NS3 Project Update -1

The Peak-Hopper: A New End-to-End Retransmission Timer for Reliable Unicast Transport

Sihat Afnan
Student ID : 1705098

OverView Of Proposed Agorithm

It essentially runs two RTO algorithms in parallel.One algorithm (Short-Term History RTO) monitors the present and short-term history in order to respond to RTT increases. The other algorithm(Long-Term History RTO) simply decays the current value of RTO, and can therefore be said to represent the long-term history.

$$\delta = \frac{RTT_{sample} - RTT_{previous}}{RTT_{previous}} \qquad \text{(Step 1)}$$

$$D = 1 - \frac{1}{F * S} \qquad \text{(Step 2)}$$

$$B \leftarrow \max(\delta, D * B) \qquad \text{(Step 3)}$$

$$RTT_{\max} = \max(RTT_{sample}, RTT_{previous}) \qquad \text{(Step 4)}$$

$$RTO \leftarrow \max(D * RTO, (1 + B) * RTT_{\max}) \qquad \text{(Step 5)}$$

$$RTO \leftarrow \max(RTO, RTO_{\min}) \qquad \text{(Step 6)}$$

Implementation Workflow

- Closely examine the default Implementation of RTT in NS3.
- Generate the default RTT & RTO graphs using scratch/tcp-variants-comparison.cc as the simulation script.
- Modifications to be made in rtt-estimator.cc file

  -> Add a new method named FloatingPointUpdate_PeakHopper(Time m) in RttMeanDeviation class which will be invoked from Measurement method.This method will contain the Peak Hopper implementation of RTT.

- In tcp-socket-base.cc file , EstiimateRtt(const TcpHeader& tcpHeader) method needs to be modified.
- Inside this method , RttMeanDeviation::Measurement(Time m) is called.Since our algorithm needs the previously sampled RTT too,we will pass it along with current sampled RTT m.
- We might need another method to calculate Decay factor D, fader variable F and booster variable B in tcp-socket-base.cc file.

Function calling sequence

- In the simulation script,where network topology will be built too,we will need to enable a peakHopper flag which will be used by both TcpSocketBase and RttMeanDeviation class.

```
Config::SetDefault("ns3::TcpSocketBase::peakHopper",
BooleanValue(true));
```

```
   Config::SetDefault("ns3::RttMeanDeviation::peakHopper",
BooleanValue(true));
```

- After enabling boolean flag,the EstimateRtt method in tcp-socket-base.cc will execute certain portions of code conditioned by this flag.

```
if(peakHopper){
    m_rtt->Measurement (m, lastRtt,CalculateGain());//m_rtt is an RttEstimator object
    lastRtt = m;//setting the current sampled rtt as previously sampled rtt
    m_rto = //do something
  }
```

- As long as EstimateRtt method is being called , RttTracer method will keep executing and write into the data file to generate graph later.

```
static void TraceRtt (std::string rtt_tr_file_name){
  AsciiTraceHelper ascii;
  rttStream = ascii.CreateFileStream (rtt_tr_file_name.c_str ());
  Config::ConnectWithoutContext ("/NodeList/1/$ns3::TcpL4Protocol/SocketList/0/RTT", MakeCallback
(&RttTracer));}
```

Thank You