이터 입에도 불구하고, 가상의 네트워크 토폴로지를 이용하는 다중노드 에뮬레이션 기술을 사용하기 때문에 네트워크 상태와 다른 패킷과의 상관관계에 의해 실제 노드에

서 발생하는 큐잉 지연, 큐잉 손실, 지터 등의 네트워크 특성을 에뮬레이션한다. 실제 네트워크 상의 라우터와 마찬가지로 각 가상노드가 라우팅 테이블을 가지고 있어서 실제 네트워크에서 운용되는 동적 라우팅도 적용할 수 있으므로 실제에 가까운 네트워크 환경 재현이 가능하다. 본 논문에서 제안하는 네트워크 에뮬레이션 기술은 각 호스트 단에서 에뮬레이션이 이루어 지기 때문에 대규모 사용자 서비스 테스트를 위해 호스트의 수를 확장함으로써 테스트에 필요한 트래픽을 확보할 수 있으므로, 대규모 트래픽을 서비스하는 온라인 서버 QA에 효과적으로 사용할 수 있다.

참고문헌

- [1] 한국게임산업개발원 산업정책팀, "2008 대한민국 게임백서", 한국게임산업진흥원, 2008년 8월
- [2] 조학동, "한 온라인 게임, 바뀌어야 산다 [3부] QA 강화 정책", 게임동아, 2008년 12월 24일
- [3] Wu-chang Feng, Francis Chang, Wu-chi Feng, and Jonathan Walpole, "A Traffic Characterization of Popular On-Line Games," IEEE/ACM Transactions on Networking, vol. 13, no. 3, pp. 488-500, June 2005
- [4] Kuan-Ta Chen, Polly Huang, and Chin-Lanung Lei, "Game traffic analysis: An MMORPG perspective," Computer Networks, vol. 50, no. 16, pp. 3002-3023, November 2006
- [5] Ikjun Yeom, A.L. Narasimha Reddy, "ENDE: An End-to-end Network Delay Emulator Tool for Multimedia Protocol Development," Multimedia Tools and Applications, vol.14, no.3, pp.269-296, August 2001
- [6] Luigi Rizzo, "Dummysnet: a simple approach to the evaluation of network protocols," SIGCOMM Comput. Commun. Rev., vol. 27, no. 1, pp. 31-41, January 1997.
- [7] Pei Zheng; Ni, L.M., "EMPOWER: a cluster architecture supporting network emulation," Parallel and Distributed Systems, IEEE Transactions on , vol.15, no.7, pp. 617-629, July 2004

신 광 식



- kwangsik@etri.re.kr
- 인하대학교 전자공학과 박사 졸업
- 현 ETRI HD게임연구팀 Post Doc.
- 관심분야 : 온라인 게임 테스트기술