토폴로지 정보를 이용한 다중 목적지 전송*

이승준[0] 예경욱 문수복
한국과학기술원
sjlee@an.kaist.ac.kr kwye@camars.kaist.ac.kr sbmoon@cs.kaist.ac.kr

MSCP: Topologically-aware Multiple-destination Secure CoPy

Seungjun Lee[0] Kyungwook Ye Sue Bok Moon
Dept. of Computer Science, Korea Advanced Institute of Science and Technology

요 약

인터넷에서 전 세계에 분포되어 있는 수많은 노드로 파일을 전송하는 경우 어떻게 하면 빨리, 효율적으로 전송할 것인가가 큰 문제가 된다. 전송을 순차적으로 할 경우, 수행시간이 길어지고 되고, parallel stream으로 전송하여 수행시간을 줄인 경우에도 송신자 링크의 병목현상이라는 문제가 발생한다. 본 연구에서는, 토폴로지 정보를 이용하여 지역적으로 가까운 노드들은 그룹으로 만들고, 각 그룹 간의 한 번의 전송을 통해, 전송 시간을 줄이고, 송신자 링크의 병목 현상을 해결하기 위한 방법을 제안한다.

1. 서 론

최근 인터넷에서의 많은 작업 중의 하나는 전 세계에 분포되어 있는 수백 개의 노드에 파일의 전송하는 것이다. 따라서 전 세계에 수많은 노드가 있는 환경에서, 파일을 그 수많은 노드로 전송하기 위한 빠르고, 효율적이며, 실용적인 방법이 필요하다. 전체 노드들로 파일을 전송하는데 사용할 수 있는 가장 간단한 방법은 순차적 전송이다. 그러나 이 방법은 하나의 노드에 전송을 완료하고 나서야 비로소 다른 노드로의 전송을 시작하기 때문에 매우 비효율적이다. 만약, 처음 전송을 시작하는 노드와의 링크 상태가 원활치 못하면 전송이 완료될 때까지 기다려야 되기 때문에, 전체 노드로 전송하는데 많은 시간이 걸리게 될 것이다. 다른 방법으로는, parallel stream으로 여러 노드에 동시에 전송하는 방법이 있을 수 있다. 그러나 이 방법은 송신자 링크에 병목현상이 발생시킬 수 있다. 토폴로지 정보를 이용하여 전송트리를 만들어 이러한 문제를 해결할 수 있다. 그러나 인터넷은 수시로 변하기 때문에 정확한 토폴로지 정보를 얻는 것은 현실적으로 어렵다. 따라서 우리는 문제의 범위를 PlanetLab[1][2]으로 한정하여, PlanetLab 상에서 실제 사용될 수 있고, 송신자 링크 병목 현상과 전송지연이라는 문제를 해결하는 방법을 제안하고, 구현하여 평가한 결과를 설명한다.

2장에서 PlanetLab이 무엇인가에 대해 설명하고, 3장에서는 제안하는 MSCP(topologically-aware Multi-destination Secure CoPy)에 대해 설명한다. 4장에서 성능평가결과를 설명하고, 5장에서 향후 계획에 언급하면서 마무리 한다.

2. PlanetLab

PlanetLab은 전 세계 181곳에 분포되는 있는 400여개의 노드들로 구성된 실험용 네트워크로 사용자는 세계 어떤 노드에도 접속하여 자신이 원하는 실험을 할 수 있는 환경을 제공해준다. 현재 MIT, 버클리, 프린스톤, 인텔, HP 등 많은 대학과 연구소들이 PlanetLab을 사용하여 분산 저장장치, Peer-to-peer, 네트워크 모니터링 등의 다양한 주제로 400 여개의 프로젝트가 수행 중이다. PlanetLab은 Internet-scale에서 실험할 수 있는 거의 유일한 환경이다. PlanetLab 노드는 지역, 링크 속도, 네트워크 크기 등에서 인터넷의 다양한 특성을 반영하고 있어, 새로운 형태의 인터넷 서비스를 위한 좋은 실험 환경이다.

노드는 변형된 Linux PC이고, 사용자가 모든 노드에 SSH[3]로 직접 접속하여 원하는 실험을 할 수 있다. 사용자마다 별도의 가상머신이 생성되기 때문에 해당 노드를 사용하고 있는 다른 사람의 실험에 전혀 영향을 받지 않게 된다. 즉, 하나의 노드를 실제로는 여러 사람이 사용하지만, 나 혼자만 사용하는 것처럼 보인다.

PlanetLab 노드 전체에서 실험하고자 하는 사람은 자신의 프로그램을 SCP[3]을 이용하여 수백 개의 노드들에게 모두 전송하여야 한다. 프로그램이 수정될 때 마다 수백 개의 노드에 모두 재전송해야 된다. PlanetLab 노드의 수가 수천 개에 이르게 되고, 전송해야 되는 파일 크기가 수 백 MB 이상으로 커지면 실험을 위한 프로그램 전송 작업은 많은 시간이 걸리게 될 것이고, 실험의 효율성을 저해하게 될 것이다. 따라서 전 세계적으로 분포되어 있는 수많은 노드들상에서 가능한 빨리 그리고 효율적으로 파일을 전송할 수 있는 방법이 필요하다.

3. MSCP: Topologically-aware Multi-destination SCP

효율적인 전송을 위해 우리가 제안하는 방법은 토폴로지 정보를 이용 지리적으로 가까운 노드들을 그룹으로 만드는 것이다. 그룹간에는 한 그룹의 마스터가 다른 그룹의 마스터에게 전송을 하여, 지리적으로 많이 떨어져 있는 그룹간의 전송 양을 최소화 하여 전송 시간을 줄이고, 네트워크 대역폭의 사용을 효율적으로 하게 된

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았고 대학 IT연구센터 육성 지원사업의 연구결과로 수행되었음.

다. 이런 방법에서 이슈가 되는 것은 어떠한 기준으로 그룹을 형성할 것이냐 하는 것이다. 우리는 그룹을 만들기 위한 토폴로지 정보로 노드의 (1) 도메인 네임, (2)네트워크 거리(송신자로 부터의 RTT)라는 2가지 특성을 이용하였다.

PlanetLab에서 어떤 토폴로지 정보를 이용할 수 있는가에 대해 알아보기 위해, PlanetLab을 구성하고 있는 node들의 도메인네임을 살펴보았다. 같은 네트워크 네임을 갖는 노드들이 존재한다는 사실과, 최상위 도메인 네임이 노드가 위치하고 있는 지리적 위치와 동일하다는 사실을 발견하였다.

PlanetLab에서의 모든 노드들은 고유한 도메인네임을 가지고 있으며, 한 기관이 여러 개의 노드를 갖고 PlanetLab에 참여하고 있다. 예를 들어, KAIST의 경우 4개의 PlanetLab 노드를 갖고 있고 노드의 이름은 각각 csplanetlab{1, 2, 3, 4}.kaist.ac.kr 이다. PlanetLab에 참여 하고 있는 모든 기관들은 이러한 이름 규칙을 갖고 있다. 즉, 도메인 네임을 보고 이 노드가 어느 기관 소속인가에 대한 정보를 얻을 수 있다. PlanetLab 전체 노드들중에서 같은 네트워크 네임을 가지는 노드들을 하나의 그룹으로 볼 수 있고, 같은 그룹에 속해 있는 노드들은 서로 가까운 지역에 위치하고 있다. 예를 들어, planetlab1.lcs.mit.edu와 planetlab2.lcs.mit.edu는 .lcs.mit.edu라는 같은 네트워크네임을 가지고 있으므로 지리적으로도 서로 가까운 위치(같은 학교의 같은 실험실 안)에 존재할 것이라고 추측할 수 있다.

도메인네임의 맨 마지막에는 .kr .jp .uk와 같이 지역을 나타내는 어미가 붙어있다. 이것은 서로 다른 도메인네임이라 할지라도 해당 노드가 어느 대륙에 위치하는지 알려줄 수 있다. 이 때 문제가 생기는 것이 .com, .net과 같이 지역정보를 포함하지 않는 도메인을 갖는 노드들이다. PlanetLab 상에서 이러한 도메인을 갖는 노드들은 모두 미국에 위치하고 있다. 따라서 우리는 .com, .net을 North America라는 그룹으로 분류하였다. 각 지역별로 노드들을 그룹화 시키면 인접한 노드들끼리 묶을 수 있게 될 것이다. 이것은 사용자가 모든 노드들에게 전송을 수행하는 것보다, 같은 지역에 속해있는 노드들 중 하나의 노드에게 전송하고 그 노드가 같은 지역에 속해 있는 다른 노드들에게 전송을 수행하는 것이 좀 더 빠른 전송시간을 보일 것이다. <표1>은 2003년 12월에 PlanetLab을 구성하고 있는 200여개 노드들의 실제 지역 위치와 최상위 도메인 네임을 구분한 결과이다. PlanetLab 노드들의 실제 지역상의 위치와 최상위 도메인 이름에서 유추된 지역 위치가 정확히 일치하였다.

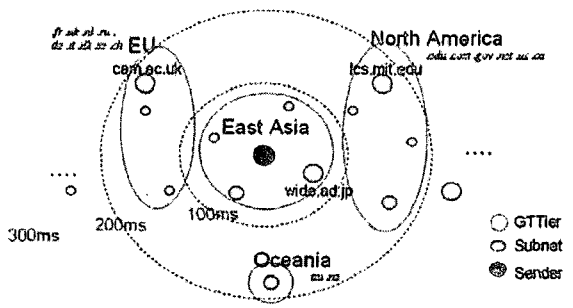
대륙	최상위 도메인
North America	.edu .com .gov .net .us .ca
Europe	.fr .uk .nl .ru .de .it .dk .se .ch .es
East Asia	.cn .kr .jp .tw .hk
Oceania	.au .nz
South America	.br
Middle Asia	.il

<표 1> PlanetLab 노드의 최상위 도메인(Top-level domain)

우리가 제안하는 MSCP가 실제로 어떻게 전송 트리를 만드는지에 대해 설명 하겠다. 첫 단계에서 목적지 노드들의 정보를 파일에서 읽어서, 같은 네트워크 네임을 갖는 노드를 하나의 그룹으로 묶고, 서브넷그룹(Subnet Group)이라 부른다. 예를 들어, csplanetlab1.kaist.ac.kr, csplanetlab2.kaist.ac.kr 은 kaist.ac.kr이라는 서브넷그룹이 된다. 서브넷그룹 중에서 임의의 노드를 그룹 마스터로 선택한다. 그 다음으로 송신자 노드에서부터 각 서브넷그룹 마스터까지의 RTT 값을 측정한다. 그룹 마스터가 응답을 하지 않아 RTT를 측정할 수 없을 때에는 마스터를 제외한 다른 노드를 목적지로 하여 RTT 를 측정한다. 이 노드도 응답하지 않으면 이 서브넷그룹의 모든 노드는 응답하지 않는다고 간주해 버린다. 여기서 RTT는 인터넷 상에서의 거리의 정보로 의미하고, 그룹의 마스터에게만 probing 패킷을 보내 RTT를 측정함으로써 오버헤드를 최소화하고자 하였다. 이렇게 측정된 값을 갖고 0~100ms, 100~200ms, 200~300ms, 300ms이상으로 각 서브넷그룹을 다시 분류한다.

다음 단계로 최상위도메인(Top-level Domain)으로 각 서브넷그룹을 묶는다. 그룹핑의 원칙은 <표1>에서 분류한 것과 같이 한 대륙에 위치하는 국가별로 그룹핑하고 동일한 네트워크 거리 갖는 대륙별 끼리 다시 그룹핑 한다. 예를 들어, 100~200ms안에 있는 .kr, .jp, .tw 을 가진 서브넷그룹과 0~100ms 사이에 존재하는 .kr, .jp, .tw를 가진 서브넷그룹은 서로 다른 그룹이 된다. <그림1>와 같이 묶인 그룹을 GTTier (Geographic TTier)라고 부르고, 이렇게 묶인 GTTier내에서 최소의 RTT값을 가진 노드를 마스터로 선정한다.

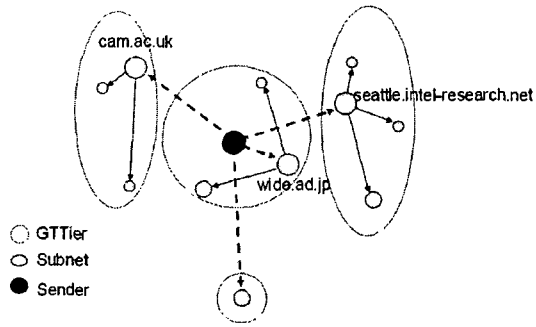
정리하면, 목적지 노드들의 리스트에서 첫째로 같은 네트워크 네임을 갖는 노드들을 서브넷그룹으로 만들고, 최상위 도메인 네임을 보고 같은 대륙에 속하는 서브넷그룹들을 GTTier 로 나눈다. 다시 GTTier는 네트워크 거리(송신자로부터의 RTT) 값을 기준으로 하여 4개의 범위 안에서의 GTTier 들로 나뉘게 된다.



<그림1> 그룹핑의 예

위와 같은 단계를 거쳐 만들어진 전송트리를 이용하여 어떤 방식으로 실제 전송을 하는지에 대해 설명하겠다. <그림2>은 실제 파일의 전송 순서를 보여주고 있다. 파일 전송 순서는 점선, 실선 순이다. 먼저 파일을 전송하고자 하는 노드는 GTTier의 마스터들에게 자신의 파일을 전송한다. <그림2>에서는 송신자 노드가 처음에 cam.ac.uk와 seattle.intel-research.net 그리고 wide.ad.jp에 동시에 파일을 전송함을 보여준다. 이후에 각각의 GTTier의 마스터는 각 서브넷그룹의 마스터에게 파일을 전송하

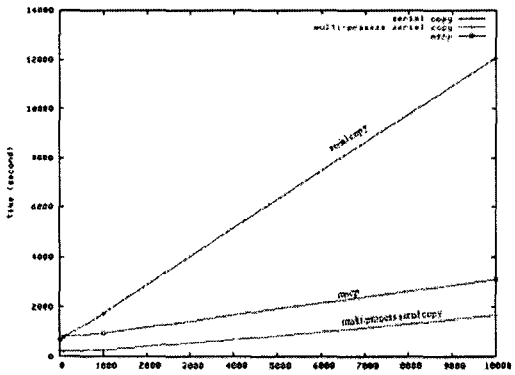
고, 서브넷그룹의 마스터는 다시 자신의 모든 노드들에게 파일을 전송하게 된다.



<그림2> 파일 전송 예

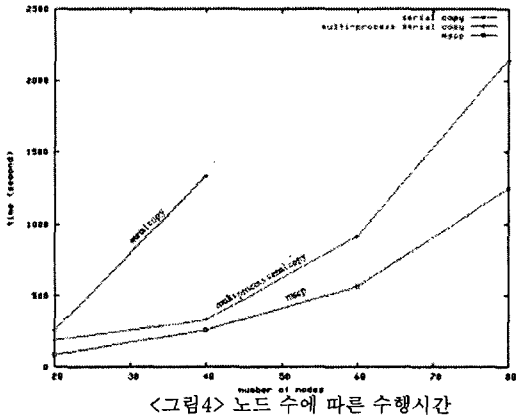
4. 성능 평가

우리가 제한하고 구현한 MSCP의 성능을 평가하기 위해, 별도로 두 개의 프로그램을 작성하였다. 첫 번째 것은 "sequential scp"로 scp를 이용하여 순차적으로 각 노드로 파일을 전송하는 것이고, 두 번째 것은 "multi-process scp"로 10개의 프로세스를 생성시켜 병렬적으로 파일을 전송하는 것이다. 목적지 노드 수, 파일 크기를 다르게 하여 sequential scp, multi-process scp, MSCP 수행하고, 각 수행 시간을 측정하였다.



<그림3> 파일 크기에 따른 수행시간

첫 번째 실험에서 PlanetLab 120개의 노드에 1KB, 10KB, 100KB, 1MB, 10MB의 크기를 갖는 파일을 전송하였을 때 걸리는 시간을 측정하였다. 측정 결과는 <그림3> 과 같다. 다른 두 프로그램에 비해 sequential scp의 경우 파일 크기가 커질수록 수행 시간의 증가폭이 커짐을 알 수 있다. 한 노드로 파일 전송이 끝날 때 까지 기다리고 난 다음에 다른 노드로 파일 전송이 시작되기 때문에 파일 크기가 커질수록 수행 시간이 길어지는 것이다. Multi-process scp가 MSCP보다 더 좋은 성능을 보였다. MSCP는 송신자가 GTTier 마스터들에게 파일을 전송한 후 다른 전송을 하지 않고 GTTier 마스터들이 전송을 마칠 때 까지 기다리게 된다. 그에 비해 multi-process scp는 10개의 프로세스들이 block 되는 일 없이 계속 전송을 하기 때문에 MSCP에 비해 전송이 빨리 끝났던 것이다. 만약, 송신자의 link bandwidth 보다 큰 파일을 전송하는 경우 MSCP가 multi-process scp보다 좋은 성능을 보였을 것이다. MSCP는 트래픽이 노드별로 고르게 분포되는데에 비해, mutli-process scp 는 모두 송신자의 링크에서 전송되기 때문에, 송신자의 link speed가 낮은 경우 MSCP보다 낮은 성능을 보일 것이다.



<그림4> 노드 수에 따른 수행시간

두 번째 실험에서는 10KB 크기의 파일을 20, 40, 60, 80, 100, 120개의 노드에 전송하는데 걸리는 시간을 측정하였다. 노드 수가 많아질수록 MSCP가 가장 적은 수행 시간을 보였다.

5. 결론 및 향후 연구 방향

우리는 PlanetLab에서 도메인 네임과 네트워크 거리라는 토폴로지 정보를 이용하여 전 세계에 분포되어 있는 수많은 노드로 지리적으로 가까운 노드끼리 그룹으로 만들고, 그룹간의 전송은 한 번 만으로 하여 전송 시간을 단축하였고, 송신자 링크의 병목 현상을 해결하는 방법을 제안하였다.

파일 크기와 노드 크기에 따른 수행시간을 비교하는 성능 실험을 하였으나, 이는 매우 제한적이기 때문에 여러 가지 상황에서 성능을 평가해 보는 추가적인 실험이 필요하다. 또, 실제 인터넷의 노드에서는 최상위 도메인이 지리적 위치와 상관없는 경우가 있다. 예를 들어, www.kaist.edu는 노드가 한국에 존재하지만, 북미에서 사용되는 .edu 최상위 도메인을 갖고 있다. 따라서 지역적 정보를 얻는 방법으로 CAIDA의 NetGeo[4] 데이터베이스를 이용하는 방법을 고려하고 있다.

참고 문헌

[1] PlanetLab, http://www.planet-lab.org
[2] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A Blueprint for Introducing Disruptive Technology into the Internet," ACM HotNets-I Workshop, October 2002
[3] OpenSSH, http://www.openssh.com/
[4] CAIDA NetGeo, http://www.caida.org/tools/utilities/netgeo/