

Traffic Control 기반의 트래픽 제어

TC소개

- TC ≠ Test Case, TC = Traffic Control
- 기본적으로 데이터 전송시, 최대 성능을 위해 커널에서는 대역폭의 제한을 설정하지 않음
- 스위치 장비에서 대역폭 제한을 설정할 경우 10/100M, Half/Full 형식으로만 제한작업이 가능
- 하지만, NIC의 link auto negotiation 기능 때문에 스위치 or 서버장비에서만 설정하면 Link Down 현상 발생
- 운영자가 원하는 세밀한 대역폭 제한 설정 불가.
- tc설정상태는 메모리에 저장되기 때문에, 장비를 재부팅하면 원래상태로 되돌아옴
- 이런문제를 해소하기 위한 방안, TC
- Iproute rpm에 포함된 명령어(iproute는 기본 패키지) Iproute 패키지에는 netstat, ifconfig, arp 의 명령어가 포함

Linux Traffic 처리 방식

PREROUTING : 외부에서 내부로 패킷이 유입될 때 적용받는 라우팅

POSTROUTING : 내부에서 외부로 나갈 때 적용받는 라우팅

Traffic Shaping 방식 소개

- Shaping?

1. Traffic Shaping : 정의한 포트/트래픽을 조절하여, 네트워크 속도를 지연시키는 방법
2. 트래픽 방향에 따라서 TBF / HTB 방식을 이용,

- TBF (Token Bucket Filter)

1. Packet을 Bucket에 담아 트래픽을 지연시키고, 정의한 양만큼 Traffic Shaping 적용
2. Inbound 트래픽에 대해 적용 가능
3. TBF Data Flow

- HTB (Hierarchy Token Bucket)

1. 기본 방식은 TBF와 동일
2. 다중 Class를 생성이 가능하기 때문에 특정 IP 혹은 특정 포트 정책을 2개 이상 생성 가능
3. Outbound 트래픽에 대해 적용 가능

HTB Data Flow

4. HTB Queue의 상관관계

Qdisc를 통해 특정 인터페이스의 Queue를 생성하고, 하위 Class를 통해 대역폭을 설정
하위 Filter를 통해 특정IP, 특정 포트에 대해서 정책 생성
Class는 Qdisc의 종속되어 있고 Filter는 class의 종속되어 Queue처리함

TBF 방식의 구성 테스트

1. TBF 방식의 Traffic Shaping (eth0 인터페이스에 최대 10Mbps, 데이터 전송간 최대 70ms의 지연 발생)

```
$ tc qdisc add dev eth0 root tbf rate 10mbit latency 70ms burst 1540
```

2. Shaping 적용 확인

```
$ tc qdisc show dev eth0
qdisc tbf 8012: root refcnt 2 rate 10000Kbit burst 1540b lat 70.0ms
```

3. Shaping 변경

```
$ tc qdisc change dev eth0 root tbf rate 20mbit latency 70ms burst 1540
```

4. 변경 적용 확인

```
$ tc qdisc show dev eth0
qdisc tbf 8012: root refcnt 2 rate 20000Kbit burst 1540b lat 70.0ms
```

5. tc정보 삭제

```
$ tc qdisc del dev eth0 root
```

HTB 방식의 구성 테스트

1. eth0 인터페이스에 최대 Limit는 100Mbps으로 설정

```
$ tc qdisc add dev eth0 root handle 1:0 htb default 10
$ tc class add dev eth0 parent 1: classid 1:1 htb rate 100mbit
$ tc class add dev eth0 parent 1:0 classid 1:10 htb rate 50mbit ceil 50mbit burst 10k prio 0
$ tc class add dev eth0 parent 1:0 classid 1:11 htb rate 20mbit ceil 20mbit burst 10k prio 0
$ tc filter add dev eth0 protocol ip parent 1:0 prio 0 u32 match ip dport 2049 0xffff flowid 1:10
$ tc filter add dev eth0 protocol ip parent 1:0 prio 0 u32 match ip dport 22 0xffff flowid 1:11
```

- destination port가 tcp/80인 경우 최대 50Mbps으로 제한
- destination port가 tcp/22인 경우 20Mbps으로 제한
- rate : 최대 허용 대역폭
- ceil : 허용가능한 버킷양
- prio : 우선순위 설정 0부터 가능하고 낮은숫자가 우선순위를 갖음

1. 적용 확인

qdisc 적용확인

```
$> tc qdisc show dev eth0
qdisc htb 1: root refcnt 2 r2q 10 default 10 direct_packets_stat 17
```

Class 적용확인

```
$> tc class show dev eth0
class htb 1:11 root prio 0 rate 20000Kbit ceil 20000Kbit burst 10Kb cburst 1600b
class htb 1:10 root prio 0 rate 50000Kbit ceil 50000Kbit burst 10Kb cburst 1600b
class htb 1:1 root prio 0 rate 100000Kbit ceil 100000Kbit burst 1600b cburst 1600b
```

#Filter 적용 확인

Match (Destination 포트 정보)는 16진수로 출력

```
$> tc filter show dev eth0
filter parent 1: protocol ip pref 49151 u32
filter parent 1: protocol ip pref 49151 u32 fh 801: ht divisor 1
filter parent 1: protocol ip pref 49151 u32 fh 801::800 order 2048 key ht 801 bkt 0 flowid 1:11
  match 00000016/0000ffff at 20
filter parent 1: protocol ip pref 49152 u32
filter parent 1: protocol ip pref 49152 u32 fh 800: ht divisor 1
```

```
filter parent 1: protocol ip pref 49152 u32 fh 800::800 order 2048 key ht 800 bkt 0 flowid 1:10
match 00000801/0000ffff at 20
```

2. 정책 변경(20Mbit -> 200Mbit)

```
$> tc class change dev eth0 parent 1:0 classid 1:11 htb rate 200mbit ceil 200mbit burst 10k prio 0
```

3. 변경 정책 적용 확인

```
$> tc class show dev eth0
class htb 1:11 root prio 0 rate 200000Kbit ceil 200000Kbit burst 10200b cburst 1600b
class htb 1:10 root prio 0 rate 50000Kbit ceil 50000Kbit burst 10Kb cburst 1600b
class htb 1:1 root prio 0 rate 100000Kbit ceil 100000Kbit burst 1600b cburst 1600b
```

NMS 연동한 Traffic Shaping 적용 확인

1. Traffic Shaping 적용전 (최대 610Mbps까지 트래픽 발생)

2. Traffic Shaping 적용 후 (100Mbps 적용시 최대 77.84Mbps 까지 나옴)

TC의 알려진 문제점

실제 트래픽 양을 설정해도, 트래픽이 처리 과정 사이에 Queue처리 하는 절차가 추가되기 때문에 설정한 양보다 전송속도가 떨어짐

TBF 설정 후 유입되는 Packet양이 Bucket양보다 클 경우 Packet Drop하는 문제가 존재

🕒Revision #3

★Created 7 June 2022 15:09:58 by artop0420

✎Updated 24 December 2023 00:51:54 by artop0420