KREONET 성능측정 BWCTL 기술보고서

BWCTL(The Bandwidth Test Controller)
Technical Report

슈퍼컴퓨팅 본부
첨단연구망센터 노민기

목 차

1장. BTC의 구조 이해

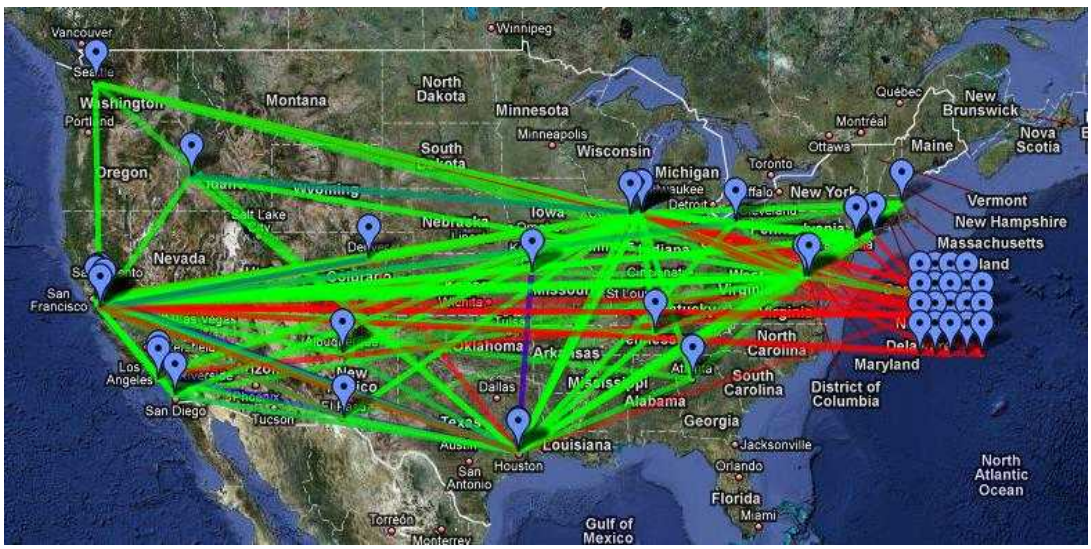
1. BWCTL(The Bandwidth Test Controller)의 개요

본 기술문서는 BWCTL(Bandwidth Test Controller)의 테스트 방법, 도구, 각 기능, 특징, 실행방법을 중심으로 기술하였으며, 또한 접근 가능한 서버와 해당 시스템의 정보 및 테스트의 각각 차트 흐름(Test Chart Flow)에 대한 단계적이고 기술적 설명을 통해 네트워크 전문가가 아니더라도 쉽게 BWCTL을 이해하고 직접 사용 할 수 있도록 하는 사용자 가이드북 제작을 목적으로 한다.

가. 개 요

BTC : Bandwidth Test Controller(BWCTL) tool은 미국의 E2E piPEs, Abilene Measurement Infrastructure 등의 프로젝트에 의해 점차적으로 업그레이드 되어왔다.

흔히 국내외 연구망을 사용하는 연구자 또는 사용자들은 자신이 접속한 현장 또는 사이트에서 데이터를 송수신하거나 협력대상으로 하는 다른쪽 사이트로의 네트워크 대역폭, 성능을 포함한 실시간적인 이용성(Available)을 확인하기를 원한다. 이를 통해 기본적인 트래픽 정보이외에 직접 연구에 사용할 수 있는 전송성능을 파악하여, 현재 자신의 네트워크의 전송성능이 정상인지, 또는 향상시킬 수 있는지, 더 나아가 향후 자신의 연구파트너를 선택할 때 협력의 조건으로 사용할 수 있다.



<그림 4> Internet2 BWCTL 운용 현황

현재 Iperf는 대표적으로 전송시 데이터의 전송속도 즉, 처리량 테스트에 공통적으로 사용되는 도구들 중 하나이다. 하지만 Iperf의 경우 사용자들이 해당 시간대에 자신의 네트워크 성능과 대역폭 수치를 사용하기에는 우수한 도구인 것은 사실이지만, 더욱 평균적인 수치를 얻기 위해서는 테스트를 수행하는 해당 시간이외에 작업시간, 데이터의 송수신이 빈번한 시간 등을 고려할 자료가 필요하다. BWCTL의 중요한 사용 목적은 앞에서 언급했듯이 해당 시간대에서 확장된 정기적인 일정관리와 on-demand 테스트를 동시에 수행하기 위한 데몬 일정관리와 해당 시간대의 네트워크를 측정하기 위함이다.

BWCTL의 기본적인 서비스정책은 다수의 노드와 회선이 연결된 각 단의 네트워크상에서 문제가 되는 부분을 찾아내기 위해 중앙부분을 중심으로 각 지점과 각 엔드포인트(endpoints)에서 사용 가능한 대역폭을 확인하는 것이다. 즉, 정확한 종단간 처리량 테스트를 해야 할 경우 BWCTL은 각 대역폭과 네트워크 테스트 방법 중 좋은 방법 중 하나이다.

BWCTL 이전에 이 유형의 테스트 방법은 직접경로를 통해 호스트를 특정하고 두 개의 엔드포인트에서 Iperf나 유사한 도구를 직접 실행하는 것이었다. 또는 NOC의 역할을 수행하는 기관에 요청해서 해당지역에 근접한 지역에서 Iperf 서버에 대한 실행을 요청하는 방법이었다.

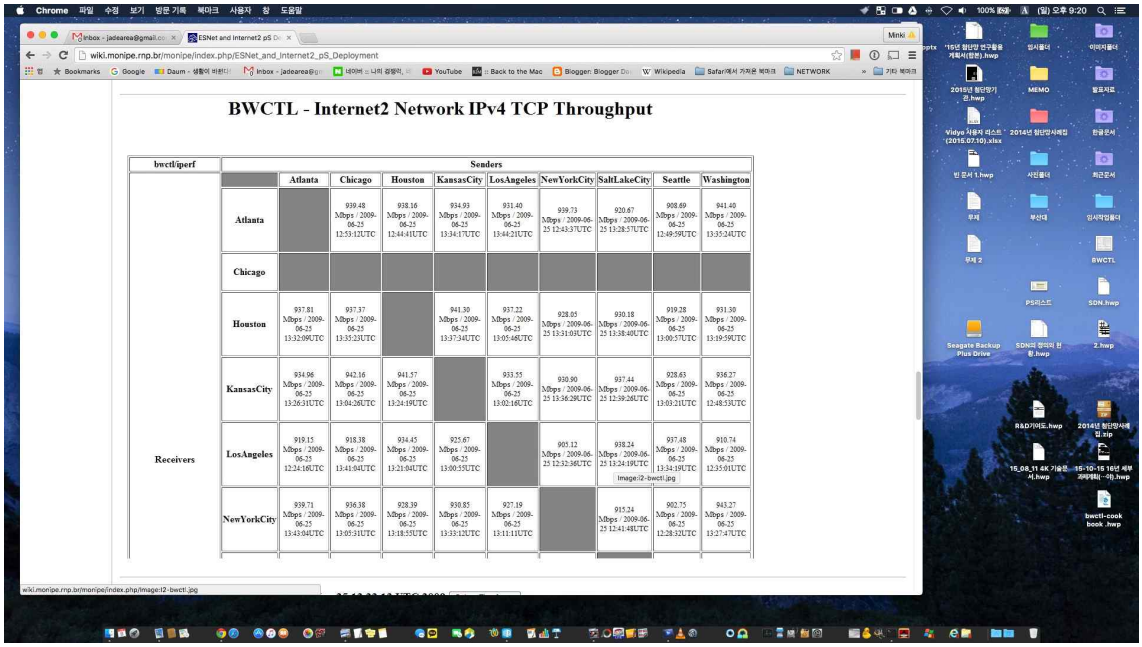
BWCTL를 통해 테스트 일정관리를 하는 것은 각각의 다른 요인 즉, 측정결과에 영향을 미치는 테스트에 대해 고려하지 않아도 된다는 의미이다. 또한 NOC로부터의 지원 없이 첫 번째로 사용가능한 테스팅슬롯을 통해 테스트 하는 것과 자동화된 테스터를 요구하는 것이 가능하다. 왜냐하면 NOC에서는 빈번하거나 또는 장시간의 테스트가 되는 것을 조정할 수 있기 때문에 즉, 대부분의 사용자들에게 그들 방식의 테스트를 실행하는 권한이 직접주어지기 때문에 좀 더 자유로운 테스트를 할 수 있다.

다음은 미국에서 해당 BWCTL을 사용한 과학 커뮤니티의 좋은 예이다:
(http://e2epi.internet2.edu/case-studies/VLBI/cs-index.html)

요약:

MIT Haystack 과학자들은 Bandwidth Test Controller 서버를 네트워크상에서의 성능 또는 사용가능한 대역폭 문제가 일어날 때를 대비하여 그들이 데이터흐름을 받는 곳 즉, 실험장비(telescope locations)에 BWCTL의 테스트구

간을 설정하였고, 그 원격위치에서 보조를 해줄 수 있는 NOC를 기다리는 대신 해당 위치에서 언제든지 테스트를 즉시 자신들이 해볼 수 있는 시스템과 구간을 설정해 놓은 예이다.



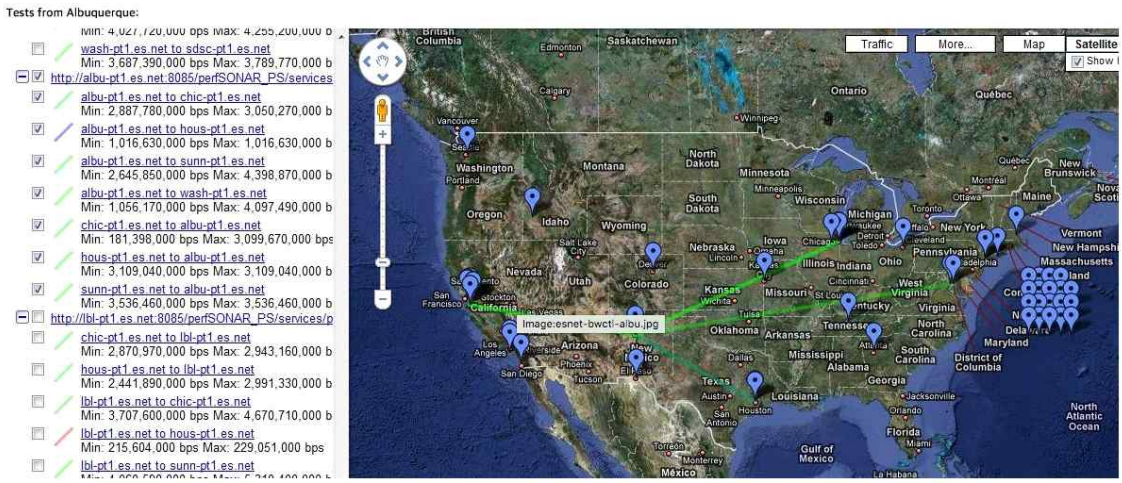
나. BWCTL의 적용

원격에서 접속이 가능한 API 인터페이스는 다른 증명과 정책기능을 통합 가능한 것이며 해당 권한은 네트워크 사용자들에게 허용하는 것이 일반적이다. 특히 Iperf는 이러한 테스트를 위한 도구로써 잘 알려져 있고 일반적으로 사용되기에 좋은 “측정기술” 이다. 하지만 Iperf는 다음과 같은 몇 가지 API의 통합화 문제를 가진다.

- Iperf 서버 초기화(포트 숫자 할당)
- Iperf 오류 상태
- 섹션의 말단

이 문제는 보통 Iperf를 실행할 때 당신의 테스트 결과에 빈번히 나타나게 된다. 수동으로 Iperf를 실행할 때 또는 테스트를 취소할 때 차이가 나며, 테스트결과는 신뢰성을 잃게 된다. 대부분 Iperf 사용자들은 장시간 테스트의 섹션을 보게 된다. Bandwidth Test Controller는 전체적인 테스트 일정을 관리하고 다음 테스트를 실행하기 위해 테스트종료와 다음 테스트의 정상작동

확인이 필요하기 때문에 해당 특정섹션을 제거해야 한다.
(참고: Iperf는 Bandwidth Test Controller와 근접하게 통합하지 않은 이후로, 처음 테스트와 다음 테스트를 연결하는 중간 세션에 신뢰할 수 없다.)



이로 인해 BWCTL은 테스트 일정관리에 대해 특별한 차이를 가진다.

- Iperf 송신자는 일시적으로 중단됨
- 섹션의 마지막을 찾기 어렵고, “scheduled” 시간 창에 문제가 있음

다. BWCTL 기능(function and feature)

1) 클라이언트(Clent)

BTC 클라이언트 응용프로그램과 BWCTL은 테스트에서 결정된 값을 요청한다. 커뮤니케이션은 오픈되거나 참여할 수 있으며 해당 요청은 타임슬롯(timeslot)에 대한 요구와 테스트의 전체적인 매개변수도 포함하게 된다.

BTC는 각면에 잠재적으로 고유하게 인증된 값을 사용하여 두 개의 관련 서버 사이에 새로운 테스트 그룹의 요청을 할 수 있다. 현재 클라이언트로부터 현재 bandwidth 테스트에 대해 BTC가 지정된 옵션의 서버를 제공하게 된다. 만일 로컬호스트(local host)에 이용 가능한 서버가 없다면, 클라이언트는 테스트 결정값이 다를 수 있다. BTC는 Iperf로부터 같은 지시의 옵션을 대부분 제공한다. 즉, 몇 가지 Iperf 옵션은 제한되어 있지만 command line options은 가능한 Iperf의 옵션과 매우 유사하다.

2) 데몬

테스트의 정책과 클라이언트간의 연결이 구현되는 곳이다: bwctld(데몬)은 테스트 과정에 대해 전형적으로 accept/fork 스타일의 데몬이다. 즉, 최근 처리과정과 자료요청에 대해 응답하고 테스트 프로세스를 수용하게 된다.

- “time slot”을 포함하는 “Iperf”에 대한 요구를 받아들인다.
- 잠정적인 예약과 거절 메시지에 대한 응답(client에 의한 예약은 “start session” 메시지로 확인된다.

3) 스케쥴링

한 타임슬롯은 시간에 의존하며 해당 테스트 시간에 리소스 할당을 필요로 한다. 그러므로 다음에 설명하는 자원할당모델처럼 BTC 스케쥴링은 현재 테스트를 중심으로 단일 테스트만 허용한다.

4) 소스 허용

테스트 데몬인 bwctld는 테스트와 클라이언트 연결 정책실행이 필요한 커넥션을 유지한다. 각 연결은 “classified”로 분류되며 각 분류는 상호간에 영향력이 있으며 영향력 제한의 설정이 가능하다.

- connection policy (allow_open_mode)
- bandwidth (allow_tcp, allow_udp, bandwidth)
- scheduling (duration, event_horizon)

라. BWCTL 테스트 구조

1) BWCTL 테스트 구조의 개요

bwctld 데몬은 classic accept/daemon으로 개발되었으며 master daemon은 새로운 네트워크연결을 대기하며, 연결된지 얼마안된 bwctld processes에 대

한 데이터를 제어한다. bwctld는 일단 외부에서 연결요청이 들어오면 bwctld가 하위 프로세스의 요청을 처리하기 위해서 단독으로 실행된다.

단독으로 프로세스 요청을 하는 외부커넥션은 로컬 호스트 또는 엔드 포인트 중 하나이며 로컬호스트 자체에 bwctld 데몬 실행이 없는 경우에는 서버에서 추가 프로세스를 생성하며, bwctld를 직접 테스트하기 위한 로컬 데몬을 추가로 실행한다.

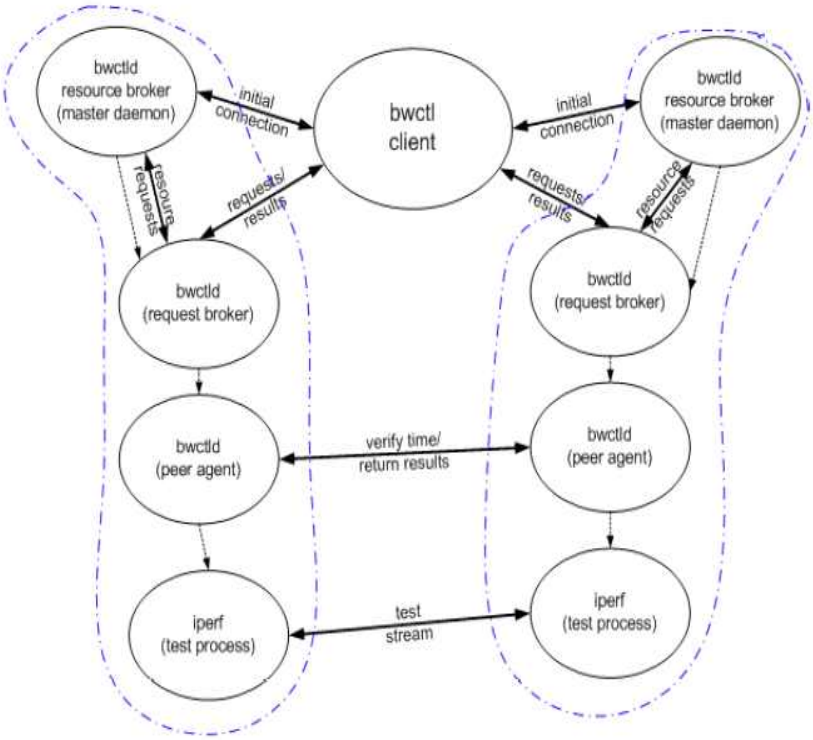


Figure 1: Control Flow for BWCTL Architecture

2) 인증 및 권한부여

bwctld 데몬과 브로커가 요구하는 the child process는 모든 encryption과 클라이언트에 대한 연결과 커뮤니케이션 문제를 다루고 내부 데이터의 활용 제한에 대해서도 함께 실행된다. 정적 데이터의 제한은 해당 노드에서 현재 일어나는 것에만 의존적이지 않다.

예를 들어, 만일 주어진 클라이언트가 상위데몬에 대해 보고 없이 UDP 테스트를 사용하는 것을 허용한다면 요청 리소스 브로커(resource bloker)는 다음과 같은 프로세스를 따라 해당 테스트의 허용을 결정할 수 있다. 요청 브

로커 프로세스의 요청이 유효한지 판단되면, 클라이언트에 의해 요청된 자원 및 시간동안 상위데몬(리소스 브로커)프로세스에 요청한다. 만일 리소스 브로커가 이용 가능한 자원이 있는 경우, 테스트 요청을 실행하고 해당 테스트 결과를 전송하게 된다.

3) 스케쥴링

클라이언트는 항상 테스트를 위한 가능자원에서 가장 기본적인 테스트 시간(time period)을 요구한다. 차례대로 테스트의 각 엔드포인트의 요청브로커 프로세스에게 테스트 시간을 요청한다. 이 시간은 2개의 타임스탬프를 상호 비교하고 정의함으로서 요청된다. 초기 시간과 제일 알맞은 첫째 bwctld 오픈시간에 대해 임시 예약을 요청하거나 'request denied' 메시지를 보낸다. 클라이언트는 다른 호스트에 브로커를 요구하며 초기시간으로서 임시시간예약 사용을 할 수 있다. 결국 이러한 브로커와 엔트포인트간의 시간예약에 대한 요청은 시간의 일치에 도달되거나 테스트가 가능한 최근시간(recent_time)에 도달된다. 만일 요청하는 브로커가 최근 요청시간 전에 이행할 수 없다면 해당 브로커에 'server too busy' 라는 메시지가 보내게 된다.

만일 클라이언트가 두 개의 서버로부터 같은 유효한 타임슬롯을 얻는다면, 'start session' 메세지에 대해 예약을 확인하게 되며, 'start session' 메시지를 구성한 타임아웃시간전과 예약시간을 받아서 bwctl 서버에 테스트 시간과 기간이 등록된다.

4) 테스트 실행

클라이언트가 한번 'start session'을 보내면 요청브로커는 테스팅의 다른 결정값을 time offset으로 바꾸는 원인이 되는 peer agent를 분할하며, 테스트의 거래값으로 사용되는 커뮤니케이션소켓을 설치한다. 만일 테스트 결정값 시스템이 적당한 시간의 근접아이디어를 가진다면 서로 통신이 가능하고, peer agent는 테스트프로세스를 분할한다. 이 테스트프로세스는 시작시간을 예정할 때까지 대기하면 command-line parameters에 대해 Iperf를 실행한다.

5) 요구사항

- Iperf version 2.0
- NTP (ntpd) 로컬시스템 시간 동기화
- NTP system calls

6) 운영문제

- 시간, 방화벽, 시간문제 등을 포함한다.
  - NTP는 시간 오류에 대해 보강할 수 있는 데이터를 제공한다. 만일 정확히 구성되어 있지 않다면 BTC는 완료하기 전에 마무리 테스트로 강제종료 될 것이며 테스트 결과를 이용하는 신뢰하지 못한 결과로 만들어지게 된다.
    (참고 : http://twiki.ntp.org/bin/view/Support/SelectingOffsiteNTPServers)

- 방화벽 문제 :
  - TCP 포트는 컨트롤 통신을 위해 허용되거나 항상 열려 있어야 한다. 두 클라이언트로부터 bwctld 서버 사이의 연결을 위한 피어를 대상으로 하며 상세한 포트번호와 프로토콜 정보는 사용자 설치를 참조하면 된다.

마. 정책 문제

정책문제는 두 가지 사안으로 크게 나누어진다. 첫째, bwctld 서버는 기본적으로 우수한 시스템과 네트워크 상에 위치하는 것이 중요하다. 이는 로컬 호스트와 네트워크소스를 사용하지 않고 bwctld 서버에서 측정된 테스트 데이터의 결과를 보호한다. 두 번째, 컨트롤은 적절히 이용 가능한 서버의 이용성에 따라 타 데몬 또는 서비스와 구분하는 것이 필요하다.

바. 보안 주의점

해당 테스트가 네트워크 보안감지 또는 모니터링시스템으로부터 3rd-party DoS source 또는 DoS target으로 인지되지 않는 것에 대해 통보와 인지가 필요하다. 테스트에서 고려해야 할 부분은 시스템 소스의 사용, 테스트자원의 저장과 용량, 그리고 네트워크 대역폭에 대한 고려이다.

가) 디도스 (DDoS)

기본적으로 네트워크 대역폭과 성능테스트가 기본적인 네트워크 사용을 방해하는 것은 충분히 고려되어야 하는 사안이다. 실제 DDos와 대역폭 테스트와의 구별은 외부에서 판단시 불분명하고, 보안시스템의 효율성에 영향을 미칠 수 있지만 현재까지는 해당 서비스에 대한 인지와 정보교환으로 해결되어야 하는 문제이다.

나) 트래픽 소스(traffic source)

구축되고 테스트가 수행되는 bwctld 서버의 다수가 직접적으로 전체 네트워크의 트래픽(tracffic)을 이용한다면 이는 기본적으로 bwctld 서버가 다른 사이트를 향해 많은 패킷을 보내는데 사용될 수 있다. 이는 매우 주의해야 할 사안이며, 특히 섹션이 인증되지 않은 모드에서의 무작위 호스트로 향할 수 없도록 확인하여야만 한다(BTC-control client로만 트패픽이 전달되어야 합니다).

다) 공격목표(Target)

만약 고의로 전체 네트워크에 bwctl에 관련된 시스템에 악영향을 주기 위해 사이트에 공격을 시도한다면 공격 패킷이 bwctld 서버를 향하는 것은 유효한 테스트 트래픽의 결과에 영향을 줄수 있다. 따라서 서버의 보안성 검토는 매우 중요하다.

마) 자원의 소비

bwctl의 해당 시스템과 네트워크의 소스 사용의 두 가지 주요한 컨트롤은 타임슬롯과 bandwidth이다. bwctld는 적절한 사용자들에게 해당 정보를 제공하기 위한 브로커와 타임스케줄 정책컨트롤을 가지고 있다. 이러한 자원의 적절한 분배는 각기 새로운 테스트 요구에 대해 IP/netmask나 AES key의 사용으로 분류함으로서 가능하다. 각 분류는 소스제한의 설정에 대해 연관이 있다.

바) 권장 정책

Abilene에서는 개방할 수 없을 때 까지 시도한다. 새로운 사용자가 UDP를 사용하는 것을 완벽히 제한하는 것을 추천한다. 적어도 bandwidth는 제한해야한다. 모든 사용자에게 AES key 입증을 요구한다.

2장. 설치가이드:
BTC서버(Bandwidth Test Control Server) 구축

1. 시스템 구성 및 요소

서버구축에 필요한 모든 것은 네트워크를 통해 다운로드할 수 있는 tar 파일에 포함되어 있다. 이 파일은 internet2 웹사이트에 저장되어 있다(아래 링크참조). 해당 소스가 지원하는 시스템의 OS는 다음과 같다.
(http://e2epi.internet2.edu/bwctl/download.html)

- FreeBSD 4.x, 5.x
- Linux 2.4, 2.6
- (Most recent versions of UNIX should work)

1) 요구사항과 권장사항

본 장에서는 BTC 서버를 설치할 수 있는 하드웨어와 소프트웨어의 요구사항 그리고 권장설정에 대해 설명한다.

가. 하드웨어 요구사항

- 권장하는 CPU와 메모리, bus 속도, NIS에 대해 엄격한 사양이 요구되는 것은 아니다.
- 하드웨어는 BTC서버가 수행하는 환경에 맞게 더욱 적합한 요구사항을 자체적으로 디자인해야 하며, 많은 멀티태스킹 테스트의 경우 고사양의 하드웨어를 요구한다.

Abilene에서는 다음에 대해 Intel SCB2 motherboard를 권장한다.

- 2 x 1.266 GHz PIII, 512 KB L2 cache, 133 MHz FSB

- 2 x 512 MB ECC registered RAM (one/slot to enable interleaving)
- 2 x Seagate 18 GB SCSI (ST318406LC)
- SysConnect Gigabit Ethernet SK-9843 SX

위치한 각 BTC 시스템 사이의 990Mbps TCP전송을 의 지원하기 위해 다음과 같이 구성된 시스템을 사용한다. 특정한 시스템 요구사항이 필요할 경우 Iperf test를 권장한다. http://abilene.ucaid.edu/observatory/

나. 소프트웨어 요구사항

- Iperf versions 1.7.0 or 2.0
- NTP (ntpd) synchronized clock on the local system
- Firewalls: leave lots of ports for communication and testing
- End hosts: TCP 및 성능 튜닝(tune)이 된 시스템을 사용하여야 한다.
  http://www.psc.edu/networking/perf_tune.htmlhttp://www-didc.lbl.gov/TCP-tuning/buffers.html

다. 네트워크 요구사항

만일 방화벽 안에 있는 시스템으로 실행하고 있다면, 테스트 통신과 성능 및 대역폭 테스트에 대해 적절한 포트를 개방하는 것이 필요하다.

- TCP에 대한 테스트로 제한한다.
- 모든 테스트에 대해 인증 값 설정을 요구한다.(AES Keys)
- AES Keys을 보호한다. 안전한 비밀번호를 사용한다.

라. 권장설정

BTC에 구축과 설정에 대해 다음과 같이 제안한다.

1) 일반보안문제

일반적으로 보안장치를 강화시키는 것에 대해 보안 패치를 최신 상태로

유지하며 시스템 내에 로컬 방화벽(IP chain등)을 실행한다. 그러나 만일 측정결과에 영향을 미치는 것이 보인다면 BTC는 peer 연결을 위해 “ephemeral” TCP port를 즉시 사용할 것을 권장한다. 로그인을제한과 로그인개방에 대해 고려해보고 최적의 상태로 유지를 권장한다.

2) BTC 보안 문제

BTC 사용에서 직접 위험에 노출될 수 있는 시스템과 네트워크 사용 소스이다. 이를 자체적으로 해결하기 위해,

- 테스트에 사용 될 수 있는 bandwidth를 제한한다.
- TCP/UDP로 실행될 수 있는 테스트 타입을 제안한다.

3) BTC 설치하기

BTC를 설치하기 위해 압축을 푸는 과정에서 해당 인증된 웹사이트에서 (http://e2epi.internet2.edu/bwctl/download.html) 제공되는 최신의 압축파일을 다운받고 tar balld을 잡아 tar file 압축을 풀고 제공된 구성 스크립을 사용하여 다음을 실행 설치한다:

```
% gzip -cd bwctl-$VERS.tar.gz ¦ tar xf -
% cd bwctl-$VERS
% ./configure --prefix=/ami
        # --prefix is only needed if you don't like the default
        # (/usr/local on most systems)
% make
% make install
```

4) 소스분할

해당 소스를 보호하기 위해 서비스를 제공하는 대상과 역할, 서비스를 얼마나 직접적으로 제공 할 수 있는지 또는 사용자들이 사용하길 원하는지에 대해 정확한 결정과 디자인을 설치와 서비스 전에 해야 한다.

- 테스트 호스트가 소비하는 또는 허용되는 자원의 전체 양 결정
- 사용자들 사이에 그 자원을 할당하는 방법 결정
- 어느 정도의 타임슬롯 배정시간과 각 사용자 그룹에 대한 서비스
- 얼마나 많은 대역폭이 사용될 것인가? 총 사용량? 그룹 당 사용량?
- 네트워크뿐만 아니라 시스템 부하가 고려되어야 함. 시스템자원이 테스트에 너무 많이 로드 된 경우 데이터의 정확성이 저하됨

BTC는 권한을 가진 자에 한해서 부분적으로 사용할 수 있도록 권한을 부여 할 수 있다.

- 사용자는 계층그룹에 분류된다.
- 하나의 상위-하위 계층은 허용되며 이용 가능한 소스의 전체적인 양으로 정의한다.
- 계층이 정의된다면, 상위계층과 다른 계층의 제한은 상속된다.
- 소비적인 소스가 요구되었을 때 계층의 제한과 모든 상위계층은 안정적이어야 한다(memory/bandwidth/timeslot).

5) 네트워크 연결

네트워크의 연결과 시스템의 네트워크 설정을 간략하게 설명한다. 모든 시스템과 BTC 종류의 DNS 매칭은 필요 없다. 연결을 분류하기 위해서는 다음 두 가지 방법이 사용된다.

가) IP/netmask

- 클라이언트의 IP 주소는 IP 넷마스크 특정 서브넷 리스트에 대해 맞춰지고 클라이언트주소에 기초한 연결 제한계층으로 등록한다.
- 가장 구체적으로 맞춰진 mask는 매칭 알고리즘으로 설정한다.

이는 routing 관점으로부터 “real” 서브넷이 되길 필요로 하진 않는다. 넷마스크는 단지 주소의 범위를 표현하는 한 방법이다.

나) 사용자이름과 AES key

- 클라이언트는 사용자 이름을 나열하고 서버는 이미 연관된 AES key를 알아야 한다.
- AES key는 symmetric 섹션 키에 사용된다.(클라이언트와 서버는 shared secret 키를 사용한다)

이는 기초적인 static symmetric 섹션 키 설정이다. BTC는 낮은 단계의 도구이고 인증 프리미티브의 사용은 매우 단순하다. 현재 인증 scheme는 쉽게 실행되고 보다 완벽한 solution에 대해 통합하기 쉽기 선택된다. 예를 들면 PKI는 Diffie-Helman style key 허용에 대해 사용되는데, 이것은 BTC 프로토콜에서 사용된 AES session을 허용하는 데에 사용된다.

2. Bandwidth Test Controller 서버 구성

bwctld를 구성하는 기본은 시스템 상에서 bwctld.conf를 설정하는 것이다. 그리고 선택적으로 bwctld.limits 파일과 bwctl.keys file을 추가로 설정한다. 이 파일은 bwctld에 -c 옵션으로 분류되는 같은 다이렉토리에 설치를 필요로 한다. 권장되는 디렉토리는 /ami/etc이다. 배포되는 bwctld-$VERS/conf sub 디렉토리에 있는 이 파일의 예시이다.

가. bwctld.conf 구성

bwctld.conf file은 bwctld daemon에 대한 구성파일이다. 이것은 listening port, the path for Iperf, error logging와 같은 서버의 기본적인 운영을 위해 사용된다. bwctld.conf file의 예시는 모든 이용 가능한 옵션에 대한 설명과 bwctld.conf 매뉴얼 페이지 주석과 옵션을 참조한다.
(http://e2epi.internet2.edu/bwctl/bwctld.conf.man.html)

대부분 설치는 다음 옵션에 따라 수정한다.

vardir	Directory where bwctld.pid file is stored
user	Specifies the uid the bwctld process will run as
group	Specifies the gid the bwctld process will run as

나. bwctld.limits 설정

bwctld.limits은 데몬을 위해 정책과 제한을 구성하는데 사용된다. 이것은 시스템관리자가 다양한 방법으로 소스를 할당하는 것을 허용한다. 정책구성은 두 가지 방법이 있다.

1) 인증

누가 요청을 하는가? 특정한 개인 사용자가 되거나 또는 특정한 네트워크로부터 요청이 오는 일반적 인증이다.

2) 허가

일단 연결이 확인된 후, bwctld의 접근을 허용할 것인가?

인증은 각 새로운 연결에 대해 제한된 유저로 등록함으로써 실행된다. 허가는 이것에 연관된 각 사용자의 제한을 설정함으로써 사용 할 수 있다. 각기 등록된 제한은 권한과 설정에 따른다. 또한 연결은 root 권한뿐만 아니라 사용자 등록 권한 제한에 인증되어야 한다.

bwctld.limits 파일 안에서 line을 등록하는 것은 주어진 연결을 등록하기 위해 사용된다. limit lines는 인증과 허가를 정의하는데 사용되며 각 권한에 따라 설정한다. 이 파일은 시스템에서 순차적으로 로딩되며 limit line의 사용을 정의하기 전에 서비스에 적용되지 않는다.

```
# total available
limit root with  \
bandwidth=900m,  \
duration=0,  \
allow_tcp=on,  \
allow_udp=on,  \
allow_open_mode=off
# Hostile
limit hostile with parent=root,  \
bandwidth=1  \
allow_tcp=off,  \
allow_udp=off
# NOC
```

```
limit noc with parent=root,  \
allow_open_mode=on
```

이 예시는 단지 가능한 서비스 세 가지만을 보인다. 주어진 limitclass을 제한 하기 위한 구성옵션의 전체 설정은 (5) 매뉴얼 페이지 bwctld.limits에 설명되어 있다.
(http://e2epi.internet2.edu/bwclt/bwcltd.limits.man.html)

보여주는 예시는 IP/netmask 등록과 특정호스트로부터 연결을 분류하기 위해 사용하는지 보여준다.

```
# loopback
assign net ::/127 noc
assign net 127.0.0.1/32 noc
# abilene nmslan (observatory systems)
assign net 2001:468:0::/40 noc
assign net 198.32.10.0/23 noc
assign net 10.0.0.0/16 hostile
```

이 예시는 어떻게 loopback 인터페이스에 대한 서버에 어느 연결이 noc limitclass에 대해 연관된 제한을 등록할 수 있는지 보여준다.
nmslan 시스템은 같은 limitclass 할당된다. 이것은 bwctl을 로컬호스트에 사용해 bandwidth 테스트를 실행하는 것은 불가능하다. 왜냐하면 동일한 기간 동안 시험의 수신기 측 및 시험 발신자 측 모두에 대한 개방 스케줄 슬롯이 필요하고 bwctld는 한번에 하나의 오픈 일정 슬롯들의 수를 제한하기 때문이다. 만일 로컬 bwctld 실행할 때 만일 로컬인증을 위해 AES bypass AES 인증을 원한다면 loopback/localhost line 설정을 적용한다.
주어진 서브넷으로부터 모든 연결을 확인할 수 있는 방법에 대해 설명하는 예시인데 인증하지 않는 한 차단된다(10.0.0.0/16 line). hostile limitclass는 open_mode 설정을 no로 허용한다. 따라서, open mode communication은 이 주소범위로부터 받아질 수 없다. 그러나 이 서브넷의 사용자가 사용하기 위해 usename/AES key 인증방법을 사용하게 된다.

다음의 예시는 특정 사용자로부터의 연결을 분류하기 위한 사용자 등록

방법이다.

```
# network admins
assign user joe root
assign user jim root
assign user bob root
# measurement geeks
assign user boote noc
```

bwctld 서버는 주어진 사용자를 인증하기 위해 필요하다. 128-bit shared key를 사용하게 되며 키 연결을 위한 사용자이름은 아래 설명된 bwctld.keys 를 사용함으로써 bwctld 에 전달된다. 사용자는 bwctld.limits file에 사용하기 위해 bwctld.keys file에 접속해야 되며 bwcltd 과정은 만일 사용자가 bwcltd.limits에 기록되거나 bwctld.keys file에 연관된 키를 가지고 있지 않을 때 보안상 차단된다.

다. bwctld.keys 구성

bwctld.keys file은 identity/AES keys pair을 저장하기 위해 사용되며 bwctld 에 사용자를 인증하기 위해 필요하다. 이 파일의 포맷은 매뉴얼 페이지 aespasswd에 설명되어 있다. bwctld.keys file의 위치는 bwctld의 -c option에 의해 조절된다.

bwctld는 대칭적인 AES keys를 인증하기 위해 사용한다. 그러므로 bwctld 클라이언트는 AES에 의한 인증을 위해 정확하게 동일한 AES key에 연결이 필요하다. 대부분 사용자는 AES key를 작동시키는 passphrase에 역할에 대해 인지하여야 한다. 또한 시스템관리자와 종단 사용자가 혼동되지 않도록 설정할 수 있다.

만일 bwctld client가 identity와 AES key를 사용하여 인증을 요구한다면 bwctld는 이 연결에 대한 차단을 위해 bwctld.limits file에 있는 directives found를 사용할 것이다.

사용자이름과 AES key Rule:

- 사용자이름은 16자로 제한된다.
- AES key는 128 bit session key이다.
- AES key는 keys file안에 암호화되지 않는다.
- AES key를 작동시키기 위해 passphrase를 사용할 수 있다.
- keys file에 실행된 passphrase 추가를 위해 AES password를 사용한다.
- Client: 응용프로그램 프롬프트는 passphrase에 대한 사용자이다.

기본적인 UNIX 보호방법은 keys file을 읽을 수 있게 승인하는 그룹허용과 특정 사용자에 대한 데몬을 위해 실행된다. 다음 예시처럼 key file이 보인다:

joe a0167ac6101b360d2f4dd164abba2337
bob 2dc36fc4807894cdfbe180b71d2b4a0f
sam 3fc763fb270ce6ba6e928bd10d4977d3

간략한 사용자이름은 hex encoded 128-bit value에 포함되어 있으며 bwctld.keys files을 유지하고 만들기 위한 가장 편리한 방법은 aespasswd 프로그램을 사용하는 것이다.

라. aespasswd

htpasswd(apache wedsever)와 유사하다. key file을 추가시키기 위해 identity를 명시하고 passphrase를 지정한다. 이것은 사용자가 128bit 전체를 기억하지 못하기 때문에 사용된다. 이는 여러 아키텍처에서 휴대용 방법으로 128bit hex key에 passhrase를 변환하는 데 사용된다. 동일한 응용프로그램은 bwctl에 대한 key file을 관리하며 bwctld에 사용된다.

key file ‘n’ option:

% aespasswd -n -f bwctld.keys demo

Additional usernames can be added by omitting the '-n' :
% aespasswd -f bwctld.keys joe

추가정보는 다음사이트를 참조할 수 있다.
http://e2epi.internet2.edu/bwctl/bwctld.keys.man.html,
http://e2epi.internet2.edu/bwctl/aespasswd.man.html,
http://e2epi.internet2.edu/bwctl/bwctld.limits.man.html.

마. BTC 실행

Bandwidth Test Controller에 대한 테스트는 두 가지 호스트를 구성해야 한다(단일 호스트에서 테스트의 endpoint 두가지를 실행 할 수 없습니다).

바. BTC Clinet 테스트 (bwctl)

첫째로, Internet2 호스트의 하나에 클라이언트 시스템으로부터 간략한 테스트가 가능하다.
% /ami/bin/bwctl -s nmsx-aami.abilene.ucaid.edu A AESKEY jimbob

bwctl은 공유된 비밀키로 사용되는 AES 키 생성에 사용된 암호에 대해 사용자 jimbob 메시지가 표시된다.

1) Bandwidth Test Controller Daemon (bwctld) 테스트

다음 단계는 로컬데몬을 실행하는 것이다. 데몬을 실행하고 '-Z' 옵션을 사용하여 테스트를 한다:

% /usr/local/bin/bwctld -c /usr/local/etc -Z

명령옵션줄의 대부분은 구성파일 매개변수를 제거하는데 사용된다. 예를 들면 데몬이 구성 디렉토리로부터 실행되지 않으면 '-c' 옵션은 거의 대부분 사용된다.

더 많은 정보는 다음을 참고할 수 있다:

http://e2epi.internet2.edu/bwctl/bwctld.man.html.

2) 두 개의 hosts, 두 개의 daemon 테스트

다음 단계는 이전처럼 원격호스트에 동일한 로컬호스트를 실행하는 것이지만, 현재는 bwctld 대신 클라이언트 응용프로그램 테스트의 로컬 엔드 포인트를 관리할 수 있는 실행이 있는지 확인해야 한다. 이 테스트를 실행하기 위해 또 다른 window를 사용해야 한다. 그럼 클라이언트를 실행하는 동안에 bwctld 프로세스 출력형태를 볼 수 있다.

% /ami/bin/bwctl -s nmsx-aami.abilene.ucaid.edu A AESKEY jimbob

만일 로컬호스트 IPmask를 사용하여 우회인증을 하지 못한다면 명쾌하게 c 옵션을 사용하여 테스트의 로컬호스트측면을 명시하는 것이 필요하다. 또한 만일 Abilene host로 로컬호스트에 대해 AESkey와 동일한 userid를 사용한다면 위와 같이 -s flag의 마지막의 인증정보를 추가하는 것 대신 테스트의 두 측면에 대해 -A flag를 사용하여 인증을 명시할 수 있다(후반부에 추가적인 설명이 있음).

3) 인증옵션 테스트

bwctl 클라이언트 응용프로그램 다중 key는 매우 유용하다. 왜냐하면 symmetric key이기 때문이며 외부조직에 대해 당사자의 “internal” key 공유가 되지 않게 하는 수단이기 때문이다.

단일 인증도메인에 대해 (AES key와 같음) 다음 명령어를 추가하고,

bwctl -A AE AESKEY myname -s hostA -c hostB

다른 도메인인증 사이의 경우에는 -s/-c 옵션의 끝에 인증정보를 추가한다.

bwctl -s hostA AE AESKEY myname -c hostB AE AESKEY othername

SAMPLE BWCTL COMMANDS	
bwctl -c receive_host	This will use your host as the sender, and run a 10 second iperf test.
bwctl -c receive_host -t 30 -f m -i 1	This will use your host as the sender, run a 30 second test, format the results in Megabits/sec. Also output results every 2 secs for both the client and server.
bwctl -c receove_host:55555 -t 30 -f m -w 16M -i 1	This will run the same test, but will connect to the bwctl daemon on the remote host that is running on the non-standard port 55555 (more on ports below). The TCP window will be 16MB.
bwctl -c receive_host -s send_host -t 30 -f m	This is very useful if you are not logged into one of the two endpoints. It runs a 30 second test from send_host to recv_host.
bwctl -L 1000 -c receive_host -t 30 -f m -i 1	By default bwctl exits if it can not get a test slot within 5 minutes. Use the -L flag for heavily used servers where you might have to wait longer.
bwctl -T iperf3 -c receive_host -s send_host -i 1 -u -b 500M	Use iperf3 instead of iperf, and do a 500Mbps UDP test.
bwctl -c receive_host -T nuttcp	run nuttcp instead of iperf
bwctl -T iperf3 -c receive_host -s send_host -o 5 -t 20	Use iperf3 instead of iperf, and omit the first 5 seconds of a 20 second test (removes TCP slowstart)

SAMPLE BWPING COMMANDS	
bwping -s send_host -c receive_host	run a ping from host A to host B
bwping -T owamp -s send_host-c receive_host -N 1000 -i .01	run owping from host A to host B, sending 1000 packets, spaced .01 seconds apart.
bwping -E -c www.google.com	run a ping to a host not running bwctld

SAMPLE BWTRACEROUTE COMMANDS	
bwtraceroute -c receive_host -s send_host	run a tracetroute from host A to host B
bwtraceroute -T tracepath -c receive_host -l 8192 -s send_host	run a tracepath from host A to host B, using a packet size of 8192 bytes. This will help find MTU issues.

3장 BWCTL MP

1. BWCTL MP 서비스

perfSONAR MDM bandwidth 컨트롤러 측정값(BTCTL MP)은 두 개의 bwctl 툴 사이에서 bandwidth 측정시 실행한다. 이는 다음 측정요구에 따라 서비스 한다.

- Achievable throughput (TCP)
- UDP throughput (UDP)

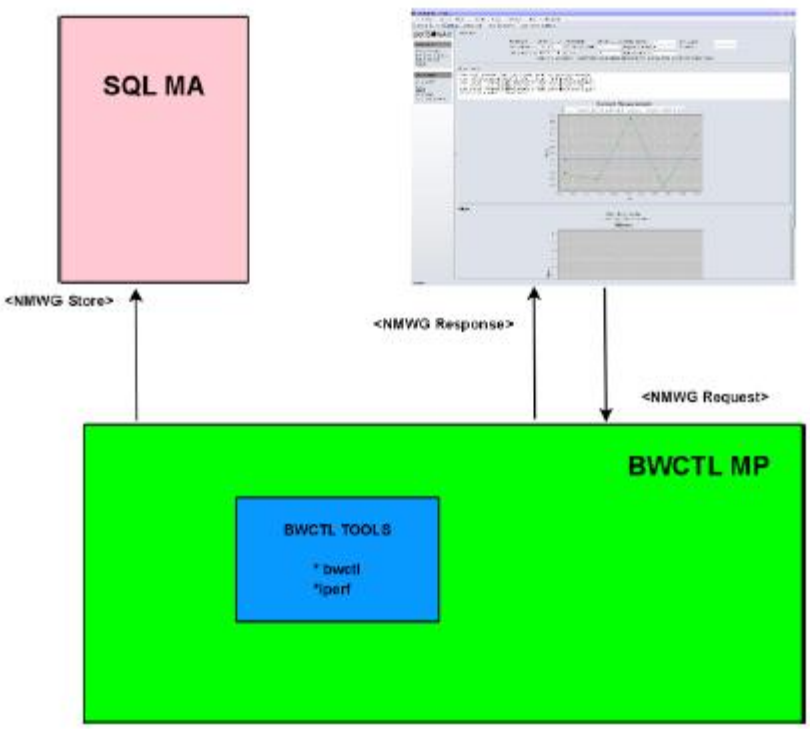


Figure 1.1 - System architecture oppd (Open Perl Perfsonar Daemon) [1]

가. 시스템구성

사용자가 perfsonarUI로부터 bwctl 툴을 실행하는 두 개의 호스트 사이에서의 처리량 측정을 요구하면 클라이언트는 BWCTL MP에 XML(NMWG schema)를 요청한다. BWCTL MP는 bwtcl tool을 사용하여 측정을 실행하고

XML response에서 클라이언트의 자료요청을 취소시킨다. 이를 “on demand

measuremnet” 라 한다.

또한, BWCTL MP는 SQL 데이터베이스에 측정데이터 저장이 가능하다. 이것은 SQL MA를 통해 이루어진다.

나. 설치

1) 지원 플랫폼

Red Hat Enterprise Linux6와 Debian 7.5

2) 사전요구 소프트웨어

BWCTL MP의 작동여부를 확인하기 위해 다음 소프트웨어가 필요하다.

- bwctl- RED Hat Enterprise Linuxd에 대한 BWCTL MP 인스톨러는 자동적으로 설치된다. 현재버전은 1.5이며 만일 bwtctl 소프트웨어를 수동으로 설치하기 원한다면 사용자 메뉴얼과 설치 포함서를 통해 상세한 내용을 확인할 수 있다.
  bwctl과 bwctl-server 설치가 필요하며 bwtcl daemon에 포함되어 있다.

  Debian 사용자: bwctl tool의 패키지 버전은 1.4이며 사용자는 다른 BWCTL MP에 대한 deployment에 대한 호환성에 주의해야 한다. 1.4버전을 설치하기 위해서는 Debian 저장소가 시스템에 구성되어야 한다. Debian 1.5버전을 사용하면 소스로부터 수동으로 설치하는 것이 가능하다.

- Perl - Linux 배포된 Perl interperter와 Perl libraries packages를 제공한다. BWCTL MP package들은 자동적으로 설치되지만, 모든 필요한 perl 모듈은 RHEL에 의해 제공되지 않으며 EPEL 저장소에서 찾을 수 있다.

- NTP - bwctl 테스트는 정확한 시간동기화에 의존적이며, ntpd 설치가 요구된다. 대부분 Linux 배포는 ntpd 패키지를 제공한다.

BWCTL MP 설치 후에는 수동적으로 설치한 bwctl 요소들이 적절한지 테스트하는 것을 추천한다. 또한 TCP 윈도우 사이즈가 제한되지 않고 정확하게 설정되어 있는지에 대한 확인이 필요하다.

3) bwctl 테스트

만일 시스템이 이미 bwctl tool을 사용 중이라면 bwctl daemon을 실행 후 같은 조건의 다른 호스트와 연결을 통해 버전과 프로세스들을 비교해 본다. 만일 모든 것이 정상이라면 BWCTL MP 설치가 마무리된 것이고 문제가 없다면 bwtcl daemon을 실행하여 테스트를 시작한다.

만약, 문제가 발생했을 경우,

- 방화벽이 실행되는 중이라면 TCP에 대한 bwctl control port(4823)을 확인하고 UDP/TCP에 대한 5000(iperf) port에 대한 allow 요청

```
# iptables -L
```

/etc/bwctl/bwctld.limits 파일이 두 선택창 사이 측정에 대해 허용하는 것들을 확인

- NTPS가 설정되어 있고 정상적으로 실행이 되는지 확인한다. 만일 시간 동기화가 정확히 구성되어 있지 않다면 bwctl tool은 측정이 불가능하다. NTP status를 바꾸기 위해서는 다음과 같이 실행한다.

```
# ntpq -p
```

- 네트워크연결 정상적인지 확인합니다. 컨트롤 케이블과 네트워크 카드등을 점검하고 만일 연결되어 있다면 Ping 테스트와같이 표준 툴을 사용하여 테스트합니다.

- bwctl tool 버전의 호환성 확인한다. 사용하는 버전을 bwctl tool 버전을 바꾸기 위해 다음과 같이 명령어를 실행하도록 한다.

bwctl -V

4) TCP Window 사이즈 설정

호스트에 bwctl을 설치한 후에는 TCP 버퍼사이즈(buffer size)가 적절히 입력되었는지 체크한다. TCP flow control과 경로(path), 지연(Latency)를 측정하지 않으면 네트워크상에서 TCP 버퍼사이즈를 네트워크 경로의 bandwidth에 상관없이 처리량이 최대치(이론치)가 제한될 수 있다. 최대 TCP 버퍼사이즈의 적절한 크기는 해당 시스템이 자동 알고리즘을 통해 대역폭(bandwidth)를 잘 사용할 수 있도록 window를 송수신할 수 있는 값까지 확장시킨다.

TCP 버퍼 사이즈 설정:

- bwtcl 호스트 장치의 개개에서는 테스트가 장치사이에서 실행될 수 있도록 설정
- 호스트사이에서 RTT 지연(latency)를 측정
- BDP값의 계산에 따라 각 호스트의 최대 TCP 버퍼사이즈 설정

다. Linux 에서의 BWCTL MP 설치

Linux를 실행한다면 RPM 배포를 사용해서 perfSONAR MDM BWCTL MP를 설치할 수 있다. 만일 Debian OS를 사용을 한다면 Debain packages를 사용하여 BWCTL MP를 설치하면 된다.

1) Debain 시스템에 GEANT Source 추가

/etc/apt/sources.list.d directory에 파일을 패치한다. 루트권한으로 접근하여 perfsonar-wheezy.list 정의 파일을 복사하기 위해 다음을 실행한다.

wget http://downloads.perfsonar.eu/repositories/deb/perfsonar-wheezy.list

다음을 실행하여 PGP key source를 저장한다.

wget http://downloads.perfsonar.eu/repositories/perfsonar.asc

apt-key add perfsonar.asc

apt-key list

시스템에서 패키지 리스트가 업데이트된다.

apt-get clean

apt-get update

정확히 패키지가 추가되었는지 확인한다.

apt-cache search perfsonar-oppd-mp-bwctl

나) Red Hat Enterprise Linux system에서 Geant source 추가

/etc/yum.repos.d directory에 배치하며 이 디렉토리에 루트권한으로 접근해야 한다. perfsonar-stable.repo라는 파일을 복사하기 위해 다음과 같이 실행한다.

32 bit(i386)을 사용 시:
wget http://downloads.perfsonar.eu/repositories/rpm/perfsonar-stable.repo

64 bit(x86_64)를 사용 시 :
wget http://downloads.perfsonar.eu/repositories/rpm/perfsonar-stable-x86_64.repo

정확히 패키지가 추가되었는지 확인한다.

yum search perfsonar-oppd-mp-bwctl

라. 패키지 설치

1) RPM distributions 사용 및 설치

Red Hat에 BWCTL MP를 설치하고 distribution를 형성하려면 다음과 같이 실행한다.

1. 호스트 BWCTL MP의 장치에 접속

32bit architecture(i3860) 사용 시:

rpm -ivh http://download.fedoraproject.org/pub/epel/6/i386/epelrelease-6-8.noarch.rpm

rpm -ivh http://software.internet2.edu/rpms/i386/RPMS.main/Internet2-repo-0.2-9.noarch.rpm

64 bit architecture(x86_64) 사용 시:

rpm -ivh http://download.fedoraproject.org/pub/epel/6/x86_64/epelrelease-6-8.noarch.rpm

rpm -ivh http://software.internet2.edu/rpms/x86_64/RPMS.main/Internet2-repo-0.2-9.noarch.rpm

2. yum을 사용하여 BWCTL MP를 설치합니다.

yum install perfsonar-oppd-mp-bwctl

3. BWCTL 서비스의 시작 및 중지

/etc/init.d/oppd [start¦stop¦restart]

2) Debain 패키지를 사용하여 설치하기

Debian이나 다른 유사한 distribution에 BWCTL MP를 설치하기 위해서 다음과 같이 실행한다.

1. BWCTL MP의 장치에 접속

2. 버전1.4부터 다음과 같이 소스(Source)를 지정한다.

   # echo “deb http://ftp.debian.org/debian jessie main“ >>
   /etc/apt/sources.list
   # echo -e “# Prefer packages comming from wheezy\nAPT::Default-Release
   “wheezy“;“ > /etc/apt/apt.conf.d/50release
   # apt-get update

3. BWCTL MP 웹 서비스를 설치

   # apt-get install perfsonar-oppd-mp-bwctl

4. BWCTL 서비스의 시작 및 중지

   # /etc/init.d/perfsonar-oppd [start¦stop¦restart]

3) 설치 테스트

BWCTL MP를 정확히 설치했는지 테스트하고 싶다면 다음을 실행한다.

1. # ps ax ¦ grep oppd

2. /usr/bin/perl/usr/bin/oppd.pl—config=/etc/oppd.conf-

pidfile=/var/run/oppd.pid

--> 정상 데몬 확인

4) BWCTL MP 구성하기

BWCTL MP는 oppd framework를 사용한다. 중요 구성파일은 /etc/oppd.conf이며 logfile에서 접속을 활성화시킬 수 있고 파일들의 경로들 설정 할 수 있다. 이 파일에는 설정할 수 있는 옵션에 대한 예가 있다.

Red hat을 시스템으로 한다면 모든 옵션에 대한 구성파일을 찾을 수 있으며 그 파일들을 /etc/sysconfig/oppd로 oppd in 한다. Debain 시스템은 /etc/default/oppd.으로 설정한다.

BWCTL MP 설정을 위해 /etc/oppd.d/bwctl.conf을 엽니다.

```
1. #
2. # BWCTL MP example configuration
3. <service MP/BWCTL>
4. # Necessary parameters for module initialisation
5. module        MP::BWCTL       # Name of module to load
6. servicetype   MP              # Service type: MP or MA
7. # Name, description, and keyword will be reported to Lookup Server
8. name          "BWCTL Measurement Point"
9. description   "Measurement Point for doing on-demand BWCTL tests"
10.      keyword       "project:mybwctl"
11.      # Further parameters
12.      metric        "bandwidth"
13.      # Measurement metric(s). More than one element definition possible.
14.      tool          "bwctl"         # Tool name
15.      # Module parameters
16.      <module_param>
17.      # Command to execute e.g. "/usr/bin/bwctl" or "/bin/bwctl".
18.      # Omitting path searches $PATH.
19.      command     "bwctl"
20.      # service
21.      # This is to get different service setting
22.      # For example eventtpye
23.      service     "bwctl"
24.      # Store functionality
25.      store       off       # Enable/disable store functionality
26.      store_url   "http://www.mySQL-MA:8090"
27.      # URL of a MA service to send the results of measurements to
28.      </module_param>
29.      </service MP/BWCTL>
```

module_param 칸에서 command 매개변수에 대해 bwctl tool의 경로를 정의할 수 있다. 이 BWCTL MP는 측정이 실행될 수 있다.

5) SQL 측정기록 데이터 저장

SQL MA와 같이 데이터베이스에 측정기록을 저장하기 위해 값을 on 으로 변경시킴으로써 25째 줄의 매겨변수를 저장 가능하게 한다. 26번째 줄의 store_url 매개변수에 대해 URL을 스토리지에 정의 할 수 있다.

BWCTL MP의 구성 후에 다음명령을 사용하여 서비스를 다시 시작할 수 있다.

```
# /etc/init.d/oppd restart
```

6) BWCTL MP 고려사항

- 방화벽 및 IP 설정

perfSONAR MDM 설정 시 기본적으로 방화벽을 사용하지 않는 것을 추천한다. BWCTL MP를 방화벽 뒤 서버에 설치했다면 특정 입력포트를 허용하는 방화벽 설정을 변경해야한다.

사용자가 정책을 설정할 때와 bwctl 테스트가 서비스부분 테스트를 제한할 때 호스트서버에서 /etc/bwctl/bwctld.limits 파일을 설정한다.
https://forge.geant.net/forge/download/attachments/491888/bwctld.limits 에 위치한 perfSONAR MDM 파일의 예를 사용권장 한다.

참 고

[1] perfSONAR. RNP, GÉANT, ESnet, Internet2. PERFormance Service Oriented Network monitoring Architecture. http://www.perfsonar.net

[2] Bandwidth Test Controller (bwctl). bwctl Version 1.5. (Bandwidth Control) http://software.internet2.edu/bwctl/. Internet2.

[3] Window-Based Transmission. eduPERT. r11 - 07 Apr 2006 - Simon Leinen. Copyright 2004 - 2009 by the contributing authors.

[4] TCP Window Scaling Option. eduPERT. r4 - 03 Jun 2008 - Simon Leinen. Copyright 2004 - 2009 by the contributing authors.

[5] perfSONAR MDM and TCP buffers, D. Vicinanza, EGI Technical Forum 2013, http://geant3.archive.geant.net/service/edupert/Resources/Documents/TCP_exercises_with_perfSONAR.pdf

[6] http://e2epi.internet2.edu/case-studies/VLBI/cs-index.html