COMP3310/6331 – Tute/Lab #4

#	
1	**Assignment preparations:** The second assignment will require you to write a web (http) client. Make sure that your earlier socket code is working properly and you can send/receive messages. The assignment will be dealing with HTTP1.0. Look up RFC 1945, and look at the protocol messages there, and compare these to what you see with a wireshark when connecting to e.g. www.bom.gov.au. Note that a connection running over https has additional security (encryption, and more interactions), and will be trickier to wireshark. If you are ready, you should modify your code to issue http HEAD requests to a webserver of your choice, and see how it responds, and then try to send http GET requests. The first lecture this week will cover HTTP.
2	**Revision and checking understanding to date** 1. A 'netstat -n' command returns a table with a row like this: *TCP 192.168.1.2:53398 150.203.56.47:80 CLOSE_WAIT* What has just happened here? (you can check the man page for netstat, and/or run it) ○ There has been an outbound TCP connection from my machine (left column), on a private network (the magic 192.168/16) to a webserver (probably) or something running on port 80 out on the public internet. The connection has concluded, our socket has closed, and we are waiting for the final packets to be acknowledged (CLOSE_WAIT). 2. Packets with a source IP address of 0.0.0.0 is used (by DHCP) to signify what? What does the address mean to the Operating System? Why do we use it that way? ○ Packets on a wire with a source IP of 0.0.0.0 imply a host doesn't know its IP address yet, so should be reached via broadcast. In the operating system the 0.0.0.0 means 'any interface', i.e. it will send from and/or listen to all interfaces (e.g. multiple wired and/or wireless), useful if you don't know which interface to use when starting up. Elsewhere a 0.0.0.0/0 forwarding (routing) prefix is also another description for the 'default' route to the wider Internet – 32bits of host address (=0 bits of network address) implies 'the whole internet'. 3. Why does DHCP have both discover/offer AND request/acknowledge? Why not just request/acknowledge? ○ Several reasons. Because a DHCP allocation is a lease, that needs to be validated. So there's a difference between starting up and knowing nothing about your situation - what's your address, where is the server - and renewing a lease when you do (probably) know them. You also need to avoid timing issues if multiple devices are starting up at the same time, and may ask for the same address. The server and the client both have to be sure that an allocation is unique. 4. Give some examples where and why a single IP address may be used by multiple (DNS) names? What about examples where a single name may resolve to multiple addresses?

	○ One interface/IP address may host several different services, e.g. a web server, mail server, etc. This gives flexibility for eventual relocation of a service to another server or interface. Having multiple addresses for a single name is usually a list of options, potentially to provide failover (try first one, if it's down try second, …), or perhaps to load-share (different one is presented first for each query). 5. In DNS, how are Zones and Domains different? ○ A domain is a specific, delegated chunk of the Domain Name System, from a gTLD or ccTLD down to e.g. anu.edu.au or lower. A zone is a set of records within a domain – it may span the whole domain or just a piece of it. 6. Why are DNS delegations legally important? ○ Domain names are tied to legal entities, e.g. corporations, countries, associations, etc. and they have trademarks over their names. Having somebody else try to register a domain name that is the same, or even similar, to an existing legal entities' name can be trademark infringement, or fraud. 7. What's the difference (and for/against) using Iterative or Recursive DNS queries? ○ Iterative, you do all the work, and get to cache the information yourself. It's potentially faster, but more work. Recursive, you ask the nameserver to do all the work for you. Easier, but potentially slower, especially if you need to query other devices in the same domain. 8. Install a tool like 'dig' (for DNS - "domain information groper") on your machine, if it doesn't already have it (see e.g. https://drjohnstechtalk.com/blog/2016/01/dig-for-windows/), and look at its man page or userguide. Use it to identify the main nameserver(s) for Melbourne University (or another university of your choice), and also identify their mailserver(s). Which DNS resource records are you looking for? Can you see the timers counting down if you make the same query some time apart? Are the references you get back to an actual machine, or an alias? How do you know? ○ For the nameserver you're looking for the SOA record, and additional NS records. Email should have an MX record. You can tell when an actual machine is identified when you see an IP address, as part of an A record, rather than a CNAME.
3	**Guest lectures**: • Reminder: Any particular real-world network systems, technologies, challenges, etc. that you'd like to have a guest lecture on?