ON THE PLATFORN FOR OPTIMAL OVERLAY CHANNEL SELECTION IN NETWORK POWERED by COMPUTING INVIRONMENT”.

R. Smelyanskiy, E. Stepanov

Moscow State University

APPLIED
RESEARCH
CENTER FOR
COMPUTER
NETWORKS

Content

- **New Age for Computational Infrastructure**
- **Network Powered by Computing : concept**
- **InOpSys platform: Intelligent automatic network transport Optimization System**
- **Problem statement and Modeling**
- **Experimental results**

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.

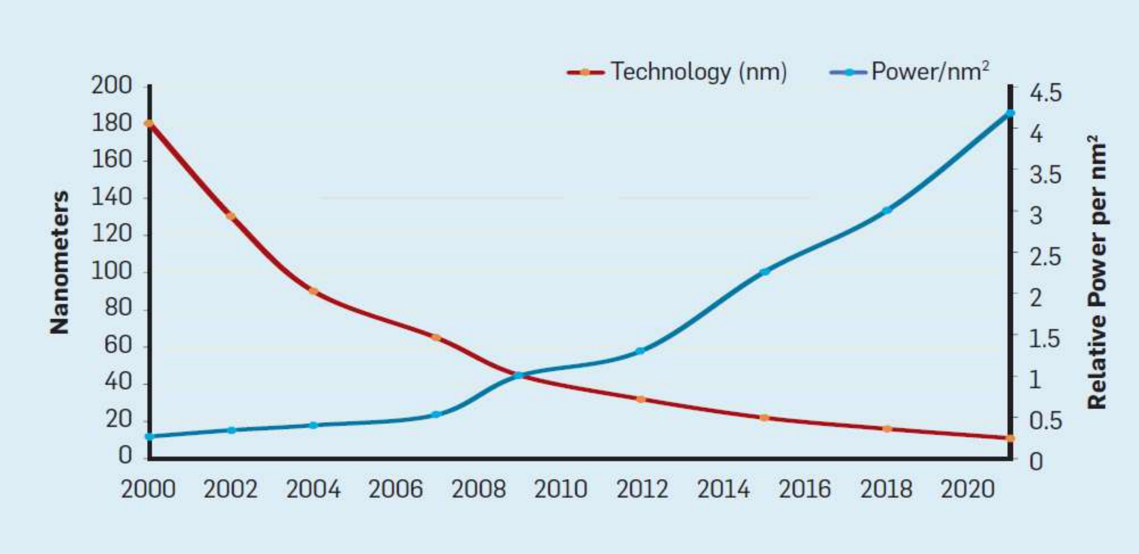
BY JOHN L. HENNESSY AND DAVID A. PATTERSON

A New Golden Age for Computer Architecture

New Golden Age of Computational Infrastructure

In 2022 the energy consumption of DC over the World was 200 GW x 24 hours x 365 days = 1 752 10^{12} Wtph

- 70-s - mainframe computer center with terminal

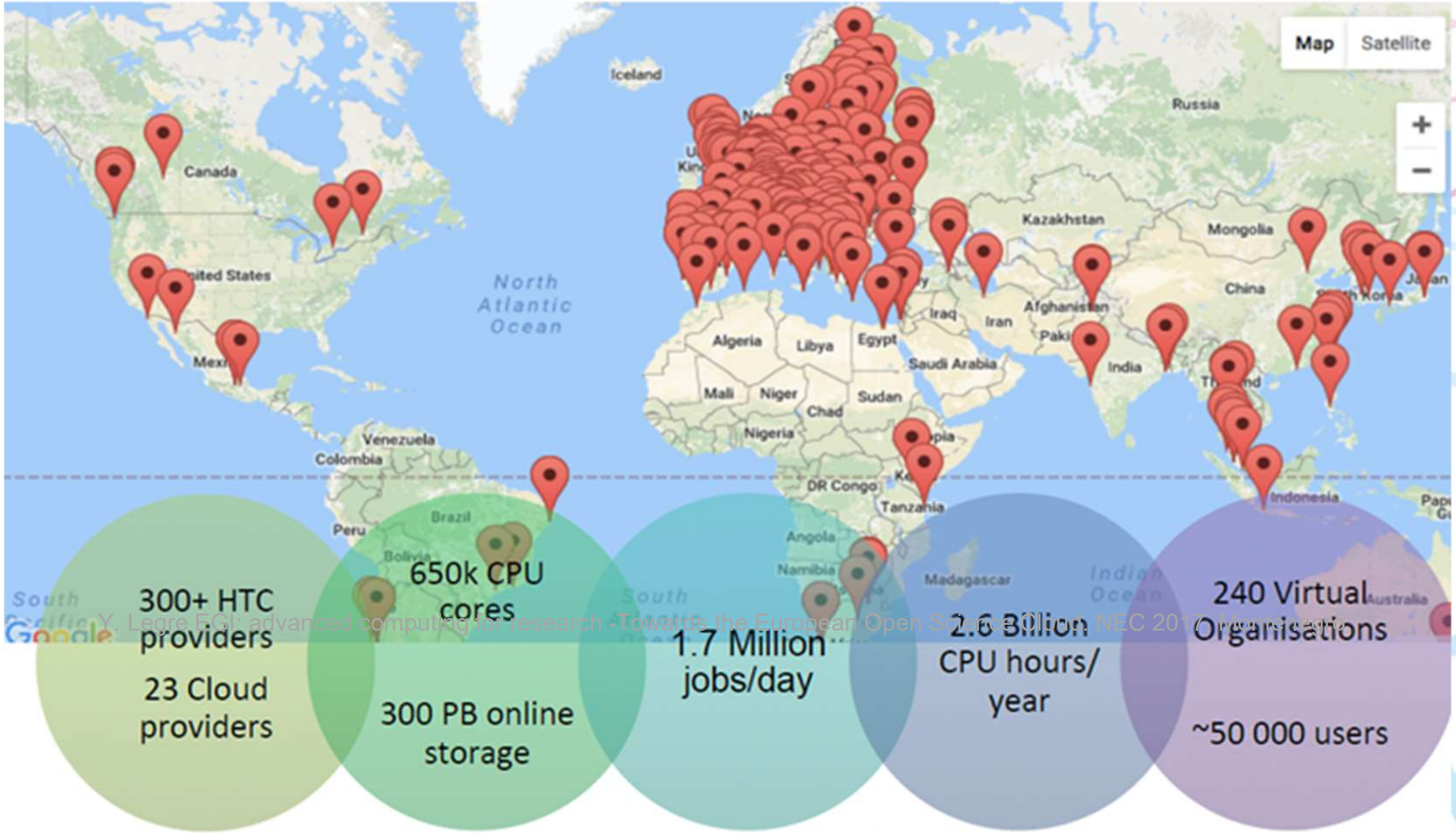


ps on Fiber-optic network

tabit-transmission-mode-fiber-standard.html

Application Requirements + Hardware Capabilities + Software Engineering

EGI Federated Infrastructure



300+ HTC providers

23 Cloud providers

650k CPU cores

300 PB online storage

1.7 Million jobs/day

2.6 Billion CPU hours/ year

240 Virtual Organisations

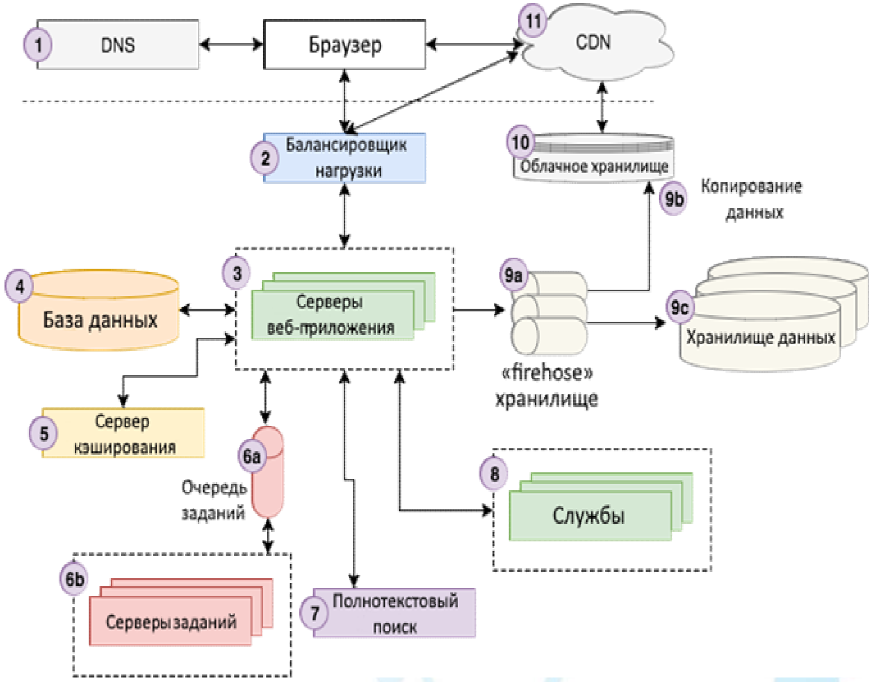
~50 000 users

Y. Legre EGI: advanced computing for research -Towards the European Open Science Cloud, NEC 2017, Montenegro

Applications suite of features

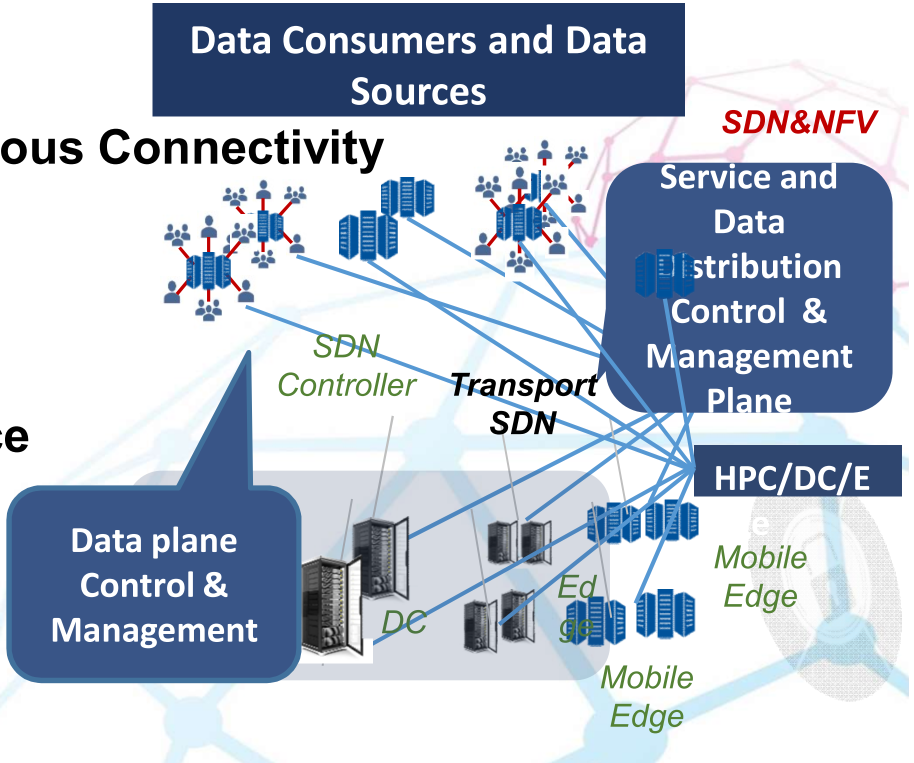
The main force of computational infrastructure developments are applications needs!

- Distributed
- Real-Time mode
- Elasticity (SLA)
- Cross-platform
- Self-sufficiency
- Interaction and Synchronization
- Maintainability



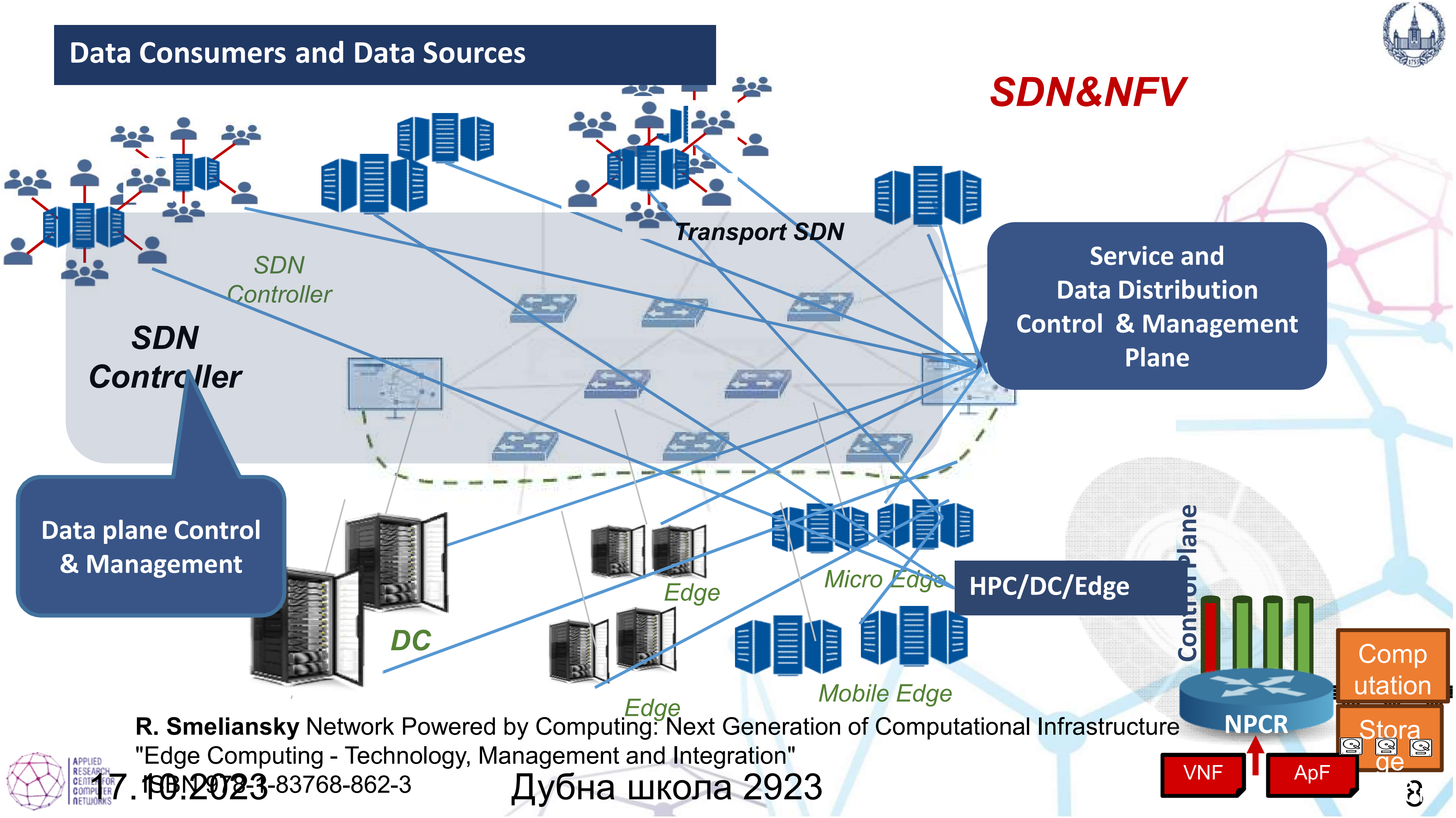
Computational Infrastructure Requirements

- Distributed Computing power and ubiquitous Connectivity
- Deterministic communication QoS
- Computing & Network Power Awareness
- Virtualization, Scalability, Serverless
- Availability, Reliability and Fault Tolerance
- Efficiency and Fairness
- Security



- 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

SDN&NFV



Data Consumers and Data Sources

Transport SDN

SDN Controller

SDN Controller

Service and Data Distribution Control & Management Plane

Data plane Control & Management

DC

Edge

Micro Edge

HPC/DC/Edge

Edge

Mobile Edge

Control Plane

NPCR

Computation

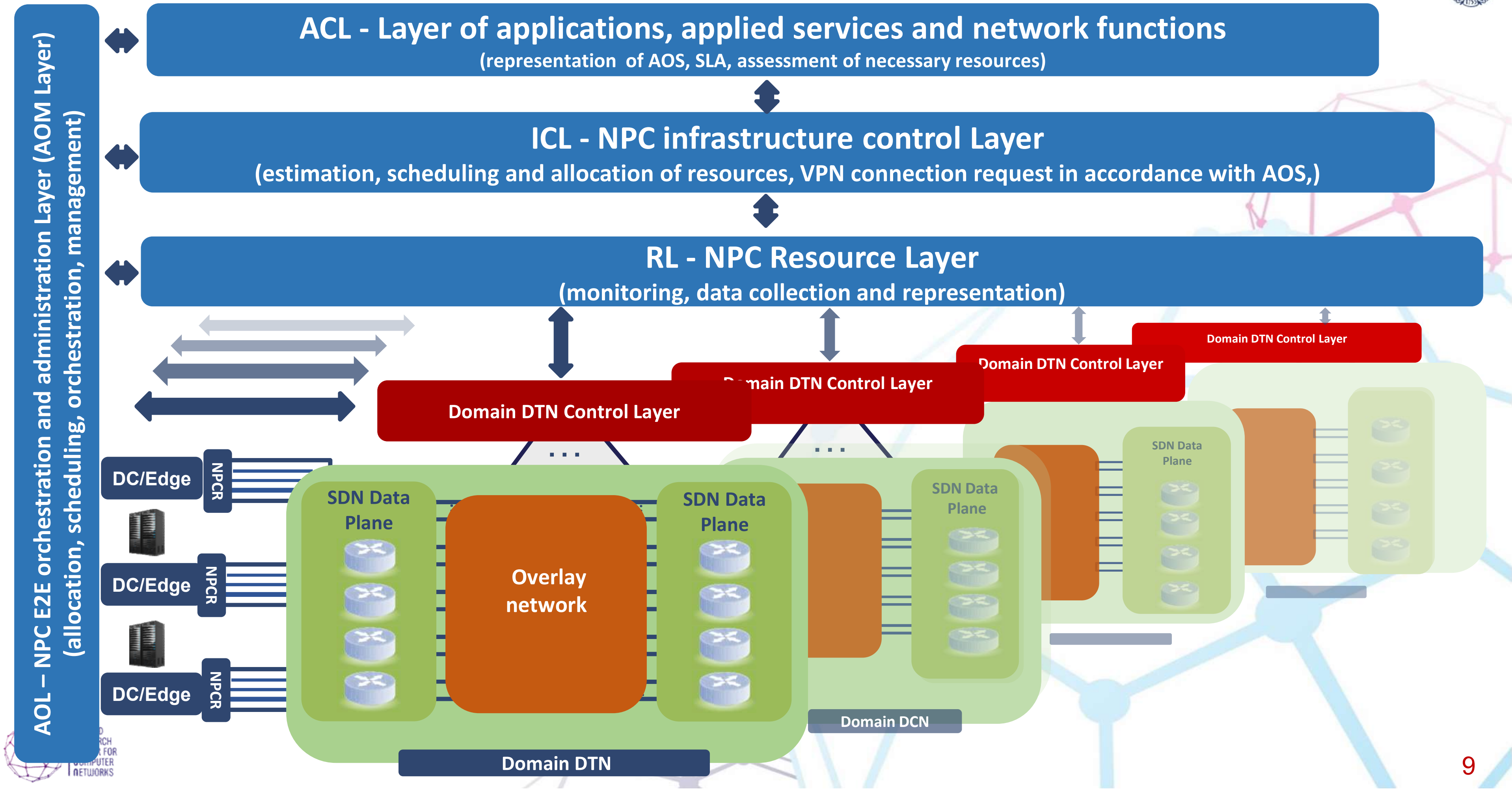
Storage

VNF

ApF

R. Smeliansky Network Powered by Computing: Next Generation of Computational Infrastructure
"Edge Computing - Technology, Management and Integration"
ISBN 978-1-83768-862-3

NPC Functional Architecture



ACL - Layer of applications, applied services and network functions

(representation of AOS, SLA, assessment of necessary resources)

ICL - NPC infrastructure control Layer

(estimation, scheduling and allocation of resources, VPN connection request in accordance with AOS,)

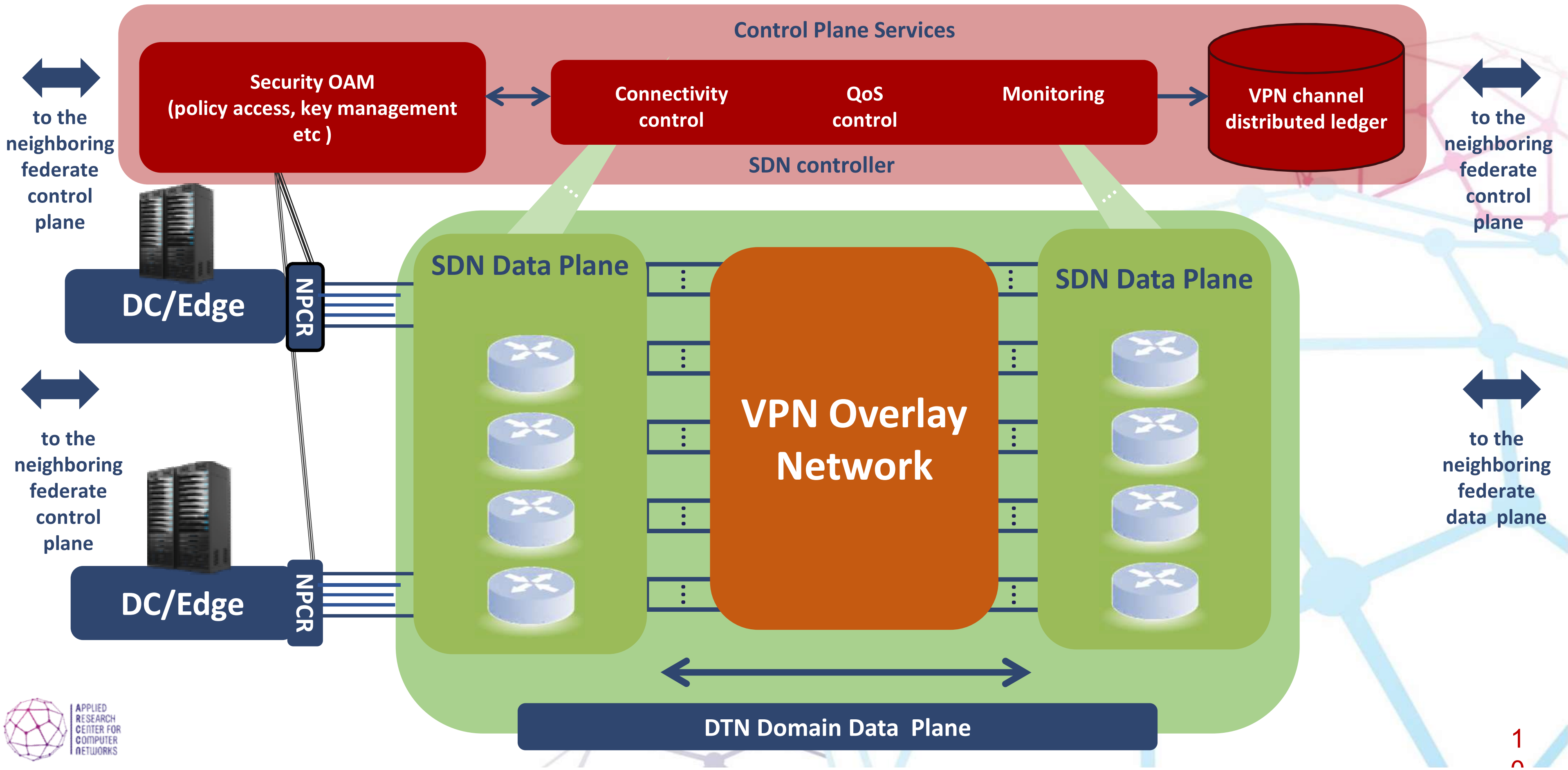
RL - NPC Resource Layer

(monitoring, data collection and representation)

AOL – NPC E2E orchestration and administration Layer (AOM Layer)

(allocation, scheduling, orchestration, management)

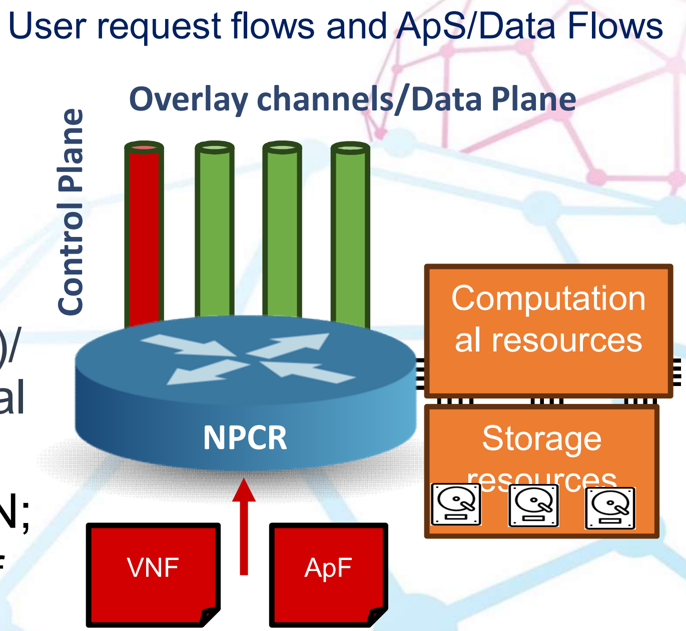
NPC intra DTN Layer



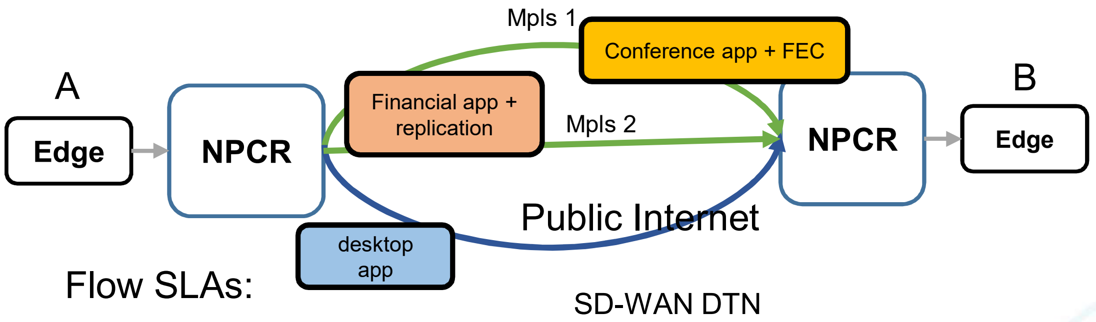
APPLIED
RESEARCH
CENTER FOR
COMPUTER
NETWORKS

Problems Road Map

- Implementation of ACL, ICL, NPC RL, NPC AOL functionality of Service and Data Distribution Management and Control Plane
- **Optimal data traffic routing control**:
    - **Selection of optimal overlay channel;**
    - Data traffic balancing;
- **Optimal allocation of application functions** (ApF)/ virtual network functions (VNF) across computational nodes (CN) of DP plane:
    - ApF execution time estimation for certain CN;
    - Selection of CN that optimal for execution of certain ApF/VNF



Intelligent automatic network transport Optimization System: Problem description



Flow SLAs:

User Requirements: A→B
- Conference app:
  - Latency: 30ms, loss: 0.01, jitter: 10ms
- Desktop app:
  - Latency: 80ms, loss: 0.02, jitter: 30ms
- Financial app: high available
  - Latency: 40ms, No loss, no-jitter

Techniques:

Available Techniques:
- FEC:
  - Redundant pkt to fix packet loss
- TCP acceleration:
  - TCP proxy with new CC algorithm

User:
- Input flow SLA requirement

Prototype:
- Monitor:
  - collet network states, links' KPI、CPU、memory etc.
- Evaluator:
  - if link can fit the flow, if not why?
- Policy generator:
  - Generate tech policies to improve the link

Results:
- Improve network to meet the SLA requirement

Links states:

Three links between A and B:
- MPLS link 1:
  - Latency: 30ms, loss: 0.02, jitter: 10ms
- MPLS link 2:
  - Latency: 40ms, loss: 0.005, jitter: 5ms
- Internet link:
  - Latency: 70ms, loss: 0.01, jitter: 30ms

Optimization Problem: Statement

Each coming flow has SLA $\mathcal{A} = (B, D, J, L)$.
Each SD-WAN channel has current channel state :
$S_r = (\underline{R}_r, \hat{R}_r, \overline{R}_\mathrm{r}, \mathrm{N}_\mathrm{r}, \mathrm{l}_\mathrm{r}, \widehat{j_r}, \overline{j_r}, \widehat{b_r}, \overline{b_r}, \mathrm{h}_\mathrm{r})$

subject to the SLA constraints:

$$\boldsymbol{\xi_b \mathcal{B}(S_r, C)} - \mathcal{B} \geqslant 0$$

minimal admissible bandwidth,

$$D - \xi_d^1 \frac{(K + \Upsilon(C))p}{K\mathcal{B}(S_r, C)} - \xi_d^2 \hat{R}_r \geqslant 0$$

maximal admissible time delay,

$$J - \xi_j \left(1 + B(h_r)^+\right) \bar{j}_r \geqslant 0$$

the maximal admissible jitter,

$$L - \mathcal{L}(S_r, C) \geqslant 0.$$

maximal admissible probability for packet loss.

The decision vector $\boldsymbol{C} = (r, f, \gamma, c, \delta)$

$r \in \{1, \ldots, R\}$ – channel number
$f \in F$ – FEC algorithm
$\gamma \in \Gamma$ – FEC algorithm parameters
$c \in C$ – congestion control algorithm
$\boldsymbol{\delta} \in \boldsymbol{\Delta}$ – congestion control algorithm parameters
$\boldsymbol{\bar{l}_r}$ *- max. probability of retransmit.*

$\mathcal{B}$ *– min admissible bandwidth*
D *– max time delay*
J *– max jitter*
L *– max loss*
p *– packet length*
$\underline{R}_r, \hat{R}_r, \overline{R}_r$ *- min, average, max RTT*
N_r *- number of sent packets*
l_r *- packet loss rate*
$\widehat{j_r}, \overline{j_r}$ *- average, max jitter*
$\widehat{b_r}, \overline{b_r}$ *- average, max bandwidth*
h_r *- current total load*
K – inf. pack. batch size
$\boldsymbol{\mathcal{E}_r} = (T_r, C_r)$
T_r, C_r *- rent period and price*
M(C) – FEC batch size

$$\min_{C} \left(\boldsymbol{Q}(\boldsymbol{S_r}, \boldsymbol{\mathcal{E}_\mathrm{r}}, \mathbf{C}) = \frac{\mathbf{C_r}}{\boldsymbol{T_r}} \cdot \frac{\boldsymbol{K} + \boldsymbol{M}(\boldsymbol{C})}{\boldsymbol{K\mathcal{B}}(\boldsymbol{S_r}, \boldsymbol{C})} \boldsymbol{p} \cdot (\mathbf{1} + \boldsymbol{\bar{l}_r})\right)$$

One packet transmission cost for the loop
«transmission – one possible retransmission» for channel ***r***

Packet Loss Codes

Suggested codes to use:

1-PR

- ⊕ - XOR of all packets
- M=1
- Restores one packet

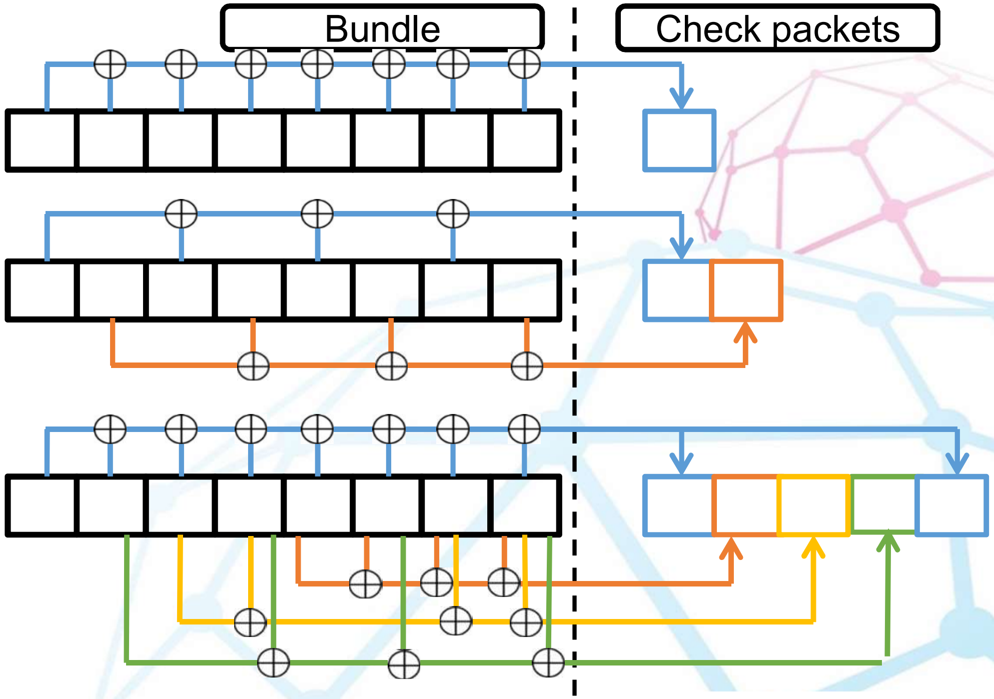
2-PR

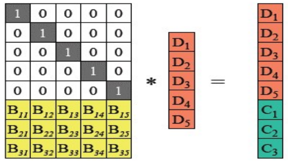
- XOR for even and odd packets
- M=2
- Restores one packet in general and two packets if they are of different parity

R-code

- Generating matrix of Reed-Muller code G with additional last line
- M= $\lceil \log_2 K \rceil + 2$
- Restores two packets

Reed-Solomon Code (RS)

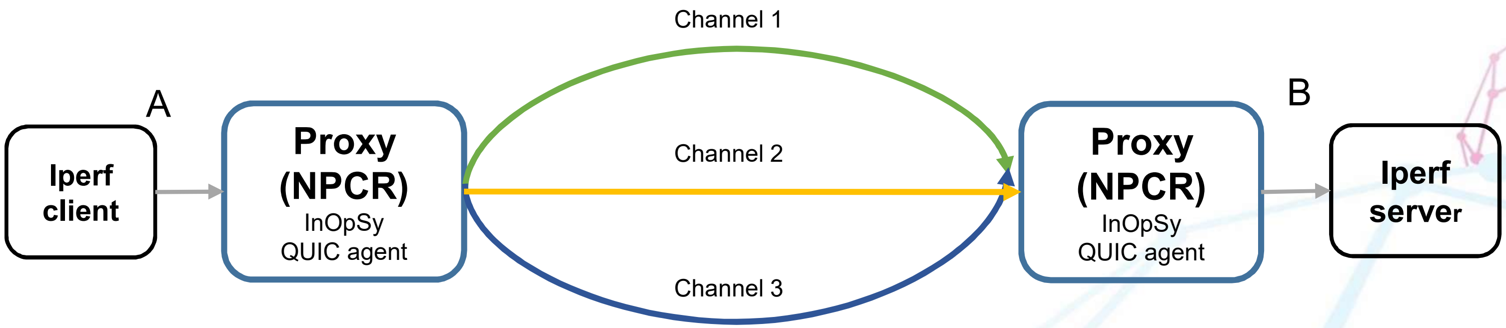




$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\ B_{21} & B_{22} & B_{23} & B_{24} & B_{25} \\ B_{31} & B_{32} & B_{33} & B_{34} & B_{35} \end{pmatrix} * \begin{pmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{pmatrix} = \begin{pmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Testbed InOpSys platform



Hardware:

- Processor: 16 CPUs: Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz
- Memory (RAM): 8 banks with DIMM DDR4 Synchronous 2133 MHz by 8GiB (64GiB total)
- OS: Ubuntu 18.04.6 LTS

Channel QoS and SLA generation:

- RTT: uniform distribution with parameters [10 ms, 100 ms].
- R: uniform distribution with parameters [25 Mbit/s, 50 Mbit/s].
- Loss: uniform distribution with parameters [0.00001 %, 5 %].
- J (jitter): uniform distribution with parameters [0.0 ms, 5 ms].

Congestion control algorithm adjustment

The decision vector $\mathcal{C} = (r, f, \gamma, c, \delta)$, where

$r \in \{1, \ldots, R\}$ – channel number

$f \in F$ – FEC algorithm

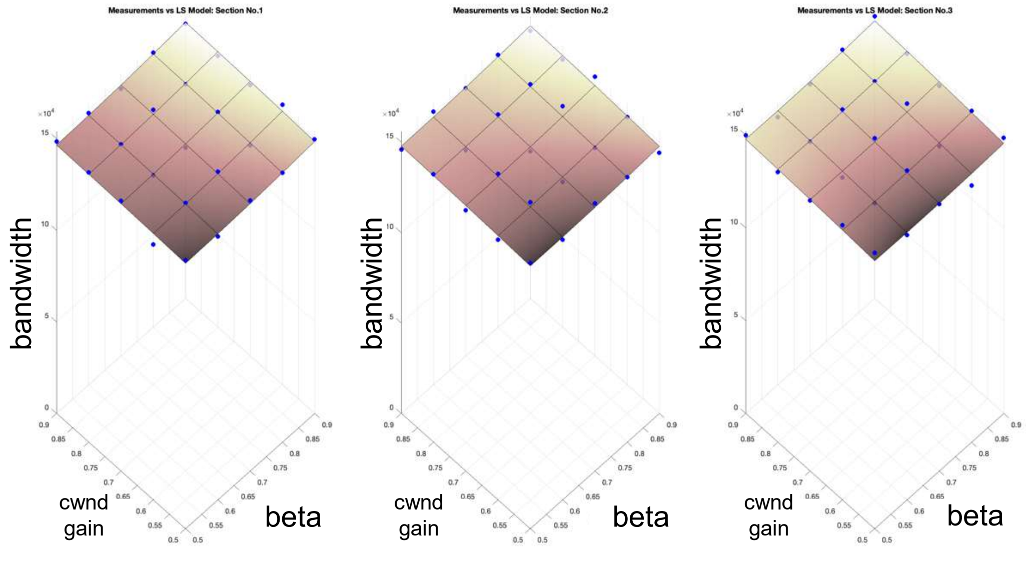
$\gamma \in \Gamma$ – FEC algorithm parameters

$c \in C$ – congestion control algorithm

$\delta \in \Delta$ – congestion control algorithm parameters

Congestion control algorithm BBR parameters :

- BBRLossThresh = 2
- BBRBeta = 0.7
- BBRProbeRTTCwndGain = 0.5
- ProbeRTTDuration = 200 ms



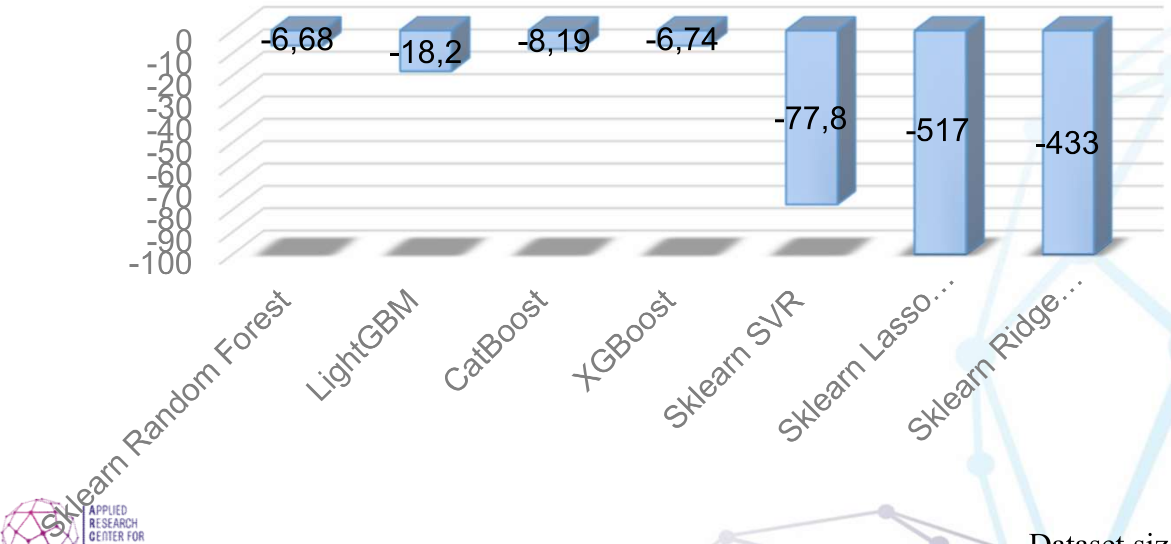
Bandwidth with BBR parameters

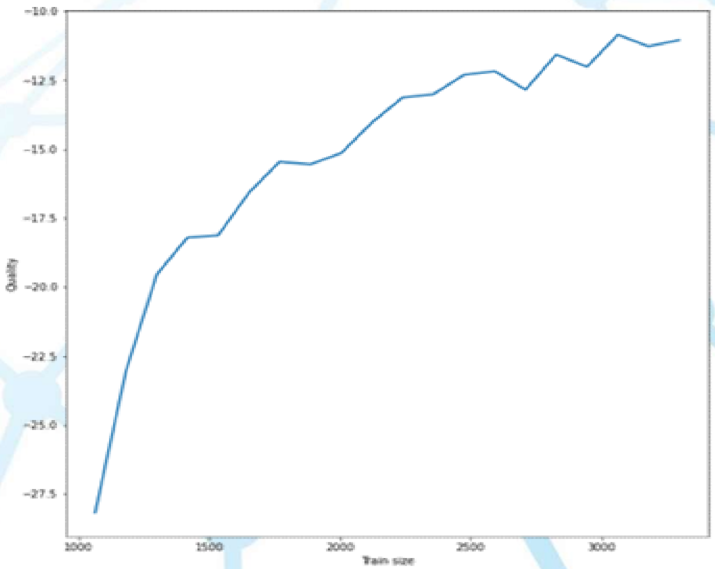
	Sender Speed (Kbit/s)	LOSS3 (%)	MinimalSpeed	MaximalSpeed	RTT ratio
Powell	41846.8102	0.4067	40165.1476	43321.0271	1.3113
Edges	42208.2246	0.3934	40600.2356	43523.0272	1.3177
Powell and Edges ratio	1.0086	0.9674	1.0108	1.0047	1.0048

ML modeling $\mathcal{B}(S_r, \mathcal{C})$ and $\mathcal{C}$ parameters

Find $\mathcal{B}(S_r, \mathcal{C})$- the bandwidth that can be reached on the SD-WAN channel with the current state Sr and chosen decision vector $\mathcal{C}$ – parameters of CC algorithm

Find $\mathcal{C} = (r, f, \gamma, c, \delta)$, *where* $\delta \in \Delta$ – congestion control algorithm parameters: BBRLossTresh, BBRBeta, BBRProbeRTTCwndGain, ProbeRTTDuration





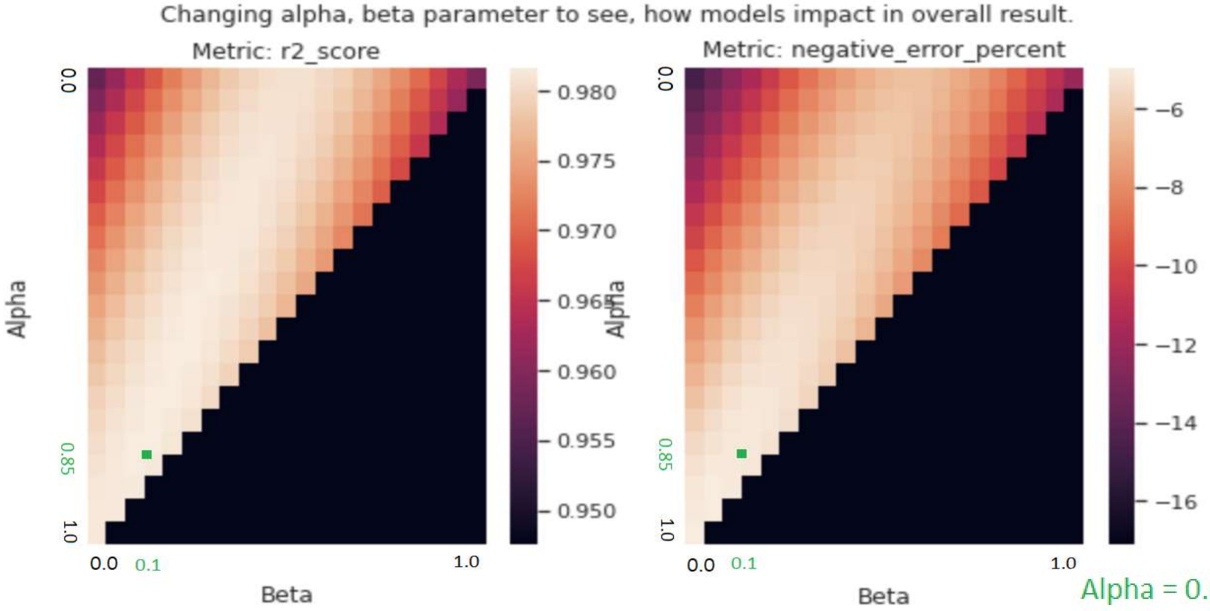
Dataset size vs model quality = deviation model vs experiment.

ML models Ensemble

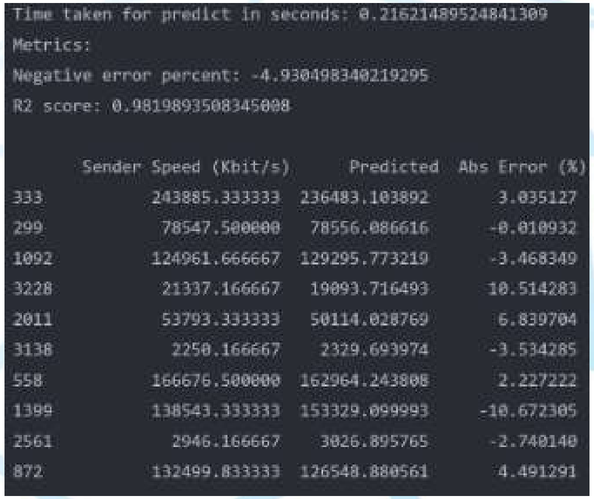
- α is changing in [0, 1]
- β is changing in [0, 1 - alpha]

Then, our ensemble have following structure:

$$MODEL1\_PRED * \alpha + MODEL2\_PRED * \beta + MODEL3\_PRED * (1 - \alpha - \beta)$$


Changing alpha, beta parameter to see, how models impact in overall result.

Alpha = 0.85
Beta = 0.1



```
Time taken for predict in seconds: 0.21621489524841309
Metrics:
Negative error percent: -4.930498340219295
R2 score: 0.9819893508345008

        Sender Speed (Kbit/s)      Predicted  Abs Error (%)
333             243885.333333  236483.103892       3.035127
299              78547.500000   78556.086616      -0.010932
1092            124961.666667  129295.773219      -3.468349
3228             21337.166667   19093.716493      10.514283
2011             53793.333333   50114.028769       6.839704
3138              2250.166667    2329.693974      -3.534285
558             166676.500000  162964.243808       2.227222
1399            138543.333333  153329.099993     -10.672305
2561              2946.166667    3026.895765      -2.740140
872             132499.833333  126548.880561       4.491291
```

$ML\ \mathcal{B}(S_r, \mathcal{C})$ model accuracy ~ 0.988

Experiments examples (bandwidth)

Channels (BW-1 experiment)				
	Mean BW (Mbit/s)	Mean RTT (ms)	Mean Jitter (ms)	Loss (%)
№0	100	30	1	0.01
№1	120	30	1	0.01
№2	100	30	1	0.01

SLA				
BW (Mbit/s)	One-way Delay (ms)	Jitter (ms)	Loss (%)	Chosen Channel
80	16	3	0.001	№1

BW-1: Three channels match required SLA, sometimes in excess

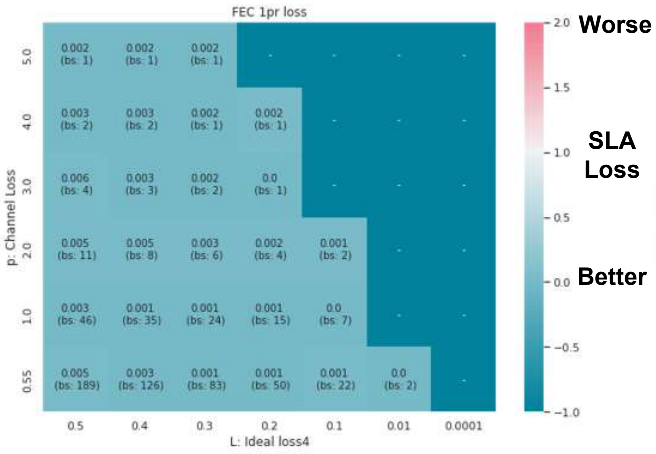
Channels (BW-2 experiment)				
	Mean BW (Mbit/s)	Mean RTT (ms)	Mean Jitter (ms)	Loss (%)
№0	80	30	1	0.01
№1	100	30	1	0.01
№2	80	30	1	0.01

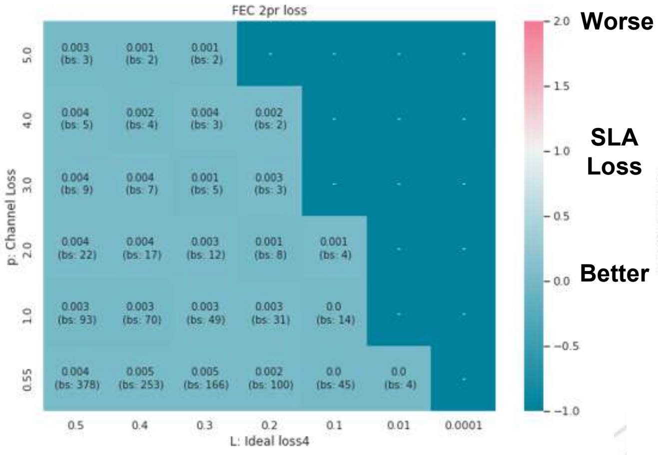
SLA				
BW (Mbit/s)	One-way Delay (ms)	Jitter (ms)	Loss (%)	Chosen Channel
81	16	3	0.001	№1

BW-2: Second channel fits SLA, others are out of SLA because of low bandwidth.

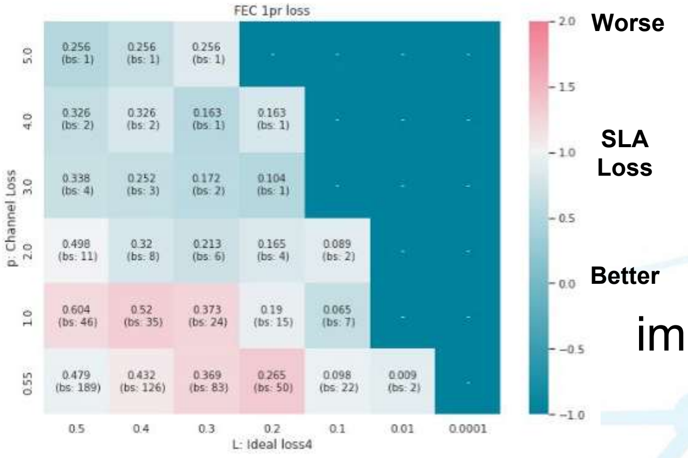
Loss evaluation: UDP vs TCP

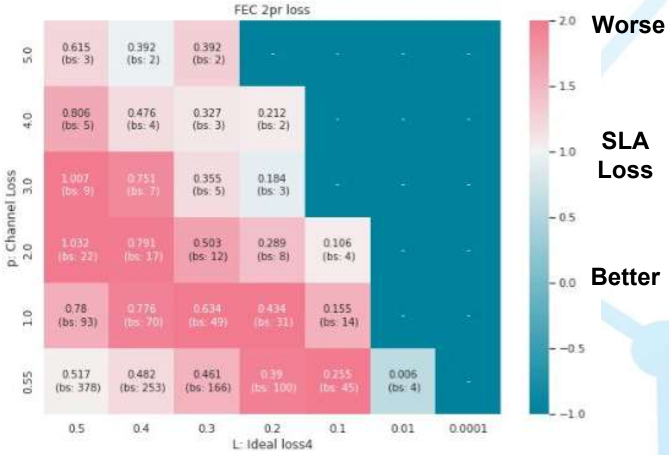
UDP





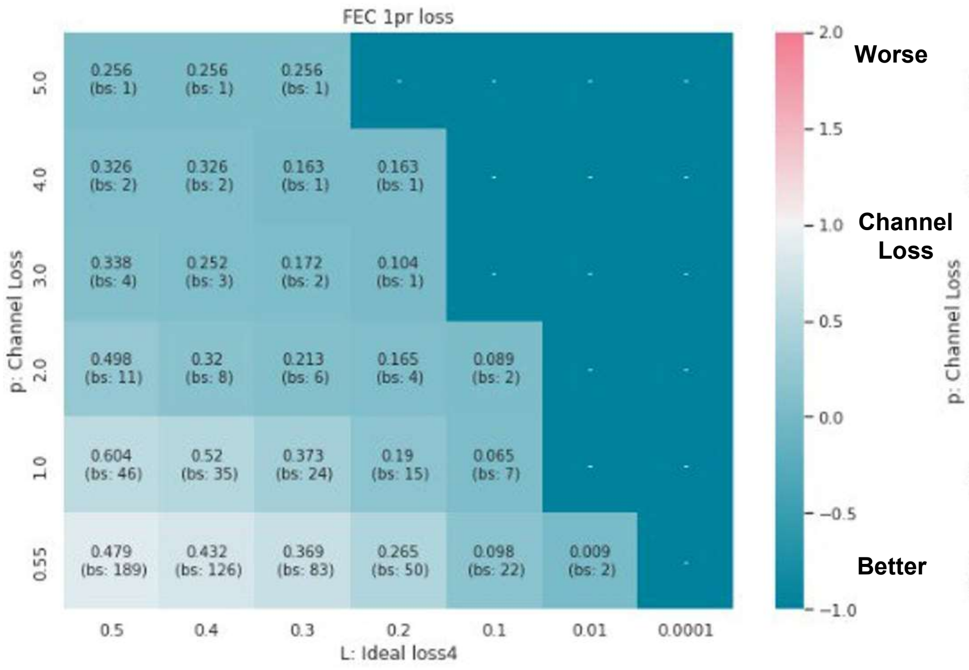
TCP





For UDP flow FEC improves the Loss near all cases.
For TCP FEC requires refining.

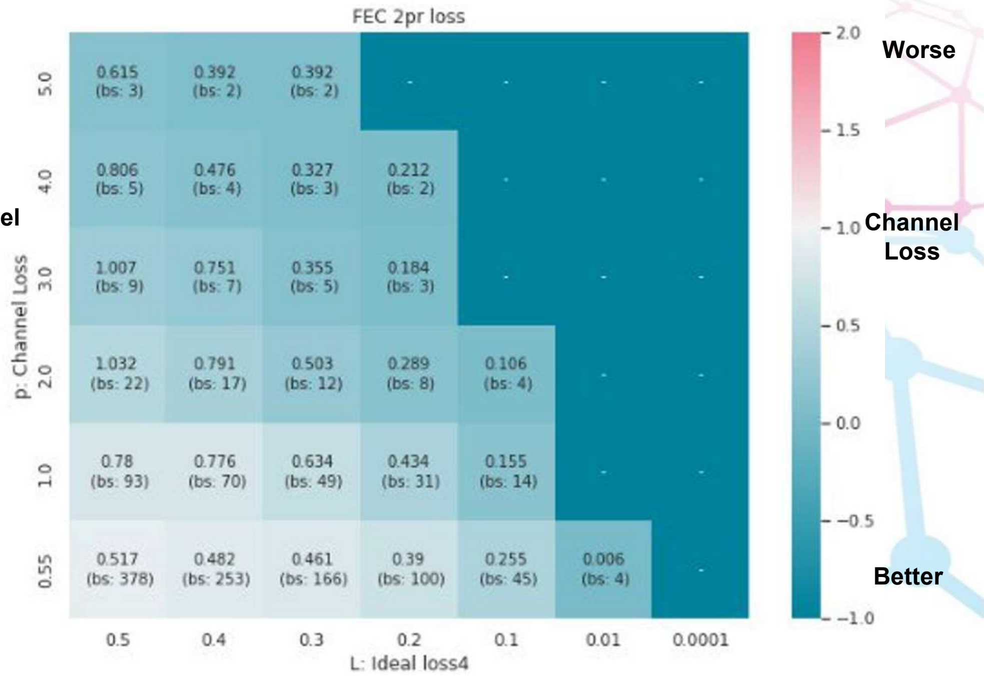
Loss improving: TCP



FEC 1pr loss

p: Channel Loss	0.5	0.4	0.3	0.2	0.1	0.01	0.0001
5.0	0.256 (bs: 1)	0.256 (bs: 1)	0.256 (bs: 1)	-	-	-	-
4.0	0.326 (bs: 2)	0.326 (bs: 2)	0.163 (bs: 1)	0.163 (bs: 1)	-	-	-
3.0	0.338 (bs: 4)	0.252 (bs: 3)	0.172 (bs: 2)	0.104 (bs: 1)	-	-	-
2.0	0.498 (bs: 11)	0.32 (bs: 8)	0.213 (bs: 6)	0.165 (bs: 4)	0.089 (bs: 2)	-	-
1.0	0.604 (bs: 46)	0.52 (bs: 35)	0.373 (bs: 24)	0.19 (bs: 15)	0.065 (bs: 7)	-	-
0.55	0.479 (bs: 189)	0.432 (bs: 126)	0.369 (bs: 83)	0.265 (bs: 50)	0.098 (bs: 22)	0.009 (bs: 2)	-

L: Ideal loss4



FEC 2pr loss

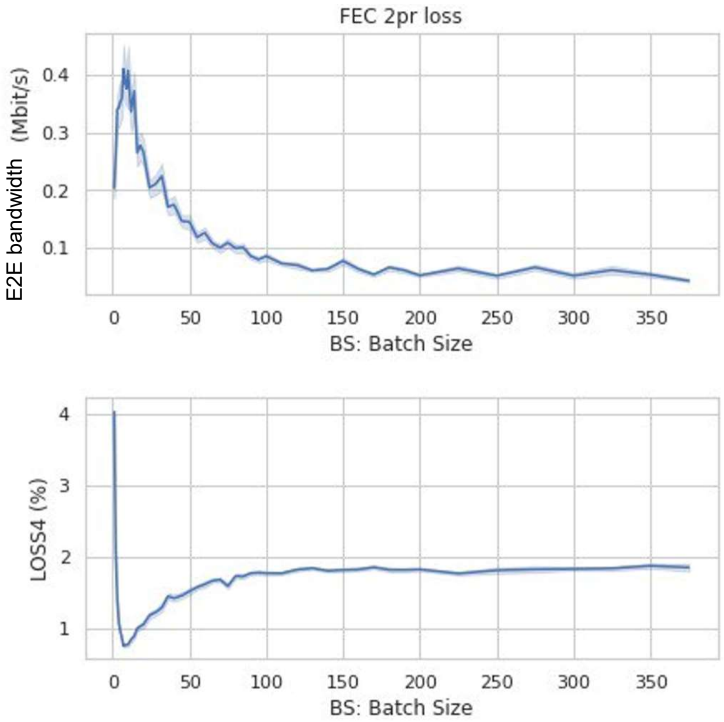
p: Channel Loss	0.5	0.4	0.3	0.2	0.1	0.01	0.0001
5.0	0.615 (bs: 3)	0.392 (bs: 2)	0.392 (bs: 2)	-	-	-	-
4.0	0.806 (bs: 5)	0.476 (bs: 4)	0.327 (bs: 3)	0.212 (bs: 2)	-	-	-
3.0	1.007 (bs: 9)	0.751 (bs: 7)	0.355 (bs: 5)	0.184 (bs: 3)	-	-	-
2.0	1.032 (bs: 22)	0.791 (bs: 17)	0.503 (bs: 12)	0.289 (bs: 8)	0.106 (bs: 4)	-	-
1.0	0.78 (bs: 93)	0.776 (bs: 70)	0.634 (bs: 49)	0.434 (bs: 31)	0.155 (bs: 14)	-	-
0.55	0.517 (bs: 378)	0.482 (bs: 253)	0.461 (bs: 166)	0.39 (bs: 100)	0.255 (bs: 45)	0.006 (bs: 4)	-

L: Ideal loss4

For TCP FEC double reduce loss at least

Why the proposed FEC mechanism does not improve the loss parameter for TCP better then double?

Congestion control window size is less than FEC batch size (zero packets) sometimes



There is an optimum batch size that gives the highest bandwidth and the lowest loss

It is promising to apply machine learning methods to find the optimal value

SLA compliance with adjusted coefficients

Constraints:

$$\xi_b \mathcal{B}(\mathcal{S}_r, \mathcal{C}) - B \geqslant 0,$$

$$D - \xi_d^1 \frac{(K + M(\mathcal{C}))p}{K\mathcal{B}(\mathcal{S}_r, \mathcal{C})} - \xi_d^2 \widehat{R}_r \geqslant 0,$$

$$J - \xi_j \left(1 + B(h_r)^+\right) \bar{j}_r \geqslant 0,$$

$$L - \mathcal{L}(\mathcal{S}_r, \mathcal{C}) \geqslant 0.$$

$$\boldsymbol{\xi_j = 1.36,\ \xi_d^1 = 2.53,\ \ \xi_d^2 = 1.27\ \xi_b = \begin{cases} 0.31, Loss < 0.1\%, \\ 0.71, elsewise \end{cases}}$$

Algorithm	BW SLA	Loss SLA	RTT SLA	Jitter SLA
InOpSys	96%	97%	90%	100%

	$\frac{RTT_{simulation}}{2D}$	$\frac{B_{simulation}}{B}$
mean	0,7911	1,3548
10%	0,5471	2,0611
20%	0,6037	1,3217
30%	0,6518	1,2879
40%	0,6977	1,2500
50%	0,8043	1,2119
60%	0,8711	1,1905
70%	0,8983	1,1716
80%	0,9335	1,1253
90%	0,9991	1,081
all	1,4329	0,7653

worse

better

By adjustment the InOpSys platform meet SLA for more than 90% of the flows

Comparative analysis

Algorithm	Correct channel	RTT ratio	Loss ratio	QUIC speed ratio	End-to-End speed ratio
InOpSys	100%	0,99	0,62	1,49	1,08
InOpSys (without FEC)	100%	1,43	0,75	0,89	0,73
Random	13,70%	2,09	7,64	0,71	0,49
Min RTT	12,60%	1,25	5,43	0,89	0,62
Min Loss	15%	2,31	0,13	1,00	0,70
vQoE	14,60%	1,26	3,65	0,91	0,62

Algorithm	BW SLA	RTT SLA	Loss SLA	Total SLA
InOpSys	96%	90%	97%	90%
Random	9,50%	65%	46%	4,70%
MinRTT	35,70%	90,50%	74,70%	25,20%
MinLoss	49,30%	61,60%	97,20%	35,60%
vQoE	42,20%	90%	76,60%	33,30%

better worse

$$vQoE = vQoE'\left(\frac{BaselineLoss}{Loss}\right) + vQoE'\left(\frac{BaselineRTT}{RTT}\right)$$

© Cisco Systems

InOpSy platform selects the channel with maximal injection speed and relatively small mean loss ratio.

Overheads: “free cheese only in a mousetrap”

Channel selection algorithm overhead

Algorithm channel selection algorithm	Execution time (ms)	RSS (Kbyte)	VMS (Kbyte)	CPU (%)
InOpSys	70,71	48,28	530,29	379,80
Random	0,02	47,88	402,78	98,70
MinRTT	0,02	47,88	402,78	99,60
MinLoss	0,03	47,89	402,76	99,50
vQoE	0,03	47,88	402,78	99,90

Resident Set Size (RSS) – the memory volume available for process in RAM.

Virtual Memory Size (VMS) is the total volume memory available a process can access

QUIC Agent operation overheads

	Client			Server		
	VMS (Kbyte)	RSS (Kbyte)	CPU (%)	VMS (Kbyte)	RSS (Kbyte)	CPU (%)
R-Scheme	14776,00	6016,00	72,30	15094,00	8284,00	44,70
Reed-Solomon	15800,50	14900,00	44,70	16118,50	16796,00	43,10
1PR	14776,00	6176,00	45,00	15094,00	8288,00	42,30
2PR	14776,00	6100,00	47,80	15094,00	8236,00	45,00
No FEC only batch	4251,00	7668,00	40,70	4513,00	7608,00	40,50
No FEC no batch	4249,00	7484,00	46,50	4513,50	7736,00	44,80

better worse

Существующий подход к организации ЦОД

Проблемы существующего подхода:

- **Высокие требования к QoS каналов связи**, для обеспечения доступности сервиса;
- **Проблемы капитального строительства** централизованного ЦОД;
- **Проблема масштабирования**, связанная со строительством новых ЦОД и ручной синхронизацией их работы;
- **Неоптимальное использования доступных ресурсов**, из-за отсутствия централизованной системы управления и

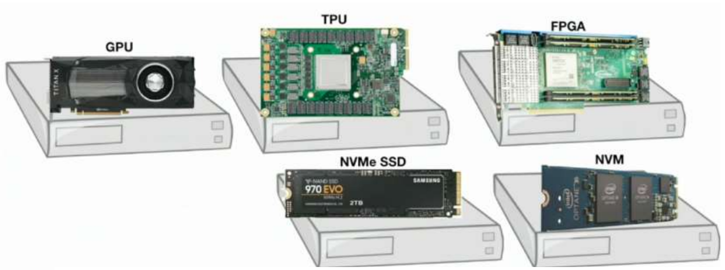
Новый подход к организации ЦОД

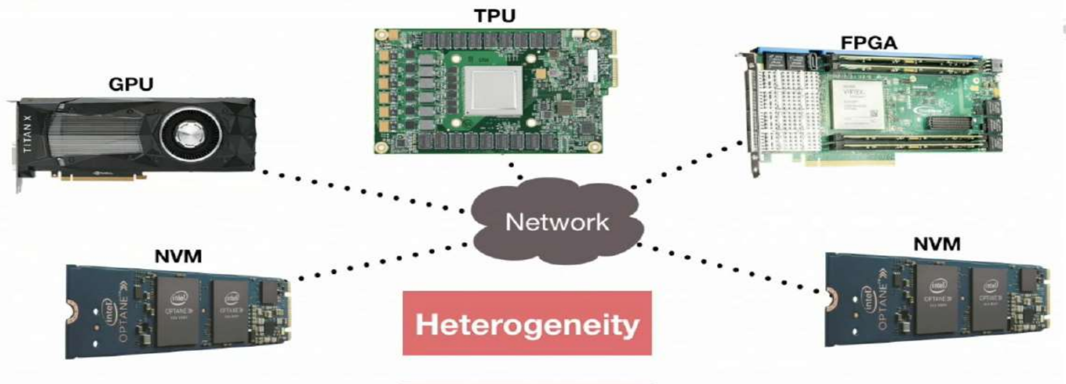
Преимущества нового подхода

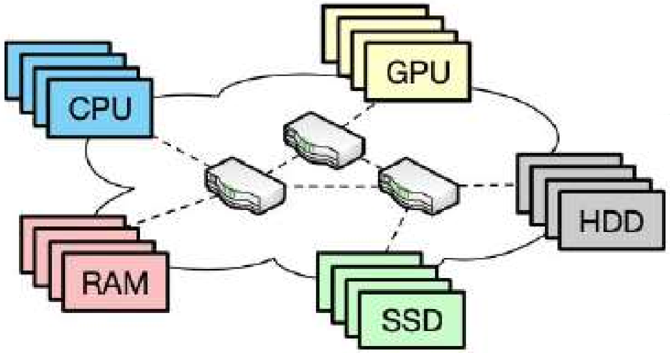
- **Снижение** требований к транспорту за счет близости экземпляра сервиса к конечному потребителю;
- **Снижение затрат** на организацию ЦОД за счет отсутствие необходимости строить централизованный ЦОД;
- **Простое масштабирование** за счет использования централизованной облачной платформы;
- **Повышение оперативности работы сети** за счет централизованной системы управления и оркестрирования и близости сервиса к клиенту.

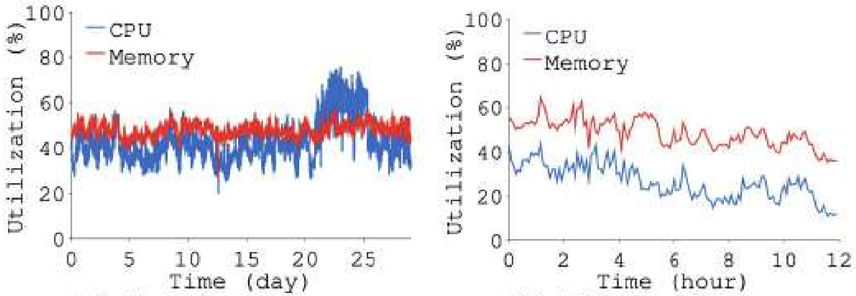
А.В.

Дезагрегированная Архитектура





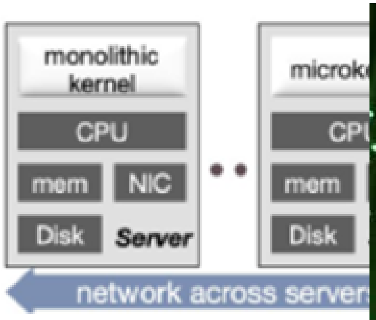




(a) Google Cluster (b) Alibaba Cluster

20% - 60% использования CPU и пам

Эволюция серверной архитектуры



Монолитная

Japan 2022 - 1,520,000 Gbps on Fiber-optic network

https://techxplore.com/news/2022-11-petabit-transmission-mode-fiber-standard.html

https://www.techpowerup.com/262237/broadcom-ships-25-6-tbps-network-switch-on-7-nm-chip#g2

Conclusion

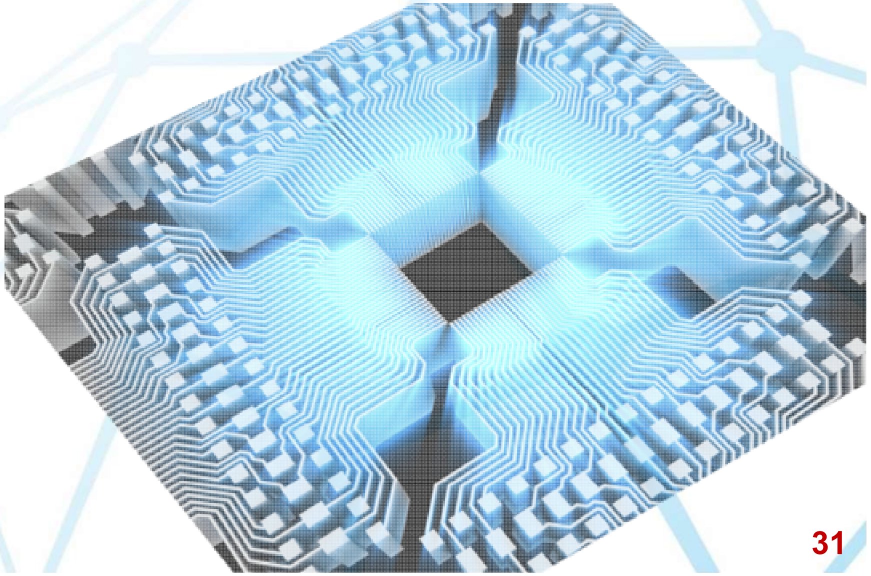
- Growth of Application requirements are the big challenges for Computational Infrastructure management and control
- Network Powered by Computing Environment
  - Distributed Computing power and ubiquitous Connectivity
  - Deterministic communication QoS
  - Computing & Network Power Awareness
- InOpSys (Intellectual transport Optimization System) platform for NPC is presented that allows automatically:
  - select the best channel from the available ones by the metrics takes into account Bandwidth, Delay, Loss, Jitter
  - adjust the parameters of selected channel to meet the SLA requirements

National TE Data Sets for ML methods have to be developed

ML technics enable NPC environment to be predictable, secure, reliable, efficient, scalable.

APPLIED RESEARCH CENTER FOR COMPUTER NETWORKS

THANKS



Contacts: smel@cs.msu.su

APPLIED
RESEARCH
CENTER FOR
COMPUTER
NETWORKS