**COSC 364**

**Internet Technologies and Engineering**

**First Assignment**

**Kayle Ransby**

34043590

**Sjaak Flick**

36406121

* A percentage contribution of each partner to the overall project. Important: this must be agreed upon by you and your partner, the relative weights will influence grading.
* Which aspects of your overall program (design or implementation) do you consider particularly well done?
* Which aspects of your overall program (design or implementation) could be improved?
* How have you ensured atomicity of event processing?
* Have you identified any weaknesses of the RIP routing protocol?

**Percentage contribution of each partner:**

Kayle Ransby: 55

Sjaak Flick: 45

**Example configuration file:**

**Router 1:**

router-id 1

input-ports 1101, 1102, 1103

outputs 6101-5-6, 7101-8-7, 2101-1-2

**What aspects of the program work well?**

Each daemon in the “network” is one big object, which makes it easier to diagnose issues related to the operation of the daemon.

**What could have been done differently?**

An error in one of the daemons could potentially happen in all of the daemons

**Weaknesses of RIP**

One major issue with RIP is the count to infinity problem,

This can be mitigated with the use of split horizon with poisoned reverse and triggered updates,

**Program testing:**

Test: Input File

Expected output:

If all of the parameters are within the ranges allowed the daemon should start and run without issues.

Ordering of the data, such as input ports, output ports and the router id should not matter

Comments should be ignored by the file reader.

If there was an issue with the input data, the demon is expected to stop running and a message printed, explaining where the error had occurred.

Some issues with input data could be that the router ID is too large or is a negative number.

Another issue could be that the input ports “well known” ports, or ports outside the upper bound.

A third issue with the input data is that the output port ID’s can have a mismatch between the router-id’s, metrics and port numbers.

Actual Output:

An issue encountered is that while the error message would be printed, there would also be another message relating to the startup of the periodic timers.

Another test was “switching off” a particular router in the network. The purpose of this test is to determine if the demons recalculate the metrics any routes affected by the router going offline.

The expected behaviour that would be seen is that any entries to that router that exist on the network would linger on for as long as the dead-timer + garbage collecter would allow for.

Any entries in the route table that went through the router that was switched off, would then have to be recalculated.

The behaviour of the routing demon when this test was performed was to be expected. When the router is “switched off”, the timeout variable ran down to 0, once that happened, then the garbage collection timer started, once that was done, the route was deleted from the table.

Another test that was performed is periodic updates.

Within a set interval, the demon would send response packets to all of it’s neighbours, telling each of them that it’s still alive, and sending the entire route table along with it.

The expected behaviour to be observed is that every so often, a periodic update packet would be sent to each of a demon’s neighbours, and in turn, a demon would receive packets from their neighbouring demons.