실효 접속의 Hop Count에 기반한 인터넷 실효 지름의 측정

이지웅, 김재균, {porce, kjkim@kaist.ac.kr}, 한국과학기술원 전기 및 전자공학과

Measurement of the Effective Internet Diameter Based on Hop Counts of Effective Connections

Jiwoong Lee, Jae-kyoon Kim, Dept. of Electrical Engineering,
Korea Advanced Institute of Science and Technologies

Abstract

인터넷이 급성장함에 따라 사용자의 더 나은 서비스에 대한 요구 역시 급증하였다. 네트웍 연구가들은 제한된 네트웍 자원 하에서 오류제어, 흐름 제어, 그리고 폭주제어를 적절히 수행함으로써 전송 품질을 향상시키는 방법들에 대해 연구해 오고 있다. 그러나 인터넷 연구의 가장 큰 난점 중 하나가 인터넷에는 네트웍의 상태를 모니터링할 수 있는 근간이 되는 방법이 기본적으로 지원 되고 있지 않다는 점이고 따라서 네트웍의 구조와 상태, 플로우의 흐름 및 통계 등에 대한 기본 자료가 부족한 실정이다. 오류 제어나 흐름 제어, 그리고 폭주제어 기술을 연구할 때 가장 필요한 자료중의 하나가 네트웍 토폴로지이다. 그 중에서도 송신자와 수신자 사이의 거리 정보인 Network Diameter가 있다. 기존 연구에서는 연구자 임의로 이 값을 할당하거나 혹은 특정 실험에서만 유효한 Network Diameter 값을 제시해오고 있었다. 이 논문에서는 이처럼 비객관적이거나 특수 값으로만 사용되어 오던 Internet Diameter를 실효 접속에 근거하여 측정하는 알고리즘을 제안하고, KAIST 에서 전세계로 나가는 실측 접속 자료로부터 접속 Hop Count의 분포를 측정하며 이를 통해 타 인터넷 연구에 적절한 자료를 제시한다.

Index terms– **인터넷, Network Diameter, Hop Count**

I. 도입

인터넷과 같은 패킷 네트웍은 토폴로지의 변화에 의해 영향을 받지 않는다는 성질을 갖는 동시에 패킷의 전달 경로에 대해 확실히 알지 못하므로 통계적 제어를 사용할 수 밖에 없다는 단점이 있다. 특히 인터넷의 경우에는 네트웍의 상태를 관측할 수 있는 기반 프로토콜의 정의와 구현이 매우 미흡하여 품질 보장과 같은 고급 인터넷 기술 연구하는데 어려움이 따른다. 인터넷이 갖고 있는 여러 결함들을 보완하기 위해 오류제어나 흐름 제어, 그리고 폭주 제어와 같은 방법들이 연구되지만 지금까지는 네트웍 토폴로지의 규모에 대해서는 항상 연구자의 경험에 비춘 적절한 가정을 사용하는데 그칠 수 밖에 없었다. 또는 존재하는 호스트에 근거하여 인터넷 규모의 확률적 분포에 대해서만 알려져 있었을 뿐 [5] 인터넷 응용 프로그램을 통해 사용되는 실제 접속에 따른 실효 규모에 대해서는 알려진 바가 없었다. 이 연구에서는 실효 인터넷 접속에 대한 Hop Count를 측정하고 그 분포를 구함으로써 현재 전세계 네트웍 실효 토폴로지의 규모를 파악하고 연구자들이 보다 객관적이고 의미 있는 정보로서 활용할 수 있는 결과를 제시한다.

II. 측정 알고리즘

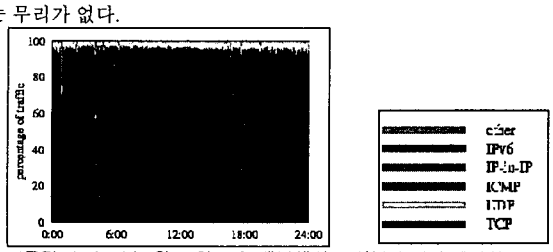
이 연구에서 제시하는 측정 알고리즘은 다음과 같다.

1) **AS (Autonomous System)의 BG (Border Gateway)나 혹은 공유 매체를 갖는 subnet 에서 TCP Connection Packet 을 검출해낸다.**
2) **Dumping 된 Packet 을 필터링하여 사용 포트와 Destination Address 를 파악한다.**
3) **그 Destination으로 1부터 증가하는 TTL 을 갖는 ICMP Query 패킷을 전송하여 각 경유 노드에서 응답한 에러 메시지를 가지고 최종 도착지까지의 Hop Count를 알아낸다.**
4) **위 1~3의 작업을 원하는 만큼의 실측 값을 얻을 때까지 계속한다.**

위의 알고리즘을 구현에 초점을 맞추어 구체적으로 설명하면 다음과 같다. 한 AS에서 Internet Application 을 이용하여 AS 외부의 호스트로 접속을 하게 된다면 그 Connection Packet 이 반드시 거치게 되는 곳은 그 AS 의 BG가 된다. 이 BG 가 속해있는 subnet에서 AS 외부로 나가는 Connection Packet 을 잡아내어 분석을 한다면 그 AS에서 사용하는 인터넷 접속에 대한 정보의 통계를 얻어낼 수가 있다. Network Diameter를 파악하기 위해서는 Connection Packet 의 목적지 주소를 필터링하고 traceroute [1]의 기본 알고리즘을 적용하여 Hop Count를 측정하면 실효 접속 Hop 수를 얻어낼 수 있다. 네트웍 지름 측정을 위해서는 트래픽의 양 정보보다는 접속 자체의 정보가 더욱 중요함을 유의한다. Connection Setup 이 이루어진 경우에만 합당한 정보의 추출이 가능하므로 TCP를 이용하는 Application 의 경우는 측정이 가능하지만 UDP를 이용하는 Application에 대해서는 실측이 거의 불가능하다. 그러나 다행히도 UDP Traffic 은 그림 1에서 알 수

[1] ———————————

있듯이 TCP Traffic에 비해서 극히 적은 양만이 사용되고 있으므로 TCP의 경우로만 제한을 하여도 전체 Network 의 내용을 파악하는 데는 무리가 없다.



[그림 1] 1998년 4월 13일 iMCI 백본에서 24시간 측정한 데이터 Sources from CAIDA

Subnet에서 TCP Dump를 할 수 있는 방법은 여러 가지가 존재한다. 여기서는 BPF(Berkeley Packet Filter) 기반의 sniff [2]를 사용하여 패킷을 잡아냈다. 그 다음으로 필요한 절차는 위의 Destination까지의 Hop Count를 알아내는 과정이다. 이를 위하여 traceroute [1] 의 소스 코드를 수정하여 특정 목적지까지의 Hop 수를 일괄적으로 측정하고 통계를 내는 tracehops 코드를 작성한다. 이 크드를 이용하여 수 시간 동안 측정으로 실제 인터넷 접속 패킷에 대한 정보들을 파악하여 통계를 내면 인터넷의 실효 지름 (Effective Internet Diameter)을 잴 수가 있다.

III. 구현 및 실험

앞에서 설명한대로, 이 실험은 sniff 와 tracehops 라는 software 를 개발하여 이루어졌다. 실험 대상은 KAIST 전기 및 전자공학과를 구성하고 있는 도메인 중 143.248.147 Domain(100Mbps Ethernet)과 , 143.248.142 Domain(10Mbps Ethernet)에서 각각 이루어졌다. 각 도메인에 대한 네트웍 정보 및 실험 내용은 아래의 표와 같다 [3].

[표 1] 실험 대상 Domain 의 특성

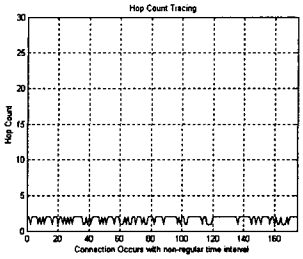
Network	Technology	Environment	Exp. Hour
143.248.147.*	100Mbps Ethernet	106 Hosts attached	12 Hours (Fri PM 1:00 ~ Sat AM 1:00)
143.248.142.*	10Mbps	41 Routers 99 Networks 244 Gateways 282 Links	21 Hours (Sat AM 2:00 ~ Sat PM 11:00)

실존하는 Port 의 수는 TCP 와 UDP를 합쳐 6만개가 넘는다 [4]. 이 모든 포트를 검색한다는 것은 실현에 있어서 엄청난 과부하가 걸린다는 문제가 있다. 다행스럽게도, 사람들이 주로 사용하는 Application 들은 제한되어 있으며, 따라서 사용되는 port 도 수 만개가 아닌 십 수개로 줄어들 수 있다는 점이 중요하다. 그리고 이러한 포트만을 검색하여도 실제 모든 포트를 검색한 것과 결과는 크게 다르지 않으면서 월등한 성능 향상을 얻을 수 있다는 것이다. 이 실험에서는 몇 개의 주요 포트만을 제한적으로 검색하였다. 검색에 해당되는 Application 들은 FTP, TELNET, HTTP, POPPASSWD, POP2, POP3, IMAP2, RLOGIN, RTSP, PNA, RealSystem G2 등이다.

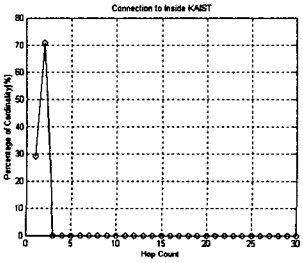
IV. 실험 결과

A. 143.248.147 Domain 에서의 실험

143.248.147 Domain 은 106 개의 Host 가 붙어있는 소규모 LAN Segment 이다 [표 1]. 여기서는 금요일 PM1:20부터 AM1:00까지 약 12시간동안의 TCP/IP Connection 을 검출하였다. 관측되는 접속에 대해 내부 접속에 대해 측정된 Hop Count를 연속적으로 [그림 2]에 나타내었다. 어떤 요인에 의해서 ICMP Packet 이 도달하지 못하거나 응답 받지 못하여 실험적으로 제한된 최대 Hops 31에 도달한 경우는 비정상 샘플로 간주하여 통계 처리시에는 제외하였다. 190개의 접속 중 비정상적인 15개의 접속 자료를 제외하고 구한 KAIST 내부 접속 통계 자료를 [표 2]에 나타내었다. 이렇게 구한 내부 접속의 평균 Hops 수는 1.7086 이었다. Hop 수가 1에서 30까지의 값 중 어떻게 분포되어 있는가를 [그림 3]에 나타내었다. 이 결과는 B-Class KAIST Domain에 적절하게 보인다.



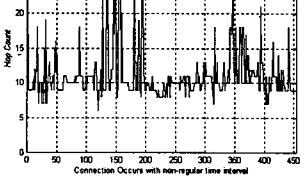
[그림 2] KAIST 내부 커넥션 Hop Count 의 시간에 따른 출력



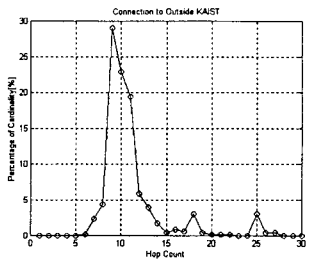
[그림 3] KAIST 내부 커넥션 Hop Count 의 분포도[%]

B. 143.248.142 Domain 에서의 실험

143.248.142 Domain 은 다소 거대한 네트웍의 한 링크이다 [표 1][3]. 실험은 주말 토요일의 약 21시간동안 관측한 것으로 이루어졌다. 21시간동안 측정한 내용을 보면 전체 Connection 개수는 3724개였고, 그 중에 KAIST 내부 Connection 은 765개로 약 20.5% 였고 나머지 2959개의 Connection 은 79.5%로 KAIST 외부로 나가는 Connection 이었다. 2959개나 되는 Connection에 대한 Hop Count를 모두 측정하는 것은 힘든 일이기 때문에(시간 소모가 극심하다) 그 중에 500개의 Connection에 대해 Hop Count를 측정하였다. 실험 A와 마찬가지로 정상적이지 않은 Hop 수 자료 44개의 Connection 을 제외하고 456개의 Connection에 대해 Hop Count를 측정하여 11.15의 Hop Count를 얻었다. 이 결과는 인터넷 상에서 애플리케이션을 활용할 때, 평균적으로 10.15개의 Router를 거쳐서 통신을 한다는 것을 말해주고 있다 [표 2].



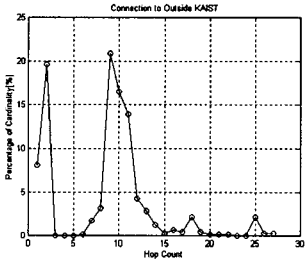
[그림 4] KAIST 외부 커넥션 Hop Count 의 시간에 따른 그림



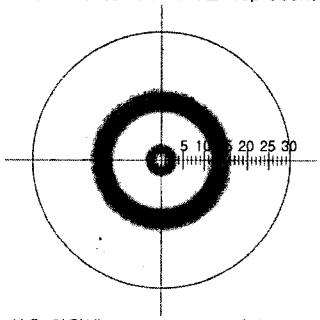
[그림 5] KAIST 외부 커넥션 Hop Count의 분포도 [%]

C. 실험 결과의 종합

두 Domain 에서 측정한 내부/외부 접속 Hop Count의 결과를 종합하여 정리하면 아래와 같다. [그림 6]은 전체적인 Hop Count 분포를 보여주고 있다. 그림에서 추정할 수 있는 바는, Hop 수가 2에서 Peak 을 이룰 때는 KAIST 내부 접속의 경우를 말하는 것이고, 9에서 Peak 을 이룰 때는 KAIST에서 서울에 있는 호스트로 접속 될 때의 Hop Count를 보이는 것이라고 해석 된다. 18에서 작은 Peak 을 이룰 때는 KAIST에서 미 서부 지역으로, 25 의 경우에는 미 동부 지역으로 접속하는 경우로 예상된다. [그림 7]은 [그림 6]을 보다 직관적으로 느낄 수 있도록 그린 그림이다. 이 원의 중심에는 인터넷에 물려있는 "자신"의 호스트가 존재하며, 원의 바깥으로 뻗어나갈수록 접속 대상 호스트가 멀리 떨어져 있는(경유 Hop Count의 관점에서) 것을 말한다. 색깔이 진한 부분은 접속 대상 호스트가 밀집 되어 있는 것을 말하며 산재해 있거나 접속이 뜸한 경우일수록 옅은 색으로 칠해져 있다. 결국 바로 [그림 7]이 현재 Internet 의 Network Diameter를 나타내는 그림이라고 볼 수 있다.



[그림 6] KAIST 내부 및 외부 커넥션 Hop Count의 분포도



[그림 7] 실효 인터넷 Hop Count 분포(실효 인터넷 지름)

[표 2] Connection 표본 정보 및 Hop Counting 결과

Network	Internal Con.	External Con.	Total Sum
Total Connection Number	200	3724	3924
Total Connection to Outside of KAIST	10	2959	3059
Randomly Selected Connection		500	690
Number of Excluded Samples	10	44	54
Number of Traced Samples	190	456	636
Average Hop Count	1.7086	11.1447	8.5277
Standard Deviation	0.4557	3.7212	5.2851
Median Value	2	10	9

V. 실험상의 문제점

Hop Count를 세는데 있어 한 가지 어려운 점은 ICMP Query 가 특정 네트웍의 Firewall 을 통과할 때의 경우이다. 사설 Network에서는 HTTP Protocol 들은 자유로이 억세스가 가능하지만, ICMP Query 와 같은 경우는 보안상의 이유로 막아놓는 경우가 많다. 예를 들어 www.sony.com 과 같은 주소는 HTTP로 매우 빠르고 손쉽게 억세스가 가능하지만, tracehops를 돌리면 17번째 Hop에서 Query 가 reply 되지 않고 있음을 발견할 수 있다. 이것은 17번째 노드에 Firewall이 존재함을 암시한다. 이런 경우 목적지 주소인 www.sony.com까지의 실제 Hop Count 는 측정할 수가 없다. 그러나 143.248.147 Domain에서의 실험 결과에 따르면 대개 AS 안에서는 2 Hop 정도의 Connection 을 갖으므로, www.sony.com 의 경우도 마찬가지로, 17번째 Firewall로부터 www.sony.com까지의 Hop Count는 2~3 정도라고 추측할 수가 있다. 결국 KAIST의 143.248.147 Domain에서 www.sony.com까지의 Hop 수는 20 정도로 보는 게 타당하게 여겨진다. tracehops에서는 이러한 advanced scheme 을 사용하지 않았다. Hop Count가 31 이상으로 폭주하는 경우는 Network Routing Protocol 의 자체 결함 때문에 간헐적으로 나타날 수도 있다. 이른 바 Ping-Pong 현상이 발생하면, Network 은 올바르게 패킷을 전달하지 못하게 된다. 접속 획득 이후 Hop Count 측정 이전에 해당 호스트가 Shutdown 이 되거나 Network 에서 분리/변화가 생기는 경우에도 마찬가지로 Hop Count 폭주가 발생할 수도 있다.

VI. 결론 및 추구 연구 과제

이 연구에서는 인터넷 실효 접속을 기반으로 접속 호스트까지의 Hop Count를 측정하고 통계 낸 결과 AS 내의 Connection 은 1.73 Hops, AS 밖으로의 Connection 은 11.5 Hops, 전체로는 8.5 Hops 가 실효 평균적인 것으로 집계되었다. 추후 연구 과제로는 Hop Count의 Peak 지점들에 대한 추정이 얼마나 정확한 가를 규명하고 Hop Count 와 지리적 정보를 결합하여 보다 발전된 Internet Network Diameter를 얻어내고 전세계 네트웍 지도를 이에 근거하여 그리며 시간에 따른 네트웍 토폴로지 변동을 관측하는 것들이 있다.

VII. 참고 문헌

[1] Eric Wassenaar, ftp://ftp.nikhef.nl/pub/network/traceroute.tar.Z
[2] Samuele Zannoli, et al., sniff-1.0, ftp://righi.df.unibo.it/pub/sniff/
[3] Tknid command, Unix
[4] IANA, http://www.isi.edu/in-notes/iana/assignments/port-numbers
[5] Aiguo Fei, et al., Measurements on Delay and Hop-Count of the Internet, Globecomm 1998

이지웅은 1998년 2월에 한국과학기술원 과학기술대학 전기 및 전자공학 학사 학위를 수여 받고 현재 동 대학원 전기 및 전자공학 석사 학위 과정을 이수 중에 있다. 현재 IP Multicast 환경에서 Multimedia를 전송할 때 신뢰성을 보장하는 문제에 대해 연구중이다.