Network Management

A one week training course for 20-30 network managers.

Table of Contents

Goal

The goal of this training is to:

- Improve ability of network managers to understand and meet the needs of academic staff?
- Improve staff access to journals and the academic Internet

Aims and Objectives

On completion of this course you will be able to:

- Troubleshoot network problems effectively
- Understand traffic on an Internet connection
- Solve various common network problems
- Monitor and manage the network quality of service
- Prioritise and restrict certain types of traffic
- Maximise uptime by fault mitigation
- Encourage organisational behaviour change
- Design and build a good network, and improve an existing one
- Confidently change and manage network configuration
- Build a social network to support each other

In addition, during the course we will design and build a test network, to experiment and learn with.

Format

This training course is not just a load of powerpoints! The aim is to make it as engaging, enjoyable, practical and rewarding for the participants as possible. Therefore most sessions are planned as either “Questions and Answers”, small group discussions, or practical technical exercises.

Since this is hard work for the facilitators, we strongly recommend that two facilitators are available, and take turns and support each other.

Reducing the scope

If you need to trim this down, to run the course over a shorter period, you might want to consider:

- What makes the participants best able to meet the needs of the academic staff?

- Which exercises am I confident will work well?
- Which are essential to other essential exercises?

Preparation

Equipment

If this equipment is not available, some of the following training plans will need to be modified.

- Powerful desktops for virtualisation (at least 4 GB RAM and 2 NICs, one per group of 4? plus one for the front. with permission to reformat them)
- Cables, switches, preferably some Cisco routers (w/console cables and USB-Serial adaptors)
- Internet connection
- Projector
- A couple of network printers?
- Wireless access points with 802.1x and SNMP support (Cisco 1130/1230 with POE injectors? AIR-AP1121G-E-K9 is extremely cheap)
- Large storage for backups (2TB disk?)
- Paper and pens for all
- Whiteboards (preferably 2) and markers
- Blu-tak, paper, cards, flip charts, markers (lots of) and coloured dots
- Large printed-out example network diagram (3 copies or one laminated)
- Power strips
- UPSes

Host Requirements

- 256 IP address block (/24): 5-9 subnets with 16 IP addresses each
- Isolated network segment (not a shared broadcast domain) over ethernet
- Permission to sniff a live Internet connection
- Access to router graphs and SNMP (read-only community)
- Staff and students to interview about their interactions with IT staff
- Meals at specific times
- Access to the venue during the specified times, including evenings
- Access to the venue beforehand for preparation

Participant Requirements

- Physical campus map/diagram
- Campus network map/diagram
- Current IT policy
- Laptops with wireless (otherwise the wireless ping experiment won't work)
- Traffic graphs from router/internet connection over several days

Venue Setup

- Print some copies of the Vi Quick Reference (perhaps one for every two participants).
- Wall area set up for parking area (topics to revisit later).
- Equipment in space, but NOT connected.
- Download Ubuntu Live DVD, VirtualBox for Ubuntu, CentOS, Mac and Windows and guest additions onto the shared drive or the Class Router/Server. Ensure that participants can find and download them from their computers.

- Set up a DHCP server, and TFTP network install of Ubuntu from a server.
    - Use a small DHCP range, leaving plenty of space for private subnets
- Test that we can make a virtual machine into a router by bridging internal and external nics to the VM.
- Set up an iperf server (for clients to connect to).
- Set up Cacti and configure it to draw traffic graphs from the host's router.
- Prepare router cheat sheets
- Set up a DNS server and domain (`localdomain`) on the Class Router/Server.

Reporting

Reporting requirements currently unknown. I propose:

- Feedback from trainees about their experience of the course and ways that it could be improved, including summaries of outcomes from daily feedback sessions (positives and deltas)
- Feedback from academic staff about perceived changes to network administration, and the direction and size of those changes, after some time (perhaps a month or two).
- Number of tickets filed and time taken to resolve them in that time.

Timetable

The course is planned to run Monday to Friday, for 8 hours a day, including breaks. Days include 4 sessions of 1h 45m, giving a total of 7 hours per day. Longer days might be possible, with longer breaks so that people don't get too tired.

It's expected that the facilitators might want to prepare or rehearse the next day's materials, and participants might want to catch up or experiment, in the evenings, and an evening clinic session is provided for that.

Every day starts with a "morning walk", outside, to enjoy nature (even in the rain), introduce the day's topic, and discuss expectations with participants. Not all their expectations will be met, but it's good to get them out there.

Every day ends with a "feedback" session, where participants tell us what went well about the day, and what they'd like to change in future. This can help the facilitators to refine and adapt the course to the needs of the participants.

Time	Activity
08.00	morning walk
08.30	session 1
10.15	break
10.45	session 2
12.30	lunch break
13.30	session 3
15.15	break
15.45	session 4
17.30	feedback
17.45	end
19.00-22.00	evening clinic and preparation

Monday

08.30: Welcome and Introduction

- Introductions (go round, 15 mins)
- Make name badges (5 mins)
- Identify a shared purpose: common problems that we all face (small group brainstorm exercise and feedback, 20 + 5 mins)
- Define a scope: what is your remit? inside and outside (brainstorm, 15 mins)

09.30: Network Design

Objective: Design and build a good network, and improve an existing one

Presentation (1hr, 09:30-10:30)

- Work through the presentation *In-building Network Design* by Carlos Vicente of NSRC (*PDF <https://nsrc.org/workshops/2009/summer/presentations/day2/layer2-network-design.pdf>*_, OpenOffice or PowerPoint)
- Discuss the slides with the participants
- Aask them to put forward any important points that they want to remember when we come to build the test network, or redesign their campus network.
- Write these notes up on the board.

Break (10:30-11:00)

Capacity needs (30 mins, 11:00-11:30)

As we go through each of these points, participants respond with information about their needs in particular areas, and write down the needs on their campus maps.

- explain and demonstrate the purpose, using an example campus map (3 mins)
- computers (private and shared/labs) (3 mins)
- devices (printers, projectors, IP phones) (3 mins)
- ports (3 mins)
- wireless coverage areas (3 mins)
- number of wireless users (3 mins)
- bandwidth (on-campus and wireless; 3 mins)
- disk storage (3 mins)
- cloud applications (Dropbox, Gmail, Outlook.com, etc.) (3 mins)
- email accounts and storage (2 mins; this goes in the NOC)
- domain/authentication accounts (2 mins)

Network elements (25 mins, 11:30-11:55)

What are they? When would you use them? What types can you get? How much do they cost? What are the limitations?

- Cat 5 and 6 (2 mins)
- Fibre links (3 mins)
- Switches (3 mins)
- Routers (3 mins)
- Wireless access points (5 mins)
- Firewalls (3 mins)
- Caches (web and DNS) (5 mins)

Broadcast domains (15 mins, 11:55-12:10)

- What is broadcast traffic? (2 min)
- What is a broadcast domain? (2 min)
- Why would you have just one? (3 min) - Apparent simplicity - No routers, only one DHCP server required, mobility between zones
- Why would you have more than one? (3 min) - Security, robustness, broadcast storm control, manageability
- How would you connect them together? (3 min) - Subnets and routing

Security needs (10 mins, 12:10-12:20)

- What needs to be partitioned from what? (3 mins)
- Rogue DHCP/RA server containment (2 mins)
- Protect switches and IP phones (3 mins)
- Port security vs dumb devices (3 mins)

Protocol needs (10 mins, 12:20-12:30)

- what things need to or benefit from being on the same broadcast domain? (3 mins)
- who needs to use them? (2 mins)
- can you work around them? (3 mins)
- DHCP, proxy auto detect, Dropbox LAN sync, Microsoft domain browsing and WINS

Lunch break (12:30-13:30)

Physical layer needs (40 mins, 13:30-14:10)

- What are the long-distance connection on your site? (10 mins)
- What kinds of connections can you use? How much do they cost? (10 mins)
- Can you reduce costs by using switches as repeaters and media converters? (5 mins) - where could you put them?
- Tradeoffs: (15 mins) - why would you not run fibre from your core switch to every desktop? (3 mins) - reducing costs (fibre vs switches) (2 mins) - multiplying ports (2 mins) - management complexity (more switches) (3 mins) - redundancy (multiple paths) (2 mins) - reliability (switches that are single points of failure) (2 mins)

Reliability needs (10 mins, 14:10-14:20)

- Redundant paths - where? (3 mins)
- Ring and mesh topologies - where? (3 mins)
- Ports and LACP trunks required - where? (3 mins)
- How much does it cost? (2 mins)

Manageability needs (25 mins, 14:20-14:45)

- How much does management cost? (3 mins)
- Quantify the benefits of:
- Simplicity (fewer devices) (3 mins)
- Fewer types of equipment (3 mins)
- Remote management (3 mins)
- Centralised logging and monitoring (3 mins)
- Fewer topologies (3 mins)
- Transparency (debuggability) (3 mins)
- Eliminating NAT (3 mins)

Hierarchical design building blocks (10 mins, 14:45-14:55)

- Stars (3 mins)
- Separate edge and core (3 mins)
- Connect up and down instead of sideways (3 mins)

VLANs (10 mins, 14:55-15:05)

Advantages and disadvantages:

- remote reconfiguration (2 mins)
- device and cable cost vs. configuration cost (2 mins)
- when NSRC recommend their use (2 mins)
- how many vlans should you have? (2 mins)
- topology recommendations (3 mins) - use subsets of the same topology, not different virtual topologies

Network Access Control (10 mins, 15:05-15:15)

- What is it for? (3 mins)
- What is 802.1x? (3 mins)
- What do you need? (compatible switches and devices, a RADIUS server, certificates) (2 mins)
- What are the alternatives? (port security; advantages and disadvantages) (3 mins)

Break (30 mins, 15:15-15:45)

Requirements for the test network (30 mins, 15:45-16:15)

Requirements gathering for the test network (brainstorm, 25 minutes, essential)

- What resources do we have? Inventory of equipment (group inspects and shouts out, we write it down; 5 mins)
- What are the use cases? What requirements do they create? (5 mins)
    - a shared wireless network for people to use
    - groups of 4
    - to be able to connect a router, some laptops, a server, and an access point at each desk
    - internet access
    - remote access across the lab (IP addresses)
    - practice subnetting, routing, monitoring and filtering traffic
- Split into groups of 4 and discuss how to meet these requirements (5 mins; move around and assist if necessary)
- Each group nominates a member to report back (go round groups, 1 minute each, 10 mins)
- Add any missing requirements: (5 mins, essential)
    - Need one switch per desk
    - Requirements for cables (power and data: length, safety, appearance)
    - Redundancy? Fault tolerance?

Design the test network (1hr, 16:15-17:15)

- What we want you to do: (show a reference diagram on the wall, role play, especially the swapping of participants; 5 mins)
- Split into groups of 4

- Work out a plan that meets the requirements, draw a physical-space network diagram (20 mins; move around and assist if necessary)
- Short break between sessions (5 mins)
- Make sure the diagram is understandable, includes all necessary info to implement (5 mins)
- Two people go to different groups, critique their diagram (5 mins)
- Swap over, the other two go to different groups and do the same thing (5 mins; do we actually need to do this twice, as planned here, so that everyone has a go at critique?)
- Go round, report one thing that you noticed or learned (30 secs each, 10-15 mins total)
- Rejoin and improve your group's diagram if necessary (5 mins)
- Go round the groups, each one quickly explains their changes (2 mins per group, 10-15 mins total)

Feedback (15 mins, 17:15-17:30)

Tuesday

08.30: Network Design, continued

Synthesize (30 mins, 08:30-09:00)

- Synthesize the designs into a single network plan (draw up on a sheet, brainstorm; 15 mins; how likely is this to actually work? main goal is to reach a shared vision/understanding of the network, so everyone can help build it. Need to carefully control time and shepherd)
- Negotiate to remove as many differences as possible from the reference plan.
- "Participatory Budget" (allocate equipment to plan, maybe VLANs; 10-15 mins)

Build the test network (1h 15m, 09:00-10:15)

- Break up into groups of 4 (2 mins)
- Distribute equipment to the right places, connect power and network cables (30 mins)
- Shuffle groups so each has at least one person with Cisco experience (5 mins)
- Prepare to configure the routers (linux/pfsense/vyatta/cisco): put up a cheat sheet, get access to console (10 mins)
- **TODO cisco (or vyatta) router setup cheat sheet**
- Configure the routers and test (20 mins)
- Swap two people with another group and test their configuration (10 mins)
- No VLANs yet!

Break (30 mins, 10:15-10:45)

Redesign your campus networks (1hr, 10:45-11:45)

Note: you might skip this as it's a repeat of the previous practical, but on the participant's own network instead of the test network. however it does introduce useful concepts such as:

- Long distances
- Large scale wireless networks and coverage
- Multi-level hierarchies of connection (versus meshing/horizontal connections)

Work in pairs (Owner and Drawer), choose one of your networks (the Owner), redesign it from scratch:

- Draw a physical building diagram (10 mins), including:
    - physical layout map (approximate, with sizes and distances)

    - peak numbers of cabled and wireless end-user devices in each location
    - wireless access points (position and coverage)
    - bandwidth expectations, with contention ratios
- List requirements as before (15 mins), including:
    - subnets and addressing
    - specific devices (servers, routers and switches)
    - end-user devices attached to each switch and AP
    - lengths of cable runs
    - link types and bandwidths
    - redeployment of existing equipment
    - cost of new equipment.
- Draw a network diagram (10 mins)
- Pair up with another group, check over and critique both designs (10 mins)
- Go round, tell us one thing you've learned (30 seconds each, 15 mins total)

Note: Would be good to discuss how to restructure an existing network, with minimal or planned downtime, but that's not included. (How to make incremental improvements: risks of changing a network; loops, redundancy, IP range changes, multi-homing)

Also, switches and VLAN configuration and testing has been left out, but may be necessary (cheat sheet; 30 mins)

Improving an Existing Network (45 mins, 11:45-12:30)

How do we get from here to there?

- In groups of 4
- Study diagrams of old campus networks
- Look at how to add links without introducing loops, or managing the loops
- List the links to move, servers to move, IPs to change
- Estimate and schedule downtime

Further study

- Practical: implement a DNS server, add reverse DNS for network devices (1 hour; instruction sheet)
- Practical: set up netdot, document our network, locate a given end device - NSRC materials: text

    or PDF.
- Practical: implement a RADIUS server and wired and wireless NAC

Lunch (1hr, 12:30-13:30)

13.30: Troubleshoot network problems

Preparing our servers (30 mins, 13:30-14:00)

- Boot from the network, go through the questions, start installing Ubuntu (20 mins)
- Why are we using Ubuntu for this course? What else could we have used? Pros and cons? (5 mins, brainstorm)
    - It's free
    - You can use it yourselves
    - It has a friendly user interface
    - It's reasonably easy to install software
    - Similar to Debian, which is better for servers but less user-friendly

    - Linux got more votes than FreeBSD at the end of AfNOG 2013 (http://www.ws.afnog.org/afnog2013/sse/survey/2013_exit_survey_results.pdf)
  - How did we do the network installation? PXE, TFTP, HTTP, configuration. Can demonstrate this later. (http://tinderblog.wordpress.com/2009/04/29/ubuntu-live-cdnetwork-boot/)
  - (setup may continue while we talk about virtualisation)

Virtualisation (40 mins, 14:00-14:40)

- What is virtualisation? Run one or more independent virtual computers on a single physical computer (2 min)
- Why is it useful? We want to run clients, servers and routers on the PCs we have available (2 min)
- Why VirtualBox? (2 min; show slide 3 from afnog 2013: http://www.ws.afnog.org/afnog2013/sse/virtualisation/afnog_2013_virtualization_kvm_cw_130610.pdf)
- Which version of VirtualBox do you need? Start downloading it (3 min)
- What is virtualised? What is the virtual hardware? How does it work? (2 min; CPU, memory, disk space, CD-ROM drive, network)
- What else do you need? an ISO image, free disk space. Start downloading the ISO image. (2 min)
- Install VirtualBox, start it up (5 mins)
- How do we create a virtual machine? (10 mins to go through the options and create the first one)
- Install Ubuntu ISO in the virtual machine (10 mins to get started)
- Install Vyatta/pfSense in a virtual machine (**TODO** add time for this)

Introduction to Unix 1 (2 hrs, 15:45-17:45)

- These materials are provided or based on the work of NSRC, please give them credit.
- Show the Introduction to Commands presentation from Afnog 2013 Unix Intro. (1 hr)
- Demonstrate the use of tab completion in the shell on slide 17. (5 min)
- Ask participants to work through the Linux Familiarization and Commands Exercises. (1 hr)

Feedback (15 mins, 17:45-18:00)

Wednesday

13.30: Troubleshoot network problems (continued)

Introduction to Unix 2 (1.5 hrs, 08:30-10:00)

- These materials are provided or based on the work of NSRC, please give them credit.
- Show the Editing presentation from Afnog 2013 Unix Intro. (30 min)
- Ask participants to work through the Editing Exercises - Source can be downloaded in MS Word format if you need to modify this exercise. (1 hr?)
- Network Performance Definitions & Metrics: (presentation, also covers common unix tools, try to fit this in if you can)

Break (30 mins, 10:00-10:30)

IP addresses, subnets and routing: (35 mins, 10:30-11:05)

- Useful materials about IP addresses and network masks.
- What is an IP address? How long is it? (2 mins)
- What is a subnet? What is a network mask? How does the binary representation work? (10 mins)
- What is your IP address and subnet mask? (ifconfig; 2 mins)
- Split into pairs, calculate some subnets (lowest and highest address given an IP address and a netmask) (20 mins)
- What is a default gateway? What are the requirements? (must be on the subnet; why?) (5 mins)
- What is your default gateway? (use the route command; 2 mins)
- How do you do this on Windows? What does the output look like? (4 mins)

Ethernet addresses (20 mins, 11:05-11:25)

- What is an Ethernet address? How long is it? (2 mins)
- What happens when you ping from one computer to another on an Ethernet network? (ARP; 10 mins)
    - Role play "who has 192.168.1.4?" "192.168.1.4 is at second bench, 8th person"
- What networks do and don't have Ethernet addresses? (only 802.3 Ethernet, 802.11 wireless and 802.16; 2 mins)
- What is your Ethernet address? (ifconfig) (2 mins)
- How would you check it on Windows? (2 mins)
- How do you find someone else's Ethernet address (4 mins; explain how "arp -a" and "arping" work; participants get the MAC address of another IP on the network)
- How do you go the other way, from MAC address to IP address? (sniffer, ping scan + arp, DHCP leases; 3 mins)

Duplicate IP addresses and DHCP (15 mins, 11:25-11:40)

- What happens if you have duplicate IP addresses on the network (think about the ARP process; some hosts end up talking to the wrong host and get lost) (2 mins)
- How do you keep track of who has what IP address? (2 mins)
- How do you configure machines automatically? (DHCP)
- In pairs, create a duplicate IP address on the network, and list both the MAC addresses (10 mins)

Traceroute (15 mins, 11:40-11:55)

- How do we use Traceroute? What destination do we use? What does it tell us? (10 mins)
- Try a visual traceroute to www.ischool.zm on http://en.dnstools.ch/visual-traceroute.html (2 mins)
- Which hops are out of place? (3 mins)
- How does traceroute work? (15 mins)
    - set up 5 people as routers, one as the sender, one as the destination, one as a packet
    - the packet carries a header: a card with a destination address and a TTL written on it
    - every router needs to decide whether to forward or drop the packet
    - if it forwards, it must decrease the TTL by one
    - if it drops, it should send a time-exceeded packet back to the source
    - what TTL does a packet need to get through?
    - what happens if we send two identical packets with different (short) TTLs? (we get two time-exceeded messages back, with different sources)
    - how do we distinguish the replies? (UDP destination port numbers)

    - what is the effect of a router filtering out ICMP packets?
    - so how does traceroute actually work?

Troubleshooting network connectivity (20 mins, 11:55-12:15)

- Imagine you can't ping an address on the internet.
- What steps does the packet have to go through? (5 mins)
- What could go wrong along the way? (5 mins)
- How would you identify each problem? (5 mins)
- How could you eliminate a whole bunch of them at the same time? (2 mins)
    - try another test to the same destination over a slightly different route;
    - try pinging a different address
- What else could cause "ping www.google.com" to fail? (3 mins)
    - DNS (try pinging a well-known IP address such as www.google.com)
    - Responses take over 4 seconds (on Windows)

Good troubleshooting technique 1 (15 mins, 12:15-12:30)

You can use some slides from the Solving Network Problems Presentation.

- What is troubleshooting? (2 mins)
    - Who knows Sherlock Holmes? What would he say about solving difficult problems?
    - Identify the problem
    - By manual, logical deduction
    - To help us fix it
- What is troubleshooting NOT? (5 mins)
    - monitoring (how do they differ? monitoring is ongoing, provides baseline and change data to assist troubleshooting)
    - management (is planned; troubleshooting is for emergencies)
- What do we have to do?
    - Respond to a problem (1 min)
    - Identify possible causes (1 min)
    - Eliminate causes (1 min)

Lunch Break (1hr, 12:30-13:30)

Good troubleshooting technique (25 mins, 13:30-13:55)

- How do we respond to a problem? (brainstorm, 5 mins)
    - Don't panic (because we'll make mistakes and forget what we've changed)
    - Understand the problem
    - Reproduce it
    - Find a quick test (why? because we'll need to check many times to see if we've fixed it)
    - Find a workaround (why? to help people work until the problem is fixed, and take some pressure off you)
- How do we identify and eliminate possible causes? (15 mins)
    - You can't print to a particular printer: your job disappears.
    - What are the possible causes? There are a lot!
    - List the chain of events that happens when you click the Print button.

- How do we eliminate possible causes?
    - What was the last thing changed? (reverting it may be a good candidate for solving the problem quickly) (1 min)
    - What slight changes can we make to the quick test? (5 mins)
        - print to a different printer
        - print a different file, for example a test page
        - ping a different IP address or hostname
        - run the same test on a different computer
        - what do these tell you?
    - Is it plugged in? (follow the chain physically, checking that equipment is on and links are up)
    - Make a backup (be able to quickly undo your changes)
    - What is “known good”? (Try swapping devices/links, one at a time, with known good/working ones)
    - Change one variable at a time (and make notes)
    - Do no harm (make sure you can quickly undo whatever you’ve done, and it won’t cause bigger problems later)

Troubleshoot network problems

Simulating Problems (30 mins, 13:55-14:25)

- Want to practice solving some problems in the lab
- What can we simulate?
    - Look at the list of problems from day 1, and brainstorm ways to simulate them

Building a virtual network (1h 15m, 14:25-15:40)

- This is optional, only need to do it if people want to simulate network problems (but that’s quite likely)
- We may need to cover VLANs first, if we don’t have a second NIC in the servers (1 hour extra?)
- Make the virtual machine into a virtual router: (1h 15m)
    - Plenty of network debugging practice here!
    - Split into groups as necessary to configure one machine each
    - Add a second NIC to the virtual machine
    - Bridge the internal and external physical NICs with virtual NICs 1 and 2
    - Statically assign external IP address to the
    - Configure a subnet on the internal NIC (should have some small subnets ready)
    - Install and configure a DHCP server
    - Connect the internal NIC to the switch, external to upstream
    - Configure the upstream router to route the subnet back via the virtual router

Break (30 mins, 15:40-15:55)

Simulating latency and packet loss (45 mins, 15:55-16:40)

- What is latency? (5 mins)
- Measuring latency (5 mins)
    - What are we measuring? Round trip time.
- tc commands to add latency to an interface: (5 mins)

    - tc qdisc add dev eth0 root netem delay 97ms
    - tc -s qdisc
    - tc qdisc del dev eth0 root
- tc commands to add packet loss to an interface: (5 mins)
    - tc qdisc add dev eth0 root netem loss random 50%
- what does it feel like? (15 mins)
    - take in turns to go round the group
    - one person sets the router for packet loss, latency, both or neither
    - the others try to work out (guess or measure) which it's set for
    - what does it feel like? how does it affect page loading? (slow loading, randomness)
- what else can we simulate with netem? (5 mins)
    - See the netem manual for details.
- How would you make it permanent (5 mins)

Simulating limited bandwidth (1 hr, 16:40-17:40)

- Why would we want to do this? (5 mins) - So we can see how much bandwidth we need to load pages - See the effects of different bandwidth settings - Need to restrict bandwidth to own the queue, for traffic engineering
- Measuring bandwidth (5 mins)
    - what are we actually measuring?
        - available capacity, not total capacity
    - speedtest (5 mins)
        - requires a desktop computer (not automatable)
        - allows testing in both directions
        - no control over test period or bandwidth used
        - sometimes fails over slow and unreliable links
    - iperf (5 mins)
        - requires a server
        - allows testing in both directions
        - exact control of test period and bandwidth used
    - wget (5 mins)
        - no server required
        - only tests in one direction
    - abget (15 mins)
        - no server required
        - can test in both directions
        - fiddly to install, doesn't work in all cases
- What happens if we all test at the same time? (5 mins)
    - Why did this not happen with ping?
- tc commands to limit bandwidth on an interface: (5 mins)
    - tc qdisc add dev eth0 root netem rate 256kbit
- Measure the results; is it what you expected? (15 mins)

Feedback (15 mins, 17:40-17:55)

Thursday

Troubleshoot network problems (continued)

Competition with other users (15 mins, 08:30-08:45)

- Only makes sense with limited bandwidth
- Use ab to generate concurrent streams: (5 mins)
    - ab -c 2 -n 10000 http://192.168.0.1/largefile
- What effect does it have? (5 mins)
- What happens when you vary the number of concurrent streams (ab -c parameter)? (5 mins)

Solving common problems (1 hr, 08:45-09:45)

How would you attack them? What would you look for? Can you reproduce it on our test network?

Look at the problems reported on the first day, and analyse a few of them (about 15 mins each) or pick some from this list:

- Pages don't load at all (IP and DNS settings, firewall; 15 mins)
- Slow access to Google (HTTP and DNS speeds)
- Slow downloads of academic journal articles
- Creating, tracking down and stopping a rogue DHCP server (practical)
- What data do you need to argue successfully for more bandwidth?

Leave out questions about traffic engineering and monitoring for now, as we hope to cover these later:

- Monitoring your Internet connection (packet loss, latency, throughput, queues)
- Bandwidth use from facebook, entertainment sites (traffic shaping)
- Restricting bandwidth use by certain websites (squid delay pools, dansguardian)
- Time-based restrictions (iptables, squid, dansguardian)
- Blocking websites (iptables firewalls, squid, dansguardian)
- Blocking traffic based on keywords (dansguardian)

Further study

- Add DNS performance testing, flood pings, ssh.

14.45: Network capacity planning

Why do we need to plan capacity? (10 min, 09:45-09:55)

- What happens as a road becomes more full? (2 min)
    - How bad is the congestion problem? Do we need to deal with it at all?
- To avoid congestion, we need to reduce (peak) demand, or increase supply (2 min)
- Increasing supply is expensive, so you need a good argument (2 min)
- Which probably means that you need to manage demand first (bandwidth management) (2 min)
- And how you do this will determine how much bandwidth you need (2 min)

What's going to change in future? (25 min, 09:55-10:20)

- More devices connected? How many and when? (5 min)
- More wireless clients? How much have they grown? (5 min)
- More crowded wireless networks? How much more? (2 min)
- Average size of web page continues to increase (http://httparchive.org; 600 to 1500 kb in 3 years is about 35% per year) (2 min)
- Increasing demand for cloud applications (dropbox, facebook, gmail, google docs, outlook.com) (2 min)
    - Dropbox grew from 50 to 100 million users in just over 1 year (https://en.wikipedia.org/wiki/Dropbox_(service)#History)
    - Has your demand for cloud services been doubling every year?

Break (30 mins, 10:20-10:50)

How much local traffic do you use? (30 mins. 10:50-11:20)

- What services do you run locally? (go round, 10 mins, keep a tally)
- How much bandwidth is available to end users? (10 mins)
    - Where are the bottlenecks?
    - In groups of 4, study your network diagrams
    - identify the local bandwidth (not internet) available to an end user at different points
    - and how many users there are at each place
    - and what the loading time would be for a local 1 MB page/email to a user in that place
    - go round the groups and report back (5 mins)
- If you don't know, how would you find out? (2 mins)

Wireless capacity planning (30 min, 11:20-11:50)

- How much bandwidth is available from each AP? (2 min)
    - up to 100 Mbps with 802.11n
- How is this shared between users? (2 min)
    - unfairly, and there's not much you can do about it; WMM + QOS may help a bit.
- How do users choose which AP to associate with? (2 min)
    - usually the strongest signal, not the least busy
- Can you move users onto a less busy access point? (5 min)
    - usually not;
    - Cisco WISM has a "Client Load Balancing" option that may enable this, but does nothing about interference
    - you can drop the power, makes clients less likely to choose that AP in the long run and reduces interference
    - especially marginal clients at the edge of signal, which cause the most interference to other APs!
- How many separate frequency bands available at 2.4 GHz (5 min) - 3 channels: 1, 6 and 11 - Interference and 802.11n make this worse - Ensure that clients can't see multiple APs on the same

  channel anywhere, especially yours!

    - Drop power if necessary to achieve this

- How many clients can associate with an AP? (2 min)
    - Cisco recommends “Ideally, not more than 24 clients can associate with the AP” but supports up to 2048
- How to increase wireless capacity: (10 min; brainstorm)
    - More access points
    - Better management (e.g. Cisco WCS/NCS)
    - Lower power
    - Non-conflicting frequencies
    - Move everyone possible onto 802.11a
        - Have a separate SSID for 802.11a so that people who switch to it will stay there
    - Identify wireless bandwidth hogs
    - Identify stations with high probe and retransmission rates, probably have marginal signal

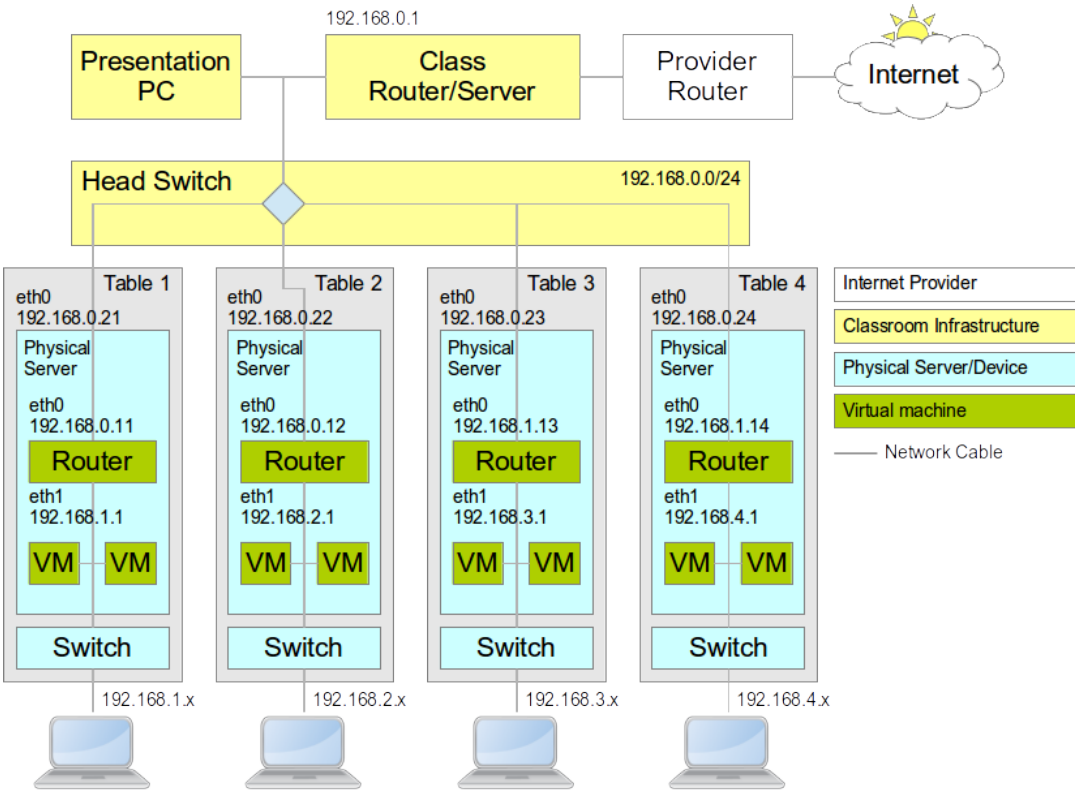
What will it cost? (10 mins, 11:50-12:00)

How will you budget for it? How much do you need to spend every year? And on what?

- 802.11g to 802.11n means replacing all APs (again!) for a 100% increase in bandwidth (4 min)
    - Every 3 years?
    - How much will that cost?
- 100baseT to 1GbaseT means replacing all switches and cables for 10x increase (4 min)
    - 10baseT released in 1990
    - 100baseTX in 1995
    - 1000baseT in 1999
    - 10GBASE-T in 2006
    - 802.3ba (100 Gbit/s Ethernet) in 2010
    - Every 5 years?
    - How much will that cost?
    - https://en.wikipedia.org/wiki/IEEE_802.3
- Bandwidth costs (peering, in the USA) have decreased at 64% per year over 17 years (2 min)
    - http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php

Network Monitoring

TODO finish and time this section



Example network diagram, repeated for easy reference

- What can we monitor? - use slides from Unit 7 Presentation
- Monitoring service availability, connectivity, bandwidth, latency, packet loss
- Install and configure Nagios to alert you of network problems (PDF, OpenOffice or Powerpoint) - Basic exercise: Install Nagios, add a host and a service, email notification - Medium exercise: Define host and service groups and check commands, SMS notification - Advanced exercise: Writing your own checks, bandwidth check, NRPE
- Monitoring Nagios on your desktop and phone
- Install and configure Cacti to monitor switch ports and routers
- Importance of network documentation - Practical using Netdot
    - Example network diagram
    - Keeping documentation
        - Swap over and use someone else's setup half way through
        - Reverse DNS, public whois data and abuse contacts
    - Network monitoring
        - nagios (fault https://github.com/aptivate/inaspmaterials/blob/master/src/Network_Management/Unit_4_Netwo
        - early https://github.com/aptivate/inaspmaterials/blob/master/src/Network_Management/Unit_4_Netwo
        - smokeping (early detection of packet loss and latency issues: query the RRD using Nagios for alerting?)
        - network traffic logging/forensics using pmacct
        - ddos detection using pmacct
            - average UDP packet size
            - number of unreplied UDP packets
                - pre_tag_map: set_tag = 3 filter = 'udp'
                - pre_tag_filter[ddos_unreplied_udp]: 3

        - aggregate[ddos_unreplied_udp]: src_host, src_port, dst_host, dst_port
        - SELECT src_host, COUNT(*) AS unreplied_udps FROM ddos_unreplied_udp AS out LEFT JOIN ddos_unreplied_udp AS return ON return.ip_src = out.ip_dst AND return.port_src = out.port_dst AND return.ip_dst = out.ip_src AND return.port_dst = out.port_src WHERE return.ip_src IS NULL;
    - number of outbound connections
        - pre_tag_map: set_tag = 5 filter = 'tcp[13] = 2'
        - pre_tag_filter[ddos_syn_packets]: 5
        - aggregate[ddos_syn_packets]: src_host
        - SELECT src_host, SUM(packets) AS syn_packets FROM ddos_syn_packets WHERE stamp_inserted > date_sub(now(), interval 4 hours) HAVING syn_packets > 10000;
    - number of remote hosts contacted (unique destination IP addresses)
        - aggregate[ddos_remote_hosts]: src_host, dst_host
        - SELECT src_host, COUNT(DISTINCT dst_host) AS dst_hosts FROM ddos_remotehosts WHERE stamp_inserted > date_sub(now(), interval 4 hours) AND
    - total volume of outbound traffic, and per destination host
        - aggregate[ddos_total_traffic]: src_host
        - aggregate[ddos_src_dst]: src_host, dst_host
    - monitoring and alerting using Nagios
- interface traffic levels: cacti
- real-time anomaly detetcion: https://github.com/etsy/skyline

Network Management

TODO finish and time this section.

- Network access control
    - Enabling port security and tracking users by MAC address
    - Build a RADIUS-compatible directory
        - Password strength, change and expiration policies
        - Importing users in bulk
    - Enable 802.1x authentication on wireless networks
    - Enable 802.1x authentication on wired ports
    - Captive portals for guests (with tickets) and self-registration
    - Integrating service authentication with the directory
        - Web servers, file servers
- Firewalls
    - general security principles: block by default, declared servers, DMZ and isolation between departments
    - firewalling individual machines (defence in depth/crunchy armadillo theory)
    - firewalls and web caches: forcing people to use the cache
    - limitations: layer 7/deep packet inspection (+DNS, HTTP proxies), website hostnames/IP addresses
    - stateless and stateful firewalls
    - iptables firewalls

        - maybe vyatta firewall? http://carbonwind.net/VyattaOFR/Firewall/Firewall.htm
    - Bandwidth self-management
        - Implementing usage reporting with pmacct
        - Implementing quotas with pmacct and iptables/vyatta
    - Antivirus
        - Build an update distribution system for AVG?
            - Need to install Windows virtual machines for this, may take a long time
    - Proxy caches
        - Internal DNS resolver/cache
        - Internal HTTP proxy/cache (Squid)
        - Blocking websites with Squid
        - Content filtering with Dans Guardian
        - Caching updates (Windows and antivirus)
    - Intrusion Detection
    - Penetration/scanning (Nessus, Inprotect, NMap)
    -

17.00: Understand traffic on an Internet connection

TODO finish this section

Theoretical and actual capacity (35 mins, 17:00-17:35)

- Why would they be different? (5 mins)
- Useful illustration
- Contention and competition (5 mins)
    - Contention is overselling of the same bandwidth, on the basis that most people won't use it most of the time.
        - How realistic is that? How heavily is your connection used?
        - Contention is a ratio, fixed by the ISP (e.g. in SLA or contract)
    - Competition is the amount of traffic already being used
        - Depends how heavily the other users are actually using their connections
- Bandwidth management by ISP (25 mins)
    - How would you detect it? (10 mins)
        - "Weird" behaviour of a connection (unusual, inconsistent)
        - Unusual: Connections reset by peer
        - Inconsistent: Slow downloads when ping times are fast
        - Inconsistent: Downloads start quickly and then slow down
        - Unusual: Some protocols blocked completely
        - Unusual: Available bandwidth drops sharply and stays down
    - Why is it a problem? (10 mins)
        - The ISP's policy is being imposed on you, and may not match yours
        - Traffic management is impossible unless you know how much bandwidth you have
            - You need queues to be on your router, which only happens if the bottleneck is there

        - ISP changing bandwidth under your feet takes that power away from you
    - It's no good having "academic freedom" if you can't exercise it
    - Why is this funny? We're complaining about being bandwidth managed, and about to do the same to our users!
- How would you deal with it? (5 mins)

    - Pretty much nothing you can do, except complain, negotiate or switch ISP!

How much bandwidth do you need? (5 mins, 08:30-08:35)

- What does it take to ensure that web pages load quickly? (2 mins)
- Bandwidth required to load a
- How many users do you have? (2 mins)
- What contention ratio do you want to offer? (2 mins)
- Also: fast, reliable DNS service and local bandwidth (think about wireless)

What's going on on your connection? (15 mins, 08:35-08:50)

- Participants who brought traffic graphs, please show them (5 mins) - Otherwise you can use the ones from Unit 7
- Can we identify any features? (5 mins)

    - Times of peak usage and little usage
    - Flat tops - connection saturated
    - Times when available (unused) bandwidth is over 1 Mbps, over 10 Mbps
- Who are the heaviest users? (2 mins)
- What are the heaviest uses of bandwidth? (2 mins)
- How much do they use? What percentage do they represent?

Further study

- Link speed, encapsulation and overhead
- MTU and MSS
- Queueing, packet loss and effects on performance throughput
- Monitoring switch ports and routers (practical with Cacti, SNMP; 1 hour)
- Network monitoring theory (TCP/IP, packets, connections, flows; 1 hour)
- Packet-level monitoring practical (sniff on teh router, generate some traffic, analyse with wireshark, swap with another group and analyse theirs, repeat if possible; 1 hour)
- Network traffic monitoring (transparent bridging, tapping, netflow; 30 mins)
- Real network monitoring (tap the university network, capture traffic, analyse and identify features, see how many you find compared to us; 2 hours)
- Classifying traffic by hand (what is significant in the trace captured above? 1 hour)
- Automated classification (iptables, argus, pmacct and snort; skip this?)
- Wireless ping experiment (1 hour)
- Bottlenecks, queues and latency (theory, practical: measure queue length with cross-traffic; 1 hour)
- Tracking down a rogue DHCP server

Tracking down an IP address (30 mins, 08:50-09:20)

- everyone writes up their used IP addresses on the board (10 mins)
- choose an IP address used by another group
- track down the physical machine (20 mins)
- if they struggle, explain how they can log into switches and look at the forwarding tables to see which MAC addresses are on which port

Bandwidth/traffic management (10 mins, 09:20-09:30)

If you want to ensure that (some) road journeys are fast, what can you do?

- Give some vehicles priority over others (e.g. emergency services)
- Keep one lane clear for priority vehicles
- Limit the number and length of car journeys
- Efficiency savings: reduce the need for car journeys (public transport, local markets and supermarkets)
- Make better use of unused capacity: encourage spreading of load into off-peak periods
- Increase the cost of petrol, or charge tolls
- Arrest people for driving slowly
- We'll come back to all of these in more detail (**TODO**)

Give some vehicles/packets priority over others (10 mins, 09:30-09:40)

- What happens when the police try to drive through a traffic jam?
- Does it help? How much?
- Why does the same apply to network connections?
    - Imagine there's a packet in the queue, being transmitted
    - Have to wait for current packet to get out of the way
    - How long does it take to send a 1500 byte packet at 1 Mbps?
    - 1500/(10^6/8) = up to 12 ms added latency (jitter)
    - Not bad, but still affects voip to some extent
- You must control the queue!
- You usually don't control the incoming queue (from your ISP)

Reserve capacity for some types of traffic (20 mins, 09:40-10:00)

- On the road: have a special lane for emergency vehicles
- On the web: reserve some bandwidth
    - one class/queue for "light" web browsing
    - one class/queue for "heavy" downloads and other traffic
    - may need others, for example voip
- How much would you need to reserve?
    - 10 Mbit (fast web browsing)?
    - 1 Mbit (acceptable speed web browsing)?
    - Per 20 computers? Per 50?
- What effect does this have on the remaining traffic?
- How efficient is this solution?
    - Could allow other traffic to use reserved bandwidth, but with a lower priority than web traffic
    - https://raw.github.com/aptivate/inaspmaterials/master/src/Network_Management/Unit_11_Technical_
- How do you ensure that only fast traffic uses the fast lane?
    - There's a limit to what we can detect and classify, in CPU time and in software capabilities
    - It's very hard to distinguish between web page accesses and downloads.
    - One way is the amount of data transferred: iptables -m connbytes
- Still need to control the queue!

Limit the amount of traffic each user can cause (10 mins, 10:00-10:10)

- Per-user quotas or bandwidth restrictions

- Ensures that users downloading heavily will only affect themselves
- Implement quotas using pmacct, argus or nfsen to track bytes transferred
- When user goes over quota:
    - Name and shame them (peer pressure)
    - Contact them (out of band management)
    - Reduce their bandwidth allocation (punishment and/or protecting the other users*)
    - Deprioritise their traffic*
    - Block them completely
    - (*) Still need to control the queue for these measures
- Self-managing
    - Provided that quota is low enough, users will self-police and keep bandwidth available most of the time
- Some users have legitimate reasons to make large downloads
    - Maybe operate quota only at peak times?
    - And/or help users to download in the background?
    - Provide a low-priority download service that doesn't count towards their quota
    - Use TOS flags to catch some downloads early

Increase the cost of use (5 mins, 10:10-10:15)

- Road analogy: increase the cost of petrol
- What are the effects?
- How would students react? Who can afford to pay?
- How would you implement it? (use the same tools as for quotas)

Further study

Look at the other alternatives above.

Break (30 mins, 10:15-10:45)

Network quality of service

TODO: plan this unit in more detail, with activities and timings

Monitor and manage the network quality of service

- Monitoring network connection quality with Smokeping: PDF, OpenOffice or PowerPoint, exercises.
- Monitoring server performance and capacity with Munin
- Build a VoIP network with Asterisk
- Measure effects of cross-traffic on call quality

What do we want to monitor?

- Statistics - Packet Loss - Latency - Bandwidth - Errors - Web page loading speed - DNS speed
- What do you want to know? - How are they right now? (is something wrong?) - How do they compare to last week/month/year? (is it getting better/worse?) - Were they bad this morning/yesterday? (why did I get so many complains) - Are they worse at certain times of day? (is something unexpected happening?)

Round Robin Databases

- **How would you collect this?**
    - Ping every 5 minutes?
    - Store for a year?
- **How much data do you need to store?**
    - 20 samples x (60/5) per hour x 24 hrs x 365 days? (2.1 million samples)
    - How are you going to display it?
    - How will you discard old data from the database?
- **Solution: Round Robin Database (RRD)**
    - Keep a certain number of data points
    - Automatically overwrite old ones
    - **Automatically maintain aggregates (minimum, maximum, average) over different time periods**
    - Draws pretty graphs

Smokeping example graph

http://oss.oetiker.ch/smokeping/doc/reading_detail.png

How to read it.

Maximise uptime by fault mitigation

TODO: plan this unit in more detail, with activities and timings

- What creates downtime?
- How long is the downtime? (detection + diagnosis + repair + restoration)
- How can we detect faults quickly, especially if they recur?
- How can we diagnose common faults more quickly?
- How can we repair (patch) them quickly?
- Can we create alternative/backup systems? What kinds are there?
- Can we quickly switch over to a backup system?
- Can we offer users a self-help backup option? (pool computers, backup connections, pool printers)
- Practical: building a redundant Ethernet link (switches, LACP, STP)
- Practical: building a redundant router (CARP/VRRP)
- Practical: building a redundant Internet connection (policy routing and metrics, what happens when you switch over?)
- Practical: load shedding (putting some network traffic onto a backup connection)
- How can we prevent them from happening again? (belt and braces approach)
- Scheduling downtime, keeping users informed

Web Proxies and Caches (30 mins, 10:15-10:45)

Introduction (25 mins, 10:45-11:10)

- What is a web proxy? (2 mins)
- Forward, reverse and open proxies (3 mins)
- Why use web proxies? (5 mins)
- What is a web cache? (3 mins)
- Why use web caches? (2 mins)
- Why not to use web caches? (1 min)
- Not transparent (2 mins)
- Effectiveness is falling (2 mins)

- Hardware requirements (2 mins)
- Single point of failure (2 mins)

Getting started with Squid (40 mins, 11:10-11:50)

- Basic installation (10 mins)
- Configuring your browser (2 mins)
- Testing the installation (5 mins)
- Access control by IP address (5 mins)
- Why do you deny me? (2 mins)
- Reading the logs (5 mins)
- Don't deny me! (5 mins)
- Reloading and restarting Squid (3 mins)
- Reverse proxies and open proxies (2 mins)

Cache Size (6 mins, 11:50-11:55)

- Introduction (2 mins)
- Disk cache size (2 mins)
- Memory usage (2 mins)

Squid Access Control (1 hr, 11:55-12:55)

- Introduction (2 mins)
- Access control elements (5 mins)
- ACE types (5 mins)
- The `srcdomain` ACE: a special case (2 mins)
- ACEs with multiple values (3 mins)
- Access control rules (3 mins)
- Rules with multiple ACEs (3 mins)
- Rule processing examples (5 mins)
- Access control practice (30 mins)

Lunch Break (1hr, 12:55-13:55)

Web Proxies and SSL (10 mins, 13:155-14:05)

- The Problem (5 mins)
- What can we do about it? (3 mins)
- HTTP and CONNECT requests (2 mins)
- Results of blocking SSL requests (5 mins)

Forcing people to use the proxy (45 mins, 14:05-14:50)

- Network setup to enable pfSense (10 mins)
- Firewalling with pfSense (10 mins)
- Proxy auto configuration (3 mins)
- Creating a PAC file (10 mins)
- DHCP server settings in pfSense (5 mins)
- Testing Proxy Auto Configuration (5 mins)

Proxy Authentication (50 mins, 15:50-15:40)

- Introduction (2 mins)
- About RADIUS (5 mins)

- Setting up a RADIUS Server (2 mins)
- Installing FreeRADIUS on pfSense (5 mins)
- Configuring FreeRADIUS (10 mins)
- Adding Users (5 mins)
- Testing RADIUS Authentication (10 mins)
- Squid RADIUS Authentication (10 mins)

Break (30 mins, 15:40-16:10)

Squid Delay Pools (50 mins, 16:10-17:00)

- Introduction (10 mins)
- Classes of delay pools (10 mins)
- Request routing (5 mins)
- Limitations of pools (5 mins)
- Simple example (10 mins)
- More advanced configuration (10 mins)
- Questions (10 mins)

Friday

08.30: Encouraging organisational behaviour change

Explore some interactions with staff or students (1 hr, 08:30-09:30)

- We're going to try out what happens when we interact with our users
- Rather than put people on the spot, we'll go round volunteering a difficult (but real) question/interaction with a user.
- The people interacting are anonymised, for example "a sysadmin" and "a student"
- The participant gives us the context and question/first line of the interaction (1 min x 30 ppl)
- Write these up on the board
- Then we choose 5 to investigate in more detail: (5 min x 5 = 25 mins)
- The group offers possible answers/comebacks (1 mins)
- The particiant gives us the actual answer/comeback (1 min)
- Ask the group to answer these questions by brainstorming:
- How does the user feel after this interaction? (1 min)
- How could the interaction have gone better? (1 min)

How do you interact with users? (1 hr, 09:30-10:30)

- Split up into groups of 4
- Discuss/brainstorm on these questions (20 mins):
- Choose someone to feed back to the group
- What do users want?
- What are they frustrated about with OUR service/support?
- What can we do to give better service (NOT just fixing the immediate problem!)
- Move around the groups checking that they're making good progress.
- Groups feed back their results (3 mins x 8 groups = 24 mins)
- Write these down (probably 2 ideas x 8 groups = 16 ideas)
- Pick some interesting ones and discuss how to implement them (5 ideas x 3 mins)
- Ensure that ways of reducing downtime are discussed:

    - responding quickly
    - providing alternatives

- being proactive to prevent failures
- keeping users informed

Break (10:30-11:00)

Do you know what users want? (1 hr, 11:00-12:00)

Let's go talk to some!

- Have prearranged interviews with approx 15 people for this time slot
- Split up into pairs and go interview people (35 mins)
- Ask them what frustrates them about their IT department, listen and make notes
- Propose ideas for ways to improve the service that they receive, get their feedback
- Bring back a report of what you learned about these users
- Report back: who you interviewed and what you learned (1 min x 15 groups)
- Brainstorm: any other ideas about how to meet the needs of these users? (5 min)
- Get people to write notes on a lopad

How to affect user behaviour (25 mins, 12:00-12:25)

- Introduce supply and demand (5 mins)
- Introduce the Tragedy of the Commons (5 mins)
- What kinds of resources in our organisations does this apply to? (brainstorm, 5 mins)
- Expected answers include: bandwidth, shared computers and printers, any limited resource (not easy to obtain more of)
- Imagine that we had not enough computers in this classroom. What would you do to ensure that everyone gets fair access to them? (5 mins)
- We've come up with some rules. What is this called? (policy). What is a policy? (discuss; 5 mins total)
    - How is it different from strategy?
    - How is it different from regulations?
    - Who creates and owns it? (show slides from INASP PDW unit 2, slides 5-6)

Lunch break (1 hr, 12:30-13:30)

Alternatives to policy (15 mins, 13:30-13:45)

What are the alternatives to creating a policy? What advantages and disadvantages do they have?

- fair access vs legitimate access: 5 minutes each, or only educational sites? how to enforce? bring up the discrimination game?
- charging: highly effective, can fund improvements, highly damaging to educational and research objectives;
- make use cases faster: many small tweaks on internet connections (DNS and caching), improve speed and experience, limited benefits
- remove inefficiencies: viruses, worms and spams, reload/fix printer faster; encourage/enable printing at different times; spread out dissertation printing.

What makes a good policy? (15 mins, 13:30-13:45)

Brainstorm; use the slides from PDW Unit 2 as prompts if necessary.

- Who needs to be involved in making policy? (5 mins)
- What encourages people to actively support the policy and follow it?
- **Purpose**: Linked to a wider objective (e.g. What is the institutional objective? How does the policy support it? Why is this the best way to support it?)

- **Consultation**: Developed by a good process (everyone is consulted, objections taken into account or refused with good reasons clearly stated, improves buy-in and sense of ownership; subject to regular review)
- **Benefit**: People support it, can see how it benefits them (defined positive purpose: e.g. “everyone can use the internet for academic purposes and it will work”)
- **Clarity**: Communicated and understandable (policy summary is clearly visible to users, well written, no technical or legal jargon, short, easily read and digested by users)
- **Authority**: Clear ownership (e.g. the Vice Chancellor, not the IT department) gives it authority and allows it to be enforced.
- **Flexible**: E.g. “the IT department is allowed to define rules for access to printers to ensure fair use” and not “printers are only to be used for 5 minutes per day”
- How can we help our university and our department? (negotiating policy, being proactive)

How good is your policy? (1 hr, 13:45-14:45)

- Examine participants’ policies (15 mins)
    - To what extent are they enforced? Is that a good thing?
    - To what extent are they kept up to date?
    - How would you change/improve them?
- Work in pairs to suggest improvements to a policy (15 mins)
    - Use your own, or one of the examples in Unit 2.
- Look at the problem scenarios again: how is this supposed to be resolved? can policy help? (15 mins)
- Practical: in pairs, negotiate a policy with each other (15 mins)

Request Management (15 mins, 13:45-14:00)

Help desks and ticketing systems

- How can we help users to help themselves and bother us less? (brainstorm, 5 mins)
    - training to help themselves
    - training to diagnose some faults
    - encourage “local experts” in departments as a first line of response/diagnosis
    - provide tools and resources for people to help themselves (e.g. paper)
    - provide a forum for discussion, asking and answering questions
- How do users report problems right now? How could it be better?
    - (training, self-help services? brainstorm; 5 mins)
- Who knows about Stack Overflow/Server Fault? How does it work? What’s it good for? (brainstorm, 5 mins)
    - users submit questions
    - users try to answer other peoples’ questions
    - get points for correct answers and upvoted ones
    - prompting if creating a question similar to an existing one
    - helps people to find answers themselves, without waiting/queueing
    - reduces load on IT support staff
- Practical: install Askbot (**todo finish this**)
    - Have a look at the website in pairs, try to work out what’s needed, shout out ideas/commands/packages (brainstorm, 10 mins)
    - sudo apt-get install python-setuptools python-mysqldb mysql-server
    - sudo easy_install pip virtualenv askbot
    - cd /var
    - sudo mkdir -p django/askbot1

- cd django/askbot1
- sudo askbot-setup
- sudo python manage.py collectstatic
- sudo telinit q
- sudo service mysql start
- mysqladmin -u root -psecret create askbot
- mysql -u root -psecret
    - grant all on askbot.* to askbot@localhost identified by “secret”
- sudo python manage.py syncdb

How can we be effective, help and support each other? (15 mins)

Brainstorm for ideas about forming a support network:

- Why would we do it?
- How much time can we allocate?
- What can we get funding for?
- What works well? - Exchanges? - Visits? - Emails? - Online forum? - Conference call?
- How can we ensure that it happens?
- How can we involve new staff as they arrive?

Confidently change and manage network configuration

TODO: plan this unit in more detail, with activities and timings

- What is the configuration? How big is it?
- Practical: set up RANCID to download and version configurations - PDF
- Practical: make a change to another group’s router, then swap and they have to identify what it was and reverse it
- Practical: set up network topology monitoring, swap with another group, make a change to their topology, swap back, identify and undo the change, feed back
- Practical: backup and restore a Unix virtual machine with duplicity
- Practical: backup and restore a Windows virtual machine with system imaging