Flow Monitor

Model Description

The source code for the new module lives in the directory `src/flow-monitor`.

The Flow Monitor module goal is to provide a flexible system to measure the performance of network protocols. The module uses probes, installed in network nodes, to track the packets exchanged by the nodes, and it will measure a number of parameters. Packets are divided according to the flow they belong to, where each flow is defined according to the probe's characteristics (e.g., for IP, a flow is defined as the packets with the same {protocol, source (IP, port), destination (IP, port)} tuple.

The statistics are collected for each flow can be exported in XML format. Moreover, the user can access the probes directly to request specific stats about each flow.

Design

Flow Monitor module is designed in a modular way. It can be extended by subclassing `ns3::FlowProbe` and `ns3::FlowClassifier`.

The full module design is described in **[FlowMonitor]**

Scope and Limitations

At the moment, probes and classifiers are available for IPv4 and IPv6.

Each probe will classify packets in four points:

- When a packet is sent (SendOutgoing IPv[4,6] traces)
- When a packet is forwarded (UnicastForward IPv[4,6] traces)
- When a packet is received (LocalDeliver IPv[4,6] traces)
- When a packet is dropped (Drop IPv[4,6] traces)

Since the packets are tracked at IP level, any retransmission caused by L4 protocols (e.g., TCP) will be seen by the probe as a new packet.

A Tag will be added to the packet (`ns3::Ipv[4,6]FlowProbeTag`). The tag will carry basic packet's data, useful for the packet's classification.

It must be underlined that only L4 (TCP, UDP) packets are, so far, classified. Moreover, only unicast packets will be classified. These limitations may be removed in the future.

The data collected for each flow are:

- timeFirstTxPacket: when the first packet in the flow was transmitted;
- timeLastTxPacket: when the last packet in the flow was transmitted;
- timeFirstRxPacket: when the first packet in the flow was received by an end node;
- timeLastRxPacket: when the last packet in the flow was received;
- delaySum: the sum of all end-to-end delays for all received packets of the flow;
- jitterSum: the sum of all end-to-end delay jitter (delay variation) values for all received packets of the flow, as defined in **RFC 3393**;
- txBytes, txPackets: total number of transmitted bytes / packets for the flow;
- rxBytes, rxPackets: total number of received bytes / packets for the flow;
- lostPackets: total number of packets that are assumed to be lost (not reported over 10 seconds);
- timesForwarded: the number of times a packet has been reportedly forwarded;
- delayHistogram, jitterHistogram, packetSizeHistogram: histogram versions for the delay, jitter, and packet sizes, respectively;
- packetsDropped, bytesDropped: the number of lost packets and bytes, divided according to the loss reason code (defined in the probe).

It is worth pointing out that the probes measure the packet bytes including IP headers. The L2 headers are not included in the measure.

These stats will be written in XML form upon request (see the Usage section).

References

[FlowMonitor] G. Carneiro, P. Fortuna, and M. Ricardo. 2009. FlowMonitor: a network monitoring framework for the network simulator 3 (NS-3). In Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '09). http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7493

Usage

The module usage is extremely simple. The helper will take care of about everything.

The typical use is:

```
// Flow monitor
Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();

Simulator::Stop (Seconds(stop_time));
Simulator::Run ();

flowMonitor->SerializeToXmlFile("NameOfFile.xml", true, true);
```

the `SerializeToXmlFile ()` function 2nd and 3rd parameters are used respectively to activate/deactivate the histograms and the per-probe detailed stats.

Other possible alternatives can be found in the Doxygen documentation.

Helpers

The helper API follows the pattern usage of normal helpers. Through the helper you can install the monitor in the nodes, set the monitor attributes, and print the statistics.

One important thing is: the **`ns3::FlowMonitorHelper`** must be instantiated only once in the main.

Attributes

The module provides the following attributes in **`ns3::FlowMonitor`**:

- MaxPerHopDelay (Time, default 10s): The maximum per-hop delay that should be considered;
- StartTime (Time, default 0s): The time when the monitoring starts;
- DelayBinWidth (double, default 0.001): The width used in the delay histogram;
- JitterBinWidth (double, default 0.001): The width used in the jitter histogram;
- PacketSizeBinWidth (double, default 20.0): The width used in the packetSize histogram;
- FlowInterruptionsBinWidth (double, default 0.25): The width used in the flowInterruptions histogram;
- FlowInterruptionsMinTime (double, default 0.5): The minimum inter-arrival time that is considered a flow interruption.

Output

The main model output is an XML formatted report about flow statistics. An example is:

```
<?xml version="1.0" ?>
<FlowMonitor>
  <FlowStats>
  <Flow flowId="1" timeFirstTxPacket="+0.0ns" timeFirstRxPacket="+20067198.0ns" timeLastTxPac
  </Flow>
  </FlowStats>
  <Ipv4FlowClassifier>
  <Flow flowId="1" sourceAddress="10.1.3.1" destinationAddress="10.1.2.2" protocol="6" source
  </Ipv4FlowClassifier>
  <Ipv6FlowClassifier>
  </Ipv6FlowClassifier>
  <FlowProbes>
```

```
  <FlowProbe index="0">
    <FlowStats  flowId="1" packets="3735" bytes="2149400" delayFromFirstProbeSum="+0.0ns" >
    </FlowStats>
  </FlowProbe>
  <FlowProbe index="2">
    <FlowStats  flowId="1" packets="7466" bytes="2224020" delayFromFirstProbeSum="+1994153892
    </FlowStats>
  </FlowProbe>
  <FlowProbe index="4">
    <FlowStats  flowId="1" packets="3735" bytes="2149400" delayFromFirstProbeSum="+1387315263
    </FlowStats>
  </FlowProbe>
  </FlowProbes>
</FlowMonitor>
```

The output was generated by a TCP flow from 10.1.3.1 to 10.1.2.2.

It is worth noticing that the index 2 probe is reporting more packets and more bytes than the other probes. That's a perfectly normal behaviour, as packets are fragmented at IP level in that node.

It should also be observed that the receiving node's probe (index 4) doesn't count the fragments, as the reassembly is done before the probing point.

Examples

The examples are located in *src/flow-monitor/examples*.

Moreover, the following examples use the flow-monitor module:

- examples/matrix-topology/matrix-topology.cc
- examples/routing/manet-routing-compare.cc
- examples/routing/simple-global-routing.cc
- examples/tcp/tcp-variants-comparison.cc
- examples/wireless/multirate.cc
- examples/wireless/wifi-hidden-terminal.cc

Troubleshooting

Do not define more than one **`ns3::FlowMonitorHelper`** in the simulation.

Validation

The paper in the references contains a full description of the module validation against a test network.

Tests are provided to ensure the Histogram correct functionality.