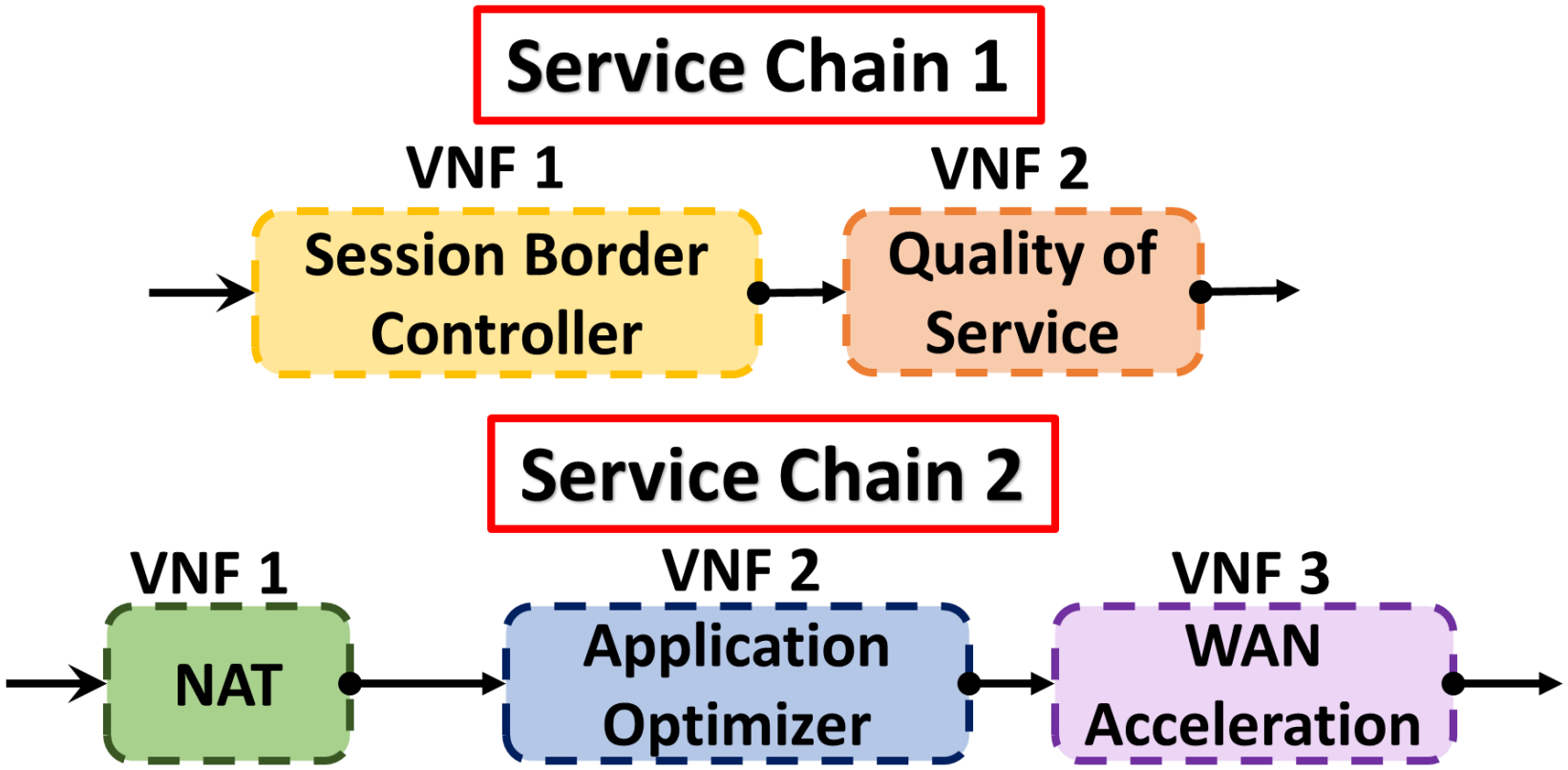
Deploying Multiple Service Chain (SC) Instances per Service Chain

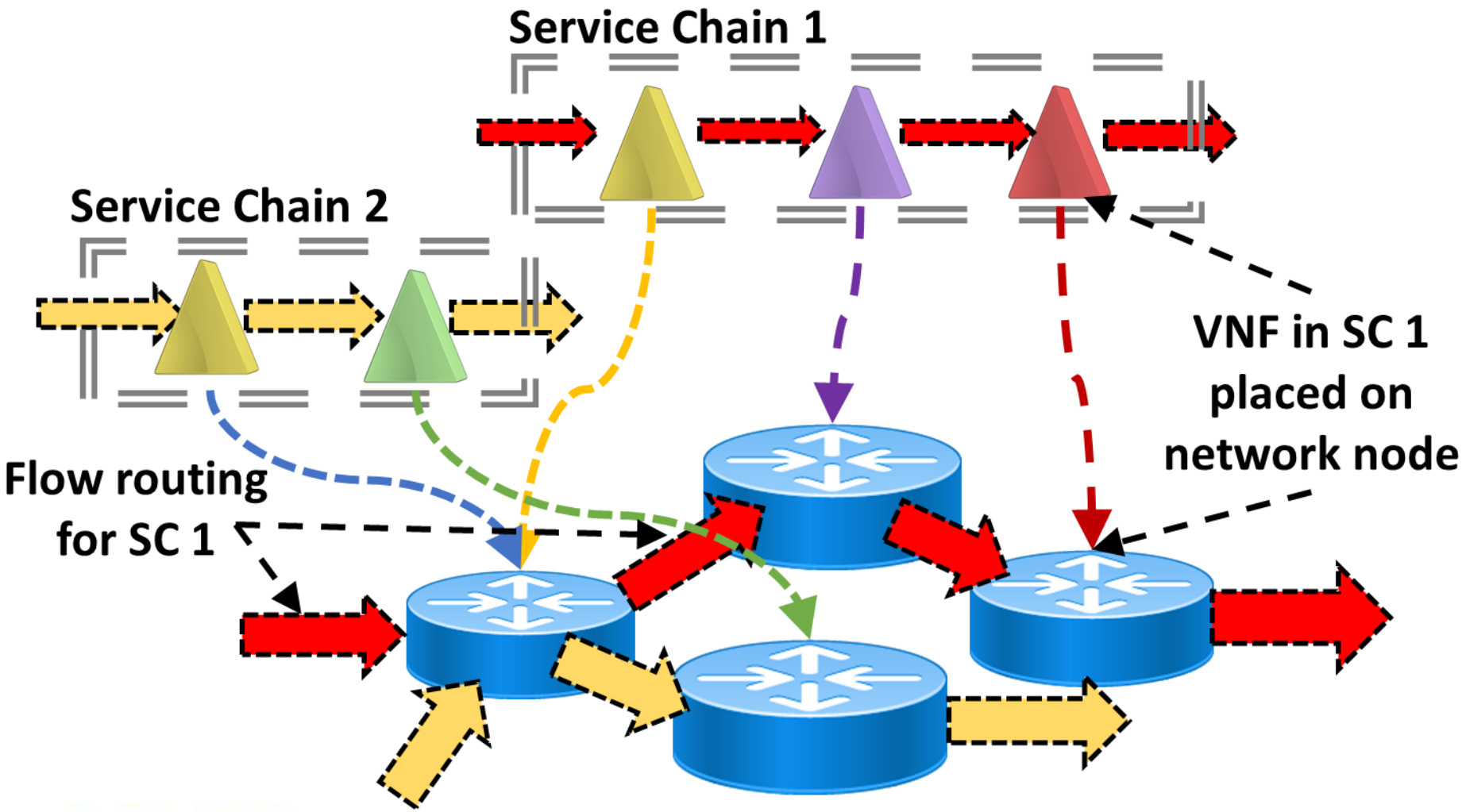
BY
ABHISHEK GUPTA
FRIDAY GROUP MEETING
APRIL 21, 2017

UCDAVIS

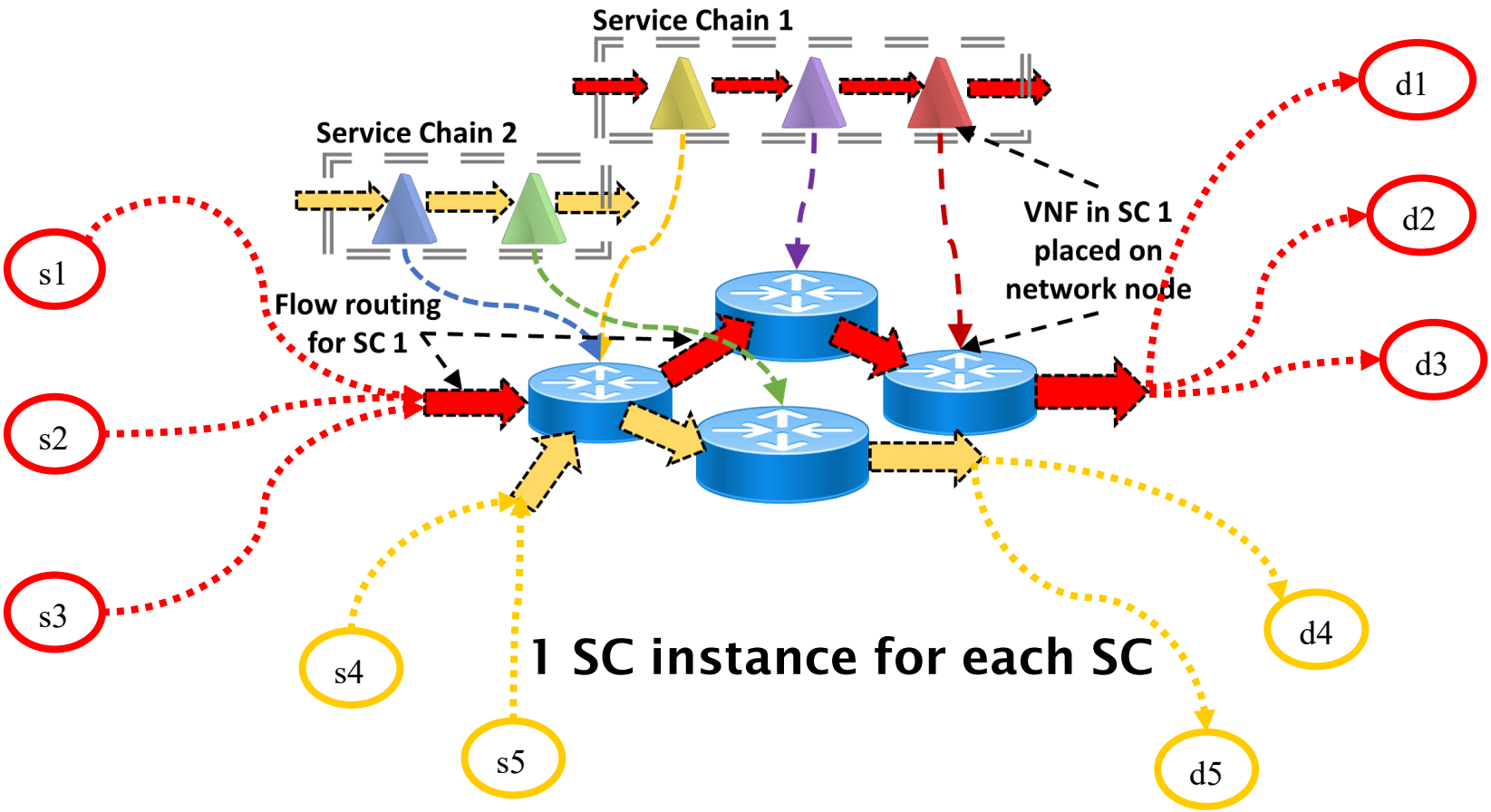
Virtual Network Function (VNF) Service Chain (SC)



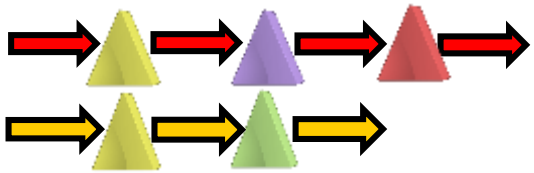
Multiple VNF SC Placement and Routing



Continued…


Service Chain 1
Service Chain 2
VNF in SC 1
placed on
network node
Flow routing
for SC 1
s1
s2
s3
s4
s5
d1
d2
d3
d4
d5
1 SC instance for each SC

Single Instance Per SC



Service Chain 1

Service Chain 2

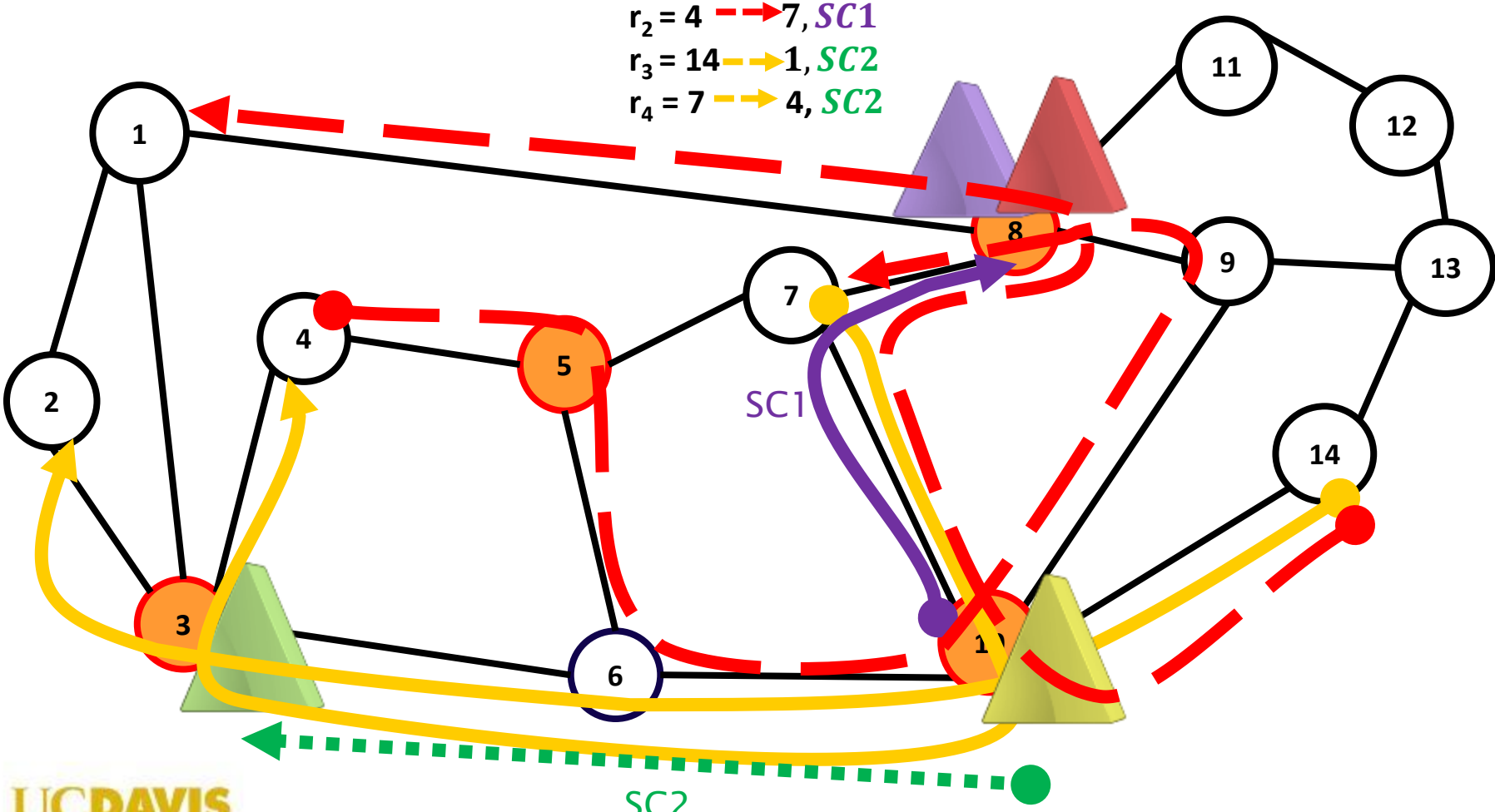
3 Demand Requests

r_1 = 14 → 1, $SC1$

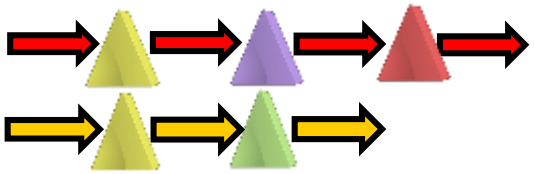
r_2 = 4 → 7, $SC1$

r_3 = 14 → 1, $SC2$

r_4 = 7 → 4, $SC2$



Multiple Instances per SC



Service Chain 1

Service Chain 2

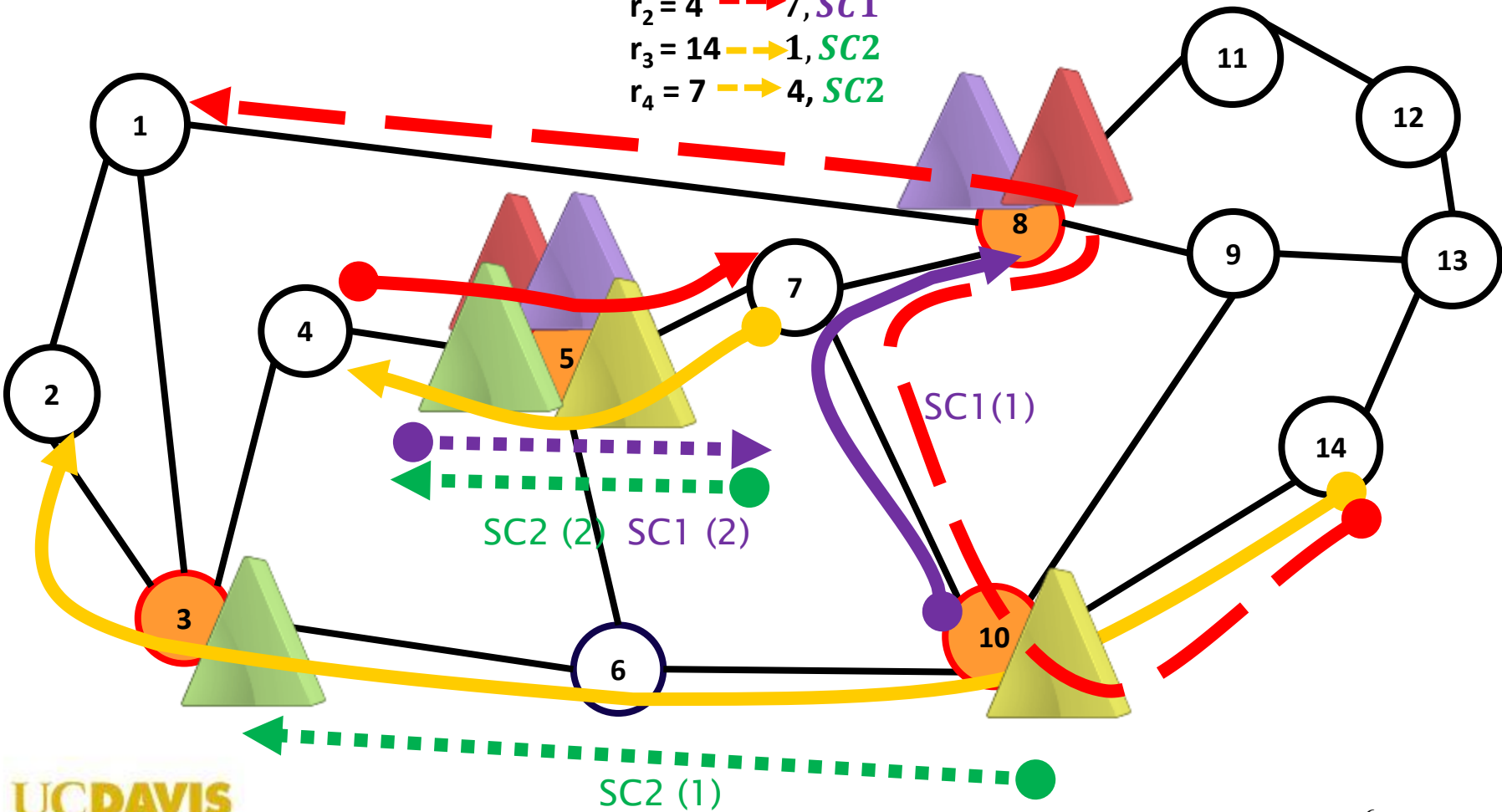
3 Demand Requests

r_1 = 14 → 1, $SC1$

r_2 = 4 → 7, $SC1$

r_3 = 14 → 1, $SC2$

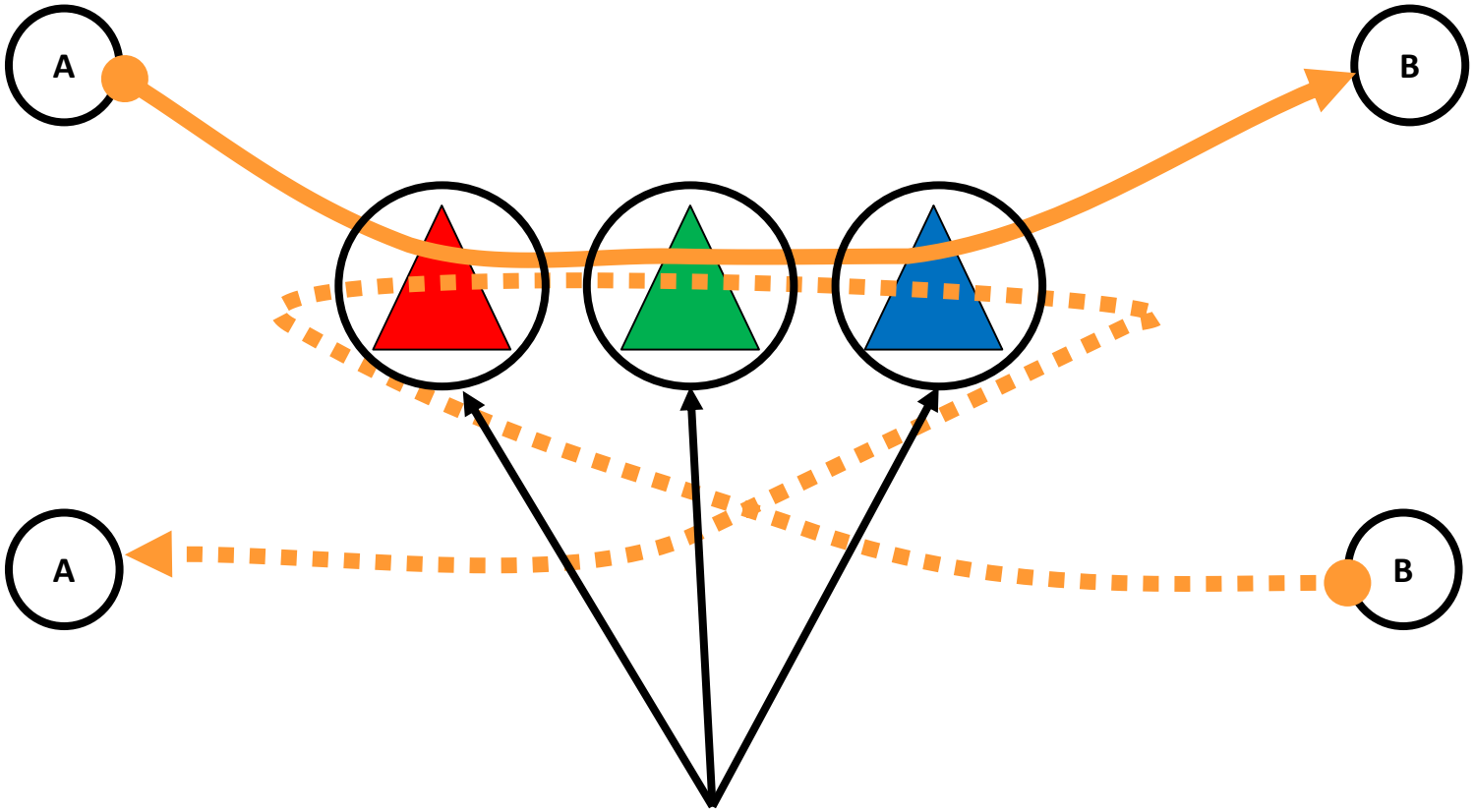
r_4 = 7 → 4, $SC2$



Inferences and Questions

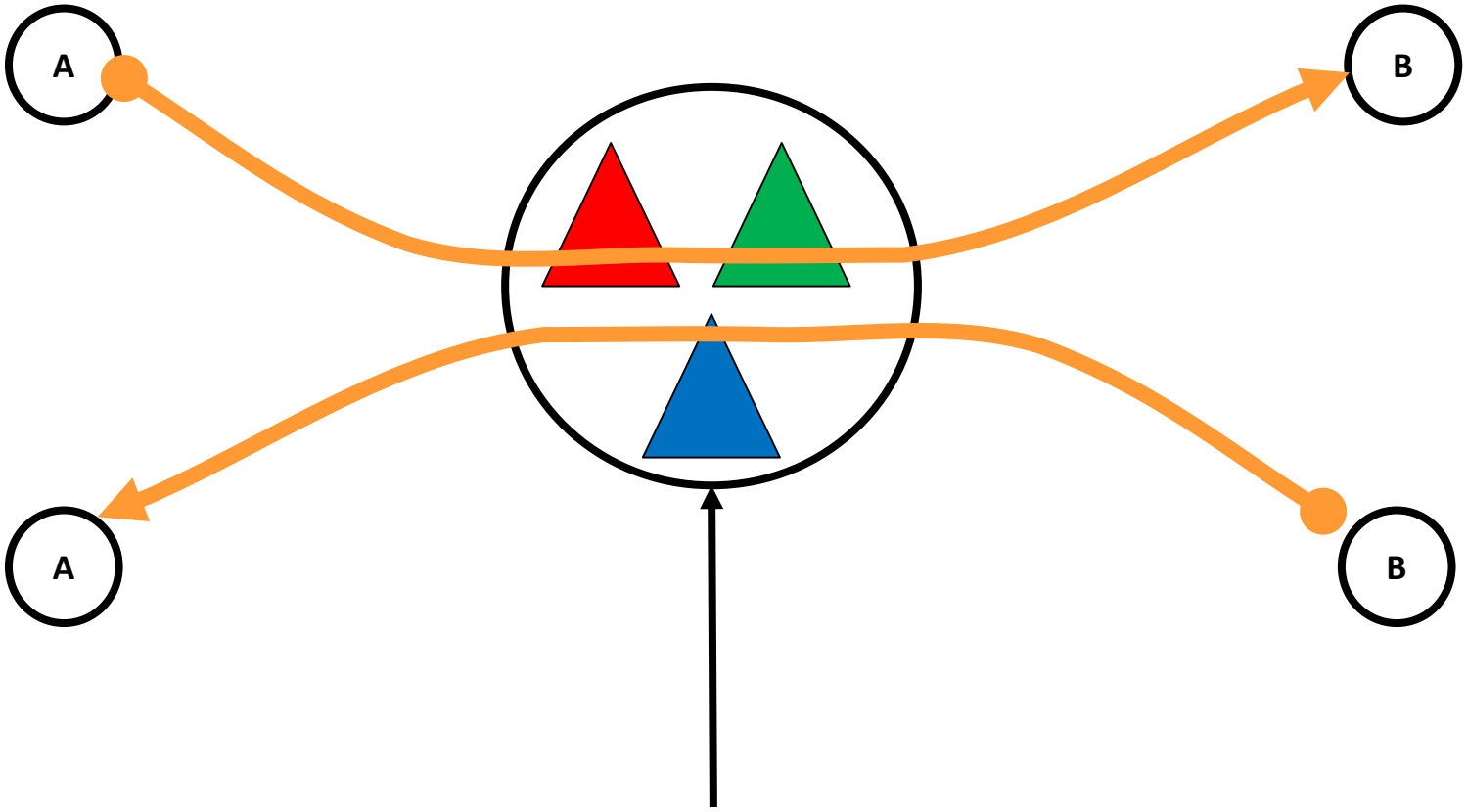
- 1 SC instance per SC leads to suboptimal results
- Having SC instances replicated on every node will lead to to optimal results
  - Large capital expenditure to make all nodes NFV capable
  - High Orchestration Overhead for large number of instances
- The question therefore becomes:
  - **How many SC instances to deploy to reduce bandwidth consumption while also reducing nodes used?**
- **We develop a heuristic to help us chose the right number of instances (SPTG)**

Issue of symmetric flows



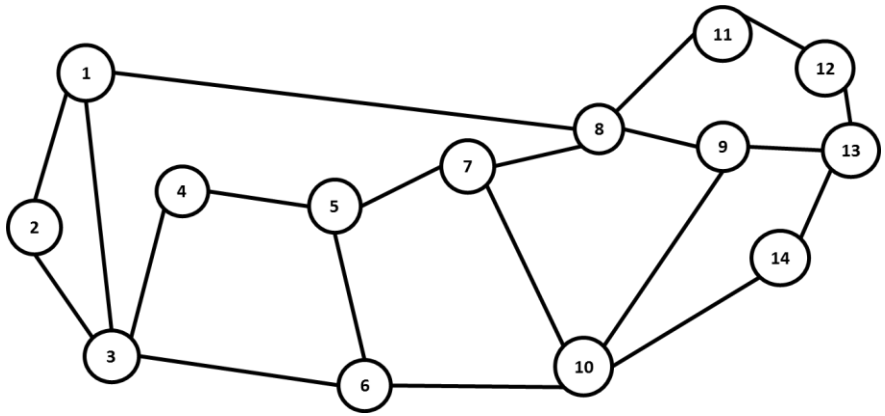
VNFs placed in different nodes

Continued…

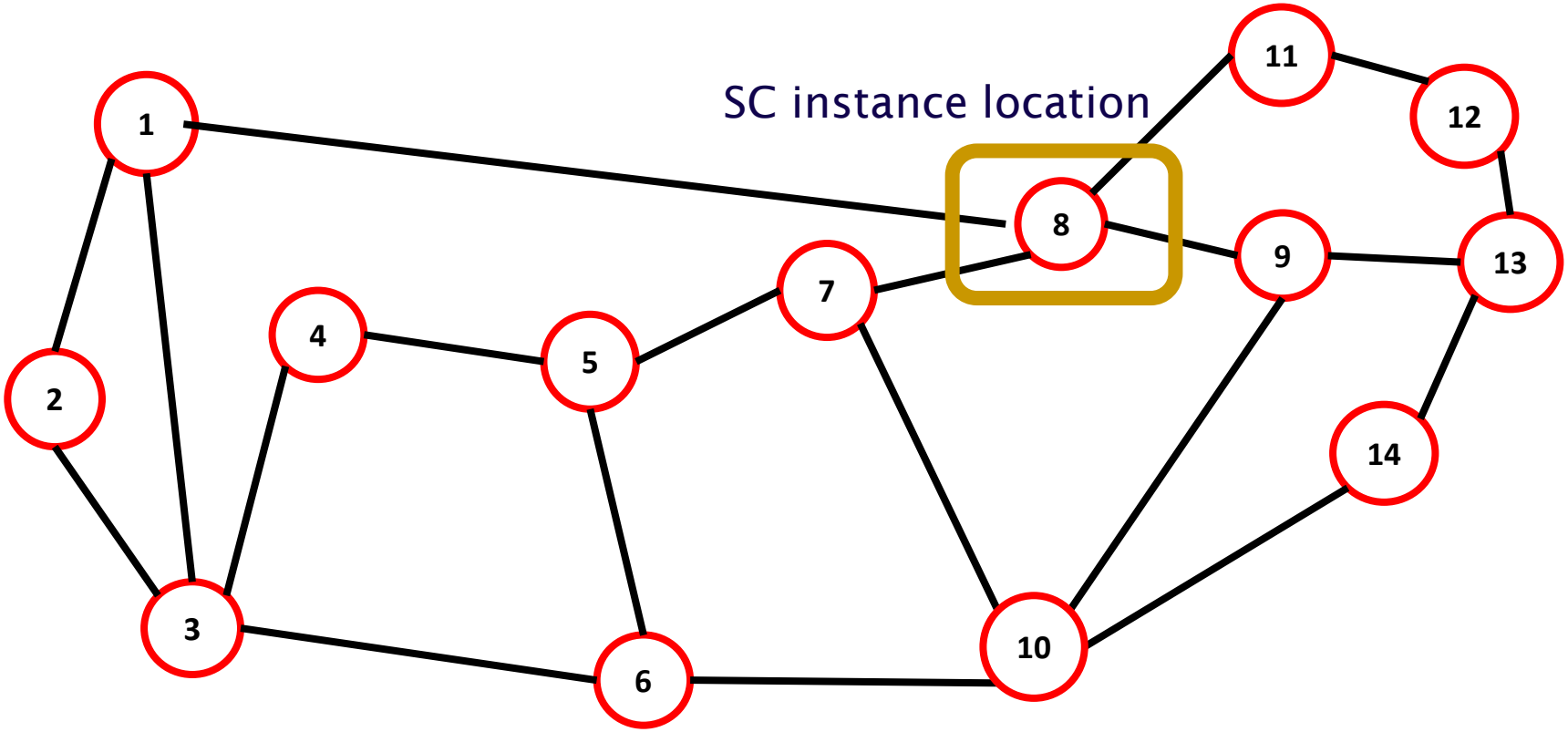


Continued…

- Placing VNFs for SC at different nodes
  - makes symmetric flow take longer path
- Placing VNFs for SC at one node
  - symmetric flow takes shorter path
  - placement and routing becomes trivial
  - chaining aspect is forgone
  - Is this more realistic?
  - Represents the case of a DC

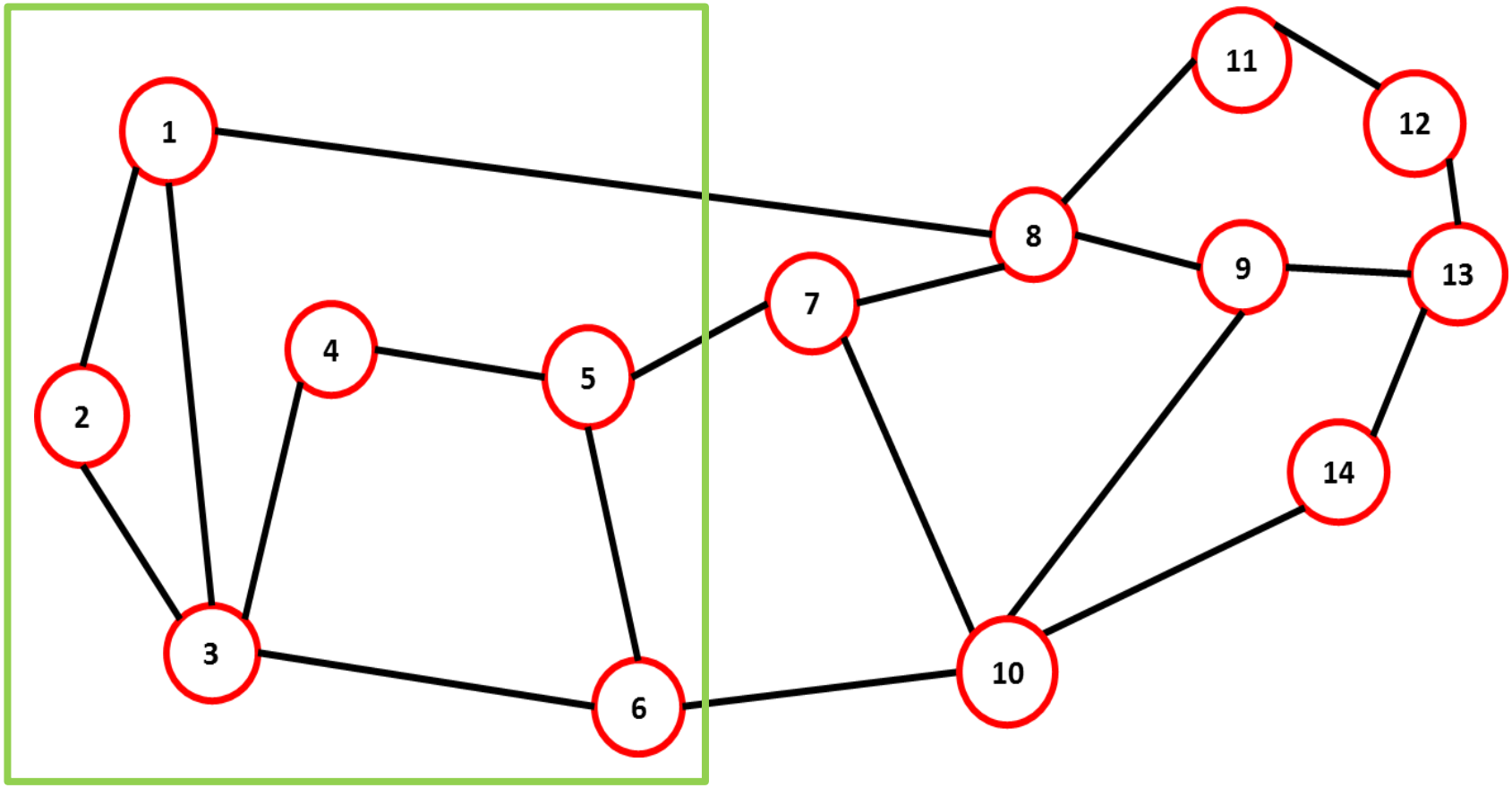


Full Traffic Matrix, 1 SC deployment, 1 SC instance



All nodes are NFV-capable.
All node pairs have requests for the same service chain.

Grouping of traffic pairs

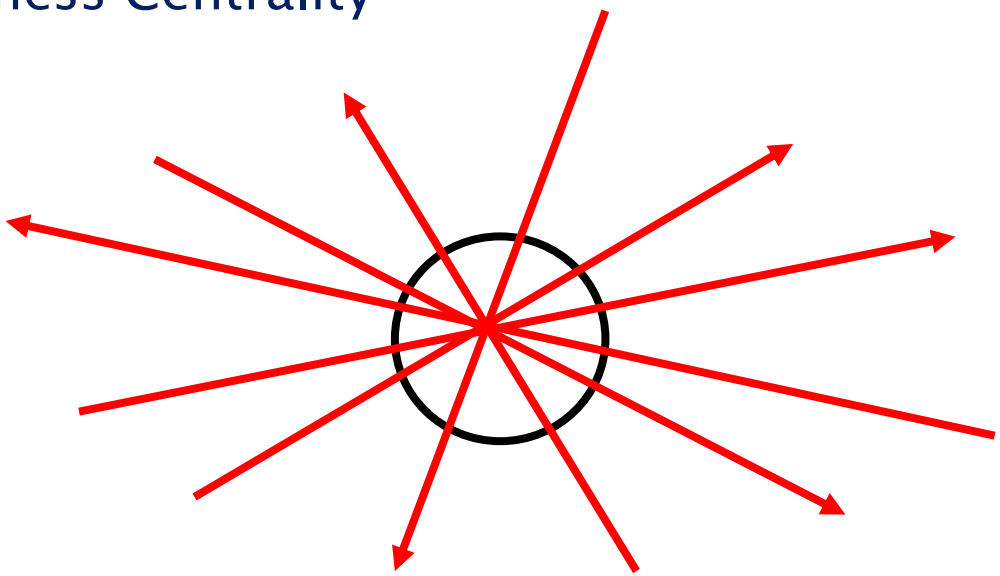


Continued …

- Create traffic flow groups
- Assign dummy SC Id's to traffic flow groups
- Big Question: **How to do we make traffic groups?**
- Model accounting for traffic groups becomes quadratic. Subsequent, linearization reduced the scalability of the model
- We, therefore, use a heuristic to do make the traffic groups

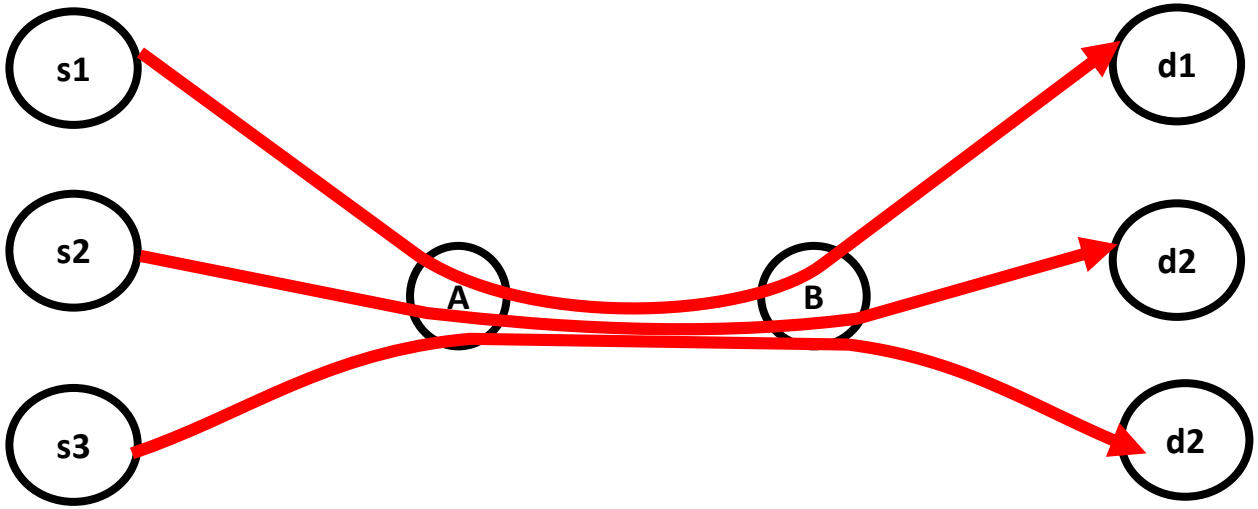
Grouping traffic flows around a node

- Betweenness Centrality



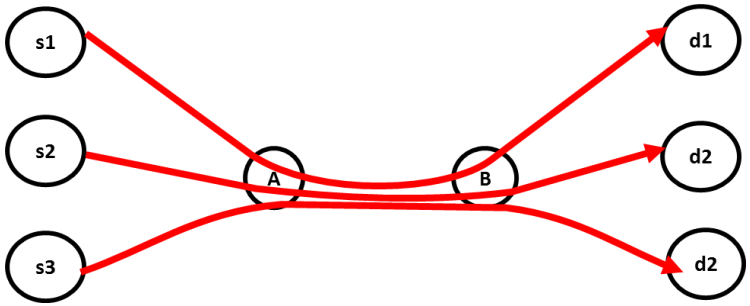
Group around node pairs of the graph

- A and B can also be source and destination
- Done for each SC



Continued…

- Ordered node pair with highest traffic flow count passing through on shortest paths



- Traffic flows which share sub-paths in common

- Deploying one SC instance for each such group

Shortest Path Traffic Grouping (SPTG)

- Given: the number of instances for a SC, all node pairs in a graph G

- The heuristic will:
  1. Find the node pair with the largest number of (s, d) pairs
  2. This becomes another (s, d) pair group
  3. All the (s, d) pairs in the group are removed from the global (s, d) pair list
  4. Repeat step 1 to 3 until number of instances is reached
  5. Iterate through the remaining (s, d) pairs:
     1. Find best group based on which path length through node pair
     2. Add (s,d) pair to that group

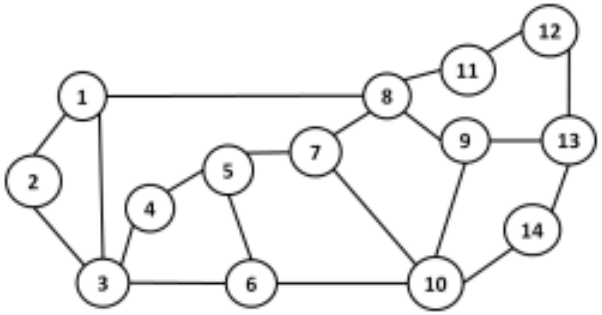
2 phase model

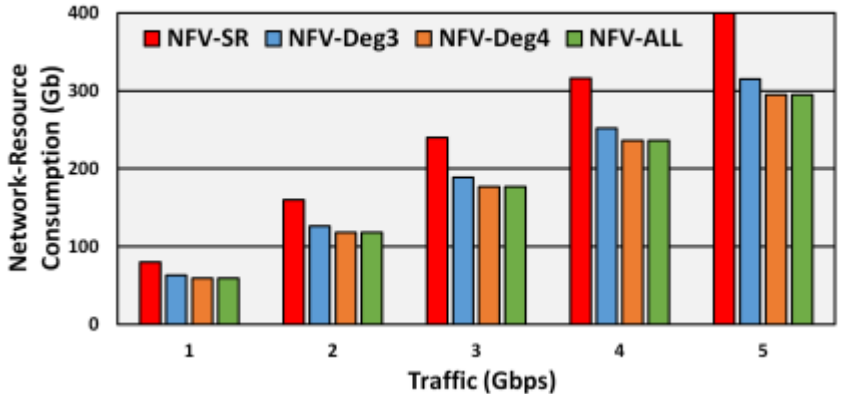
- 1st phase
  - Apply SPTG for each SC and create the required number of groups
  - Assign dummy SC ids to groups of (s,d) pairs

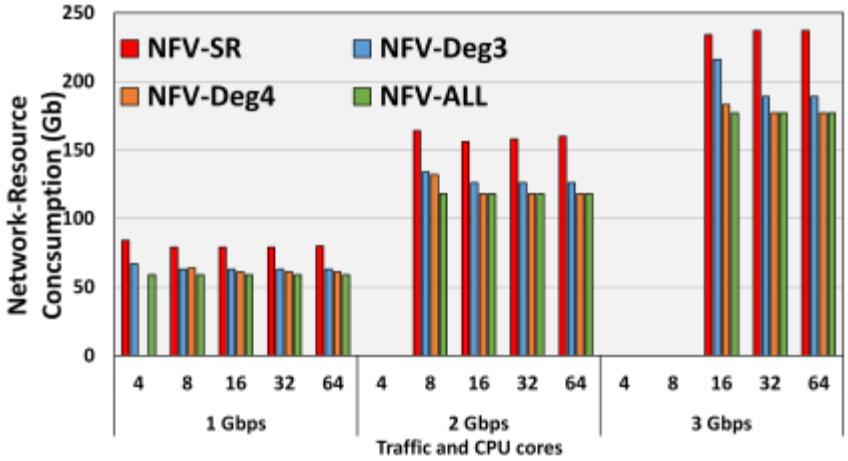
- 2nd phase
  - Use the columen generation model which decides on 1 SC instance per SC
  - ***Also we can control the number of nodes that can host VNFs, we refer to this number as 'K'***

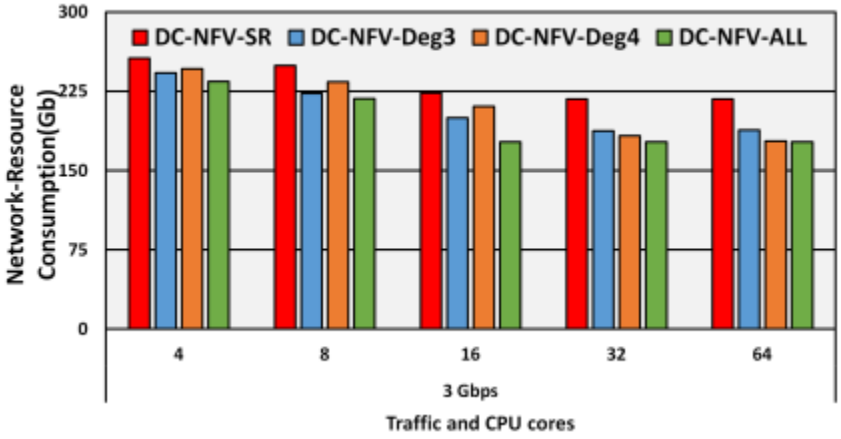
Previous Results

- 20 uniformly distributed traffic flows, 13 service chains, and 33 VNFs.







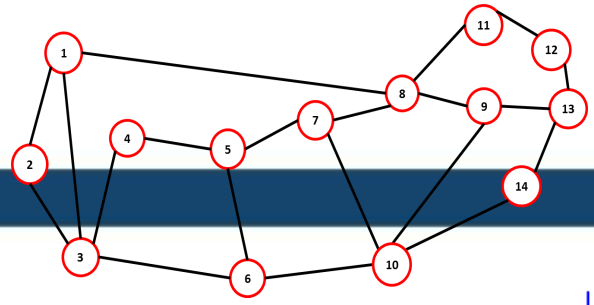


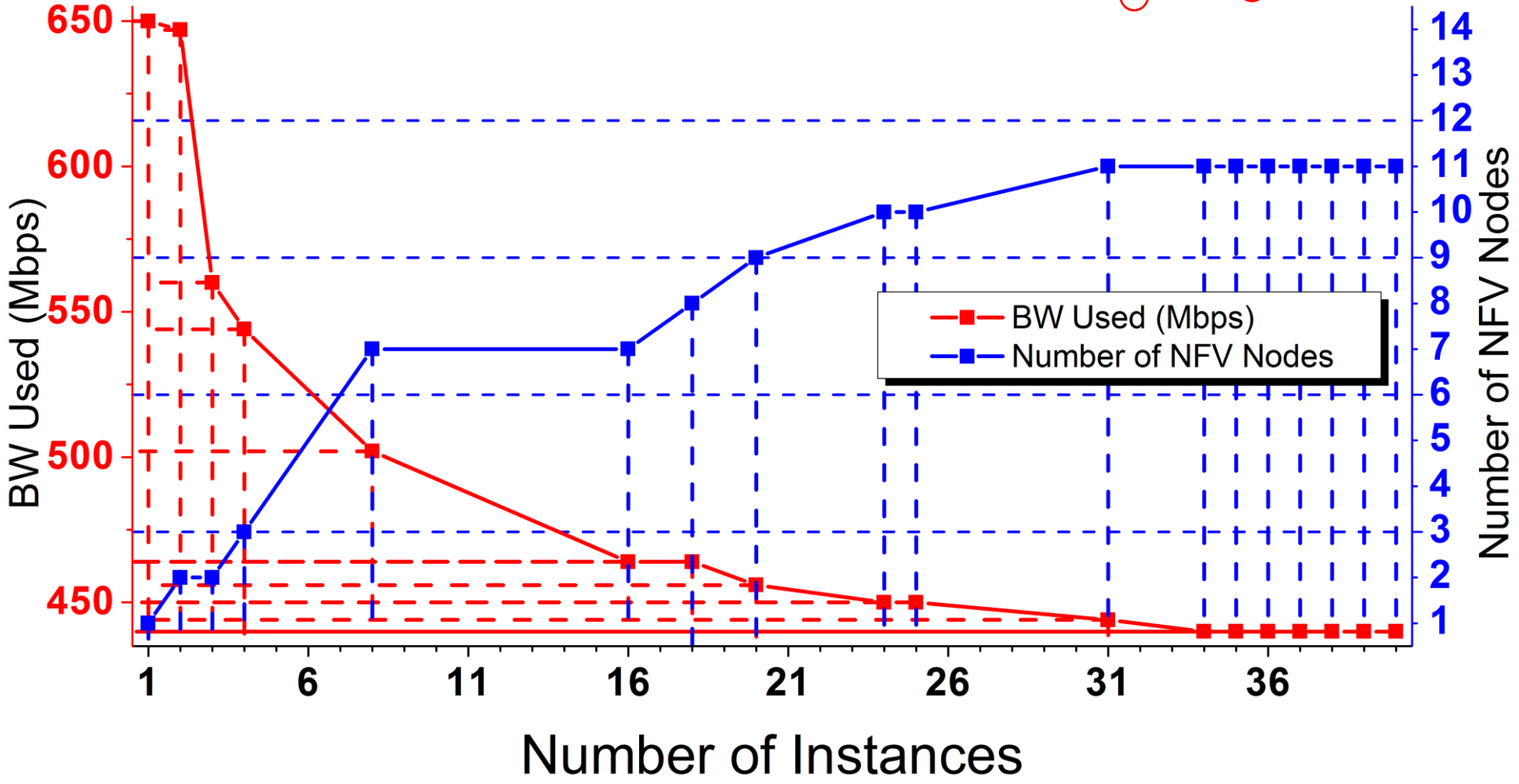
UCDAVIS
UNIVERSITY OF CALIFORNIA

Assumptions

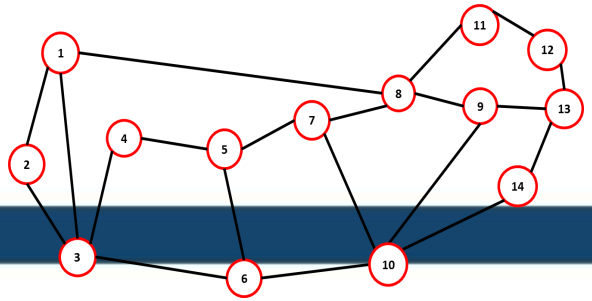
- All nodes are capable of hosting VNFs
- No CPU constraints are enforced
- No link capacity constraints are enforced
- Only one SC instance per SC model
- All traffic pairs have 1Gb traffic flow

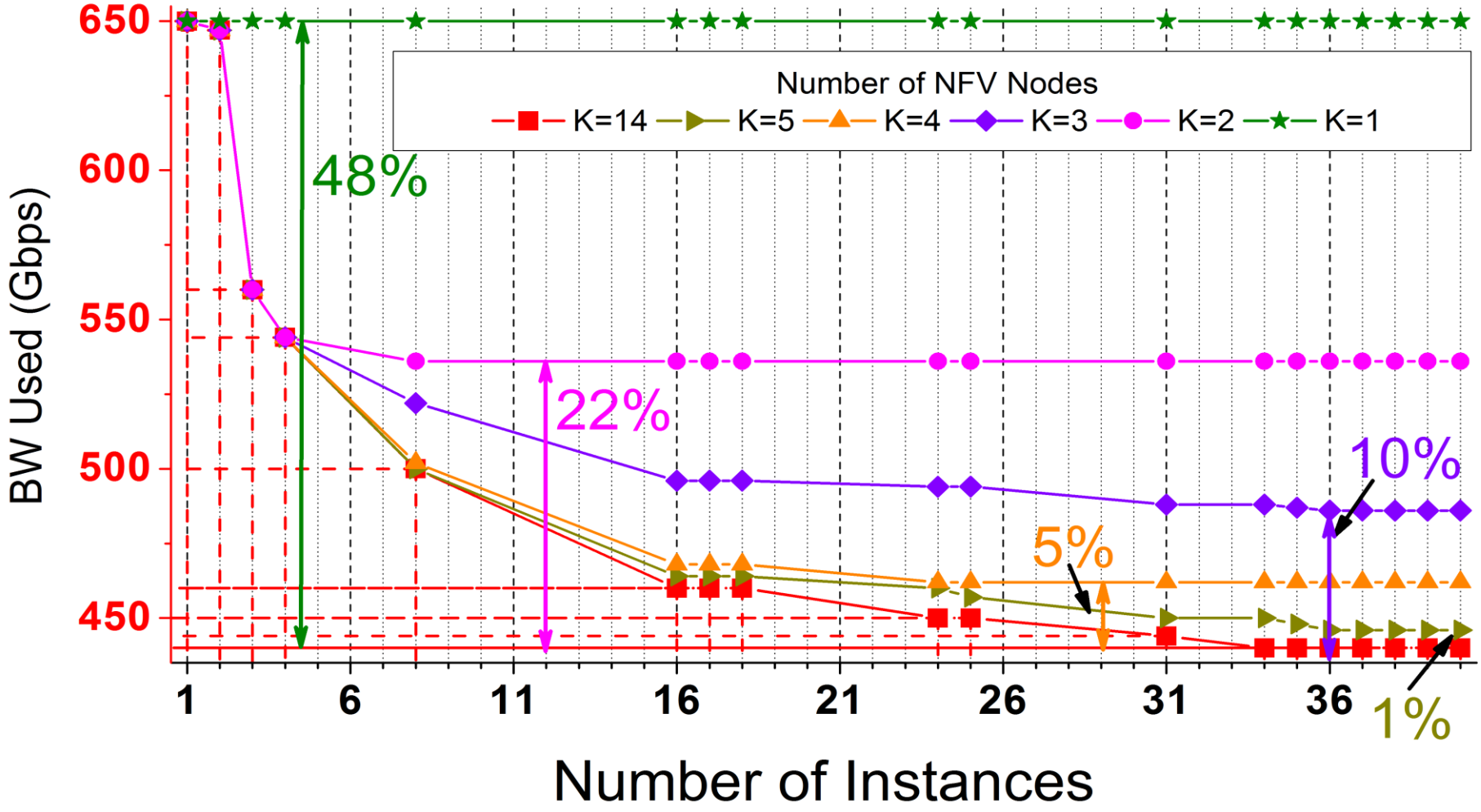
NSFNET K=14 (subset)





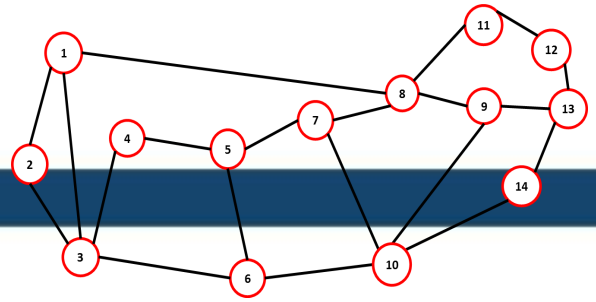
NSFNET K=14,5,4,3,2,1 (subset)

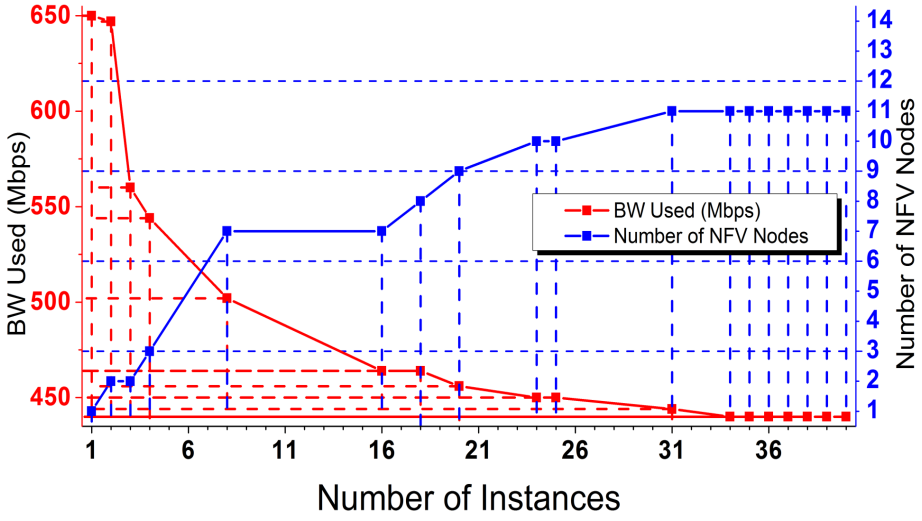


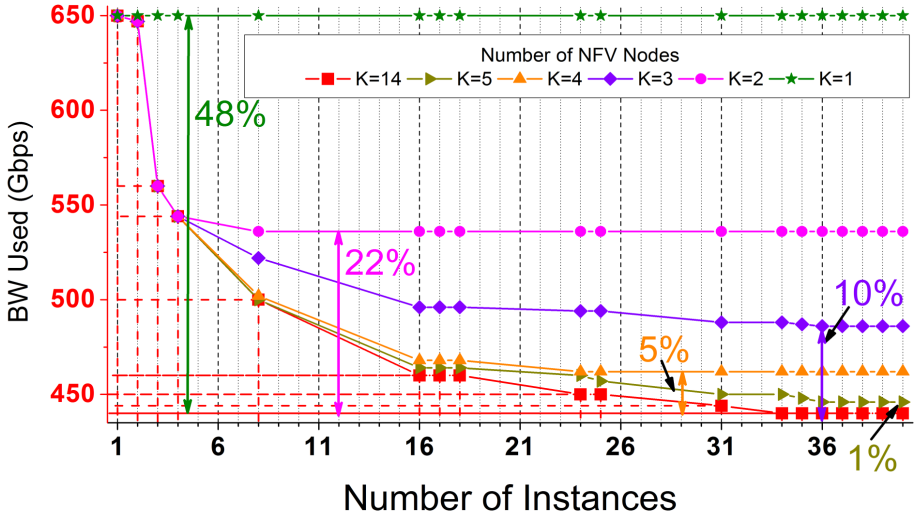


UCDAVIS
UNIVERSITY OF CALIFORNIA

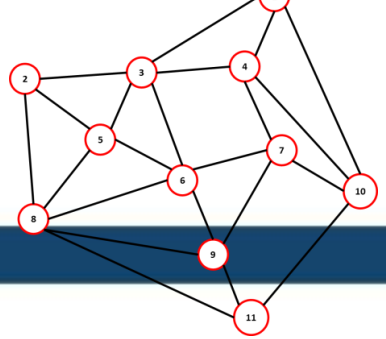
NSFNET

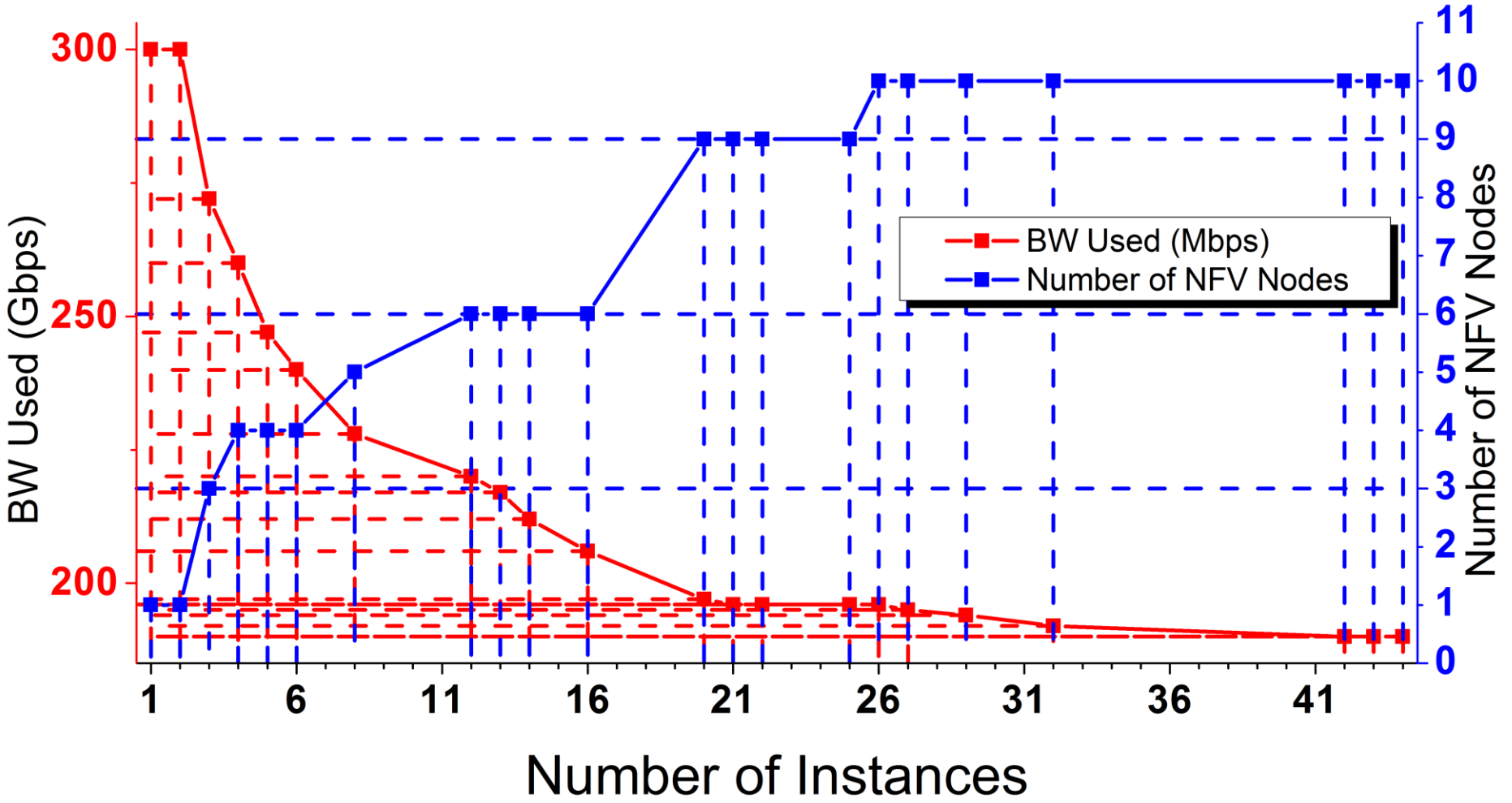

1
2
3
4
5
6
7
8
9
10
11
12
13
14


BW Used (Mbps)
Number of NFV Nodes
650
600
550
500
450
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1
6
11
16
21
26
31
36
Number of Instances

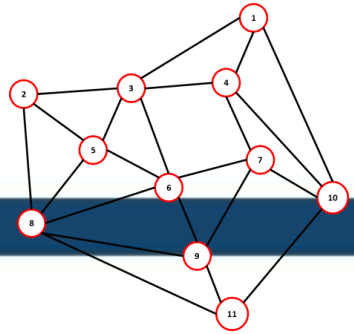

Number of NFV Nodes
K=14
K=5
K=4
K=3
K=2
K=1
48%
22%
10%
5%
1%
BW Used (Gbps)
650
600
550
500
450
1
6
11
16
21
26
31
36
Number of Instances

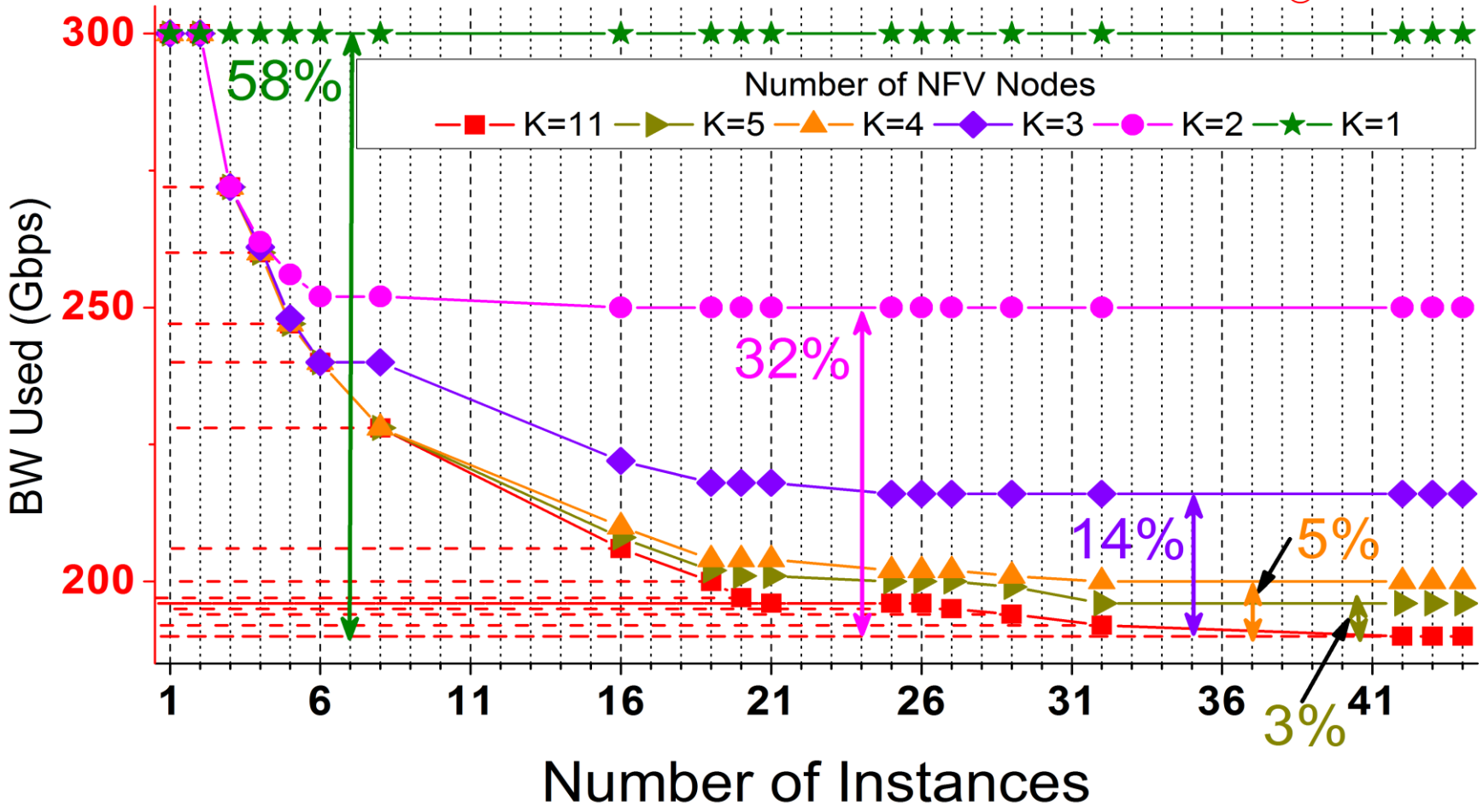
COST239 K=11 (subset)





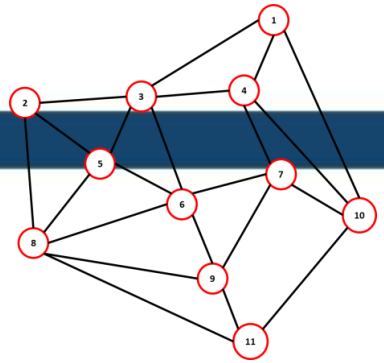
COST239 K=11,5,4,3,2,1 (subset)

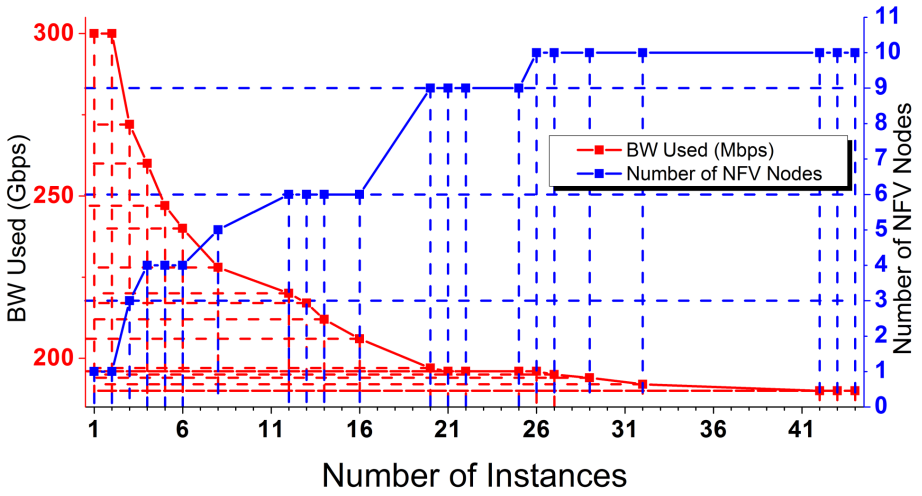


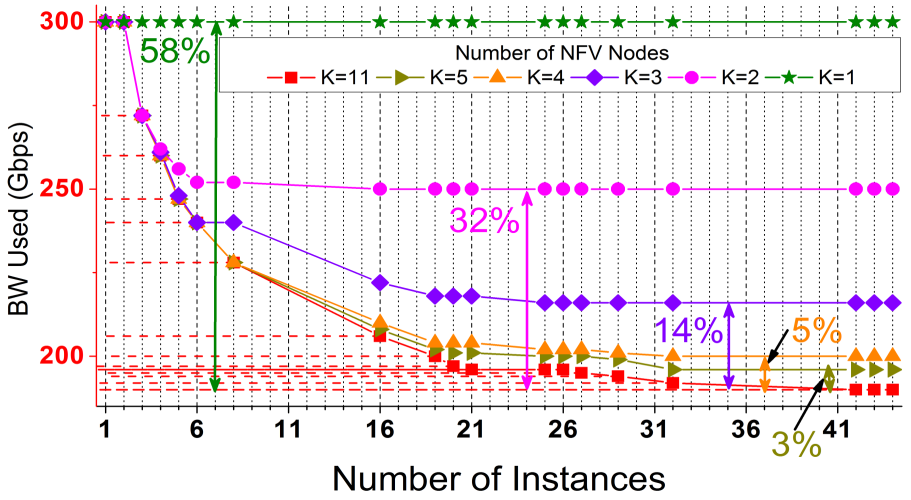


UCDAVIS
UNIVERSITY OF CALIFORNIA

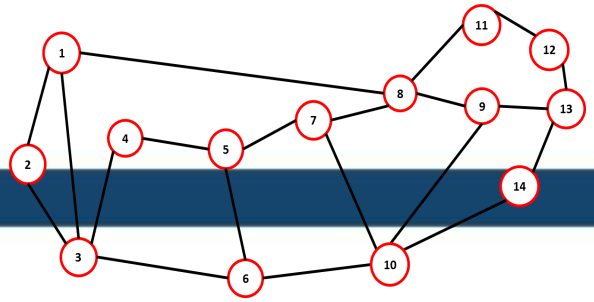
COST239

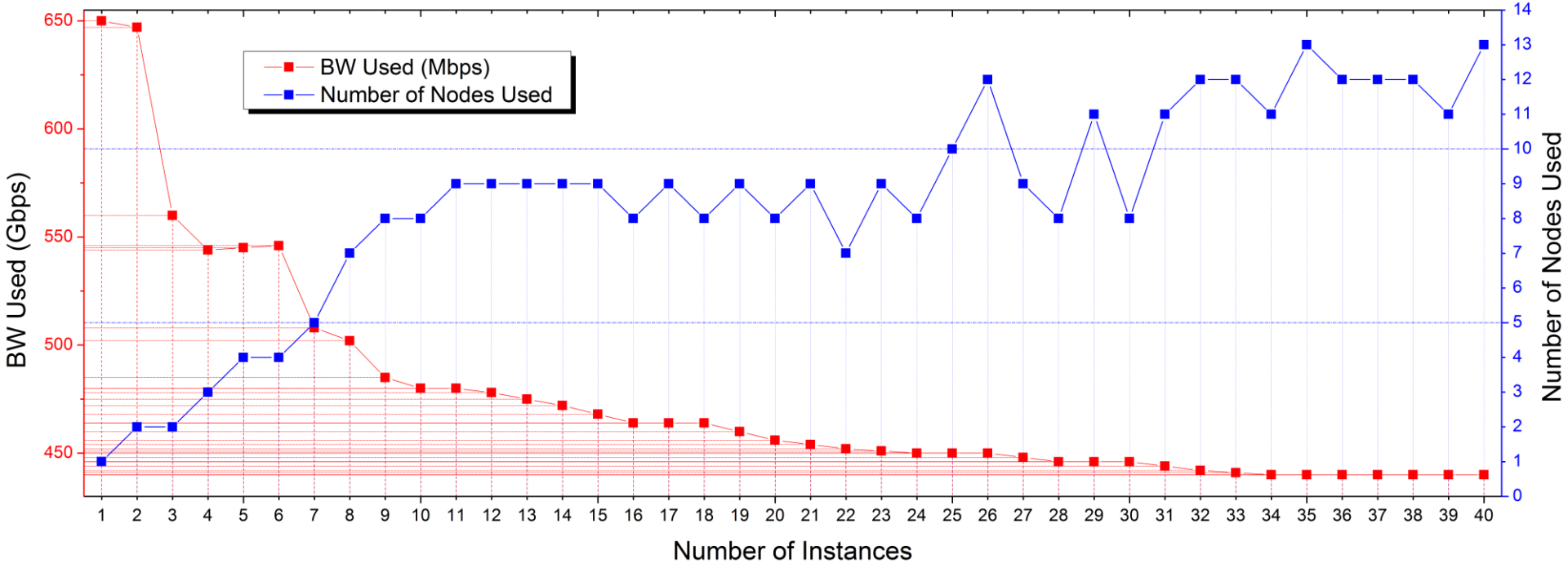

1
2
3
4
5
6
7
8
9
10
11


BW Used (Gbps)
Number of NFV Nodes
BW Used (Mbps)
Number of NFV Nodes
Number of Instances

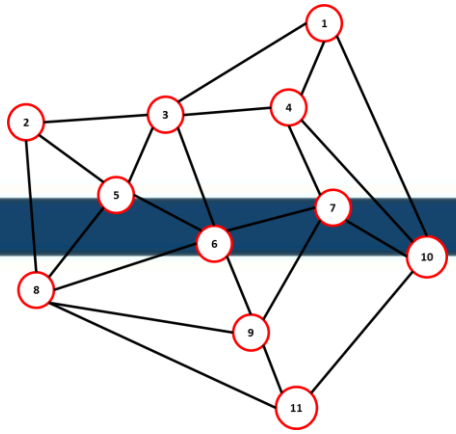

BW Used (Gbps)
Number of NFV Nodes
K=11
K=5
K=4
K=3
K=2
K=1
58%
32%
14%
5%
3%
Number of Instances

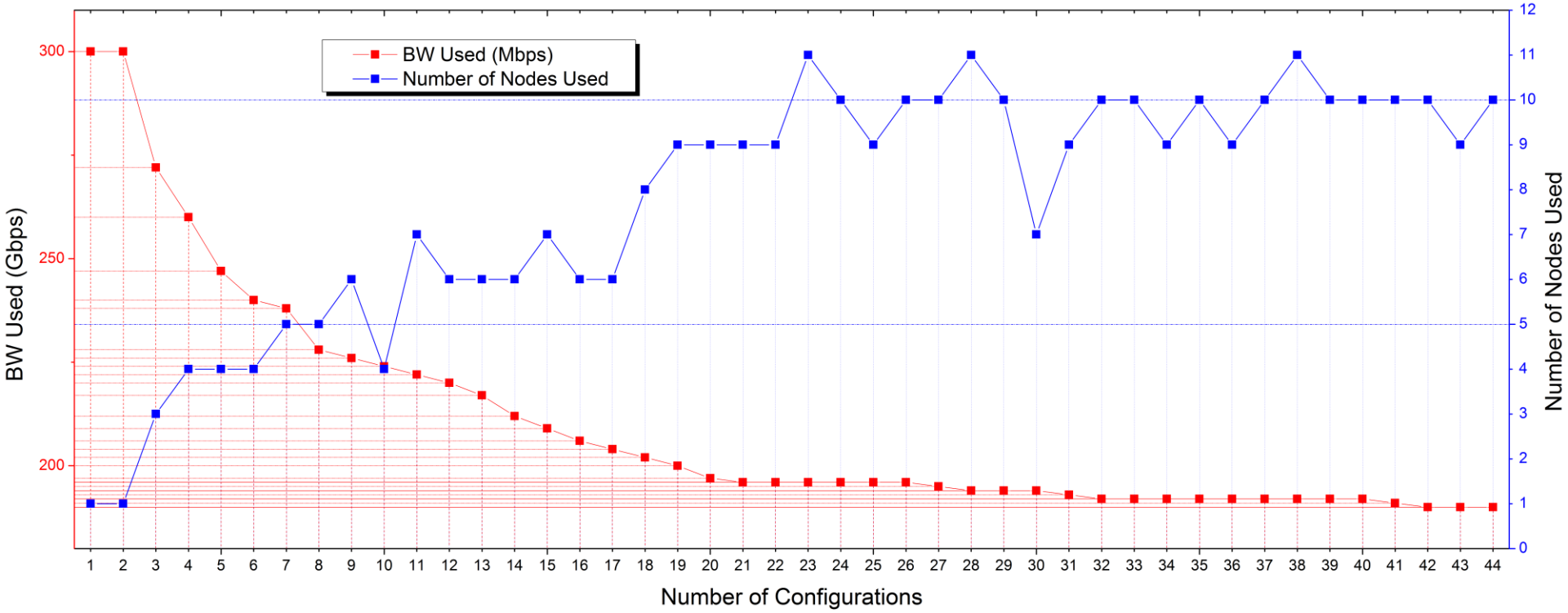
NSFNET K=14


1
2
3
4
5
6
7
8
9
10
11
12
13
14


BW Used (Mbps)
Number of Nodes Used
BW Used (Gbps)
Number of Nodes Used
Number of Instances
650
600
550
500
450
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

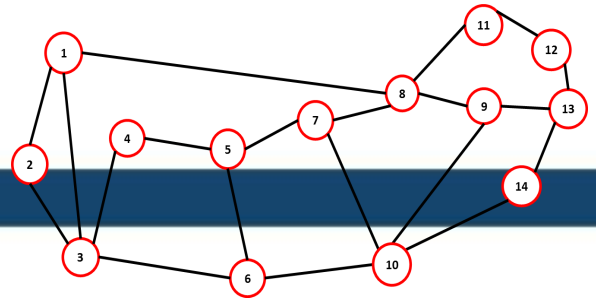
COST239 K=11

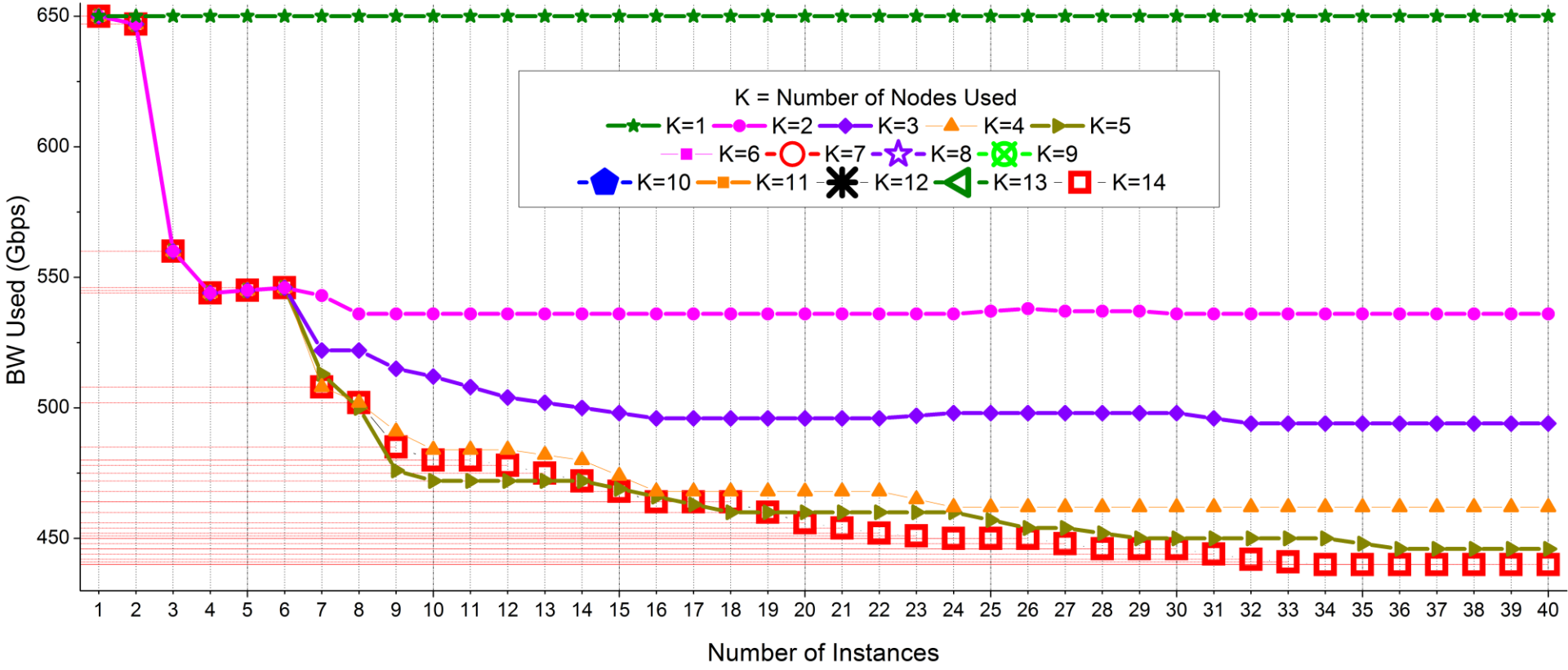




UCDAVIS
UNIVERSITY OF CALIFORNIA

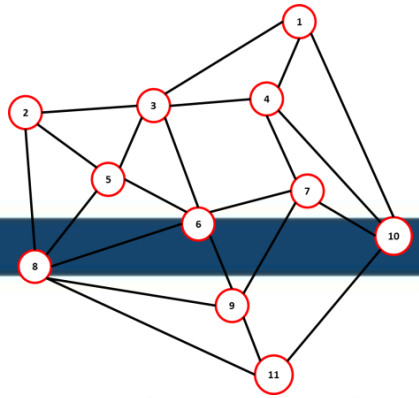
NSFNET K=14,5,4,3,2,1

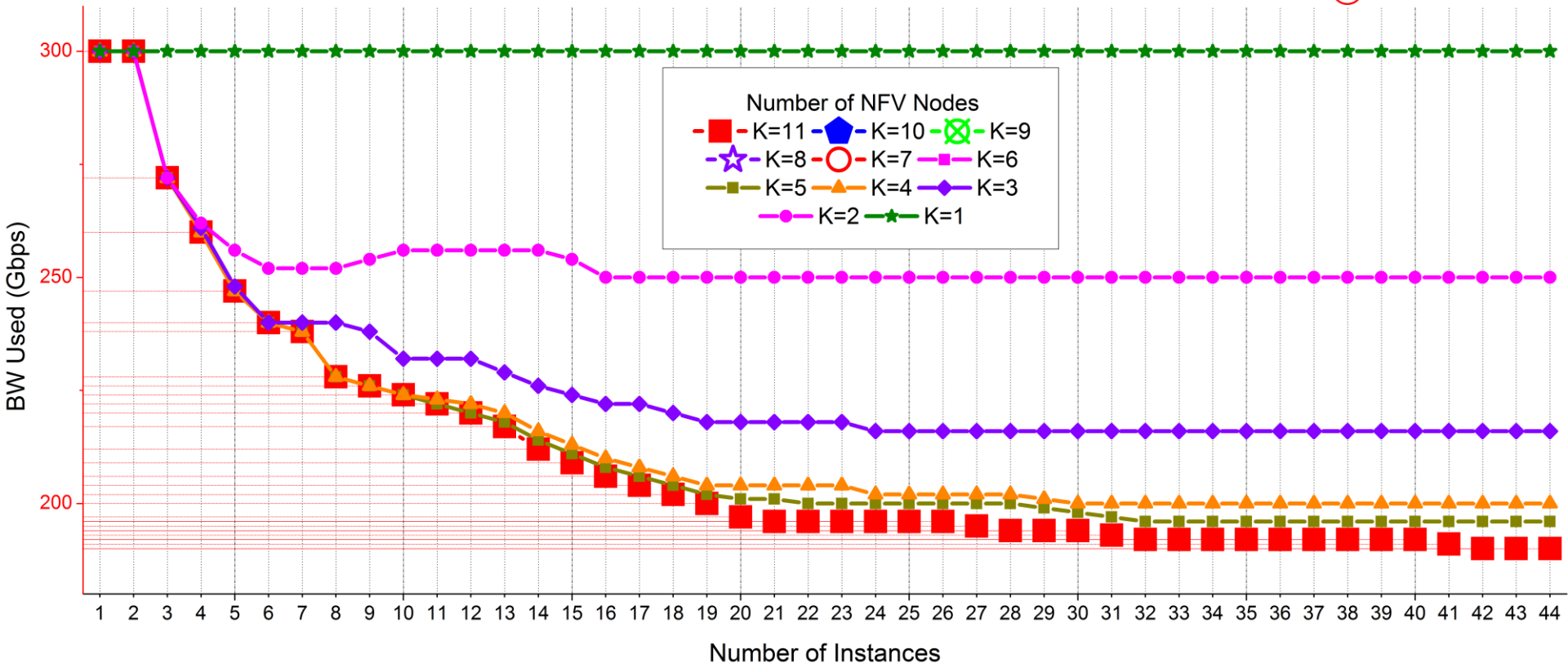




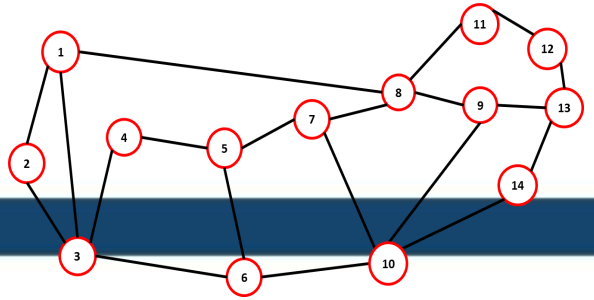
UCDAVIS
UNIVERSITY OF CALIFORNIA

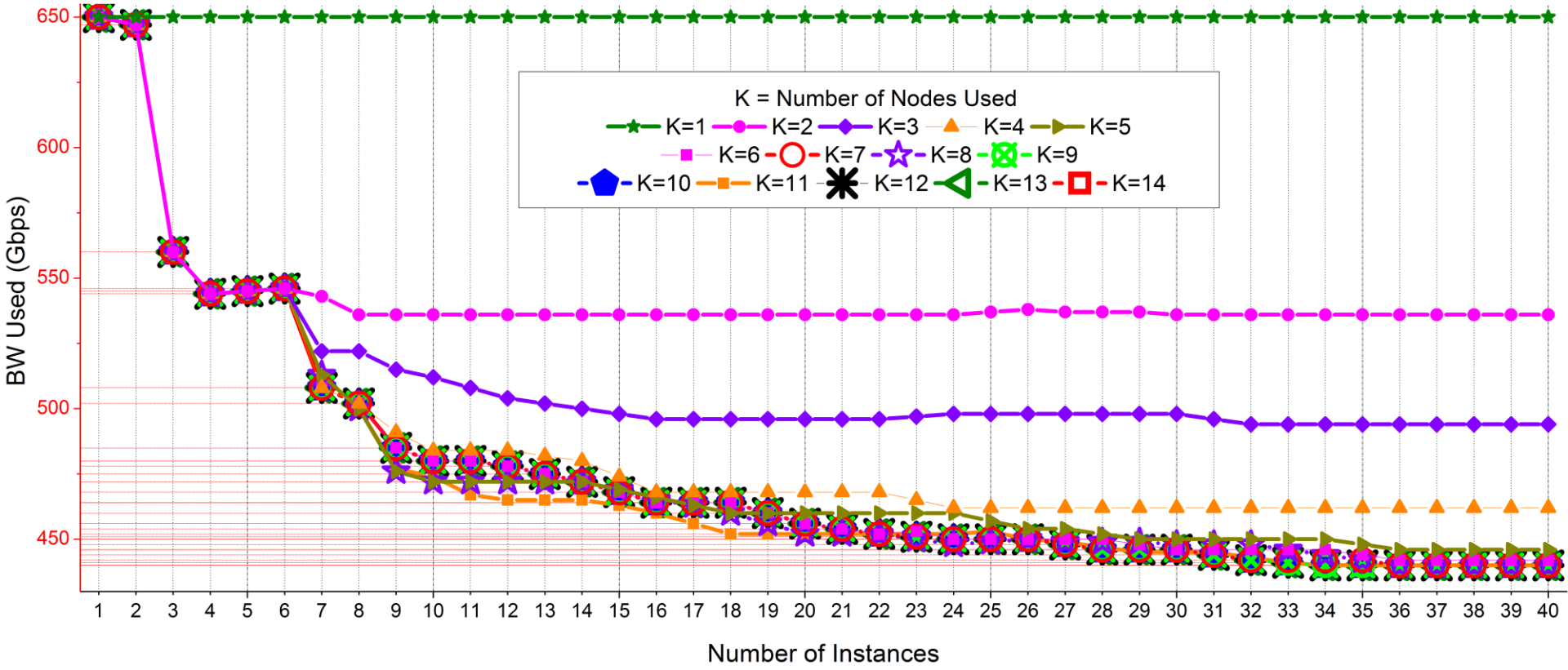
COST239 K=11,5,4,3,2,1





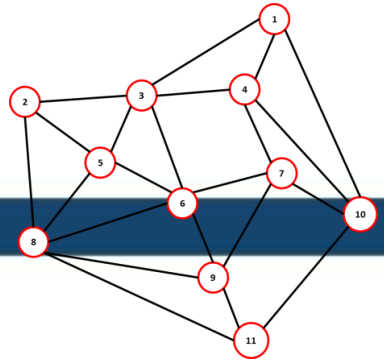
NSFNET K=ALL

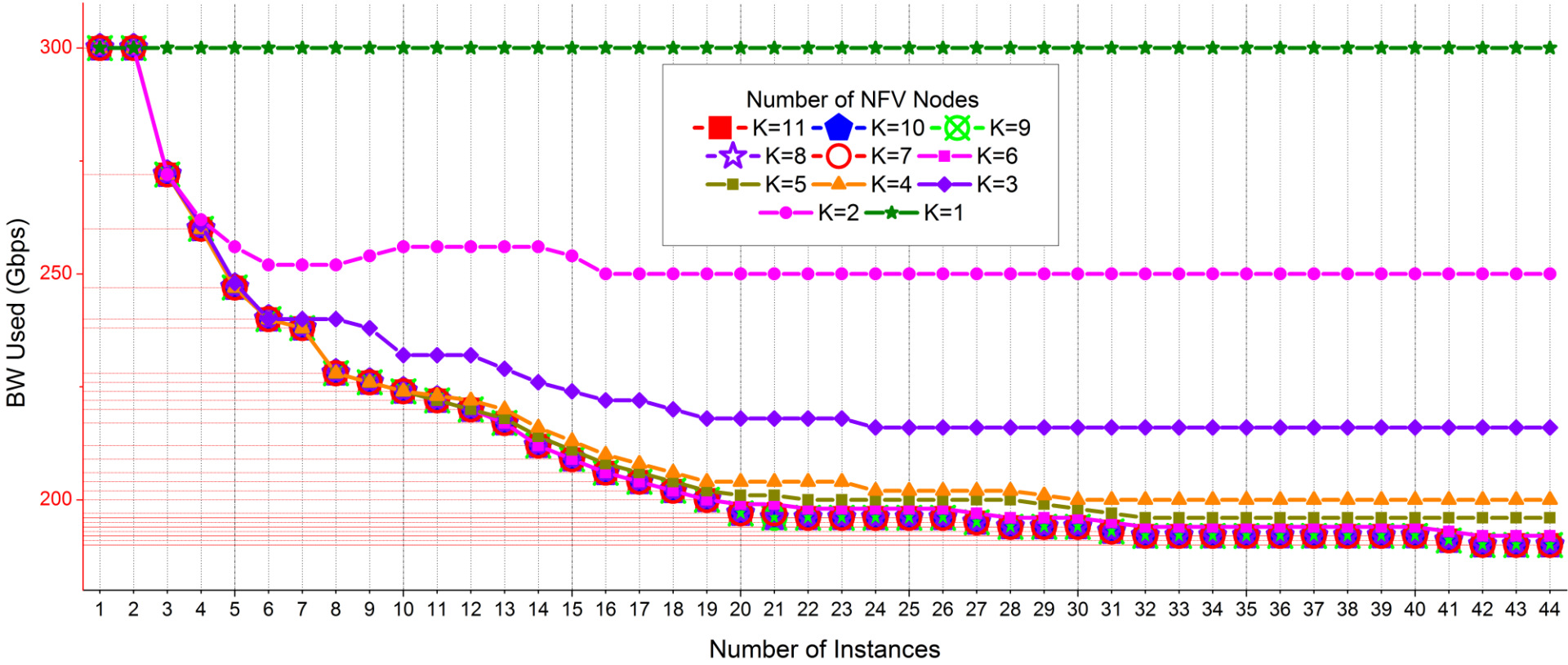




UCDAVIS
UNIVERSITY OF CALIFORNIA

COST239 K=ALL





Future Work Directions

- **Comparison of 2 phase approach with quadratic model**
- **Set of results for when CPU and link capacity constraints are enforced:**
  - Difference from current results
- **Cases where distribution of VNFs occur:**
  - Cases where CPU resources are constrained or VNF replicas (because of licenses) are enforced
  - Any additional cases?

Continued…

- **Scalability of the approach for larger network instances**
- **Account for non-uniform traffic :**
  - Matrices which are still dense but with a mixture of large and small flows. How to form traffic flow groups then?
  - Also account for multiple heterogeneous service chains
- Can the current group of traffic flows be improved?
- General graph VNFs instead of linear service chain?