NET

Networking

Goals of this lab:

- To learn how to configure network connections in a UNIX system.
- To gain experience with fundamental routing.

Prerequisites: LXB

Table of Contents

PRELAB

Exercise 1: Review and preparation

1-1 Review the documentation for the `ping` utility and answer the following questions:

(a) What is the `ping` utility used for?

(b) How does the `ping` utility work?

(c) How can you detect if the host 10.17.1.1 is up?

(d) If you want to send a ping to all computers on a subnet, how can you do it using the `ping` utility?

(e) What would happen if you could ping the address 255.255.255.255?

(f) What command-line option to `ping` causes it to print numerical addresses?

1-2 Review the documentation for the `traceroute` utility and answer the following questions:

(a) What is the `traceroute` utility used for?

(b) How does `traceroute` work?

(c) What command-line option causes `traceroute` to print numerical addresses?

(d) What is the third-to-last hop on the route to 150.203.99.8?

1-3 Review the documentation for the `ifconfig`, `route,` `netstat` and `ip` commands and answer the following questions:

(a) How do you set the address of interface eth0 to 130.236.189.14/24 (netmask 255.255.255.0) and broadcast address 130.236.189.255 using `ifconfig`? How do you set it using `ip`?

(b) How do you display the current routing table using `route`? How do you display it using `ip`? Using `netstat`?

1-4 Review the documentation for the `sysctl` command and answer the following:

(a) What is a sysctl (not what does the command do, but what a sysctl actually *is*)?

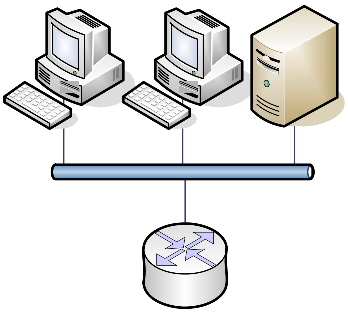
(b) In what file can you place sysctl values so they are loaded at boot time?

(c) Which sysctl controls IP forwarding?

1-5 Which addresses are your hosts supposed to use? Which names? Which netmasks and broadcast addresses? You can find this information on the course home page.

Report: Answers to all the questions above.

MAIN LAB



For this lab to work smoothly you *need* some basic understanding of IPv4, including routing, client-side DNS and Ethernet. Without it you will find the exercises very difficult.

One of the first things you have to do to make your system usable is to configure the network on the UML instances. For the basic install no networking is configured. To accomplish this assignment, you will have to understand the structure of the lab network and how your UML instances are connected to it. Moreover, you should be familiar with diagnostics tools such as `ping` and `traceroute`.

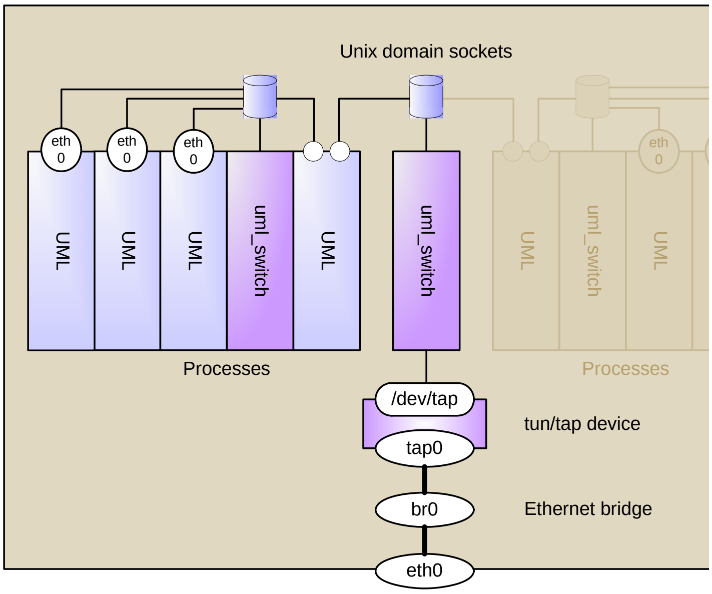
At this stage few tools are at your disposal (even `ping` and `traceroute` may not be installed) so you might be forced to do several steps without proper diagnostic tools. Because of this it is vital that you make detailed preparations in order to minimize the need for troubleshooting.

Time taken 2005: 4-12 hours, average 8 hours

Past problems: Nearly all the problems 2005 and 2006 were related to Quagga. Quagga is a quite complex piece of software, but the documentation is pretty typical of a niche product: it mostly works fine for experts, but isn't designed for newcomers. Most problems can be avoided by breaking up the problem in to tiny steps and completing one step at a time, making sure you always know what the goal of the step is, always have a way to verify that the goal has been reached and always understand how the goal was reached.

Part 1: Introduction

It may help to understand how the network connecting the UMLs to each other and to the lab network is constructed.



On your workstation each UML is executed as a number of processes. This processes use Unix domain sockets to communicate with a simulated switch (uml_switch). There is one dedicated uml_swith for each lab group that uses the computer, plus an extra shared uml_switch. The shared uml_switch is in turn connected to a Ethernet bridge on the machine through a so-called tap-device (tap0). The physical network interface (eth0) of the machine is also connected to the bridge, which allows the bridge to operate as the workstation's primary network interface.

Part 2: Setting the hostname

⚠ Before starting this part, or any later part, make sure that you have the right addresses and host names for your system. Configuring the wrong addresses can cause no end of grief for you and others. If you *do* end up using the wrong names or addresses, you will be required to change them. The information you need is available on the course home page.

The virtual systems you have started are not properly named. It is necessary for each host to know its fully qualified domain name (FQDN). The FQDN consists of the host name and a domain name. For example, www.ida.liu.se is a FQDN, but www and www.ida aren't.

There are two files that are relevant to naming a host: /etc/hostname and /etc/hosts. The /etc/hostname contains the hostname (not the FQDN) and /etc/hosts contains the host's address, FQDN and hostname.

For example, if a host has the address 192.168.12.1 and FQDN server.example.com, then /etc/hostname would have the following contents:

```
server
```

The /etc/hosts file would contain the following line (note that the order is significant).

```
192.168.12.1     server.example.com server
```

Exercise 2: Setting the hostname

2-1 Set the hostname on all your hosts. The router should be named gw, the server should be named server and the clients named client-1 and client-2. Don't forget the FQDN. Please use the recommended names – it simplifies things for your lab assistant.

Report: A test protocol that shows that the name change was successful and permanent.

The `uname` and `hostname` commands are useful for testing the hostname.

Part 3: Basic network connectivity

The first step is to establish basic network connectivity for your router/gateway. This will allow you to connect to the installation server on the distribution network, from which you can install the routing software needed in later steps.

Your router needs three network interfaces: the loopback interface (lo; which is used for network connections within the gateway), the interface connected to the distribution network (eth1) and the interface connected to your network (eth0). Each interface needs an address, netmask and broadcast address. It is important that you specify the correct netmask and broadcast address!

On Debian computers, network configuration is in the file /etc/network/interfaces. There is a man page describing the file format.

Exercise 3: Interface configuration (gateway)

3-1 Configure addresses and anything else that is needed on each network interface. The configuration must survive a restart of the system.

3-2 Bring each interface up, so that it can be used.

3-3 Use `ifconfig` to verify your configuration.

3-4 If `ping` is available, test that you can reach 130.236.189.1 from your gateway.

Report: A printout of your network configuration file(s). The output from `ifconfig`.

At this point your gateway/router should be able to connect to 130.236.189.0/24 and to your own internal network, but not to anything else (and since the clients are not configured yet, you won't be able to test connectivity to the internal network).

Exercise 4: Interface configuration (client)

4-1 Configure addresses and anything else that is needed on your clients (one client is sufficient at this time; you can do the other later).

4-2 Bring the client's interfaces up, so they can be used.

4-3 Use `ifconfig` to verify your configuration.

4-4 If `ping` is available, test that you can reach your gateway (the address it has on eth0).

Report: A printout of your network configuration file(s). The output from `ifconfig`.

At this point your gateway/router should be able to communicate with your clients, and vice versa. You still don't have Internet connectivity.

Once the interfaces are configured and up and running you should set up a default gateway on your router. The default gateway is the router to which your gateway will send any traffic it does not have a more explicit route for (when you are finished, this will be any traffic destined for outside 130.236.189.0/24).

Exercise 5: Default gateway/route (gateway)

5-1 Configure a suitable default route on your gateway.

5-2 Bring the appropriate interface down and then back up again to install the default route.

5-3 Use the `route` command to check that the default route has been installed.

5-4 If `ping` is available, test that you can reach 130.236.1.1 from your gateway.

Report: Changes to your network configuration. The output from `route`.

At this point your gateway/router should have full Internet connectivity, but your clients don't, as they're connected to a network that the rest of the world doesn't know about.

The next step is to configure a default route on your clients. The default route must be an address on a network they are directly connected to, i.e. your network, not the distribution network.

Exercise 6: Default gateway/route (client)

6-1 Configure a suitable default route on your clients.

6-2 Bring the appropriate interface down and then back up again to install the default route.

6-3 Use the `route` command to check that the default route has been installed.

Report: Changes to your network configuration. The output from `route`.

At this point, your gateway should be able to communicate with the world and your clients with the gateway. At this point your clients should be able to send data to any address, but since no router is announcing your network, replies can't find their way back.

If you get a "network unreachable" error when attempting to bring up an interface, you have probably specified a default gateway that is not on the same network as the interface you are bringing up. Check all addresses, netmasks and gateways again.

Part 4: Name resolution

At this point you should have basic network connectivity from your router to the Internet, but you can still not use names instead of IP addresses. In order to install software using APT, the preferred way of installing binary packages on Debian, you will need name resolution as well.

When learning about name resolution, it helps to know that the software that performs name resolution is called a *resolver*. Most documentation will use that term.

It also helps to know about the name service switch, which is used in most modern Unix-like systems. The name service switch (also known as NSS) is a piece of software that determines from what source names, including host names, are taken. The default setup is that all names are stored in local files on each computer. You need to change it to look for hosts in DNS.

Exercise 7: Resolver and NSS configuration

7-1 Answer the following questions:

(a) What configuration file controls NSS?

(b) What configuration file controls the resolver?

(c) What does the "search" keyword in the resolver configuration file do?

(d) Which order is more appropriate: look for hosts in local files first, then DNS or look for hosts in DNS first, then local files? Motivate your answer.

7-2 Configure the resolver on all systems to use 130.236.189.1 as its name server.

7-3 Configure the name service switch on all systems to use DNS as a source for host names.

7-4 Test that name resolution works by attempting to communicate with ida-gw.sysinst.ida.liu.se (use `ping` or `telnet` if they are available).

Report: Answers to the questions. Printouts of relevant configuration files.

At this point, your router/gateway should have full connectivity to the Internet *and* be able to use names to refer to hosts. Nothing has changed with respect to your clients. Although they may be able to contact the name server, it is possible that the nameserver doesn't know where to send the replies. Taking care of that is the goal of the next part of the lab.

Troubleshooting and testing tip: You should always use IP addresses rather than names when troubleshooting networking problems. If you use names, and experience problems, the cause of the problems could be related to name resolution (e.g. nameserver issues, resolver configuration or connectivity to the nameservers), confusing the issue.

Part 5: Router configuration

This step is perhaps the most advanced. All previous steps have to be finished and working. Furthermore, the package system (apt) on your gateway has to be configured so that it knows from where packages are supposed to be downloaded. If you have not already setup the package system, do so now and update the package list using `apt-get` update (or `aptitude`).

Routing vs. forwarding

Routing and forwarding are two separate things. In the IP world, particularly in the world of host-based routers such as the one you are building, people confuse the two. Routing is the process of figuring out where packets should be sent in order to reach their final destination. Forwarding is the process of actually accepting packets on one interface and sending then on another, chosen in accordance with the routing decisions that have been made. The point is, a host can run routing software (and thus compute paths through a network) without actually forwarding any packets.

This is important. If you set up your router correctly, but forget to turn on IP forwarding, nothing will work. Early versions of Quagga, which you will use for routing, enabled IP forwarding automatically, but current versions properly separate the two functions.

Quagga

The routing software we are using is named Quagga, and as most things in the Linux world it is open source software. Quagga is a fork of the Zebra routing software. Development of Zebra more or less halted when the main developer on the project formed a company to market a commercial version of Zebra, named ZebOS. Quagga implements several usual routing protocols such as RIP, OSPF and BGP and where applicable, the corresponding protocols for IPv6.

The configuration interface for Quagga is similar to IOS, which is used in equipment manufactured by Cisco, so by learning about Quagga, you are also learning skills that are useful on Cisco equipment. Setting up Quagga isn't terribly difficult, but you will need to read the manual. The documentation at http://www.quagga.net/ is very helpful here. You will also need to know basic networking. If you don't know the fundamentals of routing, then this would be a good time to learn. Any basic networking textbook should contain what you need.

Note that some documents suggest you use a program called `vtysh` to configure Quagga. For this lab, avoid using `vtysh`. Although it is quite convenient, you should know more about Quagga, how it works and what configuration files it uses before confusing the situation with yet another program.

Exercise 8: Quagga configuration files

8-1 Quagga consists of several separate processes. Describe how they fit together.

8-2 What configuration files does the Debian installation of Quagga use?

8-3 How do you configure Quagga interactively?

Report: Answers to the questions above.

If you want to look at a working Quagga router, connect to 130.236.189.1, on port 2601 (zebra) or 2602 (ripd) using the `telnet` command and log on using the password "zebra". You have very limited rights on these systems, so only a few commands will be available to you.

The documentation for Quagga is reasonably good, but it is fairly terse and there's nothing in the way of guidance. To use the documentation, there are two straightforward ways to go: one is to simply read the whole thing and try to remember which commands sound like they might be useful. The other, which I prefer, is to first figure out what you need your router to do (in terms of the RIP protocol), and then hunt for commands to make that happen.

Note that Quagga commands are context sensitive. For example, ripd has commands that work at the top level (simply called "command" in the documentation), but also commands that work in a "router" context (called "RIP command" in the documentation). Regular commands, such as "distribute-list" can be given at any time in the configuration file, whereas RIP commands, such as "network" commands, can only be given in the context of a "router rip" command (the context is started with the command "router rip" and terminated by any regular command).

Troubleshooting tip: To troubleshoot routing problems, it can be helpful to use a tool such as wireshark or tcpdump to see what packets are being sent to and from your router. These commands are capable of decoding RIP announcements. Wireshark is by far the easier to use, but since it's graphical it's also harder to get running in the lab environment (for starters, you'll have to install `ssh` on your router).

RIPv2

In the lab we use RIP version 2 as the routing protocol. RIP is a simple protocol suitable for small, simple networks. RIP operates by sending out announcements periodically that contain a list of address prefixes that the RIP router knows about. These may have been learned from other routers or they may be local addresses, directly connected to the RIP router.

Your router needs to announce your prefix, and it may announce prefixes it learns from other routers. It may *not* announce the default route.

Exercise 9: RIP basics

9-1 Explain, in broad terms, how RIP works.

9-2 What should your RIP router send out in the packets it sends, and to which networks?

Report: Explain RIP to your lab assistant (or if your lab assistant is unavailable, your lab partner).

Do not proceed with the following exercises until you have completed the previous exercise!

Reference documentation for Quagga is available on the Quagga home page. There are no tutorials. Note that many commands have sub-commands. For example, the `router rip` command has sub-commands such as `redistribute`, `version` and `network`, none of which are valid outside the context of `router rip`.

Exercise 10: RIP configuration

10-1 Enable RIP on the external interface of your router. Use RIP version 2 with no authentication (you may have to explicitly disable authentication). Select a sensible enable password.

10-2 (Re)start all quagga servers, wait a few seconds and examine your routing tables. If you have successfully enabled `ripd`, you should see an additional default route and possibly one or more other routes.

10-3 Connect to `ripd` interactively and run `show ip rip status` to display the current RIP status. You should see at least one routing information source.

Report: The output of `show ip rip status` and the current routing table.

At this point your router accepts routing information from other sources, but announces nothing. This means that other hosts will not know how to reach your network. In order for them to do so, your router needs to announce the prefix of your network.

By default Quagga will announce routes (using RIP) that it has learned through RIP. It will not announce anything else, such as static routes (e.g. the default route configured on the host) or prefixes belonging to network interfaces connected to the host. In order to announce such routes using RIP, you need to instruct Quagga to *redistribute* routes or prefixes into RIP.

Exercise 11: RIP announcements

11-1 Configure `ripd` to announce your network prefix.

11-2 (Re)start all quagga servers and wait a few seconds. Check that your router is still accepting announcements from other servers by using show ip rip status.

11-3 Connect to the `ripd` port on 130.236.189.1 using `telnet` and check that it has your router as a routing information source. Connect to the `zebra` port and check that it has your prefix in its routing table.

Report: Relevant output from your session with 130.236.189.1.

At this point, if everything has gone according to plan, you have full network connectivity.

Exercise 12: Final checks

12-1 Test that you have full network connectivity to your gateway and clients.

Report: Test protocols that show that your network is operating as expected.

If you want to try your hand at some more Quagga configuration, the following *optional* exercises cover a pretty important aspect. Unfortunately, additional exercises require a more complex network.

Exercise 13: Intermediate RIP configuration (optional)

This exercise is optional, but might save you a lot of trouble later. If you do not do this, it is possible that another group's configuration errors might prevent you from accessing the network. You should use the `ip prefix-list` command for this.

13-1 Configure your router to filter incoming announcements (use a prefix list). You should

(a) Accept routes for 10.17.1.0/24 (but not longer prefixes).

(b) Accept routes for the other groups' address space.

(c) Not accept anything else.

13-2 Configure your RIP router to announce a default route on your internal network *only*. This may entail setting up a distribute list for announcements (again, using a prefix list).

Report: The configuration file changes required to complete the exercise.

FEEDBACK FORM NET

Complete this feedback form **individually** at the end of the lab and hand it to the lab assistant when you finish. Your feedback is essential for improving the labs. Each student should hand in a feedback form. Do not cooperate on completing the form.

You do not need to put your name on the feedback form. Your feedback will be evaluated the same way regardless of whether your name is on it or not. Your name is valuable to us in case you have made and comments in the last section that need clarifications or otherwise warrant a follow-up.

For each section, please rate the following (range 1 to 5 in all cases).

- **Difficulty:** Rate the degree of difficulty (1=too easy, 5=too difficult)
- **Learning:** Rate your learning experience (1=learned nothing, 5=learned a lot).
- **Interest:** Rate your interest level after completing the part (1=no interest, 5=high interest).
- **Time:** How long did the part take to complete (in minutes)?

	Difficulty	Learning	Interest	Time (minutes)
Prelab				
Part 1: Introduction				
Part 2: Setting the hostname				
Part 3: Basic network connectivity				
Part 4: Name resolution				
Part 5: Router configuration				
Overall				

Please answer the following questions:

- What did you like about this lab?

- What did you dislike about this lab?

- Make a suggestion to improve this lab.

FEEDBACK FORM NET

Complete this feedback form **individually** at the end of the lab and hand it to the lab assistant when you finish. Your feedback is essential for improving the labs. Each student should hand in a feedback form. Do not cooperate on completing the form.

You do not need to put your name on the feedback form. Your feedback will be evaluated the same way regardless of whether your name is on it or not. Your name is valuable to us in case you have made and comments in the last section that need clarifications or otherwise warrant a follow-up.

For each section, please rate the following (range 1 to 5 in all cases).

- **Difficulty:** Rate the degree of difficulty (1=too easy, 5=too difficult)
- **Learning:** Rate your learning experience (1=learned nothing, 5=learned a lot).
- **Interest:** Rate your interest level after completing the part (1=no interest, 5=high interest).
- **Time:** How long did the part take to complete (in minutes)?

	Difficulty	**Learning**	**Interest**	**Time (minutes)**
Prelab				
Part 1: Introduction				
Part 2: Setting the hostname				
Part 3: Basic network connectivity				
Part 4: Name resolution				
Part 5: Router configuration				
Overall				

Please answer the following questions:

- What did you like about this lab?

- What did you dislike about this lab?

- Make a suggestion to improve this lab.

FEEDBACK FORM NET

Complete this feedback form **individually** at the end of the lab and hand it to the lab assistant when you finish. Your feedback is essential for improving the labs. Each student should hand in a feedback form. Do not cooperate on completing the form.

You do not need to put your name on the feedback form. Your feedback will be evaluated the same way regardless of whether your name is on it or not. Your name is valuable to us in case you have made and comments in the last section that need clarifications or otherwise warrant a follow-up.

For each section, please rate the following (range 1 to 5 in all cases).

- **Difficulty:** Rate the degree of difficulty (1=too easy, 5=too difficult)
- **Learning:** Rate your learning experience (1=learned nothing, 5=learned a lot).
- **Interest:** Rate your interest level after completing the part (1=no interest, 5=high interest).
- **Time:** How long did the part take to complete (in minutes)?

	Difficulty	**Learning**	**Interest**	**Time (minutes)**
Prelab				
Part 1: Introduction				
Part 2: Setting the hostname				
Part 3: Basic network connectivity				
Part 4: Name resolution				
Part 5: Router configuration				
Overall				

Please answer the following questions:

- What did you like about this lab?

- What did you dislike about this lab?

- Make a suggestion to improve this lab.