**CS3D3 Project 2 – Simple Distance Vector Routing Protocol in C++**

**Group 15:**

Jason Boyle 16327736

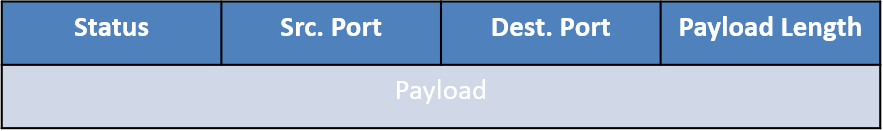
Ciaran Donegan

Samprati Jain

**Implementation:**

For this project we used **Object-Oriented C++ Programming**.

Description of the classes used in the Project:

* **DVRouter.cpp**
  + A Class to represent one of the routers in the network.
  + When a router is created, it reads in its initial distance vector table from the topology file to discover its direct neighbours.
* **DataSegment.cpp**
  + A class to represent a UDP message to be sent between the routers.
  + The following message structure was used:
  + Data Segments can either be Control Messages (‘C’ in the status field) or Data Packets (‘D’ in the status field).
  + In a control message, the payload is the distance vector advertisement from the sending router, in a Data Packet, the payload is a text phrase.

In the main file, my-router.cpp, a new DV\_Router object is created and sends and receives UDP messages to/from its neighbouring routers.

When a router receives a control message, it parses the distance vector data and uses it to check for a new least cost paths in the router’s forwarding table.

When a router receives a data packet, it is parsed and depending on the destination port:

* The router forwards the packet on to the next hop router in the least cost path to the destination if the receiving router is not the destination of the packet.
* The router prints the payload (test phrase) of the data packet if the receiving router is the destination of the packet.

For UDP Communications between the routers, we used the **C++ Boost Library** for **Asynchronous UDP Socket Programming.**

* The file, my-router.cpp runs **two processes**, each with the following function:
  + **Parent Process**: Listens for DV advertisements from neighbouring routers.
  + **Child Process**: Sends DV advertisements to neighbouring routers every 5 seconds.

**Difficulties Faced**