On traffic routing in Network Powered by Computing Environment: ECMP vs UCMP vs MAROH

E. Stepanov, *R. Smelyanskiy , A. Plakunov*

Lomonosov Moscow State University

This work was done with the support of MSU Program of Development, Project No 23-Ш03-03

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

turing lecture

4.06.2018

DOI:10.1145/3282307

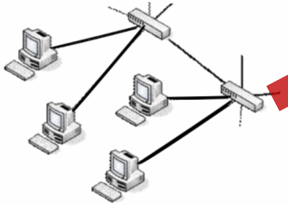
Innovations like domain-specific hardware, enhanced security, open instruction sets, and agile chip development will lead the way.

BY JOHN L. HENNESSY AND DAVID A. PATTERSON

A New Golden Age for Computer Architecture

New Golden Age of Computational Infrastructure

- the end 60-s – Computer installation with job packet processing;
- 70-s - mainframe computer center with terminal network;
- 80-s – Client-Server infrastructure with network access;
- 90-s – Servers Farm with Frontend server with access via LAN;
- 2000-s – monstrous DC with high speed WAN;
- Quo Vadis?

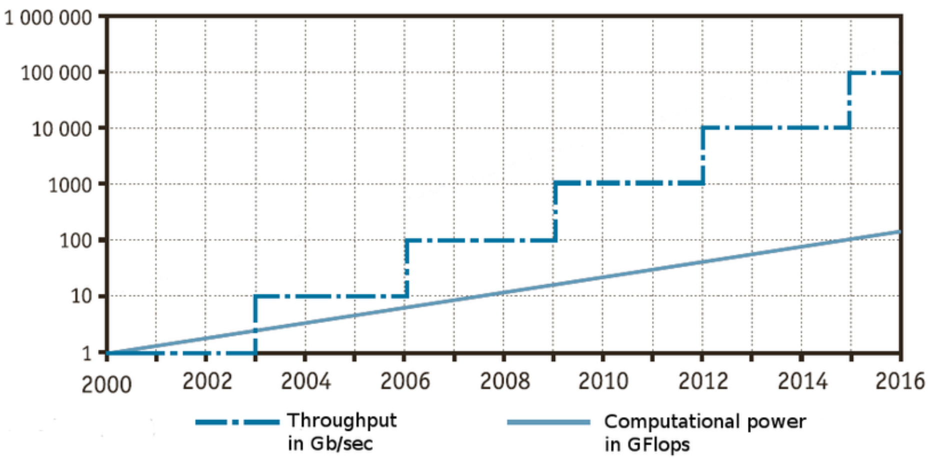


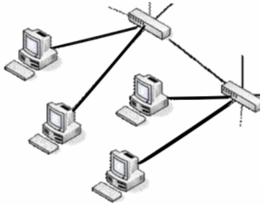
Application Requirements + Hardware Capabilities + Software Engineering

Applied Research Center for Computer Networks

New Golden Age of Computational Infrastructure

- the end 60-s – Computer installation with job packet processing;
- 70-s - mainframe computer center with terminal network;
- 80-s – Client-Server infrastructure with network access;
- 90-s – Servers Farm with Frontend server with access via LAN;
- 2000-s – monstrous DC with high speed WAN;
- Quo Vadis?



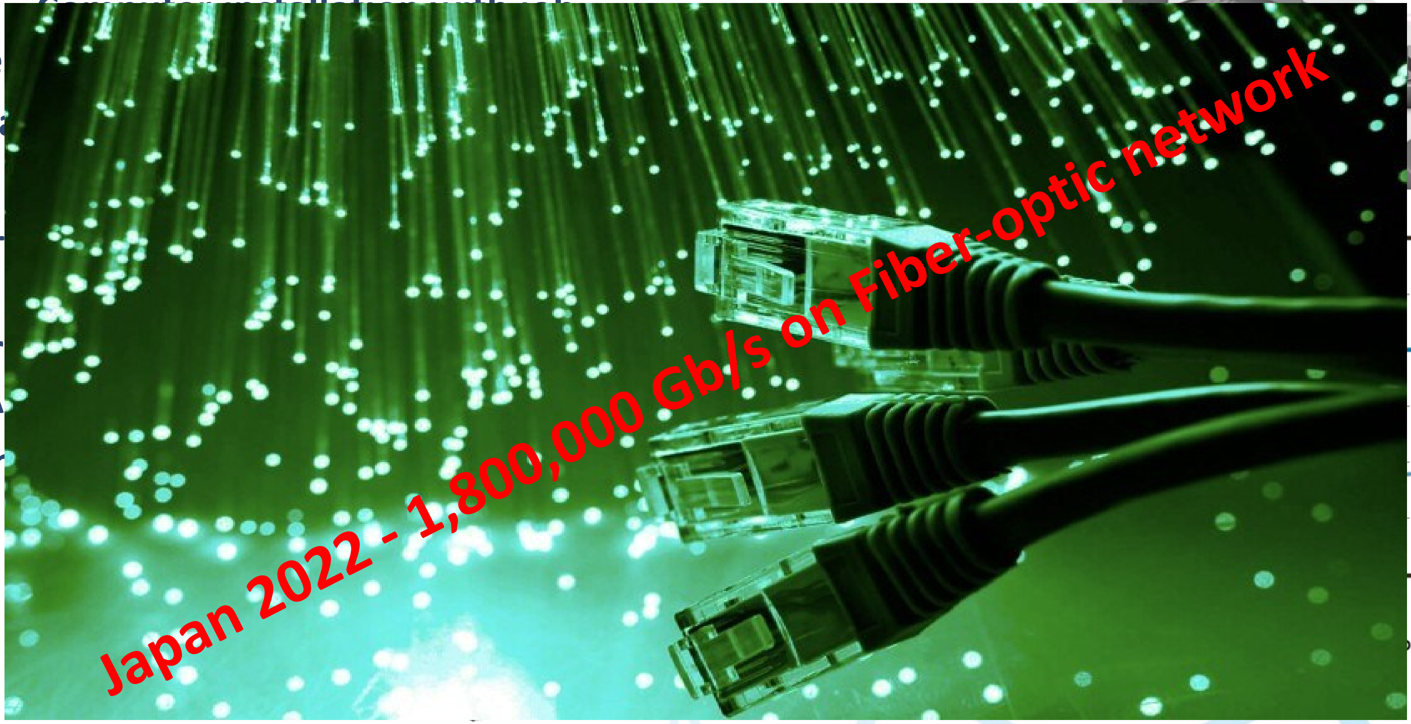


APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

Application Requirements + Hardware Capabilities + Software Engineering

New Golden Age of Computational Infrastructure

- the end 60-s
  packet proce
- 70-s - mainfr
  network;
- 80-s – Client-
  access;
- 90-s – Server
  access via LA
- 2000-s – mor
- Quo Vadis?



Application Requirements + Hardware Capabilities + Software Engineering

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

Applications suite of features

- **Distributed** –applications are composed of a set of functions/services that run in parallel on different nodes and have to integrate geographically distributed data;
- **Self-sufficient -** the application is no longer just code and source data, it is accompanied by a specification and orchestration of the components (application services), relationship topology, the determination of the required level of their performance, explicitly formulated requirements for the resources (computing, network, storage) and deadlines for their communication;
- **Elasticity** –the performance of the application changes automatically without interrupting its operation in accordance with the requirements of the SLA and the current load on it;
- **Real-Time mode** –applications are sensitive to delays and its response time is imitated;
- **Cross-platform** - it doesn't matter what software environment or hardware platform is available for the application;
- **Interaction and Synchronization -** combining the results of different stages of computations, regardless of their location, aggregation of service chains;
- **Maintainability** - updating the application does not require any action on the part of the user;

The main force of computational infrastructure developments are applications needs!

Computational Infrastructure Requirements

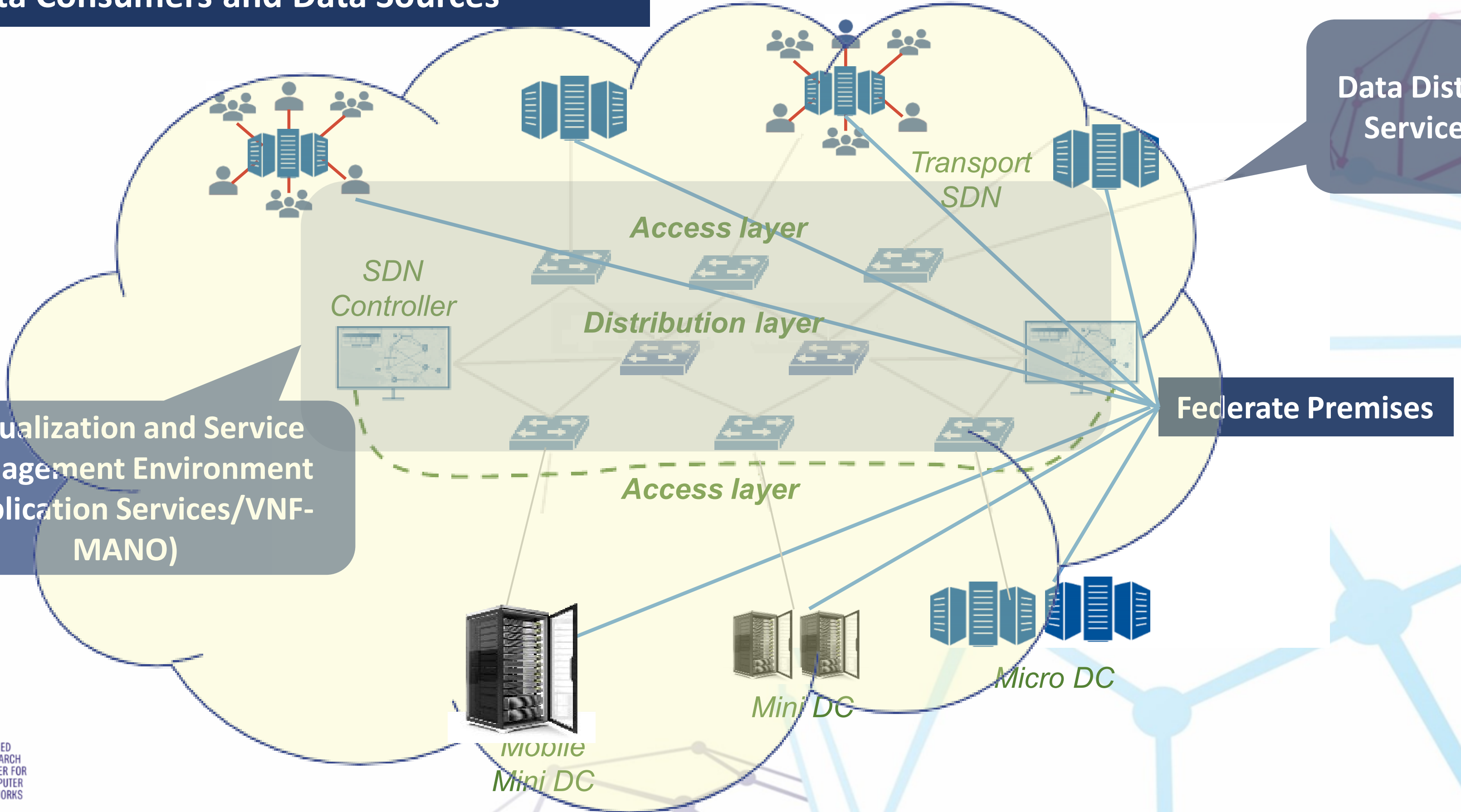
- **Behavior predictability** – predictability of delays associated with computations, transfer and access to data during the application operation, in order to manage application's execution accordingly to the requirements of the SLA;
- **Security** – it does not pose unacceptable risks to the application and its data like Confidentiality, Integrity, Availability;
- **Availability, Reliability and Fault Tolerance** - the infrastructure should be robust enough to ensure a high level of availability and operability of its services, application components, recovery of lost data in case of failures and attacks, react in real time by changes in topology, traffic flows and shape routing to ensure the fulfillment of SLA requirements;
- **Efficiency and Fairness** -the infrastructure must ensure that the application runs, delivers and processes its data by infrastructure resources, reliably, without impair other applications and their traffic;
- **Virtualization** - virtualization of all types of resources (computing, storage, network)
- **Scalability** - it should be efficiently scalable depend on the number of data, services and applications points of presence in terms of performance;
- **Serverless** – the infrastructure should automatically place application components in a way that allows them to interact according to the application structure, and in a way that ensures that the SLA requirements of the application are met, while minimizing infrastructure resources utilization.

- **The scaling range of the network service is huge and in real time, which put high demands on the algorithm time complexity.**
- **Only sub-optimal solutions are available using methods based on machine learning**

NPC: General View

Data Consumers and Data Sources

Data Distribution Service (DDS)



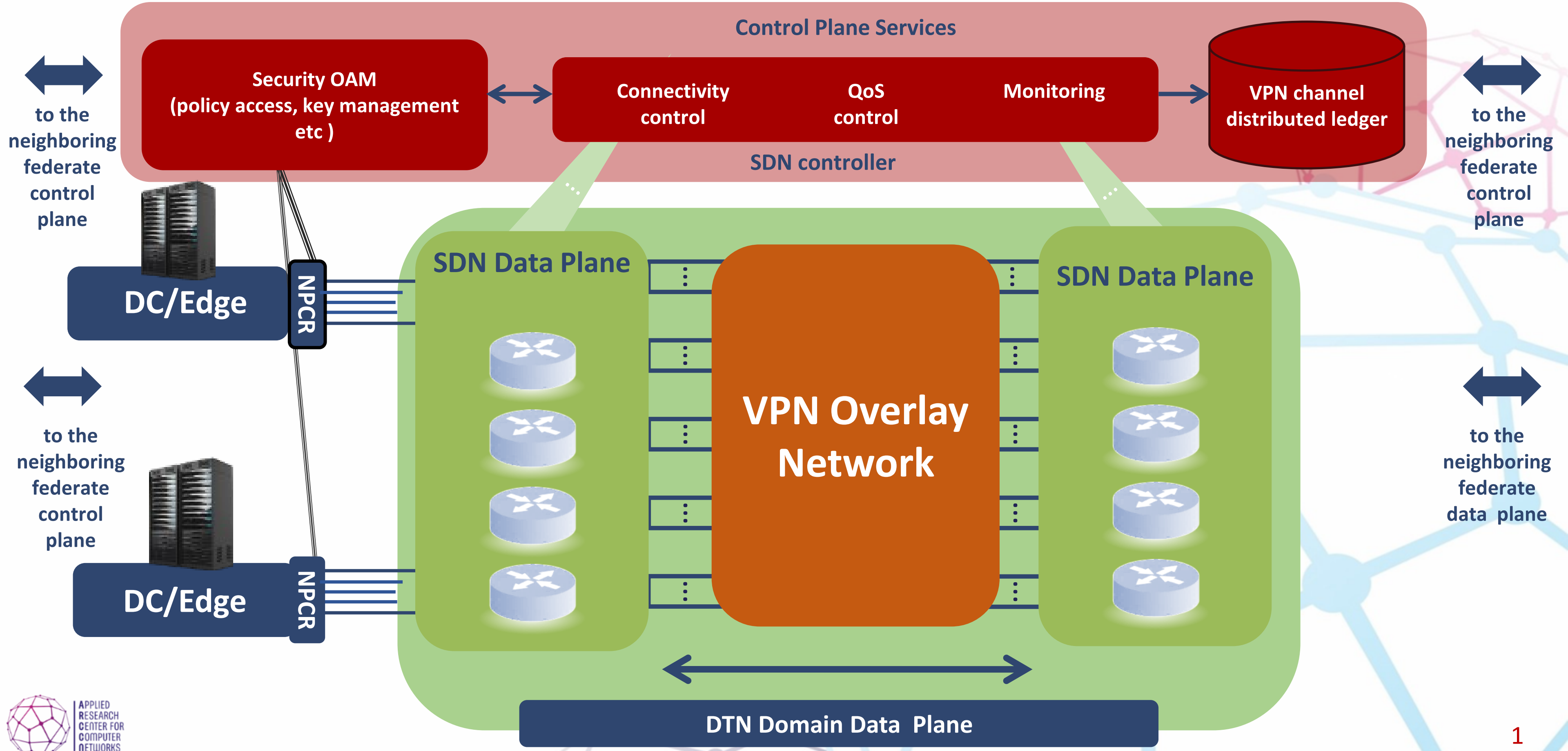
Virtualization and Service Management Environment (Application Services/VNF-MANO)

Network Powered by Computing is Super Large Scalable Computer

Fully Controllable Programmable Virtualized Infrastructure John Gage: SunMicrosystems

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

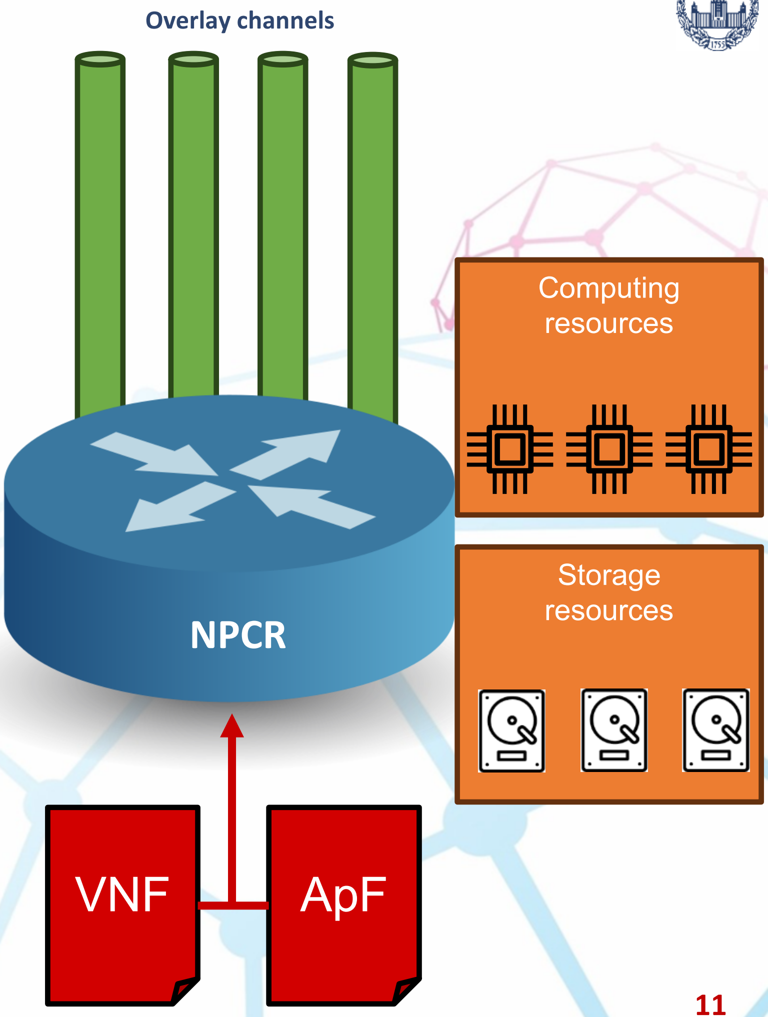
NPC intra DTN Layer



NPC Router (NPCR)

NPCR functions:

- distribution of application functions (ApF)/ virtual network functions (VNF) across computational nodes (CN) of DP plane
- decision making: is it worth to execute the certain ApF/VNF on the CN connected to this current NPCR or not;
- forwarding ApF/VNF that was not accepted by the current facility to other CNs;
- optimal data traffic routing;
- provision of the transport connection that meets the required Service Level Agreement (SLA)

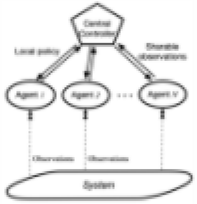


APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

Multi-agent optimal control

Efficiency → Distributed control
Accuracy → Centralized control

Centralized approach



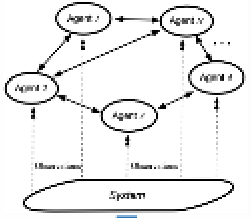
Each agent knows its local state.

The control center gathers the status of each agent.

The control center makes a decision based on the optimization policy.

Each agent is given a control action.

Distributed approach: Agent network

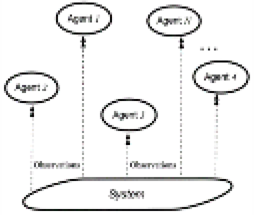


Each agent knows its local state.

Information exchange is limited to neighboring agents only.

Based on local information and information collected from neighbors, each agent decides on the optimal strategy for himself.

Distributed approach: Independent agents



Each agent knows its local state.

Each agent judges the control strategy and actions of other agents based on his experience.

The agent implements control decisions in accordance with its local optimization strategy and based on its observations.

Computing task scheduling → Dynamically tuned computing node (CN) scheduling

CN distribution: each CN decides to take a task or determines where to transfer it - a cooperative distribution of tasks between CNs.
Distributed and independent TE: each network node independently decides on the distribution of flows over available channels.

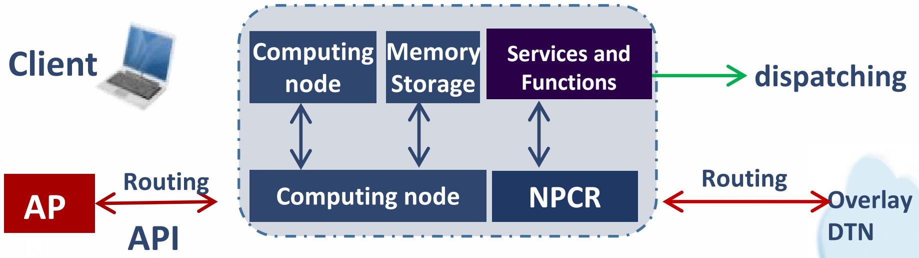
Service chain scheduling→Dynamic load of chain services in CN

Distribution of chain services:
Accounting for time constraints and interaction logic.
Maximum load of CN resources (computing & storage).
Distributed and independent TE: each network node independently decides on the distribution of flows over available channels.

Problems of Multi-agent control
Poor scaling;
There are no mathematical models that guarantee convergence to the optimal solution;
Selection of the optimization functional;
The constraint of the deviation from the optimal solution is not guaranteed.

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

Optimal SFC allocation for active mode



Problem: optimal distribution w ϵ ***W*** on NPC: ***{ cn_i }$_w$***

Necessary solutions:

- Minimizing the objective function for all $\boldsymbol{w_i}$ from ***W*** with given $p_i \in P$
- under SLA and available resource constraints

NPC = (V, A), where
V = C N U ***S N*** U ***P,*** where
CN = { cn_i = <cr, m, h>} – set of computational nodes ,
SN – set of VPN gateway ,
P – set of ***NPC*** poles.
A= {$l_{vi,vj}$ =(v_i,v_j) | v_i,v_j ϵ V } - channels set of overlay network.
Q *($l_{vi,vj}$,Δt)* ***= (B, D, L, J)*** is the function on ***A,*** *Δt* – interval of time;
W = {wi = (si1, ….., sik)}, set of SFC where ***sij*** ϵ ***AS U VNF,***
sij = <cr, m, h, Q *($l_{vi,vj}$,Δt)* ***> ;***
ET: (AS U VNF) x CN → R - estimations of the execution time of ***sij*** ϵ ***AS*** U ***VNF,*** on ***cn_j*** ϵ ***CN***

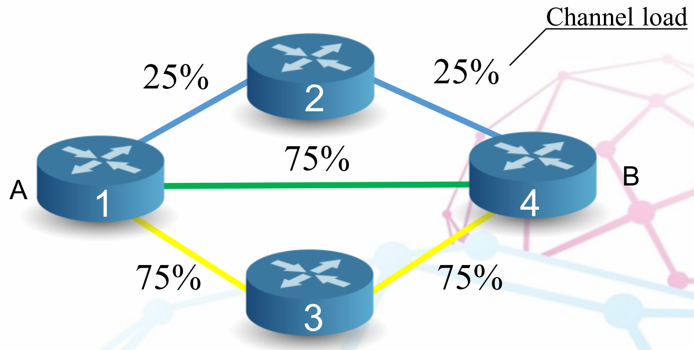
objective function

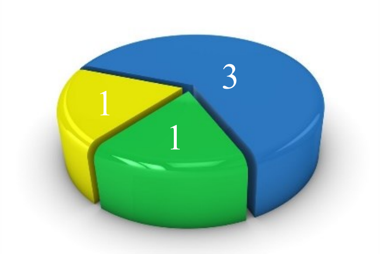
$$F = min \sum_1^{|CN|} \left[\alpha \frac{\bar{c}_i}{c_i} + \beta \frac{\bar{s}_i}{s_i} + \gamma \left(\left(\frac{\bar{c}_i}{c_i} - \Theta \right)^2 + \left(\frac{\bar{s}_i}{s_i} - \Delta \right)^2 \right) \right],$$ where:

α, β, γ – constant values;
c_i, s_i - ***cn_i*** resources are used
$\bar{c}_i, \bar{s}_i$– ***cn_i*** resources and queue length averaged over usage time;
Θ, Δ – used resources of the entire NPC, averaged over time;
$(\boldsymbol{cn_i})_{\boldsymbol{w}}$ is a path in NPC correspond to SLA(w)

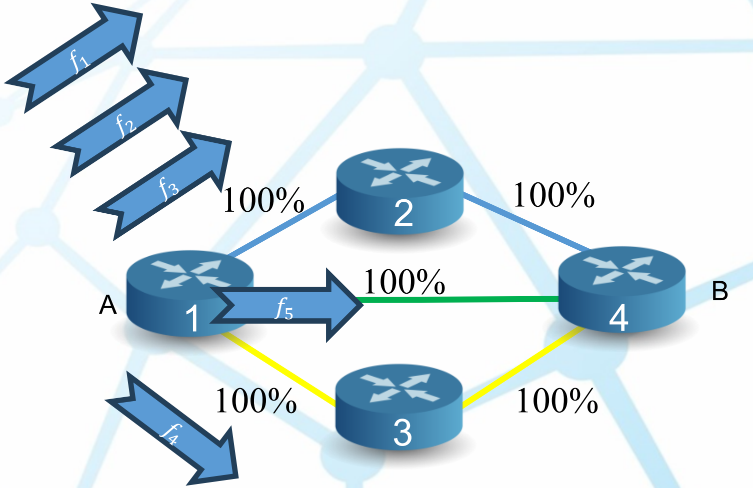
Traffic load balancing

The main goal is to find such weights, so the flow distribution accordingly to these weights provides the even channel load





$$\underset{\{R(t_i)\}}{argmin}\,\Phi \mid \Phi = \left\{\frac{1}{N}\sum_{u,v}\left(\frac{b_{u,v}}{c_{u,v}} - \mu'\right)^2\right\}, \mu' = \frac{1}{N}\sum_{u,v}\frac{b_{u,v}}{c_{u,v}}$$



Weights are updated on each NPCR based on the current channel load, current weight values and information from neighbors

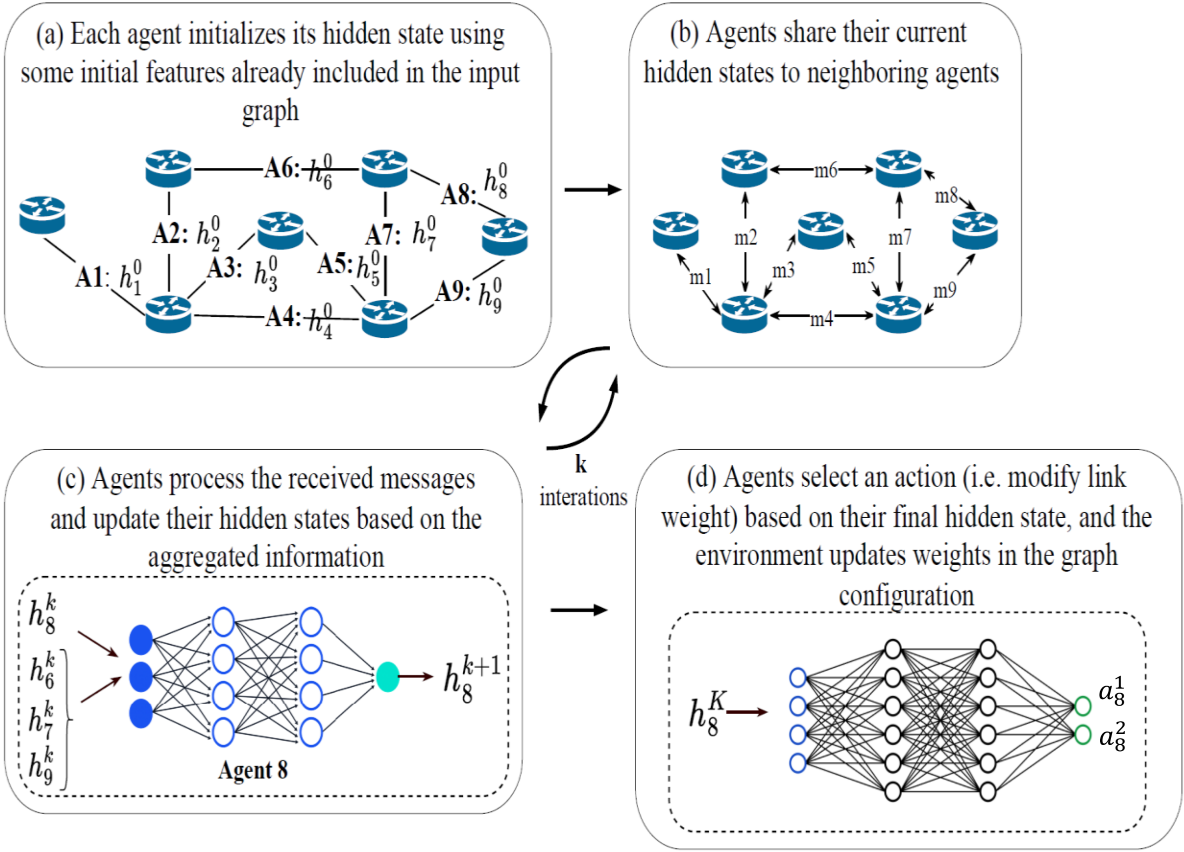
MAROH – Multi-agent Routing using Hashing

Distributed approach:

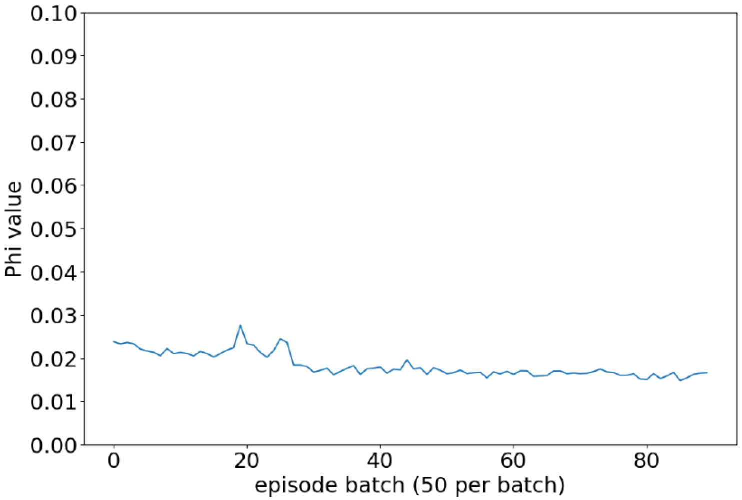
- One agent is on every NPCR
- They exchange messages with neighboring agents (1 degree neighborhood)
- Each agent can modify channel weights to minimize the goal function value

Features:

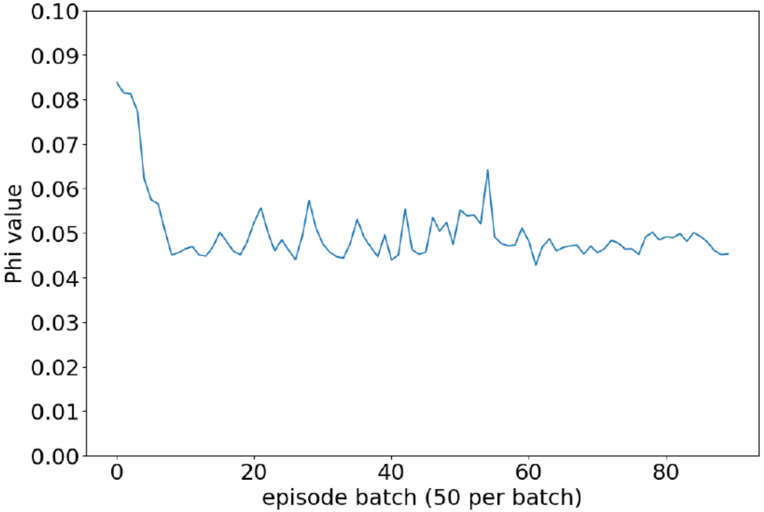
- MPNN – message-passing neural network
- State consists of occupied bandwidth and weight values
- Actions (addition and multiplication) change weights
- Reward - $\Phi(t_{i-1}) - \Phi(t_i)$



Algorithm convergence



Algorithm convergence for 40% average network channels load

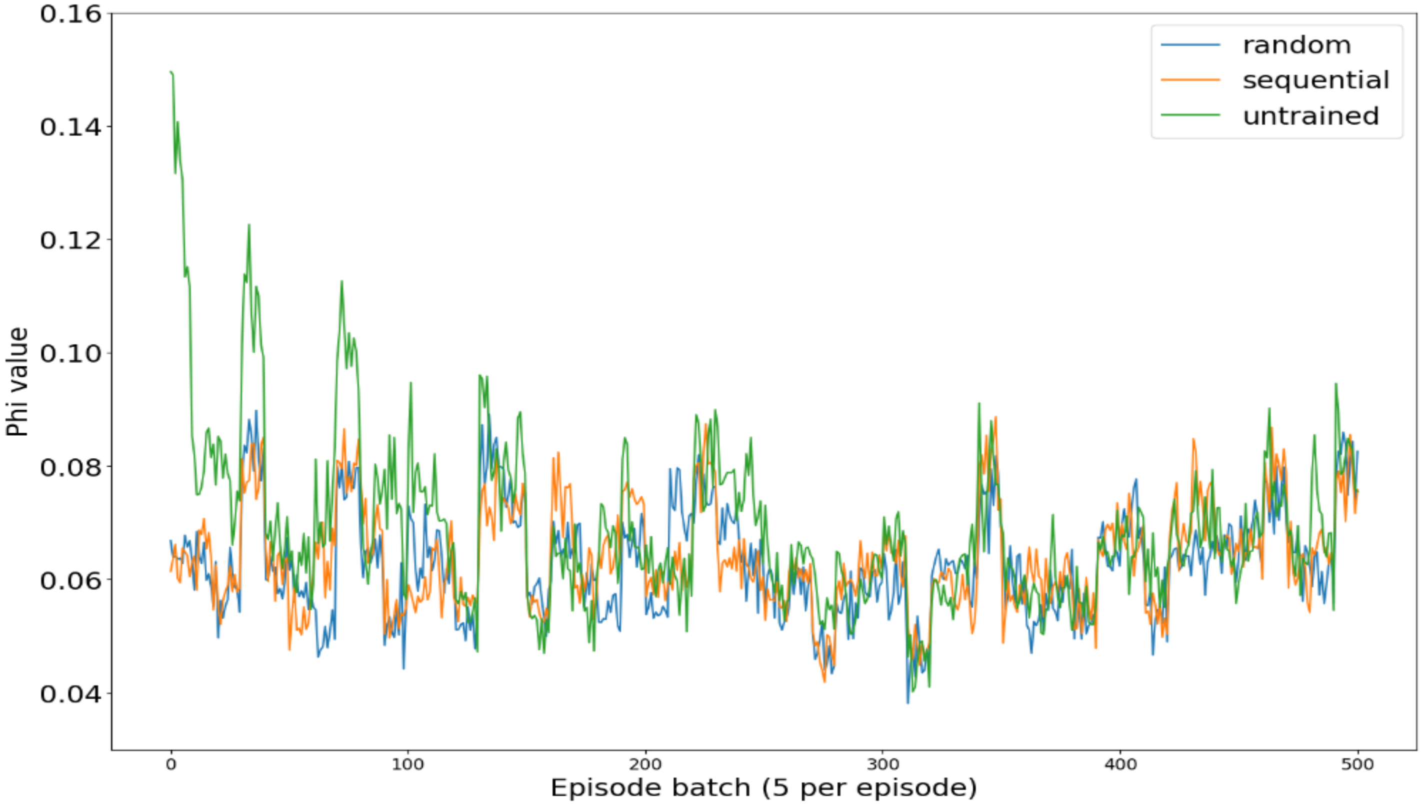


Algorithm convergence for 60% average network channels load

MAROH is more effective under high network load.

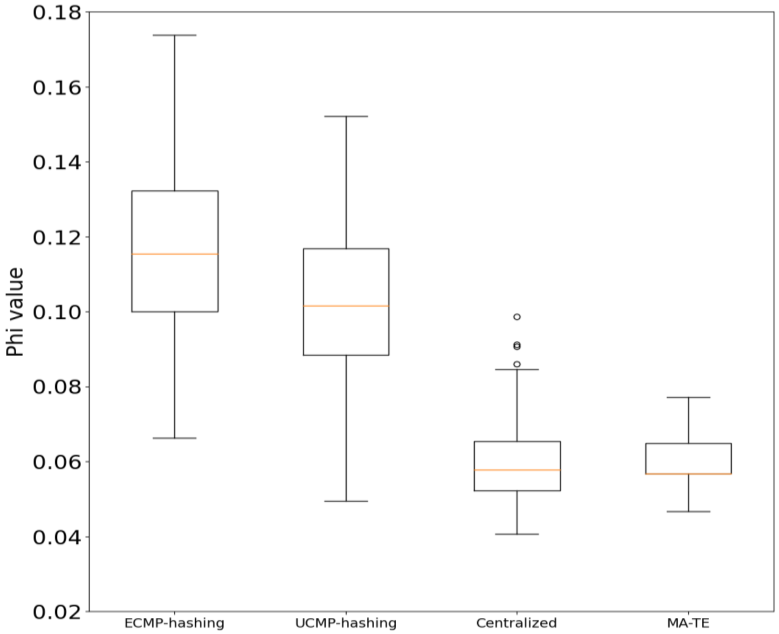
APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

Comparison of training methods



It is required 500 episodes for untrained model to get comparable results with trained models

Comparison with existing solutions



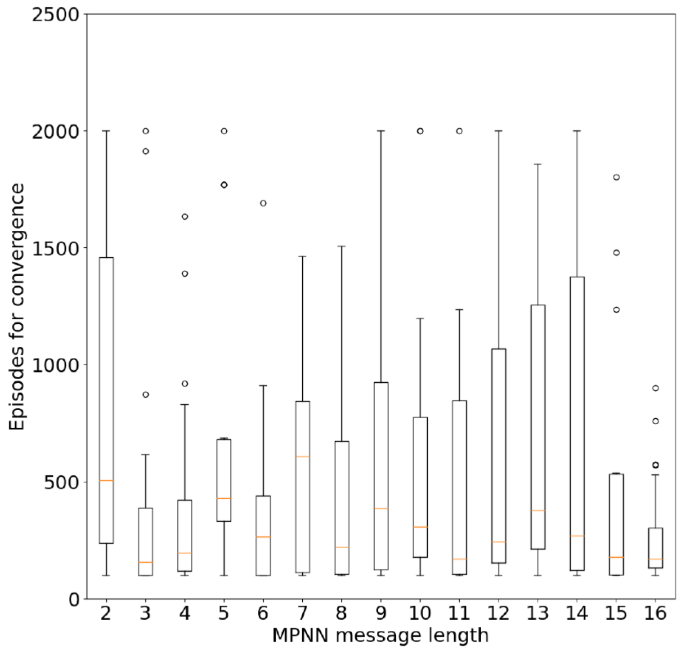
Comparison ECMP vs UCMP vs Centralized vs MAROH

1. ECMP – all weights are equal to 1. Paths are the shortest ones.
2. UCMP – all weights are calculated based on current load. There is no communication between NPCRs.
3. Centralized approach has the global NPC state as input. Heuristic algorithm was used.
4. MA-TE (Multi-agent traffic engineering) represented by MAROH shows the minimum deviation of the objective function.

MAROH approach has significantly better results compared to ECMP and UCMP and similar results compared to centralized approach.

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

Simulation results – parameters tuning



Number of episodes for convergence with varying length of the MPNN message

Experiments showed that any values of K **(number of message exchanges)** higher than **graph diameter** demonstrated poor behavior

The lowest values of M with more stable convergence speed are **M = 7 and M = 8**. The median value and range are higher compared to M = 16, but it comes with the benefit of **smaller messages**.

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

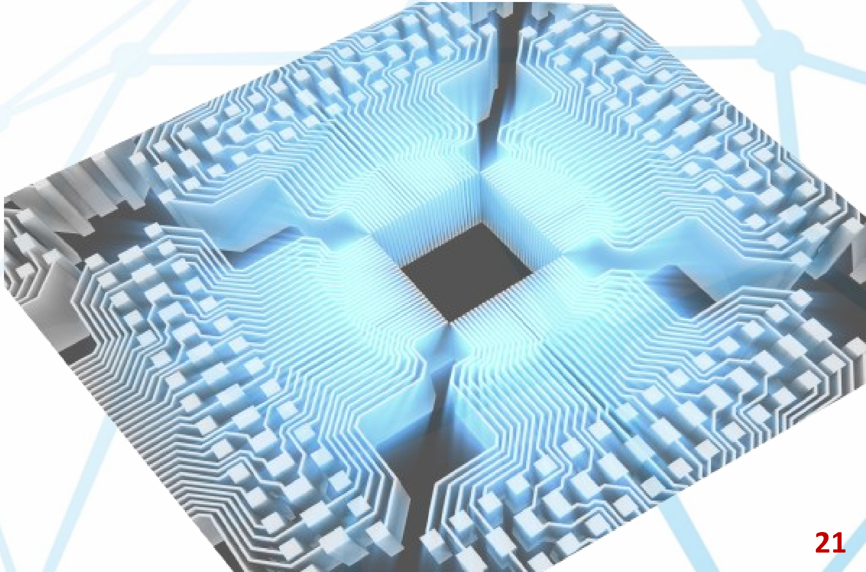
Conclusion

- Growth of network and computational performance are the big challenges for Computational Infrastructure management and control
- The Network Powered by Computing Environment is the next generation of Computational Infrastructure
- The scaling range of the network service is huge and in real time, which put high demands on the algorithm time complexity.
- Only sub-optimal solutions are available using methods based on machine learning

AI let us enable NPC environment to be efficient and scalable.

NPC with AI will make our network to be Super Large Scalable Computer – with predictable behavior, secure, reliable, fault tolerant and scalable.

THANKS



Contacts:

estepanov@lvk.cs.msu.ru

smel@cs.msu.su

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS