# COMP3310/6331 – Tute/Lab #9

Outline of Tute/Lab:

1. This is the second last week of the semester, the second last set of tutes, and there are only 4 guest lectures and one formal lecture left. All of these lectures should be posted in week 11. Yes, the guest lectures are examinable content, but not to any great depth. You will want to understand why these topics are important and relevant to the course, what the challenges were that they tackled and how these can be dealt with.
2. The last week of the semester will have some hands-on coding exercises, relating to network security (whatever that actually means, as you will see). This week is for ensuring you are on track for the third assignment and some review.
3. The third assignment is analysing MQTT performance. Take this tute time to review you have the clients working ok (Tute 8) with the demo broker (comp3310.ddns.net), and you should by now have started coding your own client to collect statistics. You can use any reasonably common programming language, and yes you can use mqtt libraries such as paho - or code your own protocol interpreter if you're really keen.
    a. **MQTTSpy:** https://www.eclipse.org/paho/components/mqtt-spy/ though a few people have noted issues getting it to run with various Java or JavaFX versions
    b. MQTT Explorer: A perhaps more modern and more reliable client at http://mqtt-explorer.com/ that has had some positive feedback.
    c. **mosquitto_pub, mosquitto_sub:** command line clients that are really useful for general analysis, except that they don't handle large volumes of messages as well as a GUI can. https://mosquitto.org/download/

**Questions from lectures**

1. What are some the network feedback sources that help tell you when things are going badly somewhere on the network?
    - ICMP - you can't reach something for various reasons; you have a poor MTU for a given network path.
    - ECN - routers tell you that congestion is looming and you can back off
    - TCP flow control information, especially the ACK clock, plus SACK information, and jitter+latency - tell you about congestion, loss, path changes

2. Why can't we use all that feedback to measure what is going everywhere?
    - They only work along the path of our application's communication. We have no control over the path, and the path may change while our application is running. TCP feedback is also a total across the entire path, without a clear indication where along the path an issue may be occurring.

3. What problems does SNMP solve?
    - It solves the inability to see the behaviour of network elements that are not on your path, and/or not involved with your application. It also provides data that is standardised and that you can aggregate. This means you can detect congestion, errors, other problems anywhere, and especially on the specific devices that may be causing it, without having to test every path. You can see up/down/change events

(e.g. interfaces going up/down) that nothing in IP will ever tell you about. However, it requires an SNMP agent (or server) to be running on every device you want to watch, and till recently the security model was pretty poor.

4. Why doesn't it solve all those problems across the width of the Internet?
    - People don't want to give access to information about their network internals, so you can only see that part of the network you control.

5. Why do we want views over time and space?
    - A view over **time** tells us what "normal" behaviour is, and so what "abnormal" behaviour is, and if it happens when we aren't looking. We can't tell whether congestion is a short term burst due to something going wrong, or a link that has just got too many users on it and needs to be expanded. A view over **space** gives us a snapshot of our network as it is currently running, and whether any issues need to be tackled right now (e.g. a link is unexpectedly down).

6. Why does an SNMP trap get sent, and by whom?
    - Sent by an agent to a manager, to alert it that something "important" has happened. Unlike standard SNMP messages that are query/response transactions initiated by a manager, a trap is sent by an agent unrequested, and may indicate one of the 6 core events have occurred (slide 16 of T8) - or one of many others that vendors have defined.

7. What's the point of a MIB?
    - A Management Information Base is the structure of a database, that can be linked with other MIBs to give us a globally unique naming scheme for information held by SNMP agents. It defines each of the fields, their structure, characteristics and relationships, as well as describing in more human terms what they hold.

8. What is a 'GetNext' used for?
    - An SNMP Get request is for a specific item in a MIB, so you need to know it exists. Items such as interface-specific data are an unknown quantity, there could be 1-100 different interfaces, each with any number of unknown characteristics stored for each one. By using GetNext, the agent walks through a MIB from a known item to the "next" one. This allows the manager to explore tables of data without knowing how many rows and columns there may be, and tables don't have to be rectangular (the same number of columns in each row).