

LAB5: More TraceSources and Trace Helpers

CS169: Mobile Wireless Networks - Winter 2018

Xukan and Thomas

Department of Computer Science and Engineering
University of California, Riverside

February 9, 2017

Table of Contents

1 More TraceSources

2 Trace Helpers

- Let's go back to mythird.cc
- Go to ns-3 doxygen and look for All TraceSources
- Look for PhyTxBegin of WifiPhy
- Look for Config Path and Callback Signature
- Let's create a trace sink function and wire it to the trace source "PhyTxBegin"
- Hint: `$ find . -name '*.cc' | xargs grep PhyTxBegin`

- Let's go to myfifth.cc
- Create a trace sink function and wire to to the trace source "TxRxPointToPoint"
- Observe txTime and rxTime of the following parameter changes
- Change the number of packets to 1
- Change packet size
- Change p2p channel data rates and delay

Previously Seen Trace Helpers


```
pointToPoint.EnablePcapAll ("second");  
pointToPoint.EnablePcap ("second", p2pNodes.Get (0)->GetId (), 0);  
csma.EnablePcap ("third", csmaDevices.Get (0), true);  
pointToPoint.EnableAsciiAll (ascii.CreateFileStream ("myfirst.tr"));
```

Trace Helpers - A Big Picture

Each helper has Methods + Filenames

	PCAP	ASCII
Device Helper	✓	✓
Protocol Helper	✓	✓

Device Helpers - PCAP

- Which device supports PCAP?
- `$ find . -name '*.cc' | xargs grep ::EnablePcapInternal`
- Different Methods to enable PCAP. 
- `void EnablePcap (std::string prefix, <Ptr >NetDevice nd, bool promiscuous = false, bool explicitFilename = false);`
- `void EnablePcap (std::string prefix, std::string ndName, bool promiscuous = false, bool explicitFilename = false);`
- `void EnablePcap (std::string prefix, NetDeviceContainer d, bool promiscuous = false);`
- `void EnablePcap (std::string prefix, NodeContainer n, bool promiscuous = false);`
- `void EnablePcap (std::string prefix, uint32_t nodeid, uint32_t deviceid, bool promiscuous = false);`
- `void EnablePcapAll (std::string prefix, bool promiscuous = false);`

Device Helpers - PCAP

- PCAP Filenames
- Common forms - `<prefix>-<node id>-<device id>.pcap`
- Explicit filenames
- `helper.EnablePcap ("my-pcap-file.pcap", nd, true, true);`

Protocol Helpers - PCAP

- Which protocol supports PCAP?
- `$ find . -name '*.cc' | xargs grep ::EnablePcapIpv4`
- Different Methods to enable PCAP.
- **Interface:** `helper.EnablePcapIpv4 ("prefix", interfaces);`
Node: `helper.EnablePcapIpv4 ("prefix", n);`
Node+Device ID: `helper.EnablePcapIpv4 "prefix", 21, 1);`
All: `helper.EnablePcapIpv4All ("prefix");`

Protocol Helpers - PCAP

- PCAP Filenames
- Common forms - `<prefix>-n<node id>-i<interface id>.pcap`
- Explicit filenames are also available.

Questions?

- Extend `mythird.cc` to display `MacTx` events of Wifi nodes (both cases where node running echo client and node not running echo client)
- Extend `myfifth.cc` to display `NextTxSequence` of `TcpSocketBase`
- On `myfifth.cc`, set error rate to 10^{-3} and then 10^{-5} and enable pcap on Internet stack. Use `wireshark` or `tcpdump` to compare the results of the two error-rate scenarios