CMPUT 313 ASSIGNMENT 2

Routing Algorithm Performance

Due: Tuesday March 1, 2016, 5:00 pm.

STEVEN CHAN

Running the program:

Single Run

Python3 hsim.py adjacency_list_file.txt

Batch Run

Python3 test.py line_list_file.txt mesh_list_file.txt star_list_file.txt complete_list_file.txt tree_list_file.txt ring_list_file.txt adjacency_list_file.txt

Purpose of program:

The goal of the program is to explore the performance of four different routing algorithms.

1. Link State Routing:
   - The Link State Routing algorithm sets up routing tables by having every node obtain link state information from every other node. Each source broadcasts its link state information by flooding.
2. Distance Vector Routing:
   - The Distance Vector Routing algorithm sets up routing tables by having every node share their estimate of delay with their neighbours. Each node sends its delay diameter times. Diameter is length of longest shortest path of network.
3. Hot Potato Routing I:
   - The Hot Potato Routing I algorithm sets up routes by having sources send route discovery packets to destinations. The source forwards a packet to a neighbour at random. At each subsequent node, check if it is destination. If not, packet is forwarded to a neighbour excluding of which it came from. After the packet reaches the destination, a reply packet is sent back. Loops are removed before the reply packet is returned.
4. Hot Potato Routing II:
   - The Hot Potato Routing II algorithm is a variation of Hot Potato Routing I algorithm. The difference is the nodes know their neighbours and if the destination is a neighbour, the discovery packet is routed directly there. Else, it will continue as in Hot Potato Routing I.

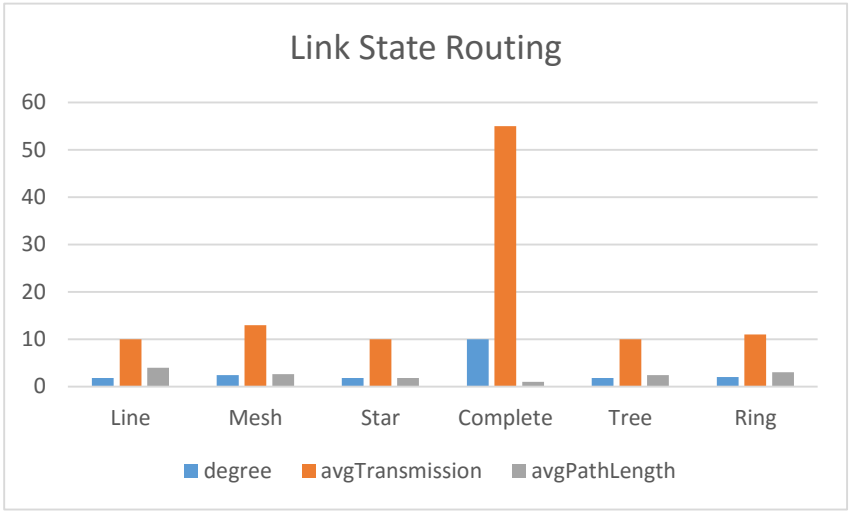
Performance metrics of interest:

a) The average number of transmissions required to set up the paths in the network (for all source-destination pairs). This is the total number of transmissions to set up the paths divided by the number of nodes

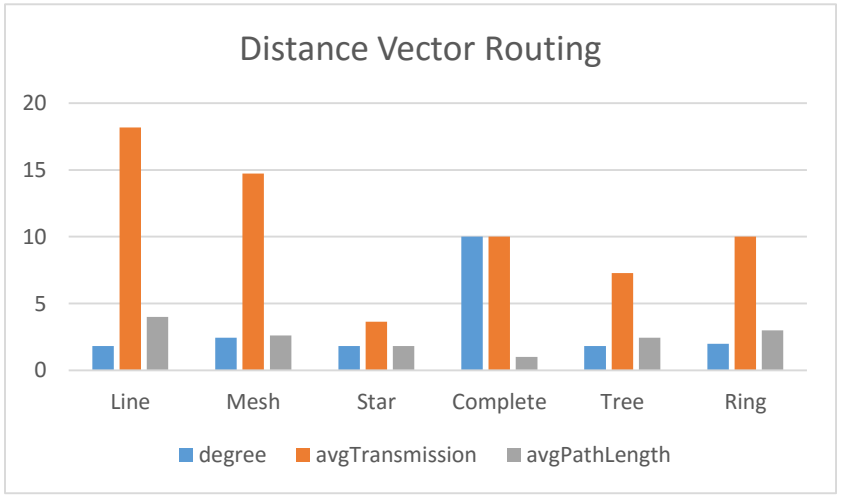
b) The average path length of the routes that are found. This is the sum of the path lengths between source-destination pairs divided by the number of source-destination pairs.

c) The degree of a node is its number of neighbours. The average degree of the network is the sum of the node degrees divided by the number of nodes. Characterizing topology.

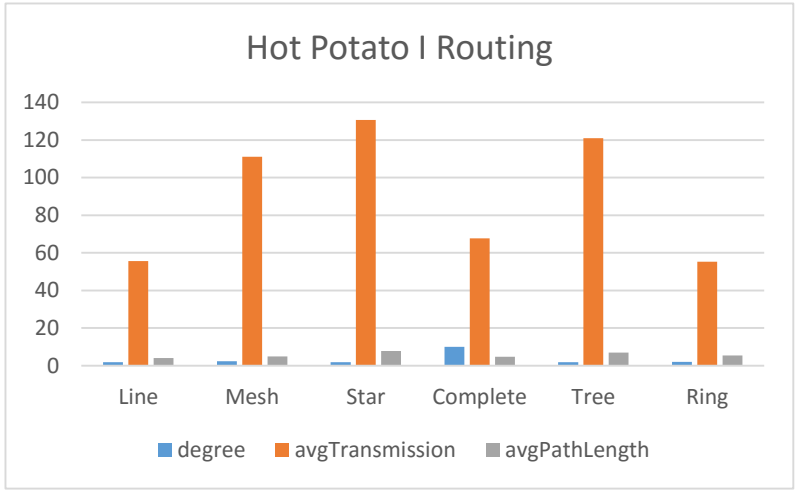
Link State	degree	avgTransmission	avgPathLength
Line	1.818182	10	4
Mesh	2.454545	13	2.609090909
Star	1.818182	10	1.818181818
Complete	10	55	1
Tree	1.818182	10	2.436363636
Ring	2	11	3



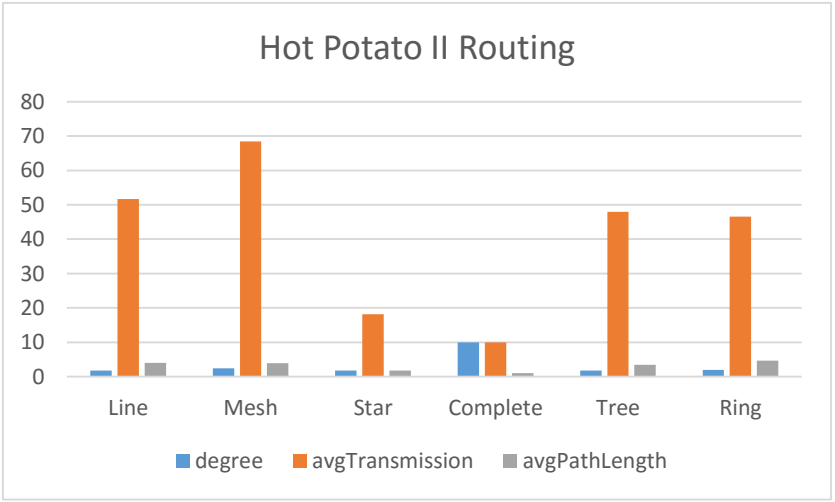
Distance Vector	degree	avgTransmission	avgPathLength
Line	1.818182	18.18181818	4
Mesh	2.454545	14.72727273	2.609090909
Star	1.818182	3.636363636	1.818181818
Complete	10	10	1
Tree	1.818182	7.272727273	2.436363636
Ring	2	10	3



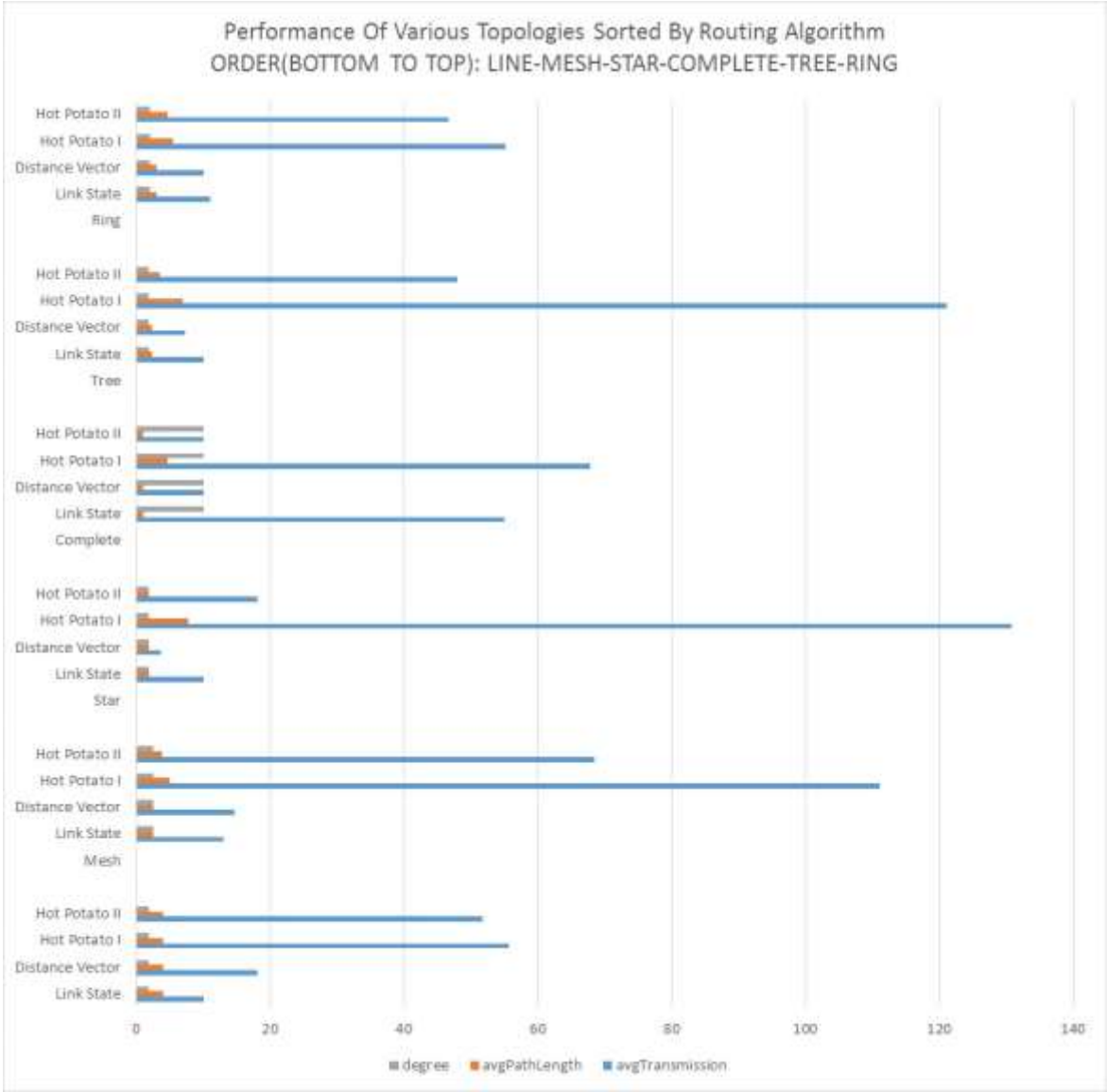
Hot Potato I	degree	avgTransmission	avgPathLength	CI_AvgTransmission	CI_AvgPathLength
Line	1.818182	55.70909091	4	(54.383793134474395, 57.03438868370743)	(4.0, 4.0)
Mesh	2.454545	111.0545455	4.970909091	(103.00914384403, 119.09994706506089	(4.637932146012983, 5.303886035805198)
Star	1.818182	130.7454545	7.745454545	(121.25866784920208, 140.23224124170707	(6.999231076939289, 8.4916780139698)
Complete	10	67.75454545	4.7	(63.227751977904596, 72.28133893118633)	(4.3597606604238885, 5.04023933957611)
Tree	1.818182	121	7.010909091	(112.3491898988663, 129.65081010113371)	(6.470813633173944, 7.551004548644237)
Ring	2	55.21818182	5.521818182	(52.66199150531365, 57.77437213105)	(5.266199150531365, 5.777437213105)



Hot Potato II	degree	avgTransmission	avgPathLength	CI_AvgTransmission	CI_AvgPathLength
Line	1.818182	51.70909091	4	(50.699357108821175, 52.718824709360646)	(4.0, 4.0)
Mesh	2.454545	68.44545455	3.909090909	(64.27625418866572, 72.61465490224342) ,	(3.672732283250621, 4.145449534931197)
Star	1.818182	18.18181818	1.818181818	(18.181818181818183, 18.181818181818183)	1.818181818181818 (1.8181818181818177, 1.8181818181818181)
Complete	10	10	1	(10.0, 10.0)	(1.0, 1.0)
Tree	1.818182	47.96363636	3.450909091	(45.27641696969697, 50.650855757575755)	(3.2552980901621034, 3.646520091656078)
Ring	2	46.6	4.66	(45.28202692240787, 47.91797307759213)	(4.528202692240788, 4.791797307759213)



Line	avgTransmission	avgPathLength	degree
Link State	10	4	1.818182
Distance Vector	18.18181818	4	1.818182
Hot Potato I	55.70909091	4	1.818182
Hot Potato II	51.70909091	4	1.818182
Mesh	avgTransmission	avgPathLength	degree
Link State	13	2.609090909	2.454545
Distance Vector	14.72727273	2.609090909	2.454545
Hot Potato I	111.0545455	4.970909091	2.454545
Hot Potato II	68.44545455	3.909090909	2.454545
Star	avgTransmission	avgPathLength	degree
Link State	10	1.818181818	1.818182
Distance Vector	3.636363636	1.818181818	1.818182
Hot Potato I	130.7454545	7.745454545	1.818182
Hot Potato II	18.18181818	1.818181818	1.818182
Complete	avgTransmission	avgPathLength	degree
Link State	55	1	10
Distance Vector	10	1	10
Hot Potato I	67.75454545	4.7	10
Hot Potato II	10	1	10
Tree	avgTransmission	avgPathLength	degree
Link State	10	2.436363636	1.818182
Distance Vector	7.272727273	2.436363636	1.818182
Hot Potato I	121	7.010909091	1.818182
Hot Potato II	47.96363636	3.450909091	1.818182
Ring	avgTransmission	avgPathLength	degree
Link State	11	3	2
Distance Vector	10	3	2
Hot Potato I	55.21818182	5.521818182	2
Hot Potato II	46.6	4.66	2



Analysis:

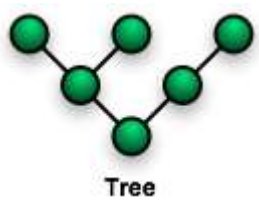
Complete Topology:

A topology in which each of the nodes is connected to each other.

Performance Ranking: Distance Vector / Hot Potato II > Link State > Hot Potato I

Distance Vector is by far the fastest with avgTransmission of 10 and avgPathLength of 1. This is in line with the structure of a complete topology. Every node is connected to every other node. As a result, the shortest path will always be 1, which is what we get. Every node can reach another with the shortest path. Link State takes more avgTransmissions as every node records the state of all other nodes via flooding before creating a shortest path. Hot Potato II is quite fast and has a significant improvement over Hot Potato I. This is due to Hot Potato II being able to directly route to neighbour destinations. Hot Potato II has avgTransmission of 10 and avgPathLength of 1 while Hot Potato I has avgTransmission 67.75 and avgPathLength 4.7. Hot Potato II is faster than Hot Potato I which makes sense since Hot Potato II is an improved version and can take shortest paths. A degree of 10 makes sense since every node is linked together.

Tree Topology:

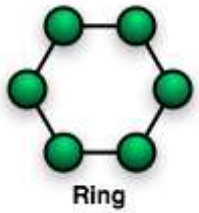


A topology that is a combination of bus and star. Connects one star network to other star networks.

Performance Ranking: Distance Vector > Link State > Hot Potato II > Hot Potato I

Distance Vector is the fastest by a bit over Link State (7.27 vs 10). This is since Link State takes more avgTransmissions as every node records the state of all other nodes via flooding before creating a shortest path and Distance Vector can simply route information straight away to neighbours for shortest path. Hot Potato II this time is slower than Distance Vector and Link State but still beats Hot Potato I. avgTransmission 47.96, avgPathLength 3.45 vs avgTransmission 121, avgPathLength 7.01. This is due to Hot Potato II being able to directly route to neighbour destinations. Hot Potato II is faster than Hot Potato I which makes sense since Hot Potato II is an improved version and can take shortest paths. A degree of 1.81 makes sense since nodes will branch out and have a parent and or child, thus roughly 2 neighbours.

Ring Topology:

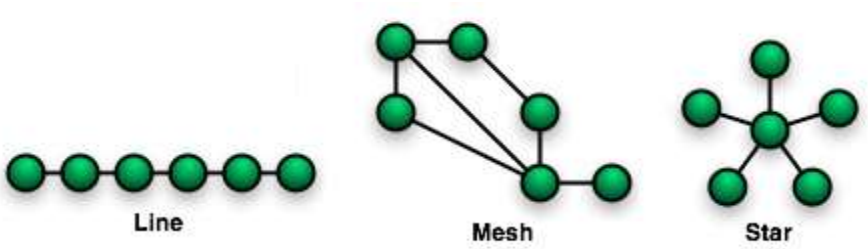


A topology that is set up to make a closed loop. Each node connects to exactly two other nodes forming a single continuous pathway.

Performance Ranking: Distance Vector > Link State > Hot Potato II > Hot Potato I

Distance Vector is the fastest but barely over Link State (10 vs 11). This is since Link State takes more avgTransmissions as every node records the state of all other nodes via flooding before creating a shortest path and Distance Vector can simply route information straight away to neighbours for shortest path. Hot Potato II is slower than Distance Vector and Link State but still beats Hot Potato I. avgTransmission 46.6, avgPathLength 4.66 vs avgTransmission 55.21, avgPathLength 5.52. This is due to Hot Potato II being able to directly route to neighbour destinations. Hot Potato II is faster than Hot Potato I which makes sense since Hot Potato II is an improved version and can take shortest paths. A degree of 2 makes sense since nodes as a ring topology, at best and at worst a node will always have two neighbours.

Other (Line, Mesh, Star) Topologies:



Line: A topology in which nodes are connected in succession to another.

Performance Ranking: Link State > Distance Vector > Hot Potato II > Hot Potato I

Mesh: A topology in which some of the nodes are connected to more than one other node.

Performance Ranking: Link State > Distance Vector > Hot Potato II > Hot Potato I

Star: A topology that connects all nodes to a central node.

Performance Ranking: Distance Vector Routing > Link State Routing > Hot Potato II > Hot Potato I

Line

- Link State Routing: 10 4.0
- Distance Vector Routing: 18.181818181818183 4.0

Mesh

- Link State Routing: 13 2.609090909090909
- Distance Vector Routing: 14.727272727272727 2.609090909090909

Star

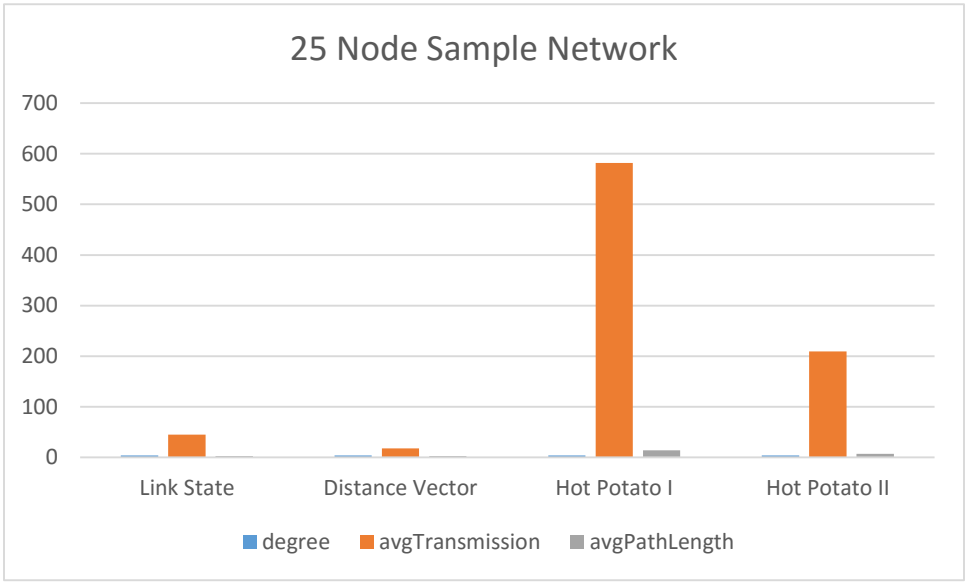
- Link State Routing: 10 1.818181818181818
- Distance Vector Routing: 3.6363636363636362 1.818181818181818

It is interesting to note that Link State came out on top for both Line and Mesh topologies. Hot Potato II continues to always beat Hot Potato I, which is in line with the implementation.

Degrees:

- Line: 1.818
  - Makes sense as always connected to 2 neighbors except the beginning and end nodes.
- Mesh: 2.45
  - Makes sense since is a variant where some nodes have more than one or two neighbors.
- Star:1.818
  - Makes sense as a central node connects other nodes, which only have the central node as a neighbor but the central node has all the other nodes as neighbors.

25 Node	degree	avgTransmission	avgPathLength	CI_AvgTransmission	CI_AvgPathLength
Link State	3.6	45	2.77		
Distance Vector	3.6	18	2.77		
Hot Potato I	3.6	581.74	14.06833333	(557.1662466653752, 606.3137533346248)	(13.301143429514807, 14.83552323715186)
Hot Potato II	3.6	209.3	6.932666667	(201.45042170592586, 217.14957829407405)	(6.6707358453845975, 7.194597487948736)



Distance Vector is by far the fastest with avgTransmission of 18 and avgPathLength of 2.77. This is since Link State takes more avgTransmissions as every node records the state of all other nodes via flooding before creating a shortest path and Distance Vector can simply route information straight away to neighbours for shortest path. Hot Potato II is slower than Distance Vector and Link State but still beats Hot Potato I. avgTransmission 209.3, avgPathLength 6.93 vs avgTransmission 581.74, avgPathLength 14.07. This is due to Hot Potato II being able to directly route to neighbour destinations. Hot Potato II is faster than Hot Potato I which makes sense since Hot Potato II is an improved version and can take shortest paths. A degree of 3.6 makes sense for the size of the network. The confidence interval of Hot Potato II is significantly smaller than that of Hot Potato I, which is quite fluctuant and broad.

Margin of CI of Hot Potato II: 217.14957829407405-201.45042170592586 = 15.6991565881

Margin of CI of Hot Potato I: 606.3137533346248-557.1662466653752 = 49.1475066692

The avgTransmissions fluctuation of Hot Potato II is only ~16 while Hot Potato I has a massive difference of 49 avgTransmissions. The variation in Hot Potato II shows to be very significant in transmitting and finding a shorter path length by being able to directly route to destination neighbours whereas Hot Potato I is unable to do so, thus has to continue to use a probability in choosing neighbours to route to, which results in this giant range in avgTransmissions.

<u>Conclusion Remarks:</u>

In general, Distance Vector Routing is the better algorithm than Link State. However, in certain cases Link State was barely behind and even beat out Distance Vector in some small tests (ex. Line, mesh). Both algorithms use shortest path but Distance Vector proved to be more efficient in avgTransmissions. Hot Potato II proved to be significantly faster than Hot Potato I in every single case tested. This is a result of Hot Potato II being able to directly route to destination neighbor nodes instead of

basing it on random probability of it hitting hoping it is the destination. If not, trace back and do again. As a result of these trace backs, Hot Potato I constantly sky rockets in number of transmissions required as shown in the graphs. Hot Potato I appears to be almost an outlier compared to the other algorithms. The general Performance Ranking is Distance Vector > Link State > Hot Potato II > Hot Potato I. There is a small relative gap between Distance Vector and Link State and a large gap between Hot Potato II and Hot Potato I. Comparing Distance Vector, Link State with Hot Potato II, Hot Potato I, it is shown that Distance Vector and Link State always beat out any Hot Potato algorithm in avgTransmissions as well as avgPathLength.