Can the Internet learn?

Patrick Carlson
Research Experience for Undergraduates
University of New Mexico

Primary UNM Mentor:
Dr. Terran Lane

Primary Simpson College Mentor:
Dr. Lydia Sinapova

NSF Grant Number: 0453545

Introduction

- Computer network – collection of nodes and edges

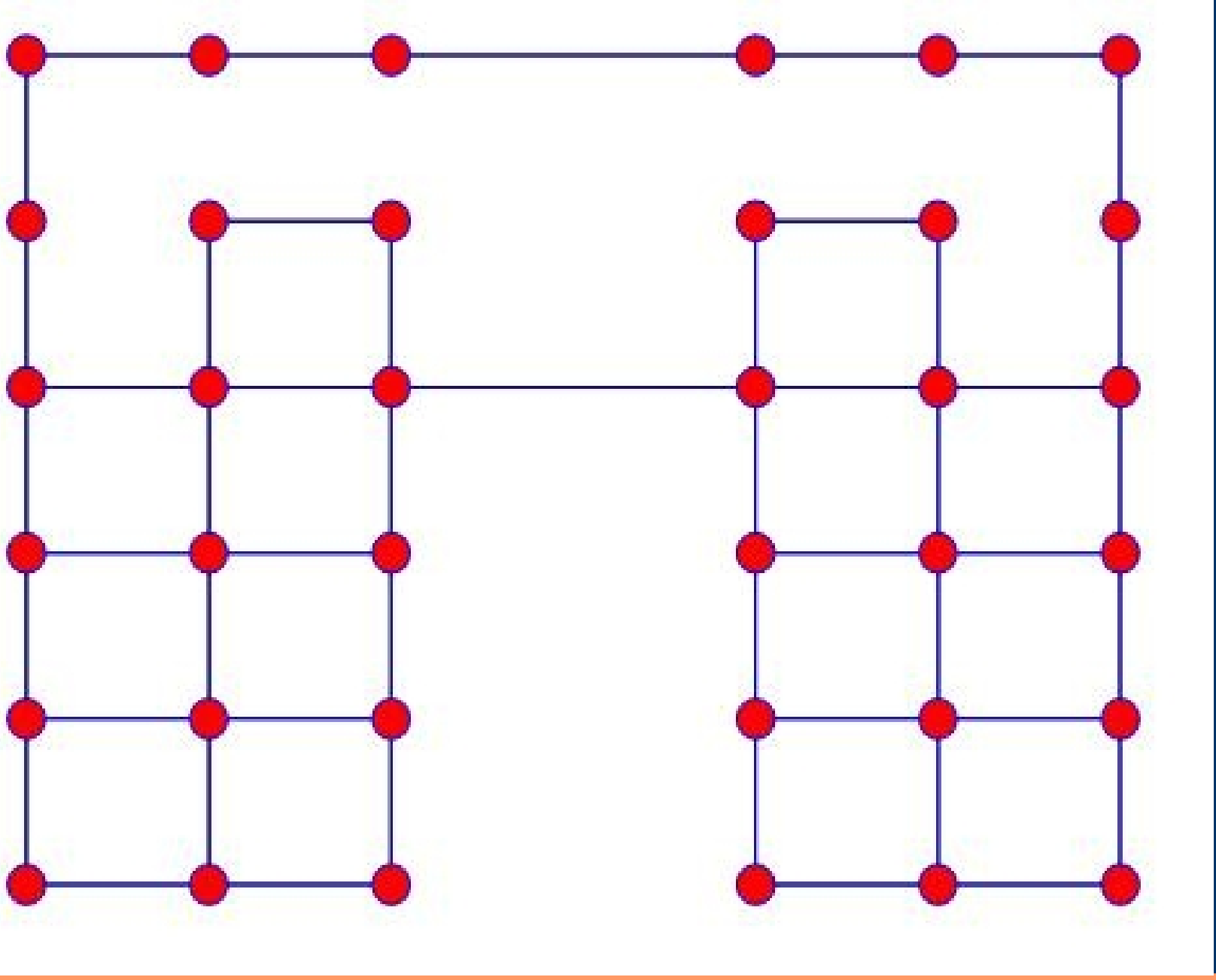
The Big Question:

How do you efficiently route packets of information through the network as quickly as possible?

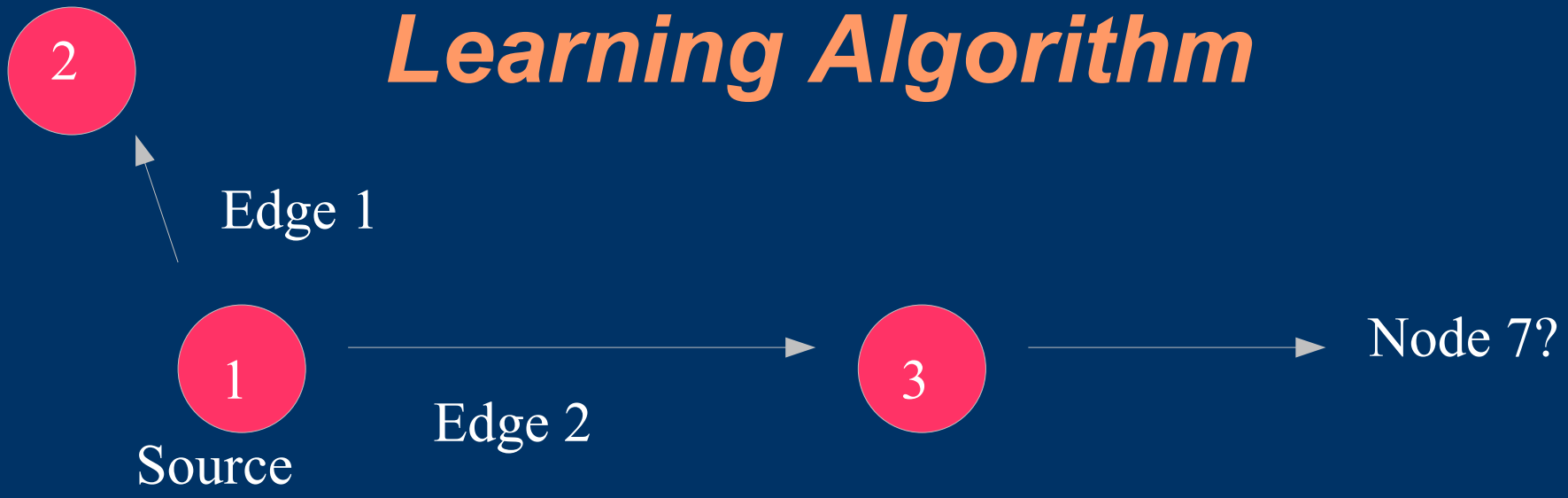
Background

- Current routing algorithms – shortest path in terms of number of “hops” through nodes
- Link-State algorithms – know global information about network graph (e.g. Dijkstra)
- Distance-Vector algorithms – only knows information about neighbors, distributed information exchange (e.g. Bellman-Ford)

The Problem...



Learning Algorithm



Q-Table stores estimated time for packet to reach destination

Columns represent each node

Rows represent each edge

0	3	5	1	9	10	13
0	7	3	9	6	6	**9**

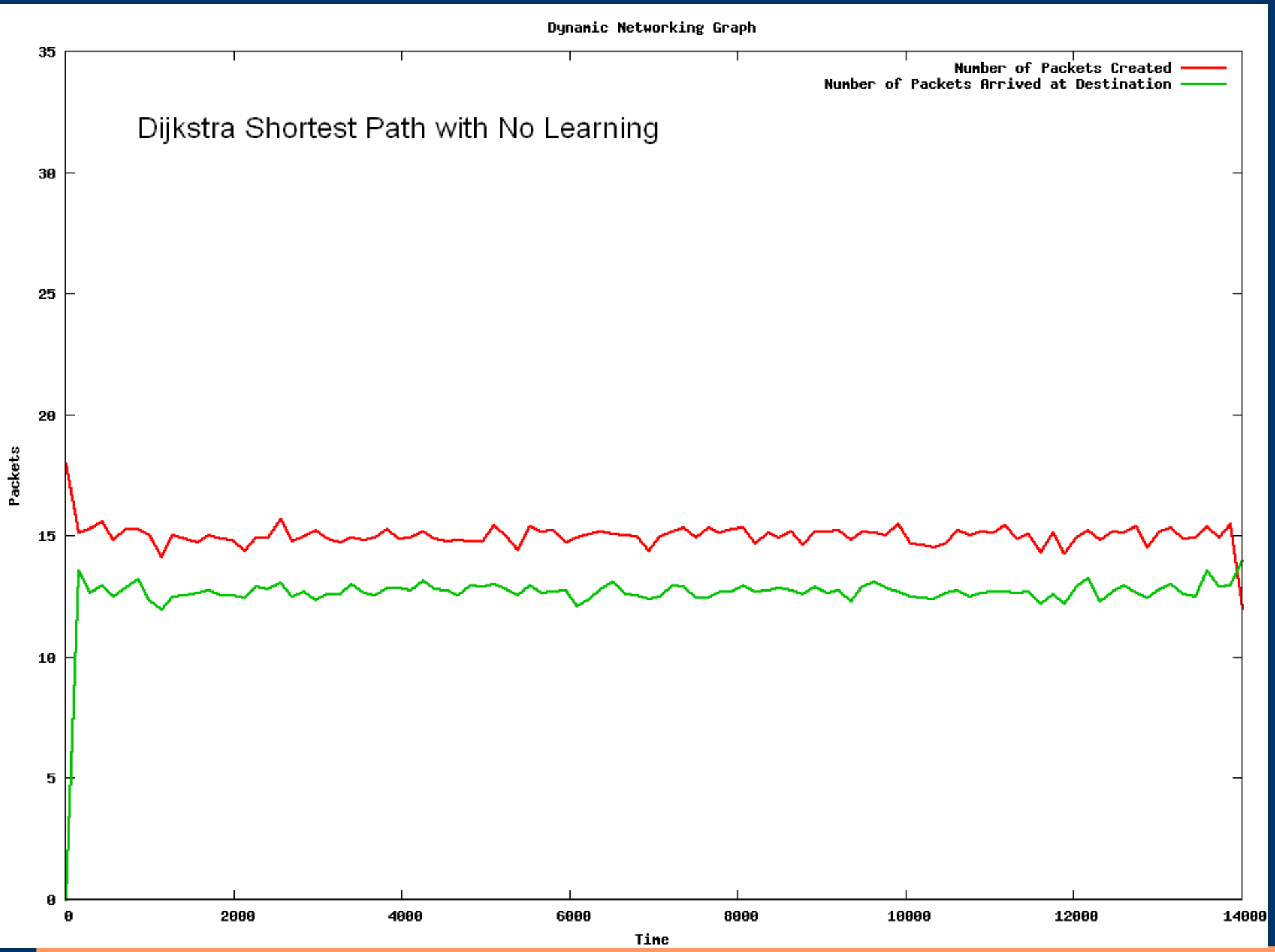
Packet gets sent!

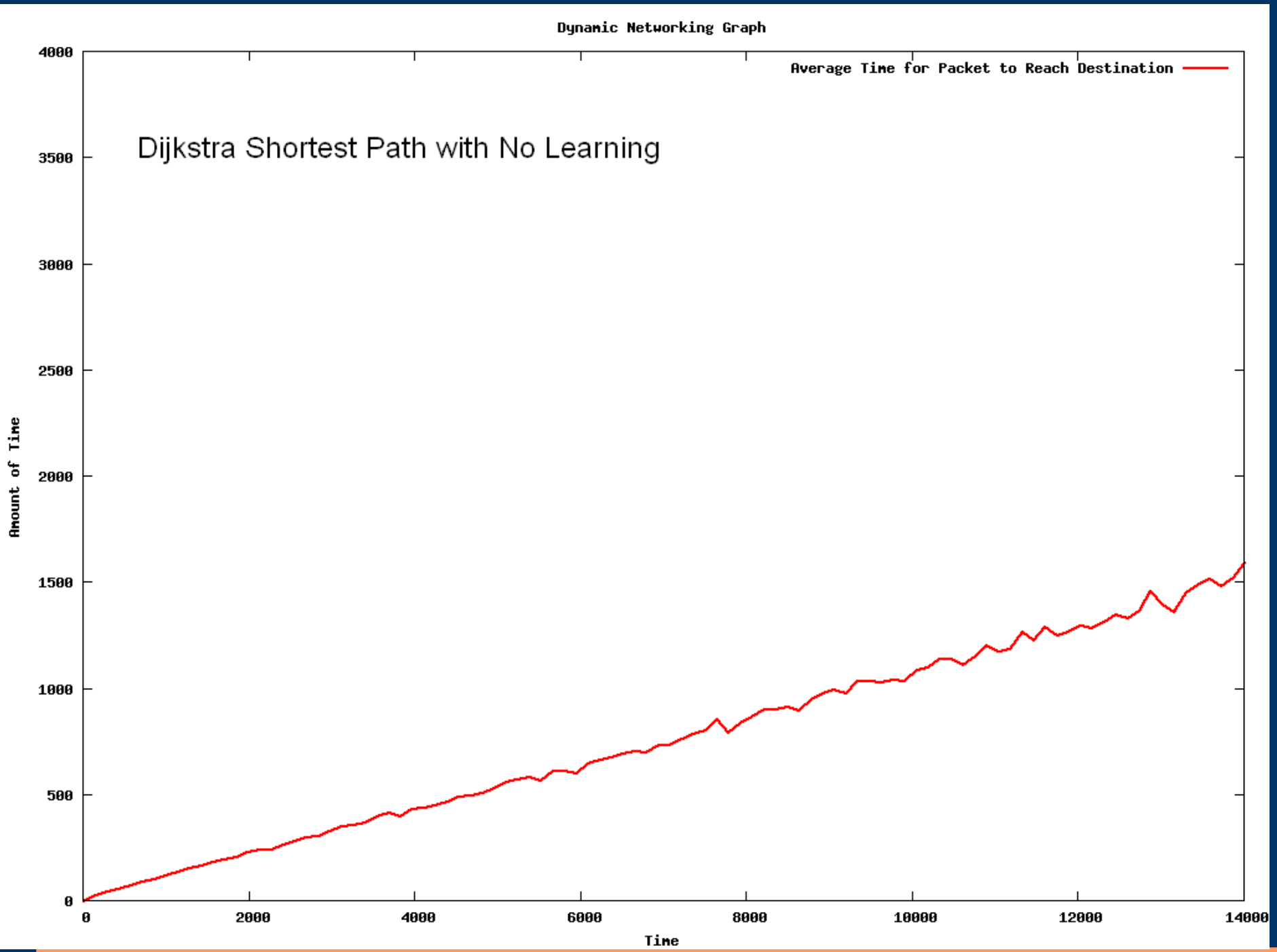
Learning Algorithm (cont.)

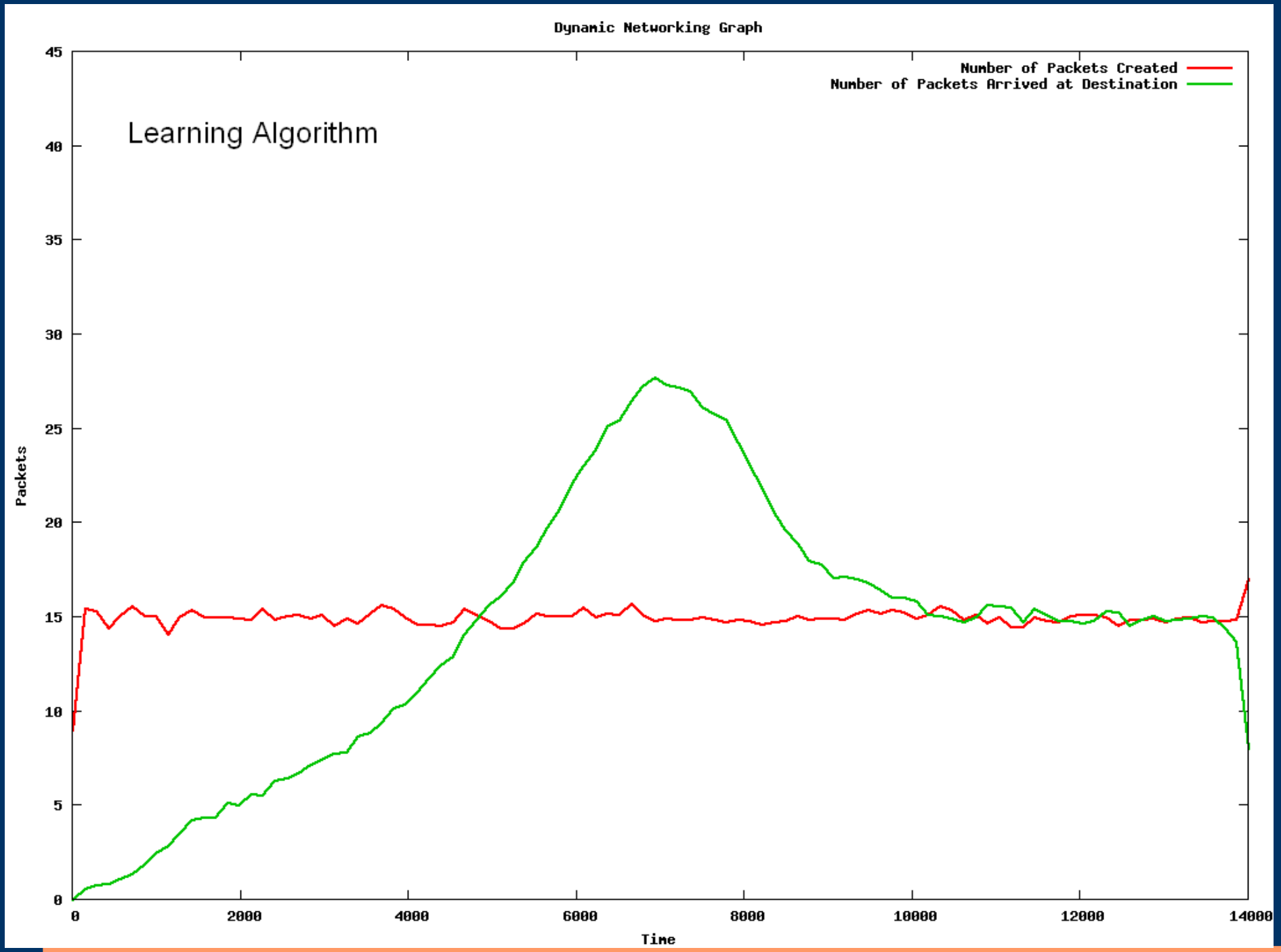
Node 1 gets back the q-table estimate from Node 3

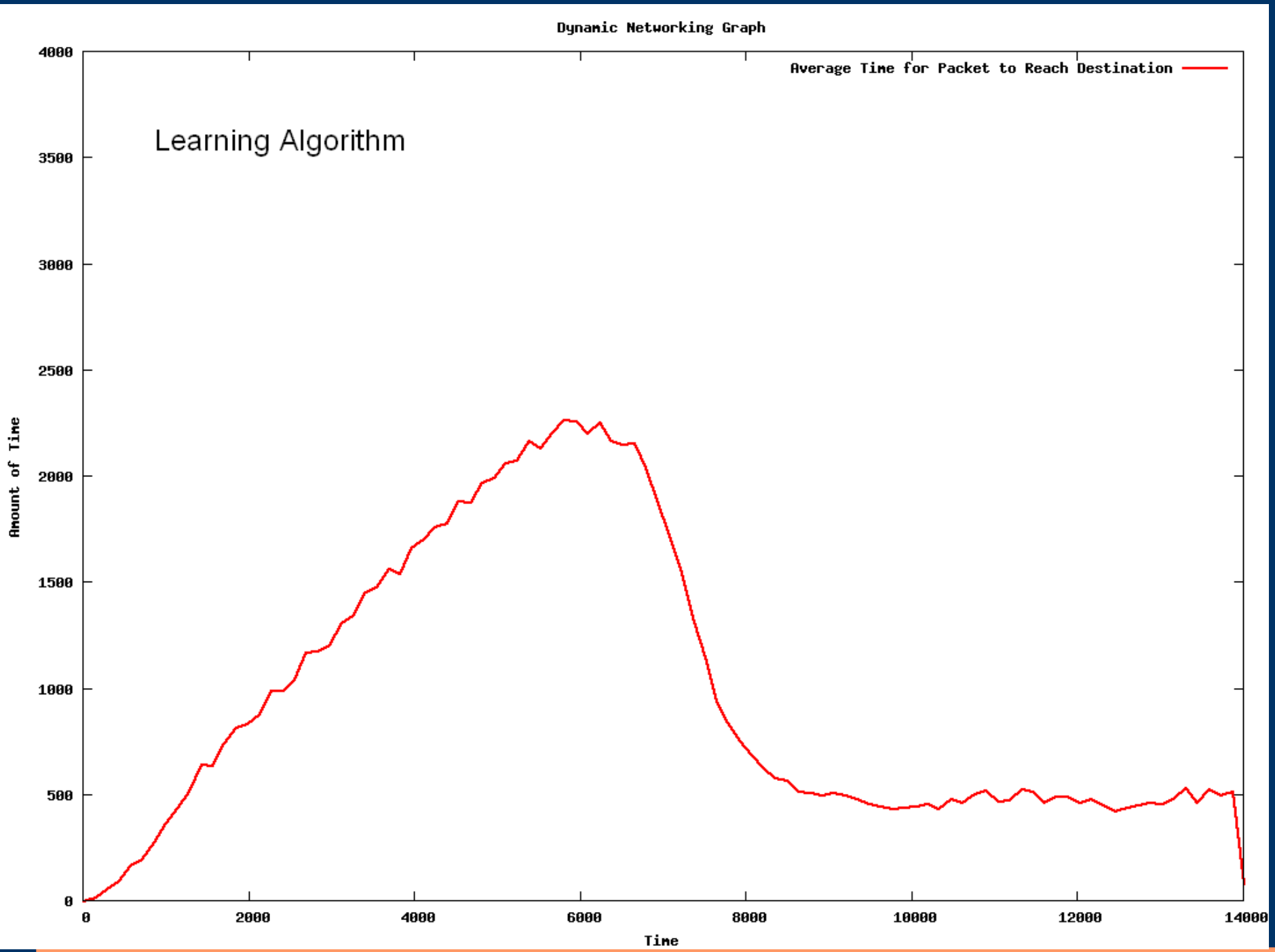
New Q-value = L (q + s + t – Old Value)

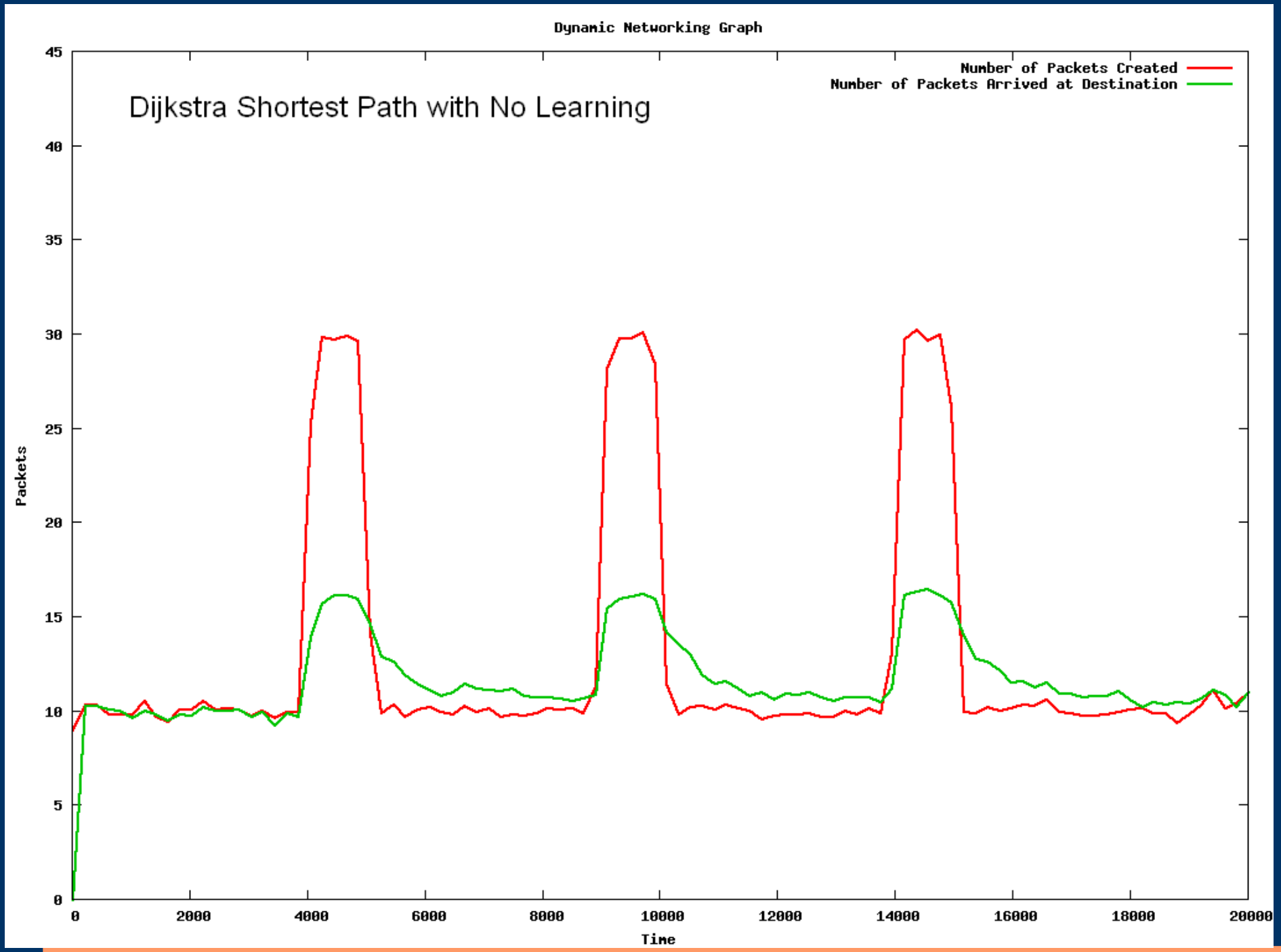
L = learning rate (0.5 in our runs)
q = time spent in queue (buffer) in Node 1
s = time in transmission between nodes
t = new estimate from Node 3

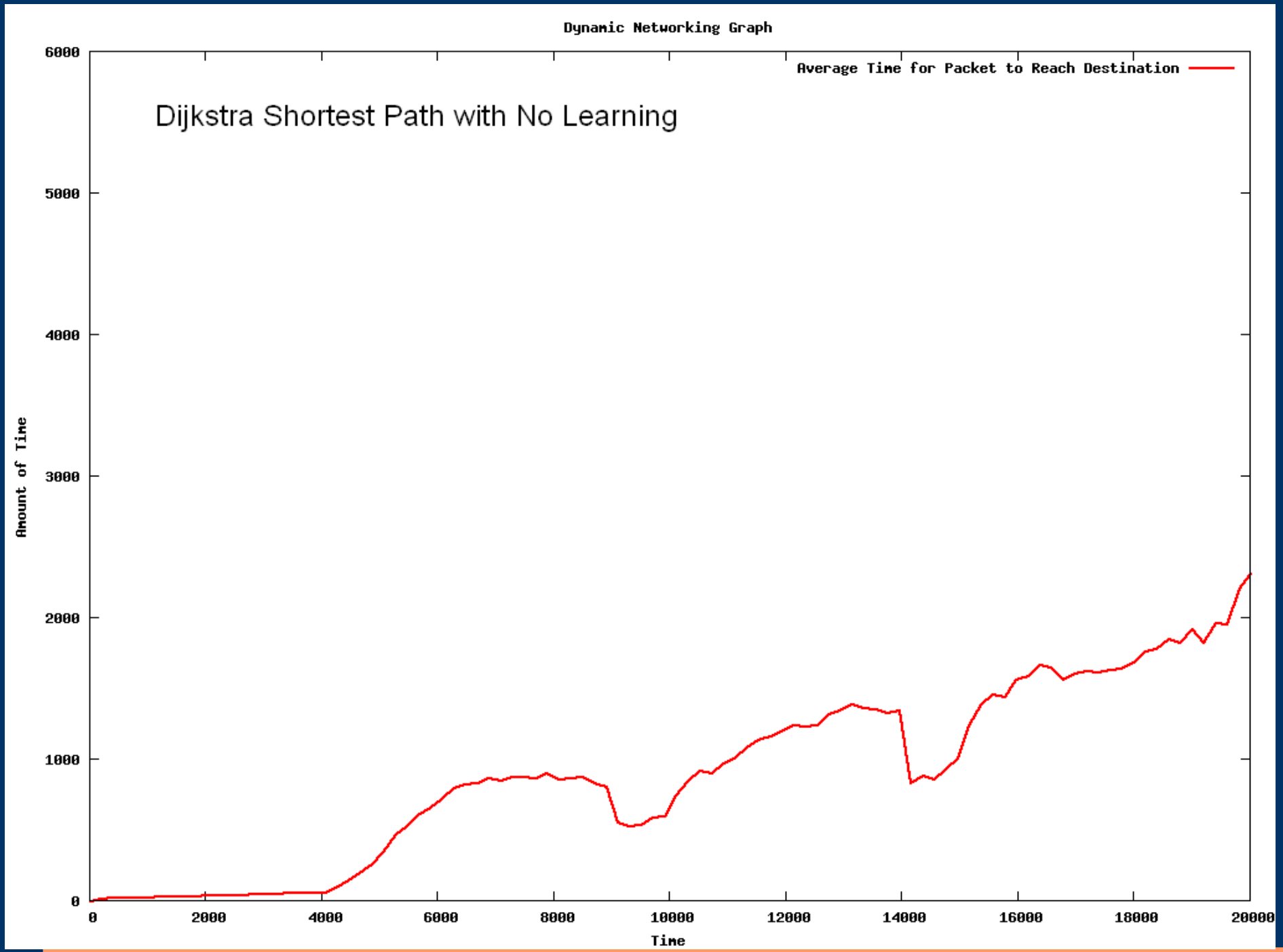

Dynamic Networking Graph
Number of Packets Created
Number of Packets Arrived at Destination
Dijkstra Shortest Path with No Learning
Packets
Time
0
5
10
15
20
25
30
35
0
2000
4000
6000
8000
10000
12000
14000

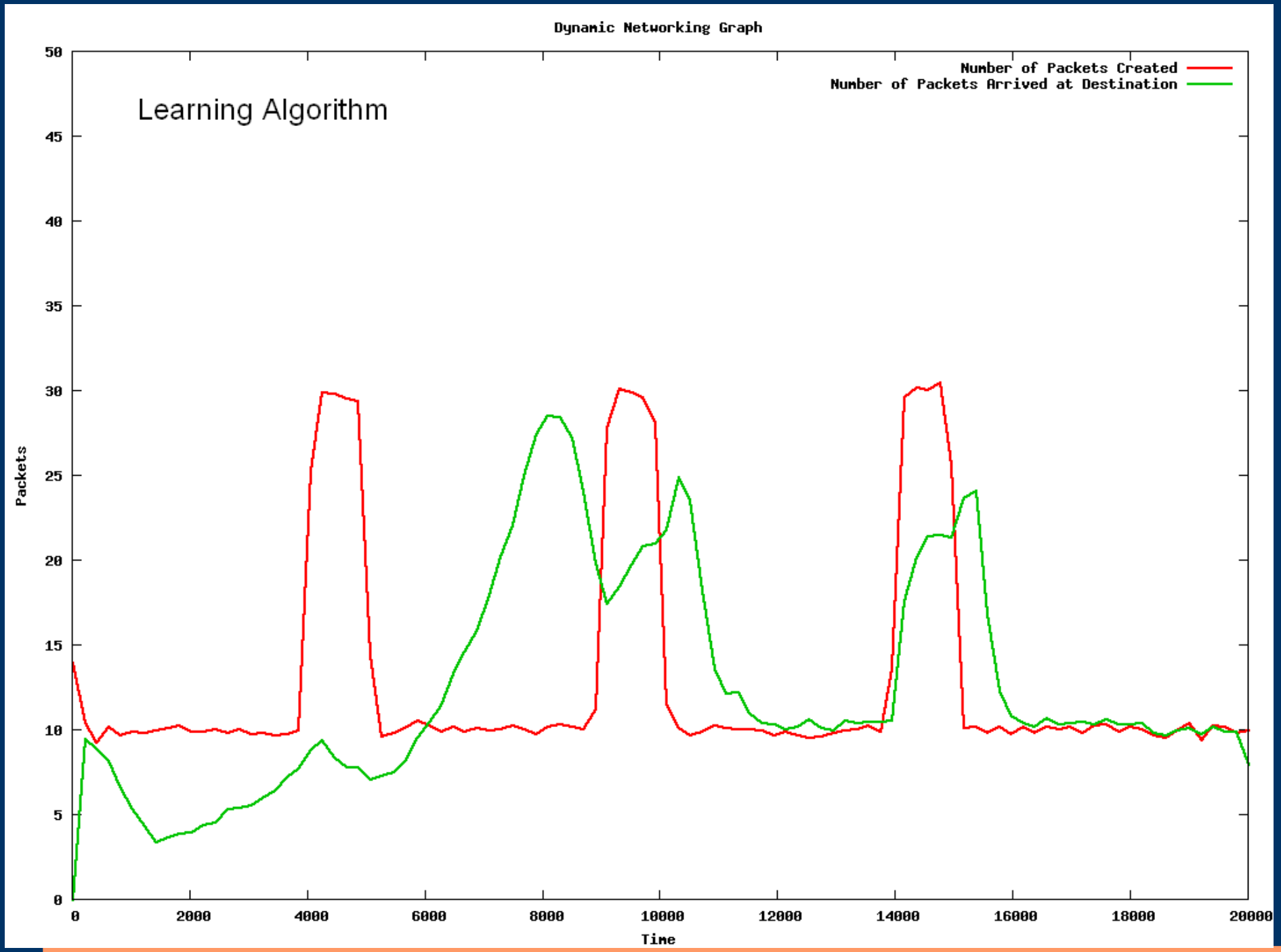

Dynamic Networking Graph
Average Time for Packet to Reach Destination
Dijkstra Shortest Path with No Learning
Amount of Time
Time
0
500
1000
1500
2000
2500
3000
3500
4000
0
2000
4000
6000
8000
10000
12000
14000

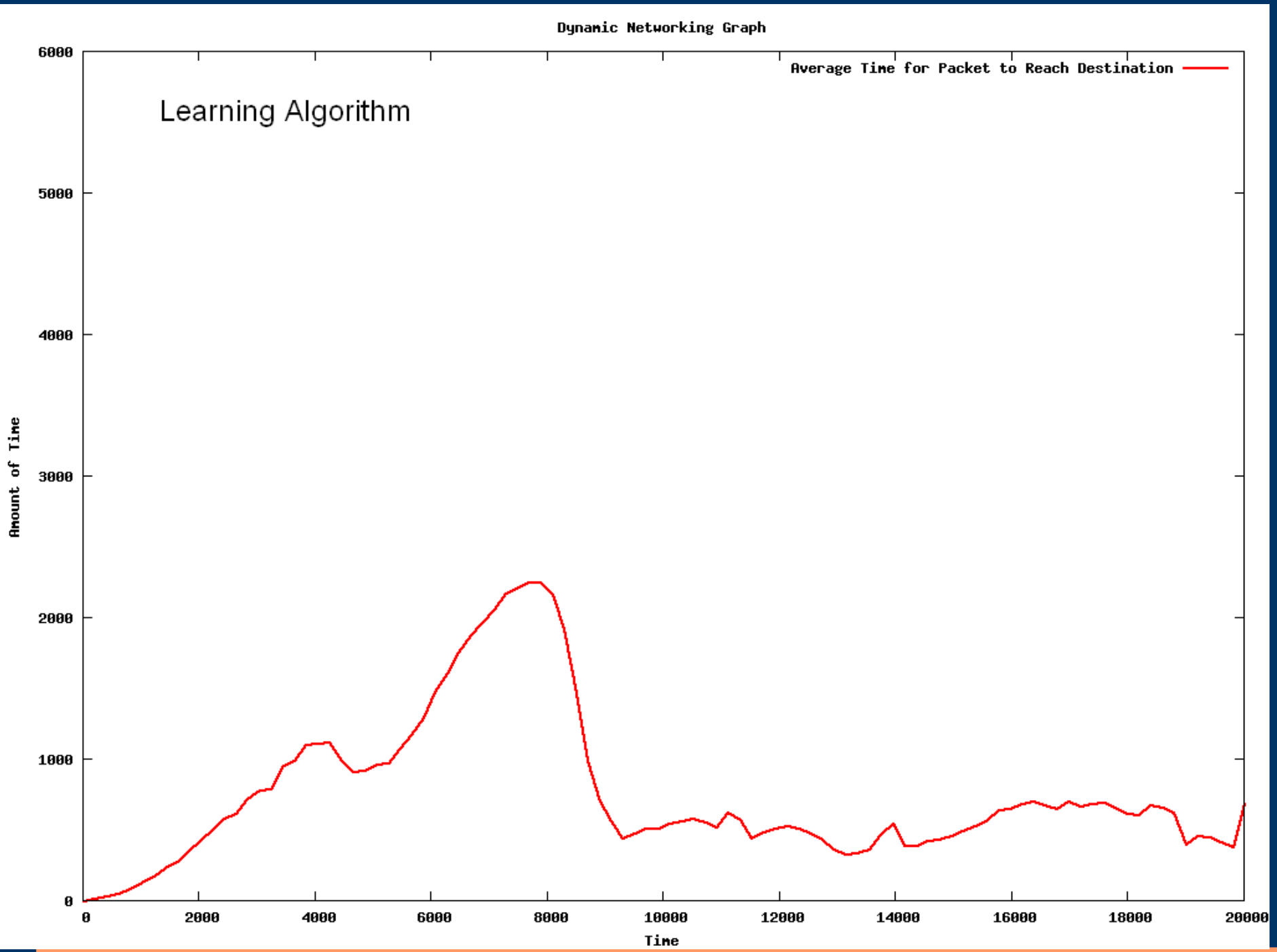

Dynamic Networking Graph
Learning Algorithm
Number of Packets Created
Number of Packets Arrived at Destination
Packets
Time
0
5
10
15
20
25
30
35
40
45
0
2000
4000
6000
8000
10000
12000
14000


Dynamic Networking Graph
Average Time for Packet to Reach Destination
Learning Algorithm
Amount of Time
Time
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
14000


Dynamic Networking Graph
Dijkstra Shortest Path with No Learning
Number of Packets Created
Number of Packets Arrived at Destination
Packets
Time
0
5
10
15
20
25
30
35
40
45
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000


Dynamic Networking Graph
Average Time for Packet to Reach Destination
Dijkstra Shortest Path with No Learning
Amount of Time
Time
0
1000
2000
3000
4000
5000
6000
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000


Dynamic Networking Graph
Learning Algorithm
Number of Packets Created
Number of Packets Arrived at Destination
Packets
Time
0
5
10
15
20
25
30
35
40
45
50
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000


Dynamic Networking Graph
Average Time for Packet to Reach Destination
Learning Algorithm
Amount of Time
Time
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000

Advantages of Learning Algorithm

- Fast in dynamic changing networks
- Works well under fairly high load
- Performs better than current shortest-path algorithm

Disadvantages of Learning Algorithm

- Requires storage of full network table, thus probably will not scale well
- Initial learning very costly

Future Work

- Change learning rate based on network load (exploration versus exploitation)
- Run on real network or simulator
- Distributed storage of Q-Tables
- Examine caching technology or P2P protocols like Bittorrent

Thanks To:

- NSF sponsored REU program (grant number 0453545)
- Dr. Terran Lane (UNM)
- Dr. Lydia Sinapova (Simpson College)
- Dr. Andrew H. Fagg
- Dr. Dean F. Hougen
- Dr. Amy McGovern
- Dennis, Will, and super-coder Laura

Thanks! Are there any questions?