Interference Aware Network Function Selection Algorithm for Next Generation Networks

Venkatarami Reddy, Gaurav Garg, **Bheemarjuna Reddy Tamma** & Antony Franklin

Networked **W**ireless **S**ystems (**NeWS**) Lab
Indian Institute of Technology Hyderabad, India

*3rd Workshop on Performance Issues in Virtualized Environments and Software Defined Networking (**PVE-SDN**), IEEE **NetSoft** 2019*

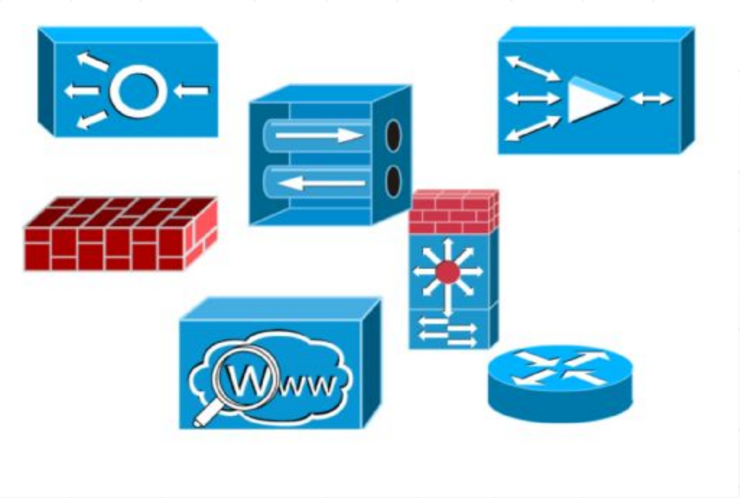
Outline

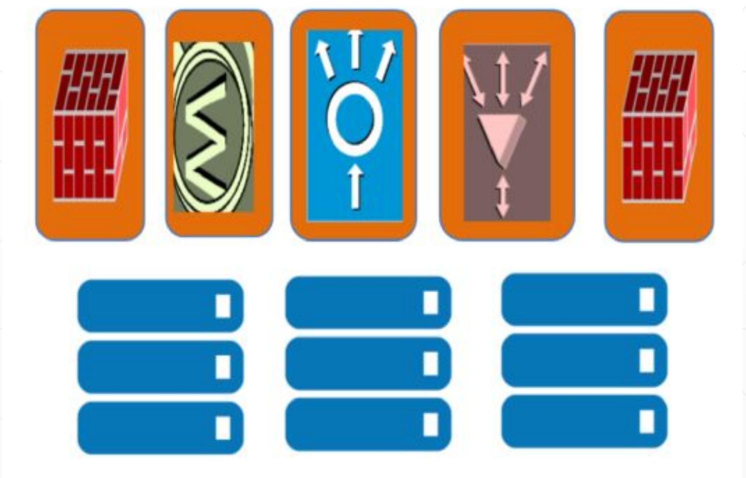
1) Motivation
2) Problem Statement
3) Proposed Interference Aware NF Selection Algorithm
4) Performance Evaluation
5) Conclusions and Future Work

Introduction

Network Functions Virtualization (NFV):

- Hardware (middlebox) ➡ Software (VNFs)
  - Easy to instantiate, deploy, and manage network functions (NFs)
  - Reduce the operational/management costs and complexities





Classic approach

NFV approach

Service Function Chaining (SFC)

- ISPs and Telcos offer a diverse set of services to users
- Traffic of each service required to pass through and processed by a set of ordered NFs or SFs called Service Function Chain (SFC)
- Each SFC request has some specific requirements such as throughput and end-to-end latency
- SDN and NFV provide flexibility and agility for deploying SFCs as VNFs on VMs/Containers on Cloud platform

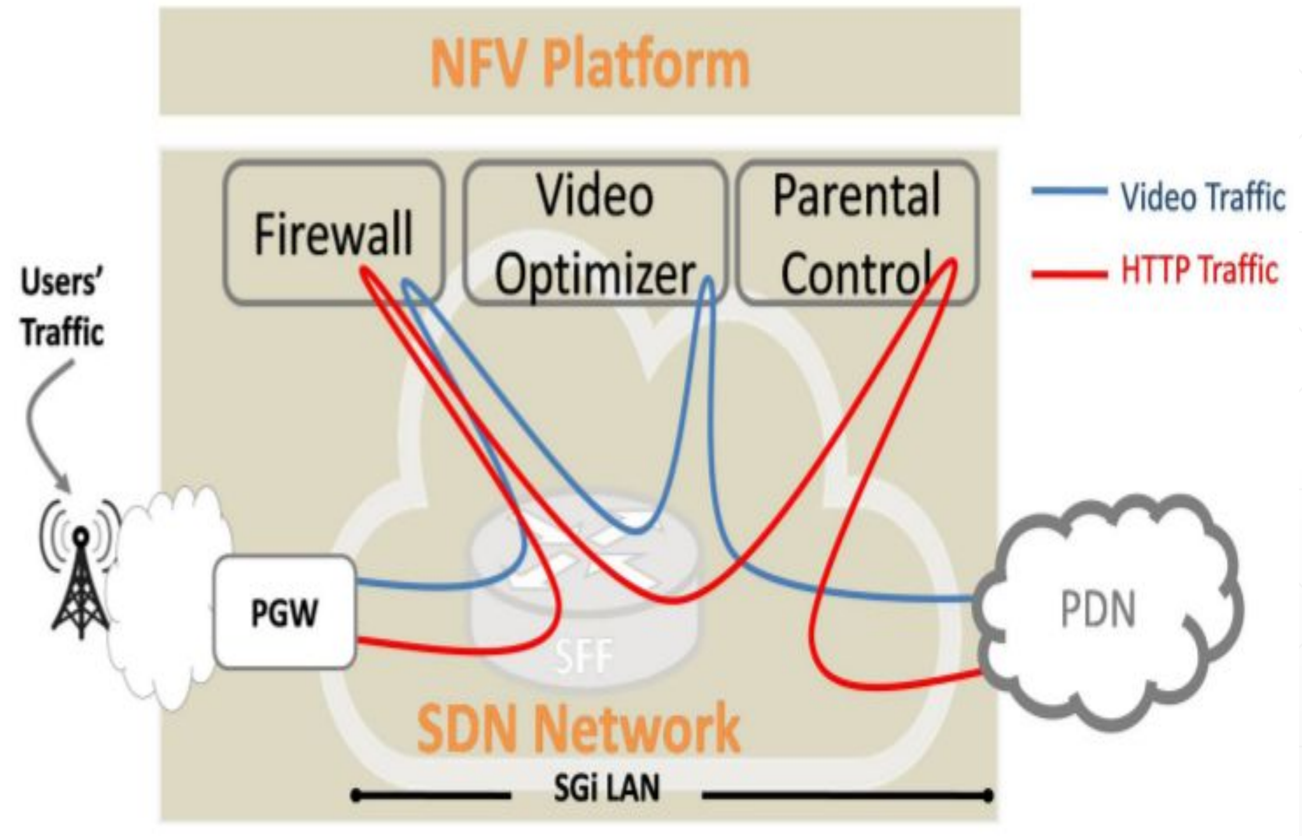


Fig. 1: SFCs in SGi-LAN of 4G

What about the performance of Network Functions (VNFs) deployed on NFV platform?

Offering Performance Guarantees is Challenging

- To save CAPEX/OPEX and reduce the communication latency, VNFs are consolidated on limited number of compute/server nodes
- VNF consolidation might cause performance degradation (in terms of throughput and latency), which is known as VNF interference
- The throughput degradation ranges from 12% to 50% as more VNFs are consolidated on the same server
- Many possible reasons of performance degradation
  - Conflicts between co-located VNFs for system resources
  - OS scheduling methods
  - I/O bottlenecks, …..
- Performance a VNF depends on other co-located VNFs on the same server

Contributions

- **Goal**
  - Design a VNF selection algorithm which increases the acceptance rate of SFC requests by considering interference effect also while guaranteeing SLAs

- **Contributions**
  - Show how VNF response time varies when it is co-located with other VNFs by conducting real-time experiments
  - Propose an Interference Aware Network Function Selection (IANFS) algorithm which uses Dynamic Programming approach and considers the expected VNF interference delay based on the co-located VNFs, along with other delays

Impact of co-located VNF (Motivational Results)

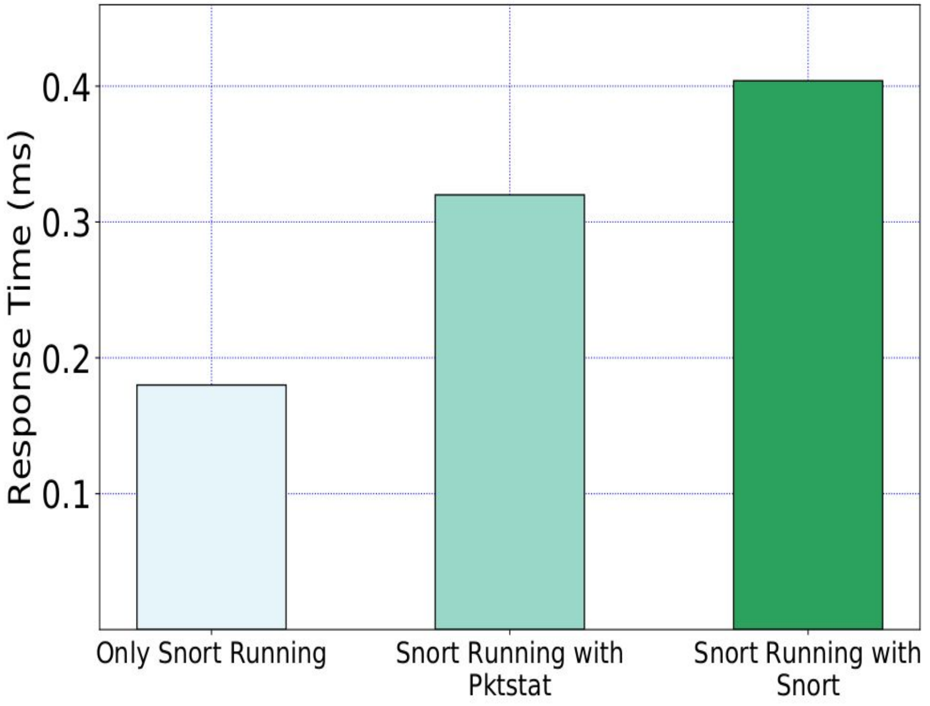


Fig. 2: Significant degradation in response time for Snort when it is runningalong with other VNFs

- Significant degradation in response time for Snort when it runs along with other VNFs on same node
- Interference delay is the difference between the response time when it runs along with other VNFs on a node and it runs exclusively on the node
- Considering interference effect is of paramount importance for critical services
- But, existing solutions in literature did not consider this interference effect while selecting VNFs while provisioning the SFC request □ SLA violations

An Example of VNF Selection for Steering SFC Request

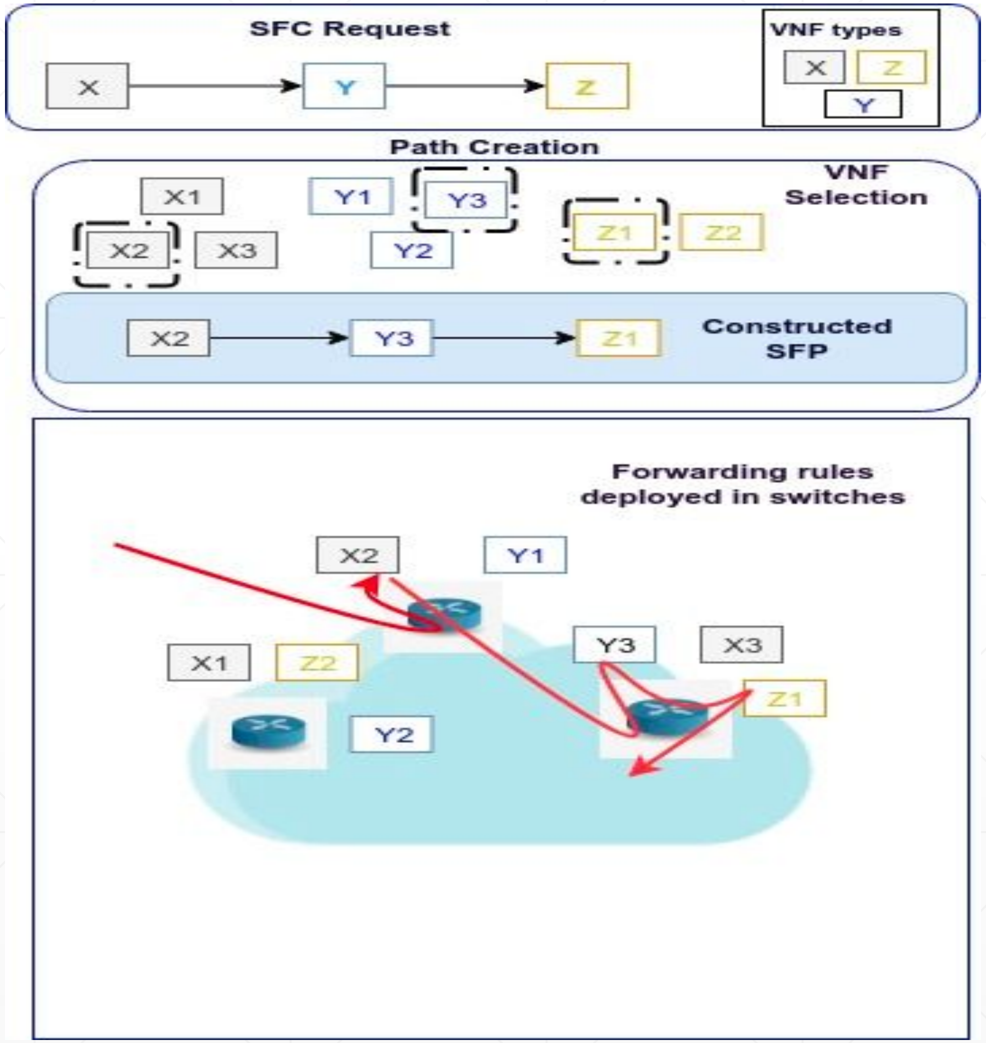


Fig. 3: Example of Steering SFC Request

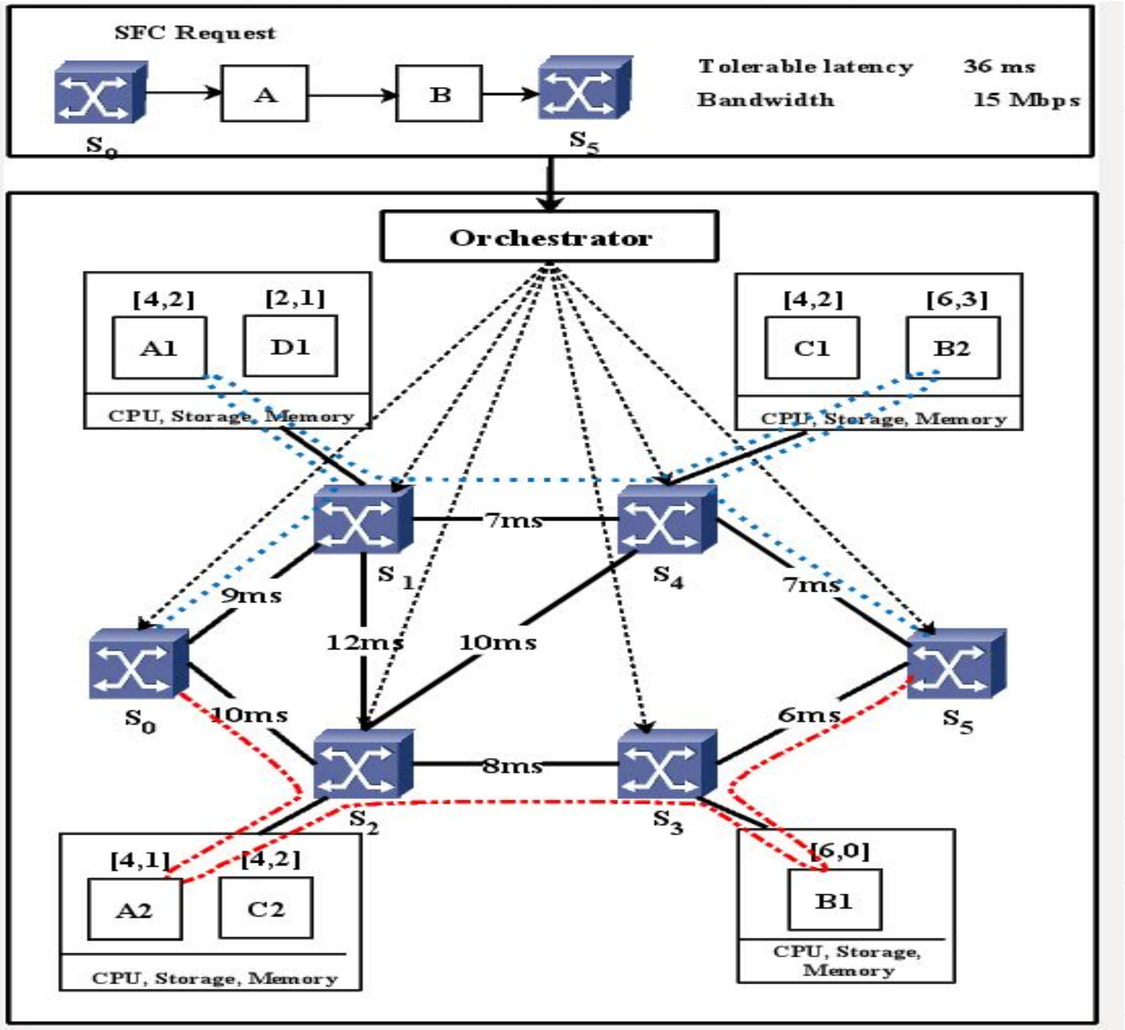
- The VNF selection can be done through different algorithms that aim to optimize either resource utilization or QoS
- Multiple instances of the same VNF are deployed at different server nodes to achieve reliability and to load balance the traffic from different locations
- Maximizing traffic throughput with SLA guarantees when steering SFC requests is still an open problem

System Model

- Network topology is modelled as an undirected graph G = (V, E)
- Each node can host multiple types of VNFs
- **Objective:** Increasing the acceptance rate of SFC requests by efficiently selecting the required VNFs in order to provision them with guaranteed SLAs
- **Input:** Service request s_m is represented as a quintuple (src_m, dst_m, tol_lat_m, sc_m, b_m)
- **Output:** Acceptance ratio and effective throughput
- End-to-End latency of SFC request (d_{es}) includes: Processing latency (d_{sf}), interference delay (d_{if}), and link/path delay (d_{pl})

$$d_{es} = \sum_{f \in sc_m} (d_{sf} + d_{if}) + \sum_{l \in path_{sc_m}} d_{pl}$$

An example of VNF selection for SFC provisioning



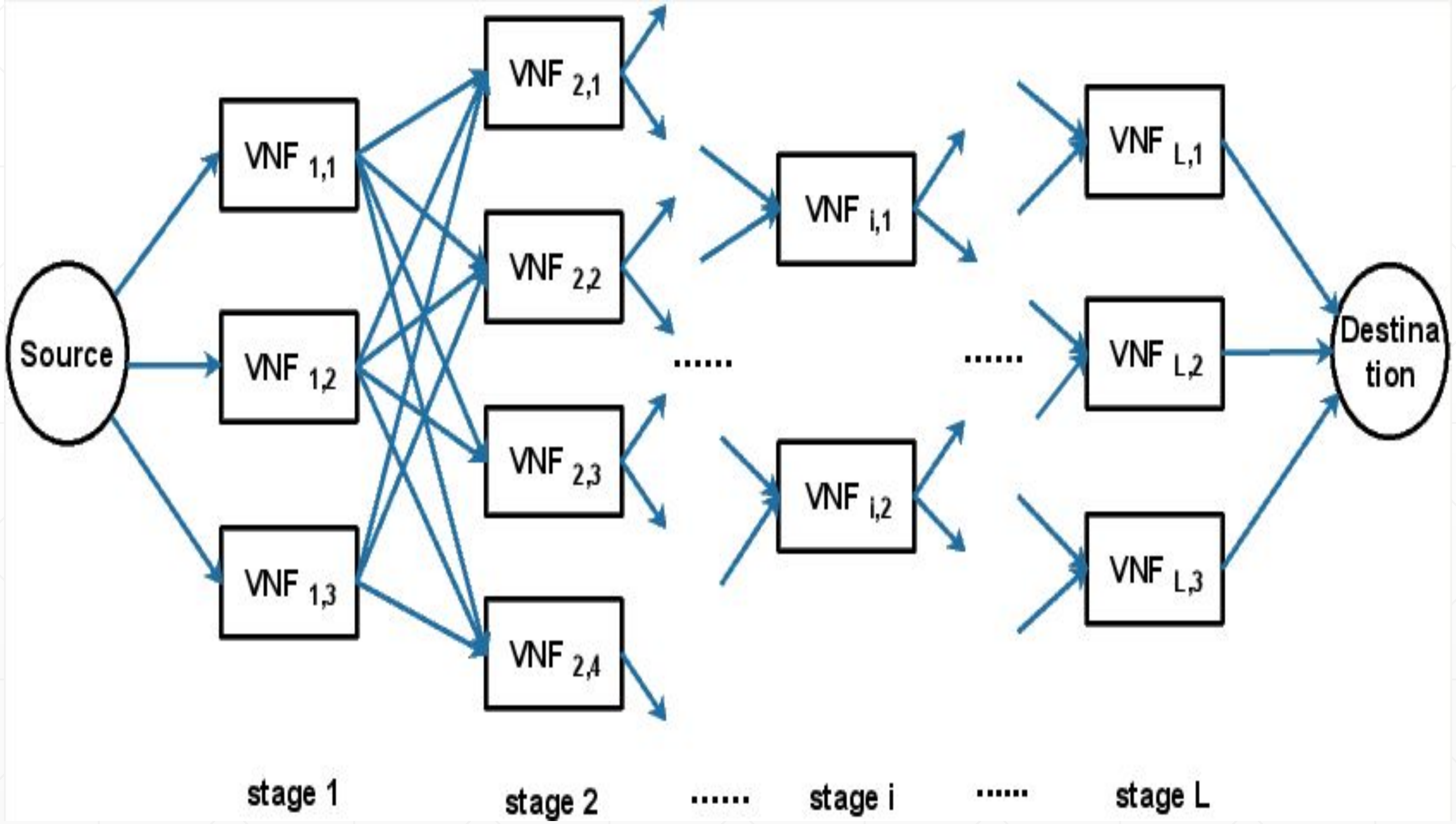
- ***Shortest path scheme:*** Chooses the path ($S_0 \rightarrow S_1 \rightarrow S_4 \rightarrow S_5$) with the end-to-end latency of 33 ms
- But it is not acceptable because after adding interference delay, the end-to-end latency reaches 38 ms which is more than the tolerable latency (36 ms)
- ***Proposed IANFS scheme:*** Chooses the path ($S_0 \rightarrow S_2 \rightarrow S_3 \rightarrow S_5$) with the end-to-end latency of 34 ms which considers interference delay also at the time of VNF selection

Fig. 4: An example of VNF selection for SFC provisioning

IANFS: A Heuristic Algorithm for VNF Selection

- NP-Hard problem
- Proposed a heuristic approach named IANFS based on Dynamic Programming
- Build a multi-stage (L-stage) graph (where L is the length of the SFC request)
- Stage i contains the set of vertices where an instance of VNF i is running
- Shortest delay path between adjacent stages is computed using *Dijkstra* algorithm

Multi-stage graph



- Time complexity depends upon the number of stages and the maximum number of nodes in any stage

- If S_{max} is the maximum number of nodes in any of the L stages, then its computational complexity is $O(L * S^2_{max})$

Fig. 5: Multi-stage graph of length L

IANFS: Flow Chart of Proposed Algorithm

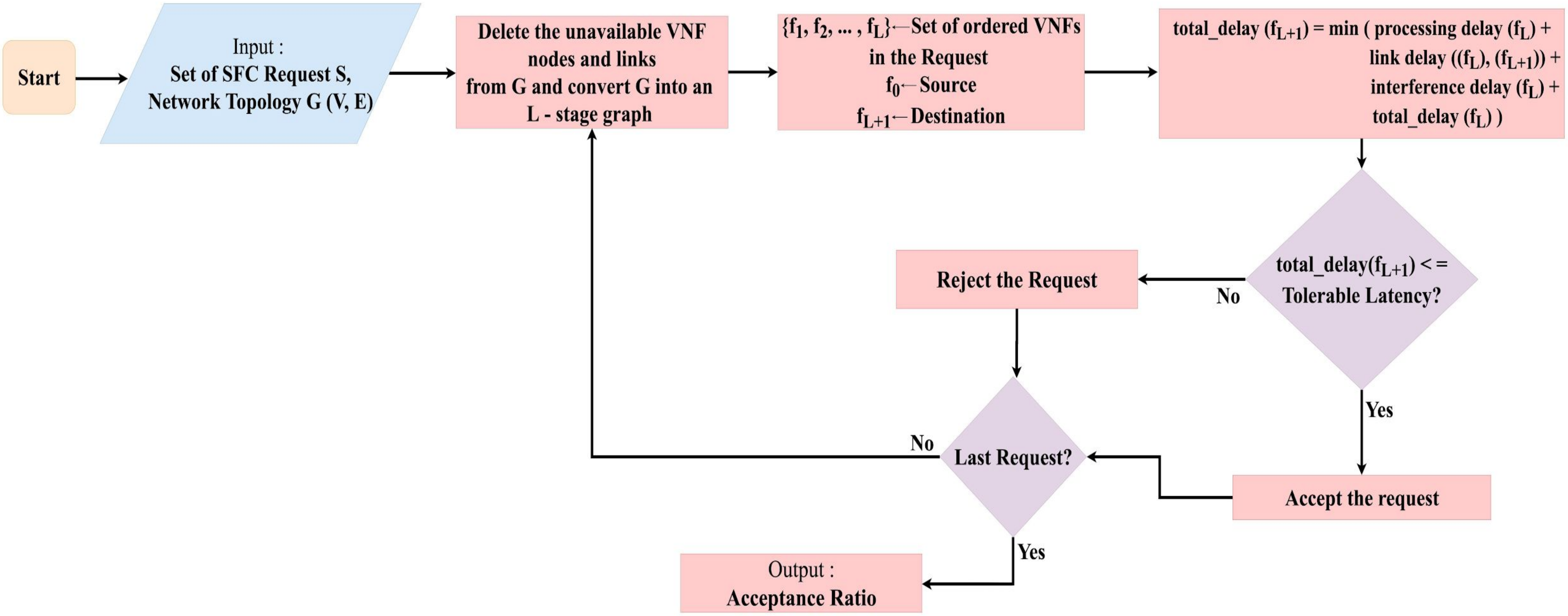
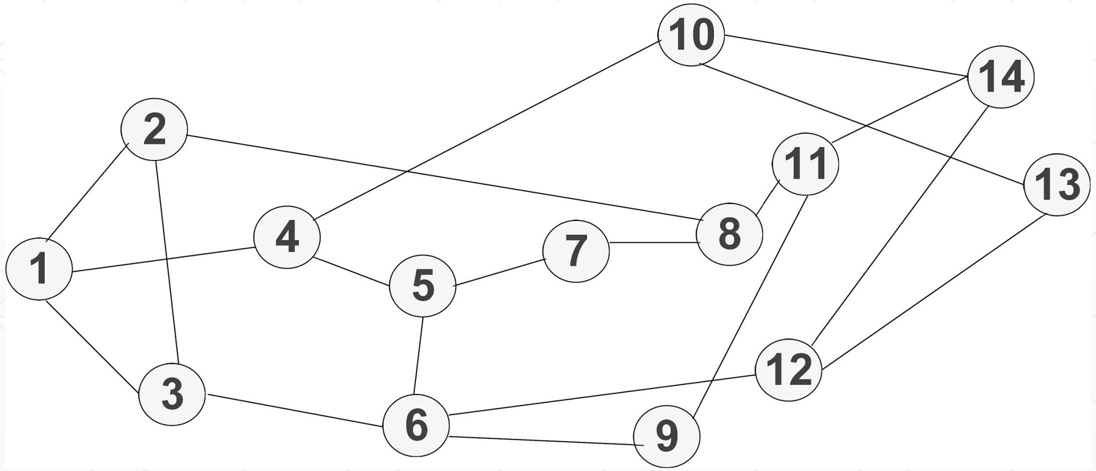


Fig. 6: Flowchart of proposed algorithm

Performance Evaluation

- Performance of the proposed IANFS algorithm on two different well known network topologies: NSFNET network (14-node, 21-link) and USA backbone IP network (USNET: 24-node, 43-link)
- To evaluate the performance, we develop a C++ based simulator
- Each experiment is repeated 100 times and results are plotted with 95% confidence interval

Simulation topologies used:



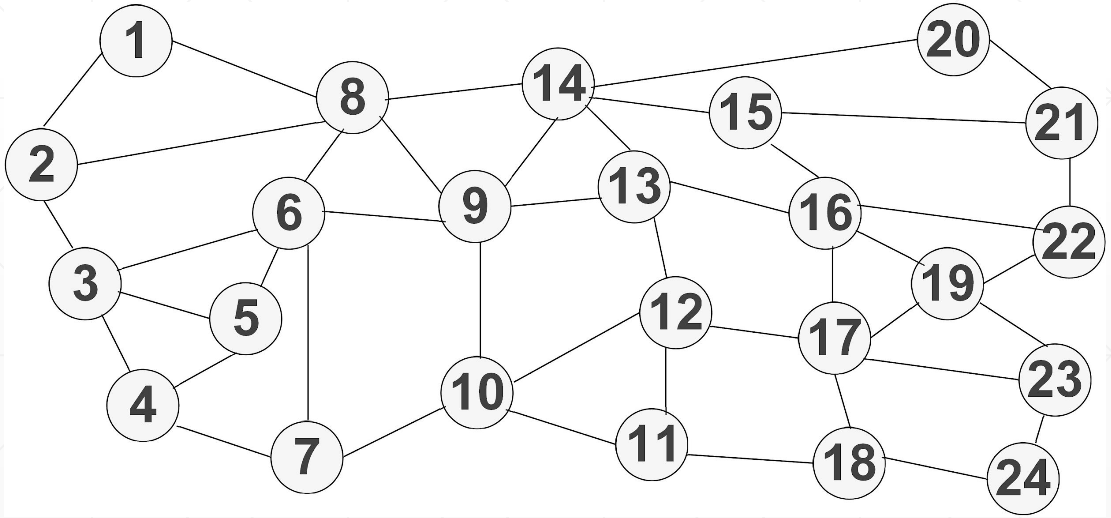


Fig. 7(a): NSFNET

Fig. 7(b): USNET

Performance Evaluation (Cntd…)

- **Performance Metrics**
  - **Acceptance Ratio :** The number of successfully provisioned SFC requests which satisfied the end-to-end latency requirement to the total number of SFC requests

  - **Effective Throughput :** Sum of throughputs of all the accepted SFC requests

- **Algorithms Compared**
  - **LSFCS (Latency Aware SFC Steering):** Provisions the SFC requests based on the available capacity of link and VNF instances without considering the interference effect

Simulation Parameters

Parameters	Range
Length of SFC requests	[1 - 4]
Requested bandwidth of SFCs	10 - 50 Mbps
End-to-End latency of SFCs	90 - 110 ms
Link delay b/w Nodes	15 - 25 ms
Processing delay of VNFs	5 - 10 ms
Interference delay due to co-hosting of VNFs	1 - 5 ms
Link Capacity	600 - 1800 Mbps
Node Capacity	600 - 1800 Mbps

Acceptance Ratio vs Service Arrivals

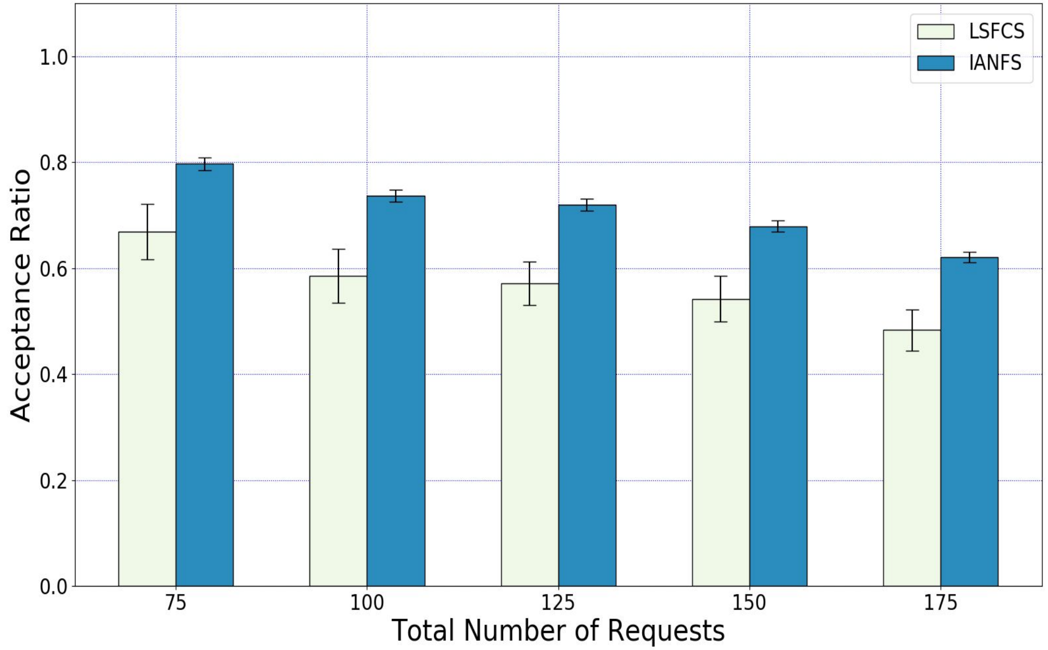


Fig. 8(a): NSFNET

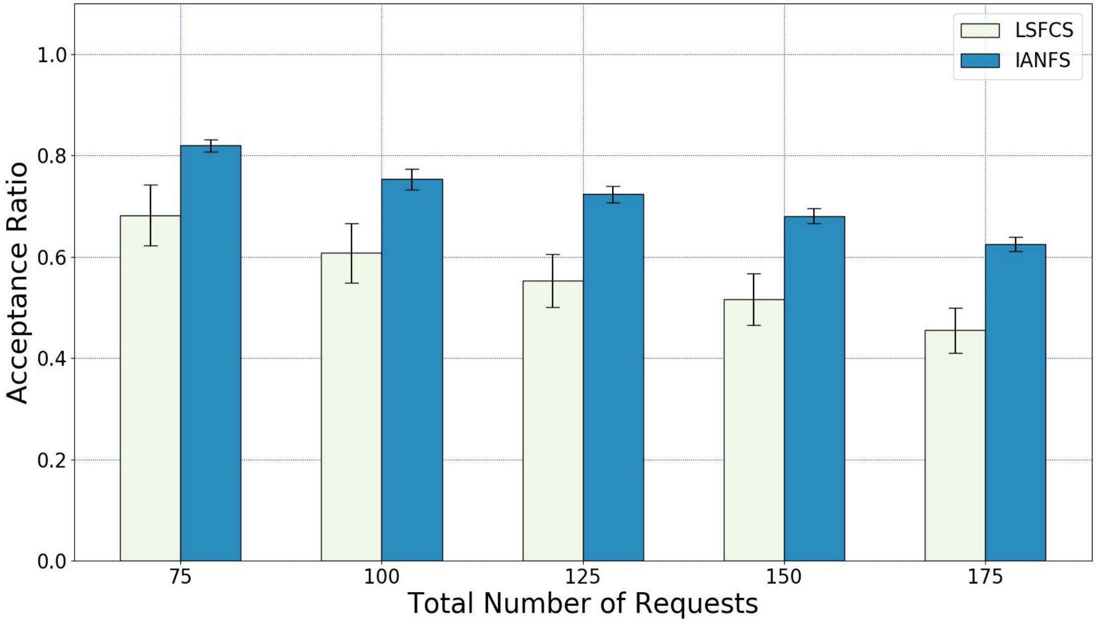


Fig. 8(b): USNET

- Length of SFC is fixed as 3
- **Observations:**
  - The acceptance ratio decreases as the number of SFC requests increases due to insufficient resources
  - IANFS always achieves better acceptance ratio than LSFCS algorithm

Effective Throughput vs Service Arrivals

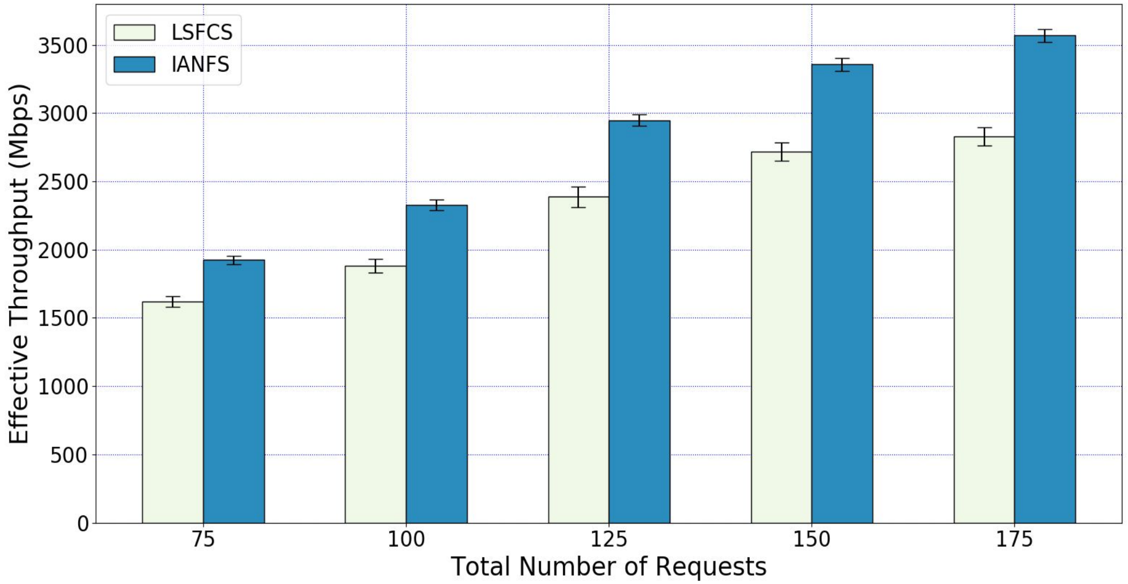


Fig. 9(a): NSFNET

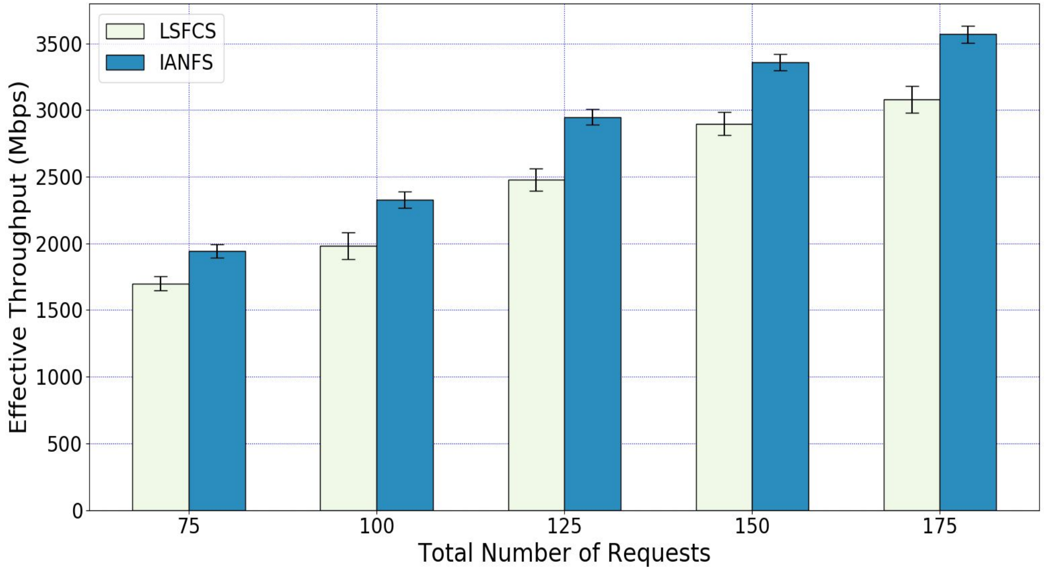


Fig.9(b): USNET

- Effective throughput increases with an increase in the number of SFC requests
- Similar to the acceptance ratio, effective throughput obtained by IANFS is always higher than LSFCS approach

Acceptance Ratio vs Length of SFC

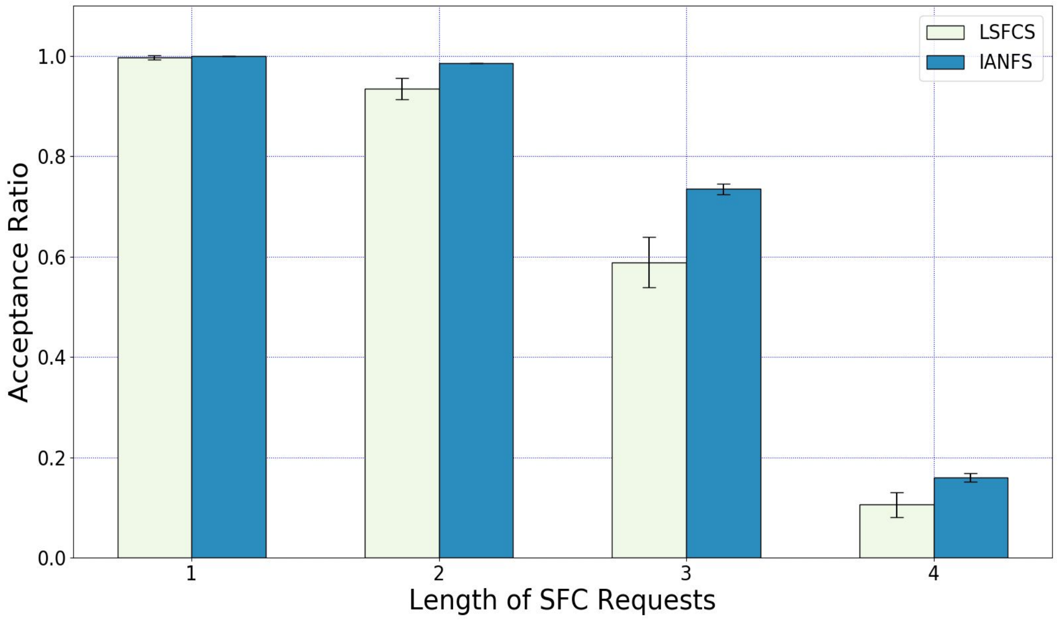


Fig. 10(a): NSFNET

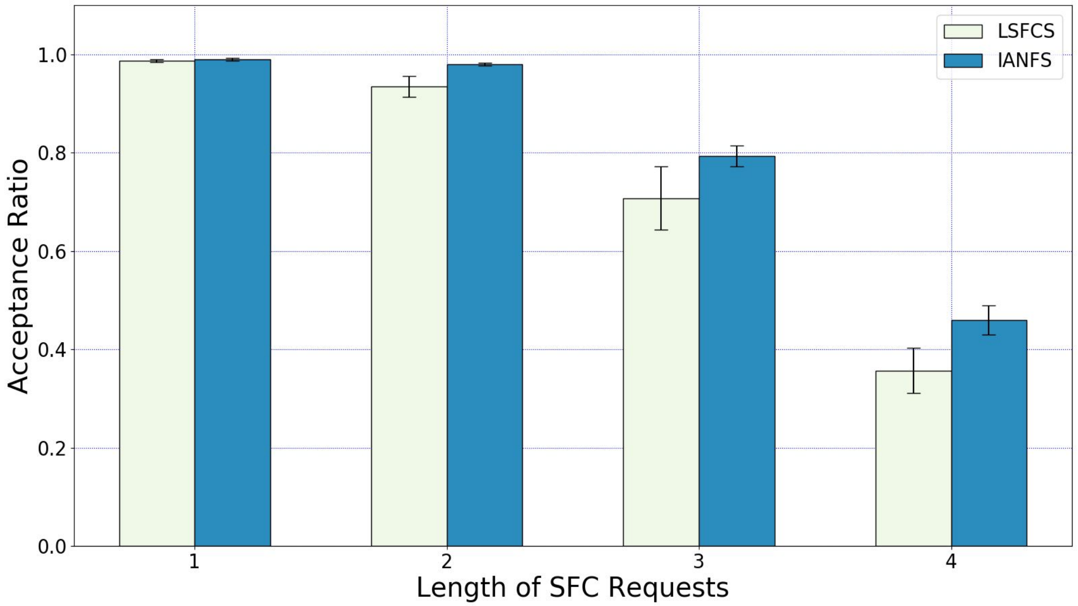


Fig. 10(b): USNET

- Number of SFC requests is fixed at 100
- **Observations:**
  - As the length of SFC increases, the acceptance ratio decreases
  - At SFC length 4, the acceptance ratio of USNET is high because more VNF instances are available when compared to NSFNET

Acceptance Ratio vs Link Capacity

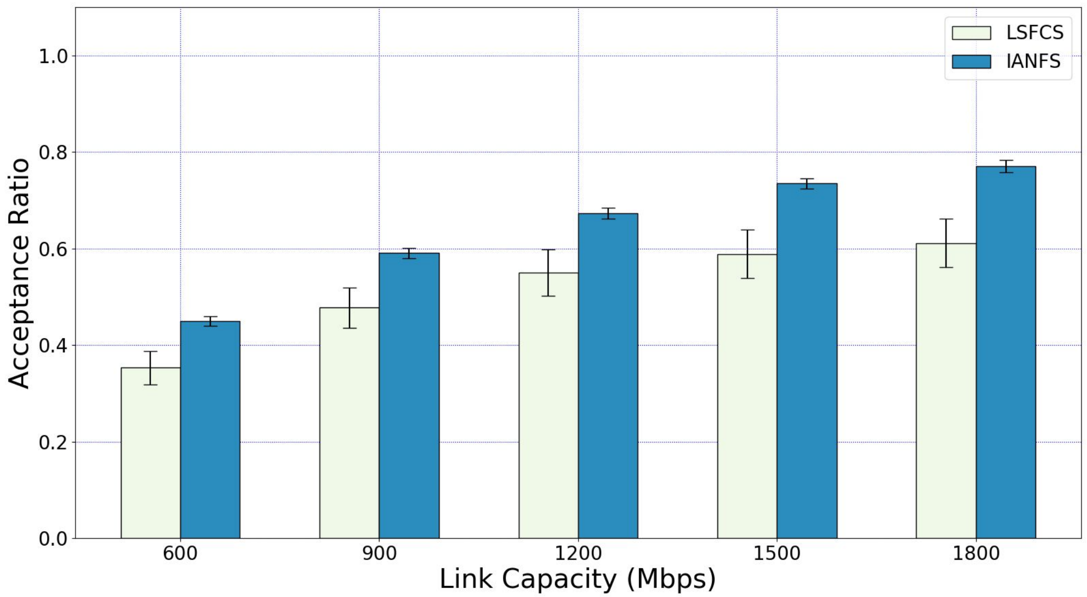


Fig. 11(a): NSFNET

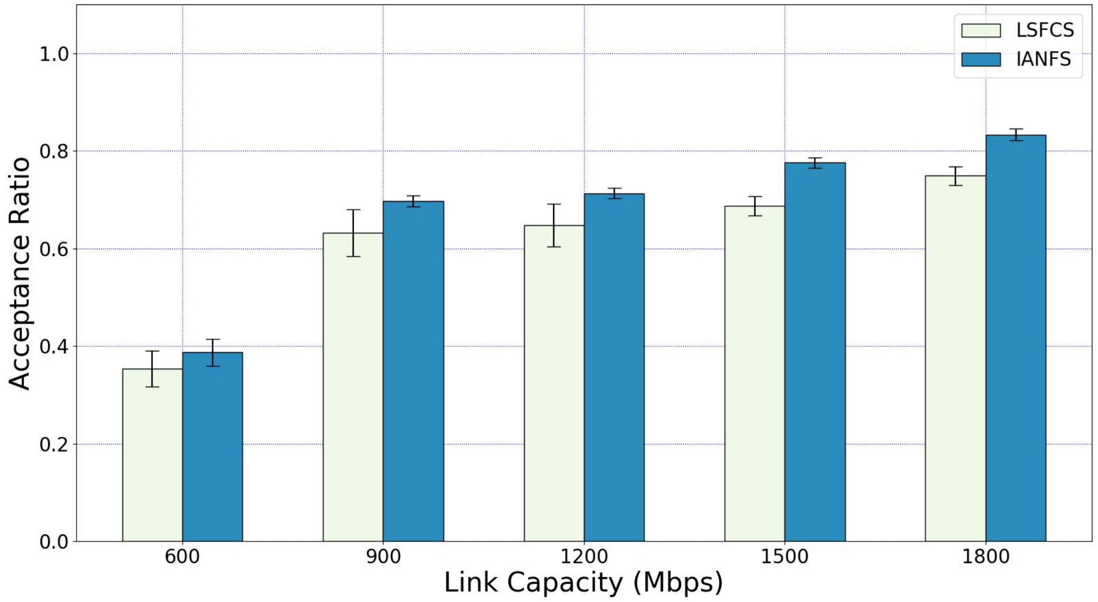


Fig. 11(b): USNET

- The impact of link capacity on SFC acceptance ratio
- Compared with LSFCS, IANFS increases acceptance ratio upto 26% on average.
- It shows that IANFS performs best with the variation of link capacity for both topologies

Acceptance Ratio vs VNF Capacity

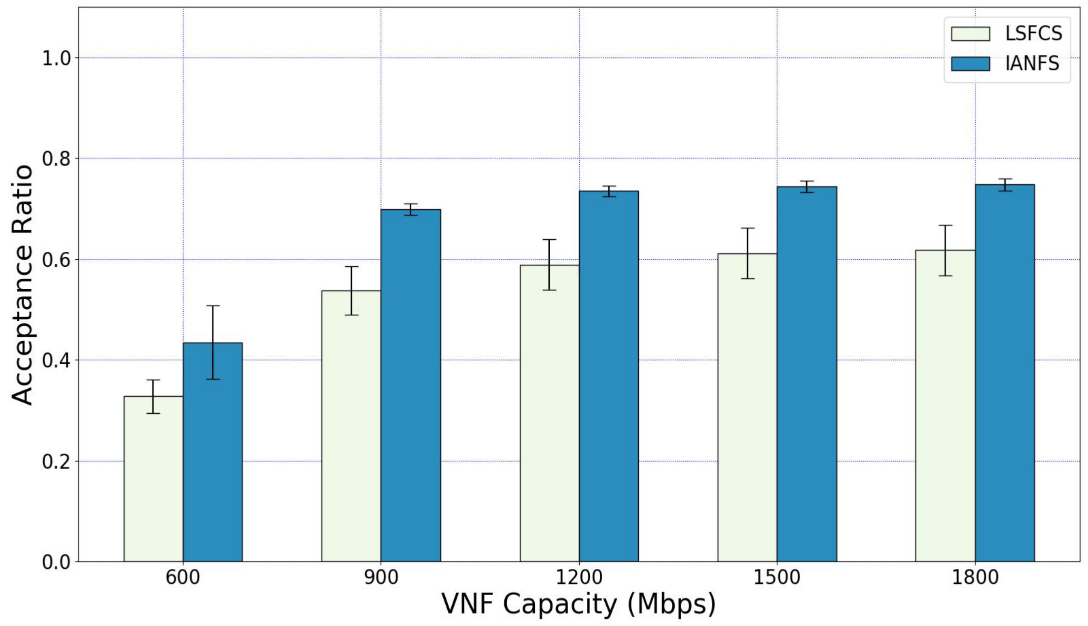


Fig. 12(a): NSFNET

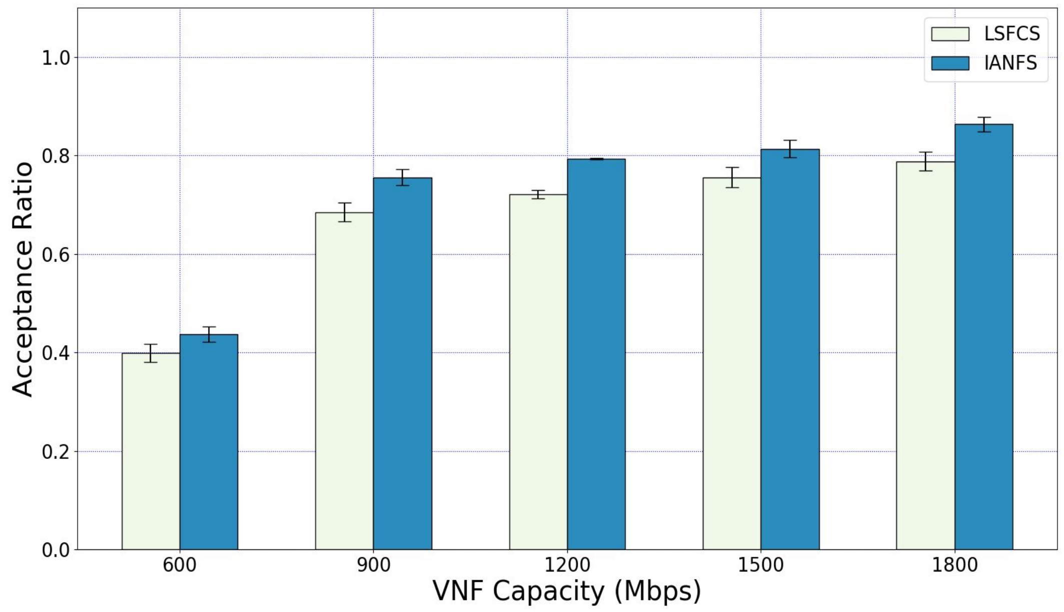


Fig. 12(b): USNET

- The impact of VNF instance capacity on SFC acceptance ratio for both topologies
- Compared with LSFCS, IANFS increases acceptance ratio up to 34% on average

Conclusions and Future Work

- Shown interference effect due to co-located VNFs by conducting real-time experiments
- Proposed a heuristic algorithm (IANFS) for VNF selection which considers interference effect also while provisioning SFC requests
- Through simulations studies, it was shown that IANFS outperforms an existing scheme (LSFCS) by 29% in terms of Acceptance ratio of SFC requests

- ***Future Work:***
    - Factoring in variation in interference effect due to change in the load of co-located VNFs
    - Experimental studies using LTE/5G-Core modules (OAI) and open source NFV MANO (OSM)

References

[1] Joel Halpern and Carlos Pignataro. Service Function Chaining (SFC) architecture. Technical report, 2015.

[2] C Zhang et al. L4-l7 service function chaining solution architecture. Open Networking Foundation, ONF TS-027, 2015.

[3] Chaobing Zeng et al. Demystifying the performance interference of co-located virtual network functions. In Proc. of IEEE Conference on INFOCOM, pages 765–773, 2018.

[4] Shundan Jiao et al. Joint virtual network function selection and traffic steering in telecom networks. In Proc. of IEEE Conference on Global Communications (GlOBECOM), pages 1–7, 2017.

[5] Abdelhamid Alleg et al. Delay-aware vnf placement and chaining based on a flexible resource allocation approach. In Proc. of IEEE Conference on Network and Service Management (CNSM), pages 1–7, 2017.

[6] Naik et al. NFVPerf: Online performance monitoring and bottleneck detection for NFV. In Proc. Of IEEE Confeence on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2016.

Acknowledgements

This work is partially supported by the projects "Visvesvaraya PhD Scheme" and "Converged Cloud Communication Technologies", MeitY, Govt. of India.

सत्यमेव जयते

Ministry of Electronics
& Information Technology
Government of India

Thank You

Email: tbr@iith.ac.in
Homepage: http://www.iith.ac.in/~tbr
Google Scholar Profile: http://goo.gl/JdgRB
NeWS Lab: https://newslab.iith.ac.in/