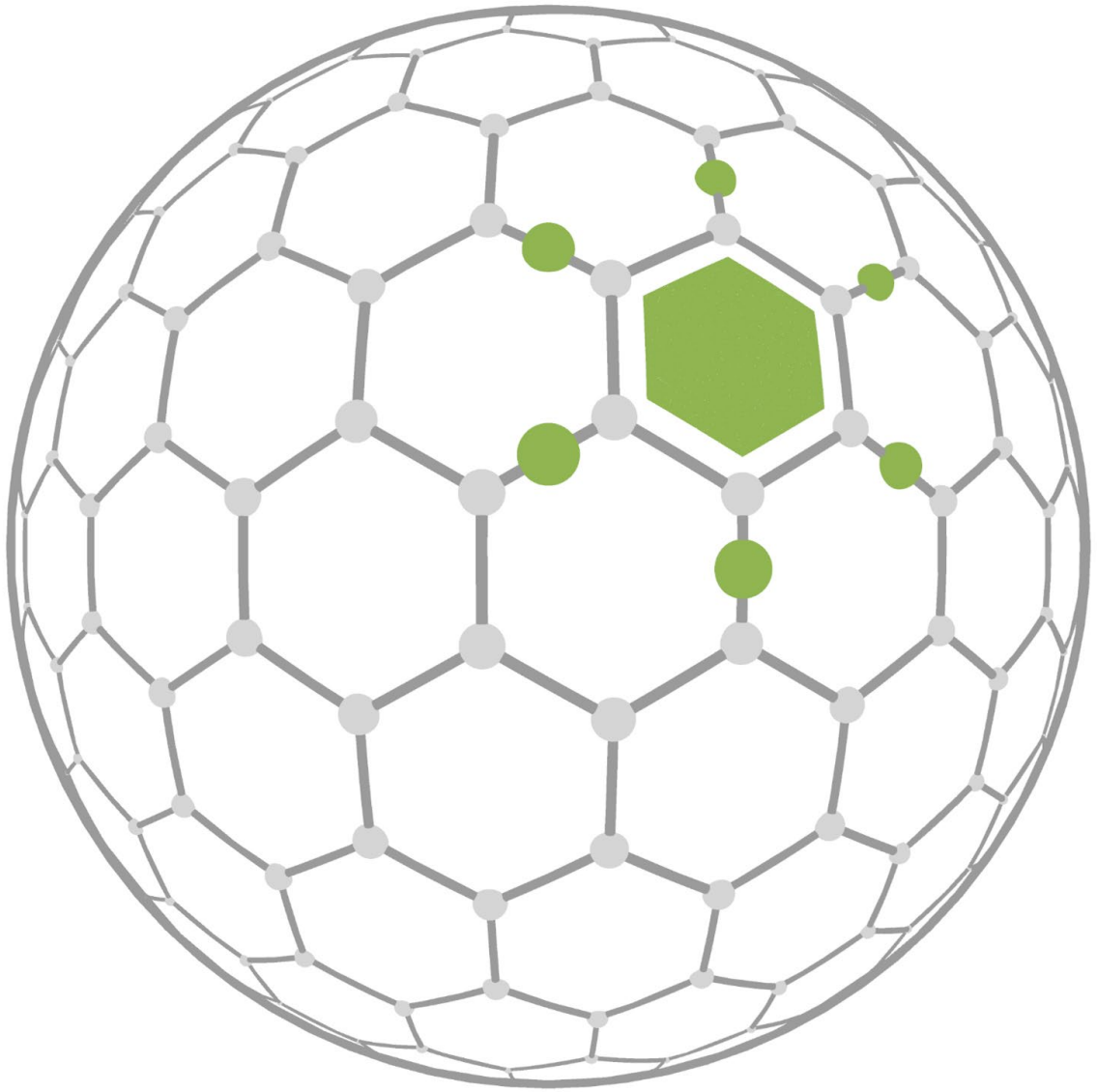


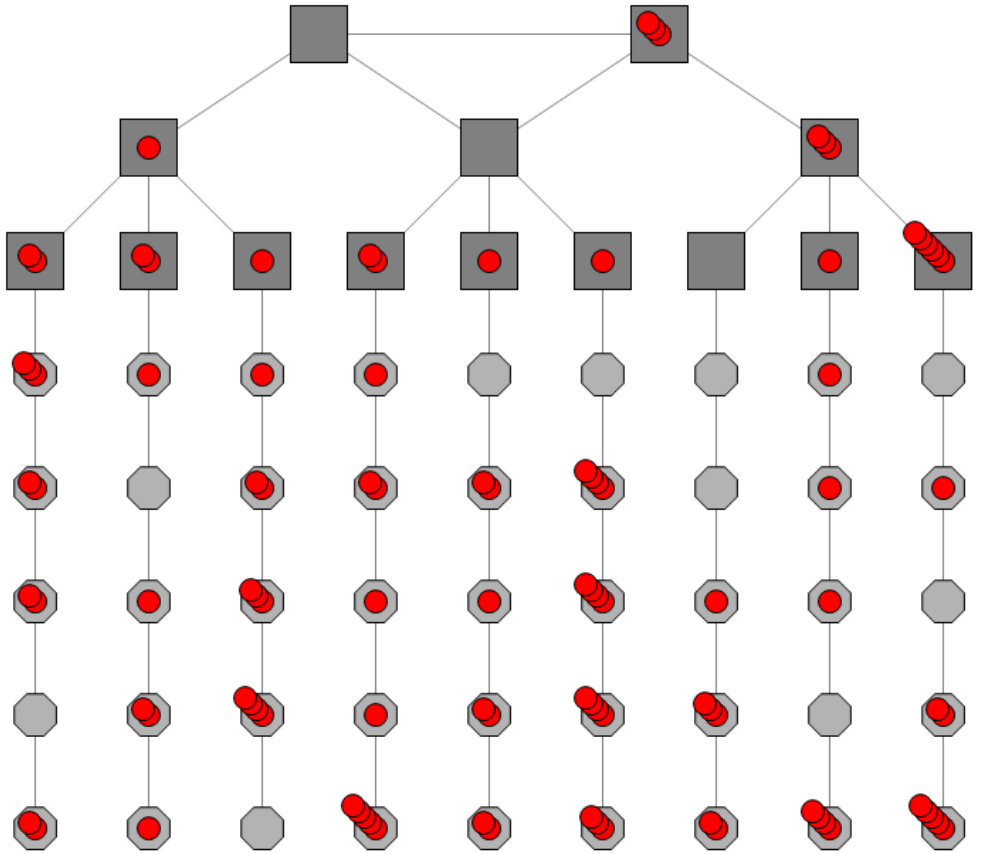
Daedaelus Routing Protocols

Demonstration of network routing protocols using Wolfram Mathematica

Clos Network Congestion

Clos Network

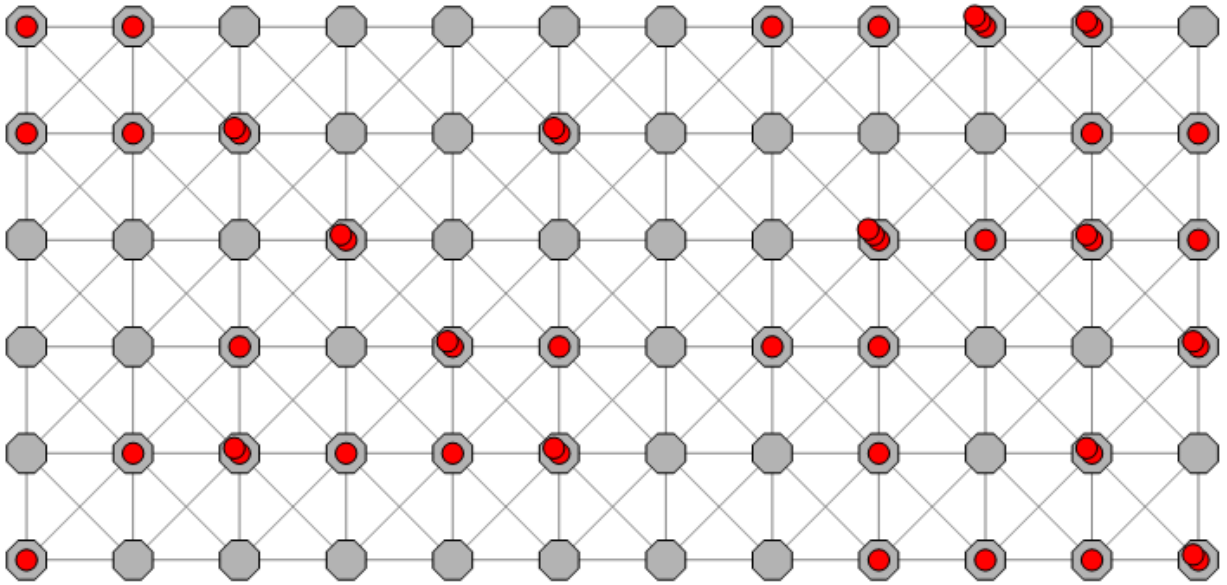
- Fat tree network topology
- Each node connected to top of rack switch
- Heavy use of buffers at the switches



Mesh Network

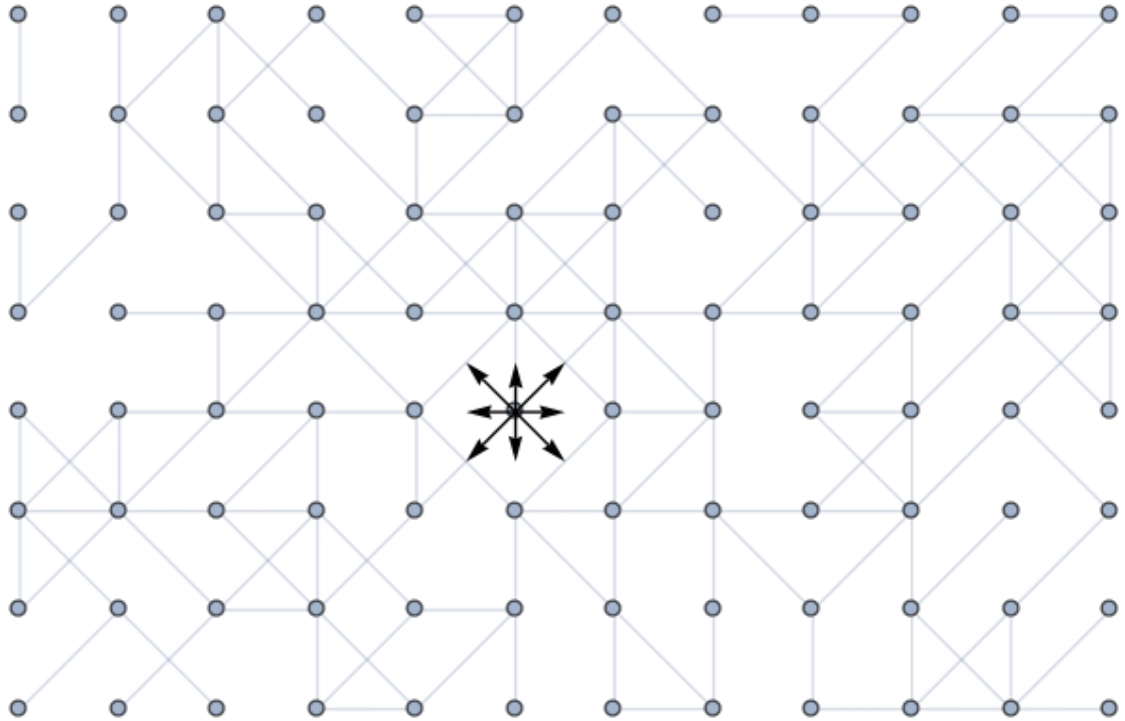
Mesh Network

- Vastly more routes possible
- Direct routing
- Less reliant on buffers at nodes



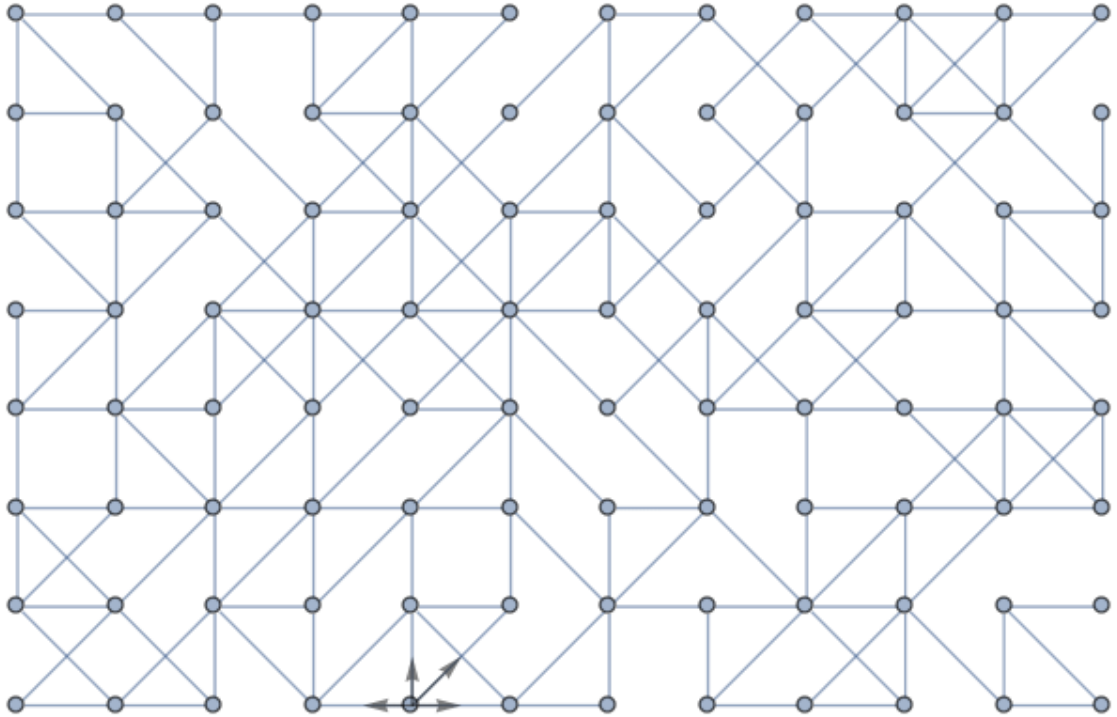
Routing Protocol (naïve)

- Source Node
  - Sends out scout packets along all available ports
- Each packet contains
  - Source
  - Destination
  - routing history as it is passed along the network
- Packet lifetime is limited from the source
  - Maximum number of hops
- Intermediary Node
  - Receives incoming scout packet
  - Duplicates that packet out all other ports
- Target Node
  - Receives a successful packet
  - Sends it back along the path to return to source
- Source Node
  - Receives successful packets
  - Records the best path to its Target Node



Routing Protocol (naïve)

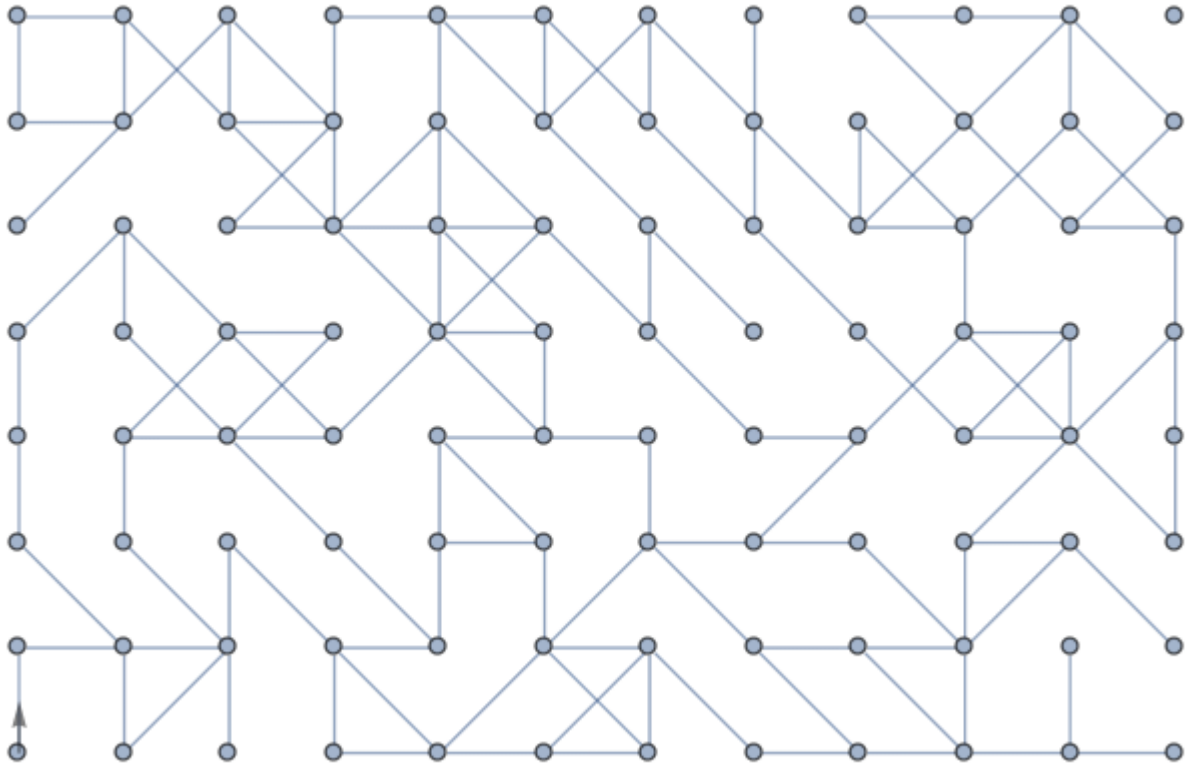
- Source Node
  - Sends out scout packets along all available ports
- Each packet contains
  - Source
  - Destination
  - routing history as it is passed along the network
- Packet lifetime is limited from the source
  - Maximum number of hops
- Intermediary Node
  - Receives incoming scout packet
  - Duplicates that packet out all other ports
- Target Node
  - Receives a successful packet
  - Sends it back along the path to return to source
- Source Node
  - Receives successful packets
  - Records the best path to its Target Node



- Exponential growth in packets
- Heavy congestion and use of buffers
- Multiple paths are returned

Routing Protocol (Buffer Optimization)

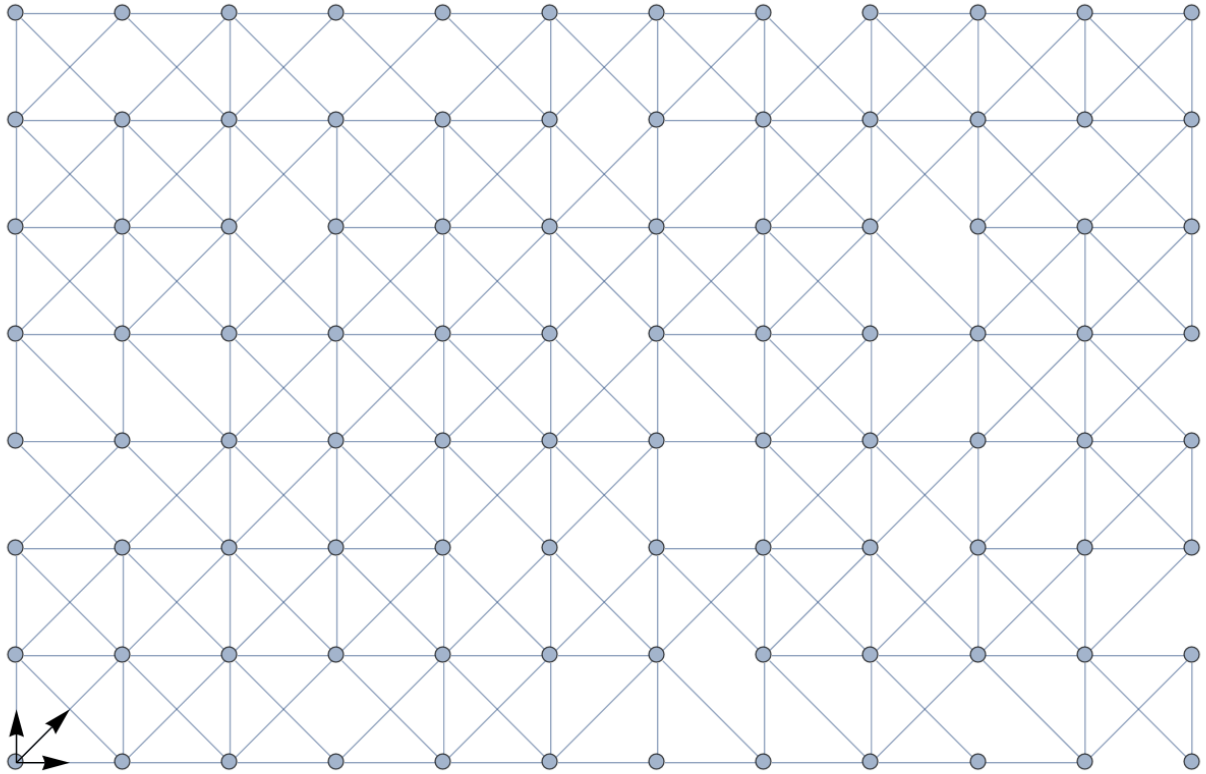
- Source Node
  - Sends out scout packets along all available ports
- Each packet contains
  - Source
  - Destination
  - routing history as it is passed along the network
- Packet lifetime is limited from the source
  - Maximum number of hops
- Intermediary Node
  - Receives incoming scout packet
  - Duplicates that packet out all other ports
  - Only duplicate the first (or optimal) packet in a buffer,
  - delete duplicate scout packets in buffer
- Target Node
  - Receives a successful packet
  - Sends it back along the path to return to source
- Source Node
  - Receives successful packets
  - Records the best path to its Target Node



- Reduced packet storm
- Can trace longer paths without overloading the network
- Buffers are minimized
- Multiple paths are returned

Routing Protocol (Memory Optimization)

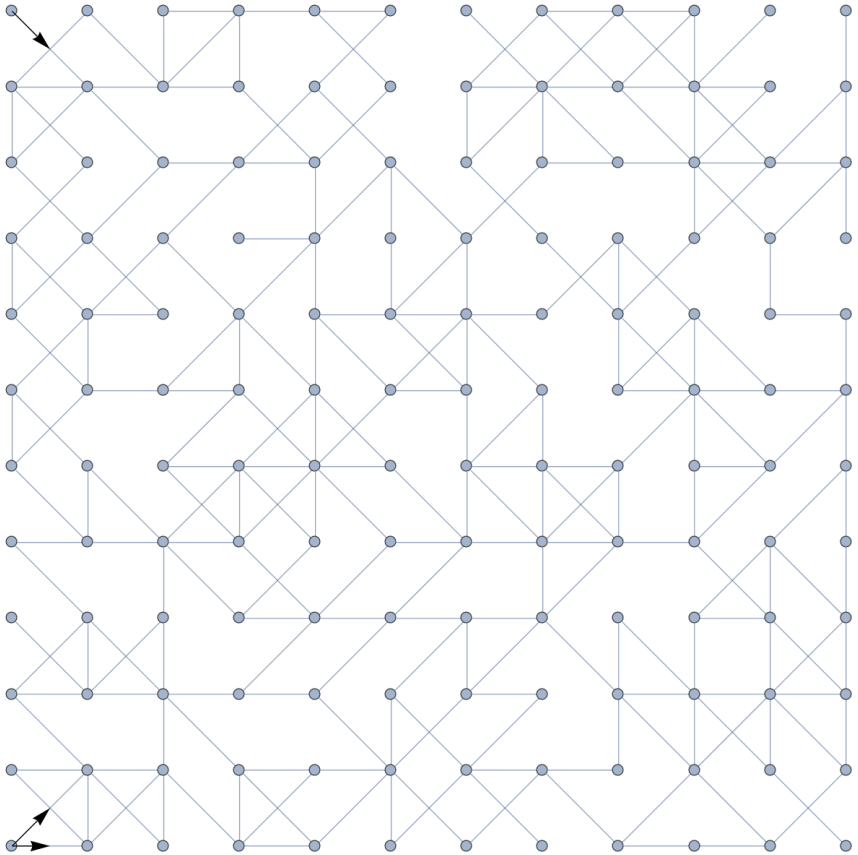
- Source Node
  - Sends out scout packets along all available ports
- Each packet contains
  - Source
  - Destination
  - routing history as it is passed along the network
- Packet lifetime is limited from the source
  - Maximum number of hops
- Intermediary Node
  - Receives incoming scout packet
  - Duplicates that packet out all other ports
  - Keep a memory of scout packets that have been seen
  - If another scout packet is seen, drop the redundant ones
- Target Node
  - Receives a successful packet
  - Sends it back along the path to return to source
- Source Node
  - Receives successful packets
  - Records the best path to its Target Node



- Minimizes packet storm
- Can trace longer paths without overloading the network
- Buffers are minimized
- Only one (optimal) path is returned

Routing Protocol (Memory Optimization)

- Source Node
  - Sends out scout packets along all available ports
- Each packet contains
  - Source
  - Destination
  - routing history as it is passed along the network
- Packet lifetime is limited from the source
  - Maximum number of hops
- Intermediary Node
  - Receives incoming scout packet
  - Duplicates that packet out all other ports
  - Keep a memory of scout packets that have been seen
  - If another scout packet is seen, drop the redundant ones
- Target Node
  - Receives a successful packet
  - Sends it back along the path to return to source
- Source Node
  - Receives successful packets
  - Records the best path to its Target Node



- Minimizes packet storm
- Can trace longer paths without overloading the network
- Buffers are minimized
- Only one (optimal) path is returned
- Robust to handle any number of scout packets for multiple sources & destinations

Routing Tree – Automatic Failover

- Pre-computed routing trees to the Target Node
- All nodes have list of routing trees
- One tree for each incoming port to Target Node
- When one incoming port fails
  - an announcement is made to the network
  - Packets enroute to the Target Node are switched to next available tree