Internet Protocol Design

Iain Phillips

Semester 1 2016

Chat program

- Supporting:
  - Point to Point Messages
  - File Transfer
  - Broadcast Messages
- Over a layer-2, mesh network
- The `icns` library

Python module

- Written for this module
  - Provides point-to-point communication between neighbours
  - Neighbours are *added*
  - Messages `sent` to individual neighbours or all
  - Messages `received` in a single queue
- But:
  - It is unreliable and drops packets
  - and might introduce delays and reordering

Download it with git from bitbucket.

Login to Linux, start at Terminal

```
cd
git clone http://bitbucket.org/iainwp/icns.git
```

Remember:

```
PYTHONPATH=$HOME/icns python yourprog.py
```

and

```
export PYTHONPATH=$HOME/icns
```

Details on LEARN.

- It's said that everything in UNIX is represented as a file.
- keyboard, screen, other devices, files etc.
- Each file is accessed through a *File Descriptor*

Each program running has by default 3 File Descriptors:

- `stdin` for input
- `stdout` for output
- `stderr` for error

In python these are in the `sys` module, `sys.stdin` etc and read from and written to using the `os` module, `os.read`, `os.write` etc.

Creating a network

A network conists of a numnber of nodes, each one:

- Creates a Network Object, `n=icns.Network(num)`
  - If several nodes are represented on the same host, then their `nums` must be different.
- Adds neighbours, `nh=n.addNeighbour(address, num)`, once for each neighbour.
- Messages can then be sent with `n.send(...)` and `n.receive(...)`

Creating a User Interface

- A UI has a file descriptor to pass text from the keyboard
- Create a UI with `ui=incs.UI("title text")`
- Add text to the window with `ui.addLine("text to add")`
- Get the file descriptor with `ui.getfd()` and read from it with `os.read(...)`

Waiting for input

This is the key part of linking the network to the UI.

- You can wait for input from:
  - The user (via the filedescriptor from `getfd`)
  - The network
- or for a timeout
- Use the `n.orfd(...)` call for this.
- details in the documentation (`pydon icns`)

Tidying up

- If your program crashes and the screen gets all mucked up, then type `reset` and hit return.
- To prevent things getting mucked up use `ui.stop()` before you exit, catch this with `try/expect`.

`icns` library

- In the lab
- Write a `netcat` equivalent
- listener (multiclient)
- sender
- Experiment and experience the unreliability.
- Then write a UI program.