

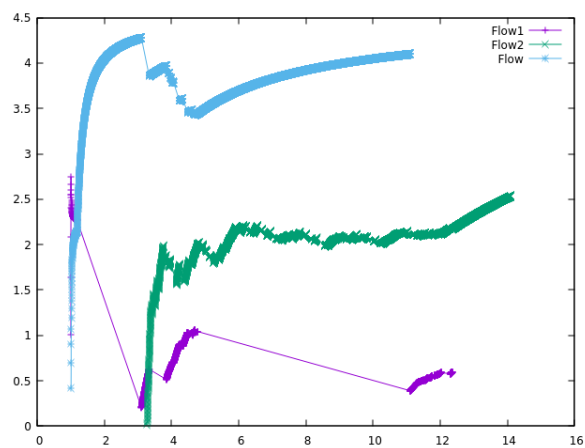
네트워크 프로젝트 과제 1

2017313260 이재민

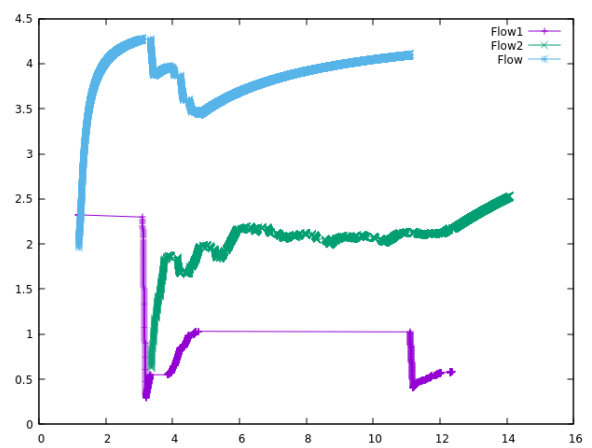
Default Flow가 그냥 Flow로 라벨되어 있음. 그 외에 Flow1, Flow2

Question1. UDP

기본 결과값을 plot 했을 때

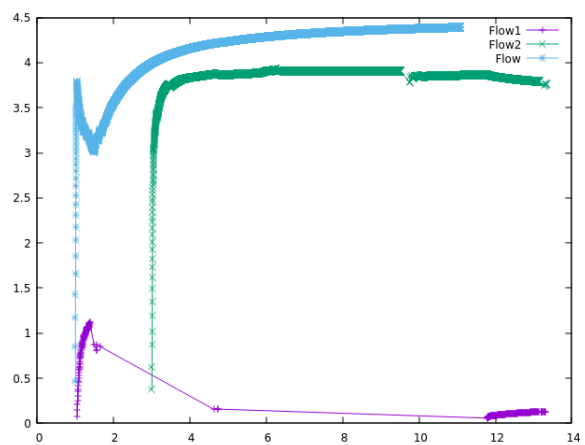


moving average를 plot 했을 때

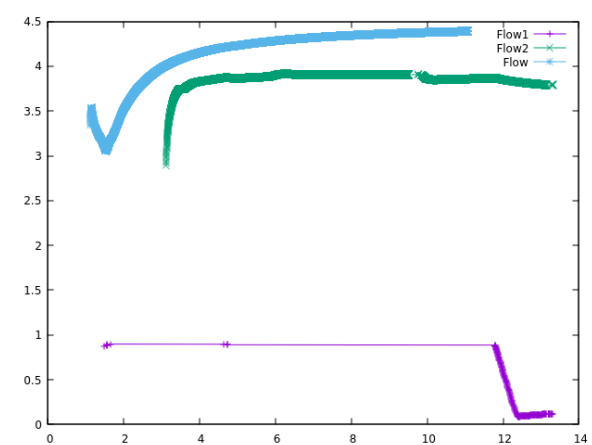


Question2. TCP

기본 결과값을 plot 했을 때



moving average를 plot 했을 때



Question3.

1) UDP인 경우

모두가 UDP가 되게 되면, Default Flow와 Flow1의 목적지가 같은데, 보낼 수 있는 패킷은 보내고 비연결지향형이기 때문에, 데이터 순서가 없고 중간중간에 Flow1의 데이터를 받게 된다. 이 순간에 Default Flow의 속도는 저하되거나 Packet Loss가 발생한다. 그리고 이미 송신된 packet에 대해 receiver가 열려있으므로, Default Flow가 끝난 뒤 처리가 된다. 흐름제어, 혼잡제어가 안되기 때문에, Bridge에서 처리가능한 한계가 있어 Flow2의 속도도 저하되고, 전송이 끝나고 13초 이후 추가적인 시간에 데이터를 받는 모습을 보인다. 또한 Packet Loss가 발생해서 중간중간에 연결이 끊기는 듯한 모습이 보인다.

2) TCP인 경우

CSMA 수용 가능 DataRate가 5Mbps인데, Default Flow는 10Mbps로, UDP로 패킷을 최대 속력으로 계속 받는다. UDP는 흐름제어, 혼잡제어가 안된다. 그렇게 되면, 연결지향형인 TCP는 Default Flow와 같은 목적지로 가는 데이터 전송이 CSMA의 충돌감지로 지연되고, 연결이 끊긴 듯한 모습을 보인다. Receiver로부터 받은 ACK가 없어, Sender는 연결이 끊어졌다고 판단해 패킷을 전송하지 않았을 수도 있다. 그래서 Default Flow가 종료된 뒤 마지막에 이미 보낸 packet에 대해서만 받은 모습으로 보인다. 혹은 Flow2가 진행중이기 때문에 CSMA의 제어로 인해 느린 속도로 처리를 했을 가능성도 있다. Flow2는 목적지 node가 다르기 때문에 전송 및 연결이 정상적으로 가능했으나, 중간에 Packet Loss가 발생한 것을 볼 수 있다.

Question4.

PacketSinkHelper로 프로토콜, 주소를 지정해주면 PacketSink 구조체 Local, Protocol에 저장된다.

그리고 PacketSink::StartApplication에서 Socket::CreateSocket으로 ptr<Socket>을 생성한다.

PacketSink::StartApplication 에서 소켓이 없다면 소켓을 새로 생성하고 Ipv4RawSocketImpl::Listen으로 패킷을 받을 수 있게 열어준다. 그리고 각종 ::Handle~ 함수로 컨트롤해준다.

Socket 내부엔 연결과 관련된 callback에 대한 변수와, m_boundnetdevice가 있다. m_boundnetdevice는 Ptr<NetDevice>의 변수이다. 이는 ns3::NetDevice를 보는 포인터이고, 이는 node type, callback, address등이 담겨있다. BindToNetDevice로 NetDevice로 바인드한다.

Socket 내부에 Send, SendTo 함수로 패킷을 전송한다. Packet이 없다면 새로 생성한다. Packet내부에는 각종 메타데이터와 버퍼가 담겨있다. 그리고 Recv, RecvFrom으로 패킷을 받는다.

Sinker를 Install한 Application이 HandleRead로 시간, Packet 사이즈, 주소, 포트, bytes 를 Socket의 RecvFrom으로 받아들인다. 이는 NetDevice에 바인드된 Socket으로, Packet을 넘겨받는다.

Onoff도 유사하게, OnOffApplication으로 연결된다. StartApplication이 실행되면, Socket이 없다면 SetSocket으로 m_socket에 Socket을 생성해준다. 그리고 SendPacket에서, Packet을 생성하고, 아까 만든 Socket의 Send 함수로 Packet 전송을 시작한다. 마찬가지로, Socket은 NetDevice와 밀접하게 연결되어있다.

즉, 네트워크 스택에서, 사용자가 입력한 명령어 아래, Socket, protocol, Address, Packet, Network Device(NIC)이 깔려져있다.