山东省计算中心（国家超级计算济南中心）
SHANDONG COMPUTER SCIENCE CENTER (NATIONAL SUPERCOMPUTING CENTER IN JINAN)

Remote Direct Memory Access (RDMA) Network:

Taking RDMA over Converged Ethernet (RoCEv2) as an example, and its watermark parameter optimization

Lizhuang Tan
Shandong Computer Science Center
(National Supercomputer Center in Ji'nan), China
tanlzh@sdas.org

1 **About of SCSC (NSCCJN)**

2 RDMA/RoCEv2

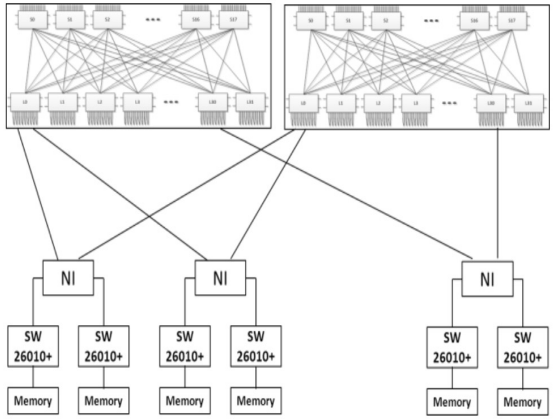
3 ByteTuning

About of SCSC (NSCCJN)

Shandong Computer Science Center (National Supercomputer Center in Ji'nan), *established in 1976, is the largest research institute for computer science and technology in Shandong Province, China.*
It is now affiliated with Qilu University of Technology (Shandong Academy of Sciences).

About of SCSC (NSCCJN)

Shandong Computer Science Center (National Supercomputer Center in Ji'nan), *established in 1976, is the largest research institute for computer science and technology in Shandong Province, China.*
It is now affiliated with Qilu University of Technology (Shandong Academy of Sciences).

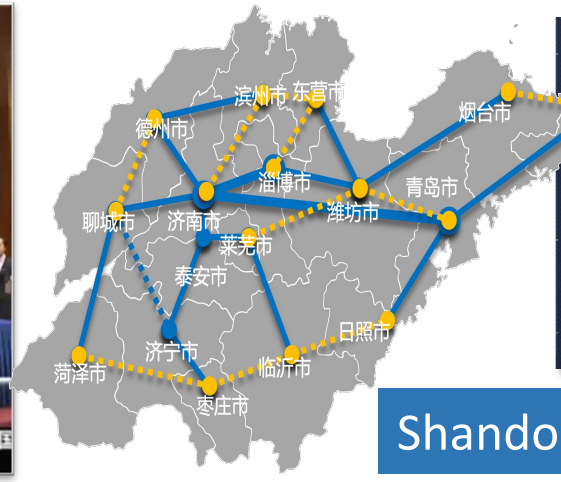


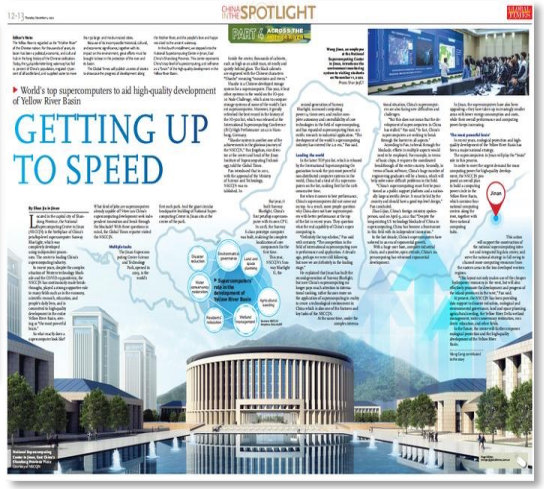
In 2011, the Sunway BlueLight was built.
The first P-class supercomputer based on Chinese-made CPUs in China

In 2018, Sunway E-class prototype
One of China's three fully domestically produced prototypes

About of SCSC (NSCCJN)

Shandong Computer Science Center (National Supercomputer Center in Ji'nan), *established in 1976, is the largest research institute for computer science and technology in Shandong Province, China.*
It is now affiliated with Qilu University of Technology (Shandong Academy of Sciences).

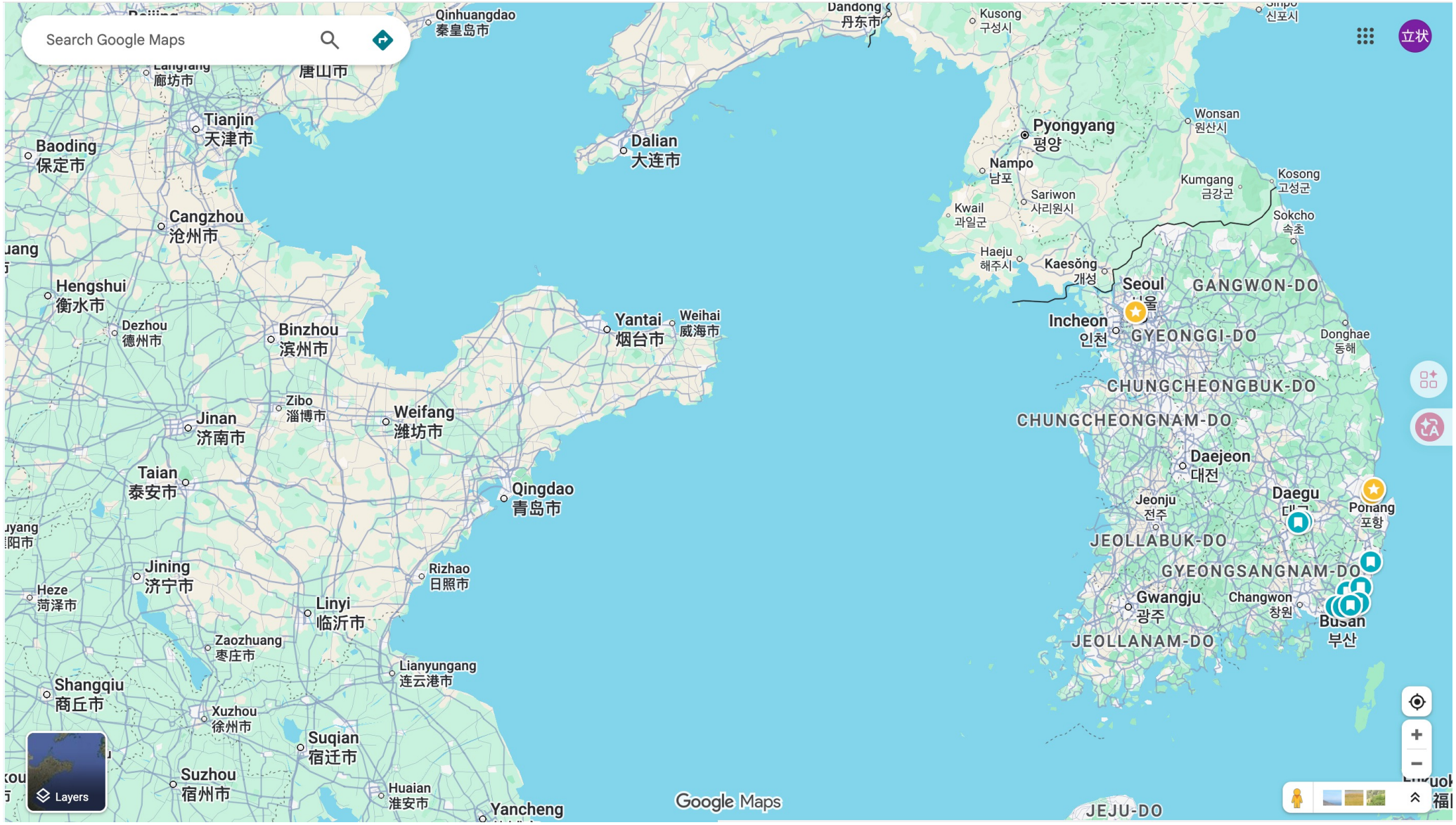




Shandong Computing Network Infrastructure

In 2024, the integration infrastructure for supercomputing and network was built.

About of SCSC (NSCCJN)



About Me

My Website: tanlizhuang.cn

Lizhuang Tan (谭立状) Home CV Teaching Publication Patent Standard Group Service Data Photo Conference

Lizhuang Tan (谭立状)

Associate Researcher, Ph.D.

Shandong Provincial Key Laboratory of Computer Networks,

Shandong Computer Science Center (National Supercomputer Center in Ji'nan),

Qilu University of Technology (Shandong Academy of Sciences)

tanlzh@sdas.org/lzhtan@qlu.edu.cn

Supercomputing Technology Park, No. 28666, Jingshi East Road, Ji'nan, Shandong, China.

About Me

Since July 2022, I have been working in Shandong Computer Science Center (National Supercomputer Center in Ji'nan), China. From October 2024 to October 2025, I am a visiting scholar in Professor James Won-Ki Hong's DPNM Lab at Pohang University of Science and Technology, Korea. From April 2021 to June 2022, I worked as a R&D intern in the High-Speed Network Team of ByteDance, working on RDMA/RoCE under the leadership of Dr. Zhuo Jiang.
In June 2022, I received my Ph.D. degree from the National Engineering Research Center of Advanced Network Technologies, Beijing Jiaotong University. In June 2017, I received my Bachelor of Engineering degree in Communication Engineering and my Bachelor of Laws degree from Shandong Normal University.
My research interest is the Network Measurement, Testing and Management, especially Software-defined Networks and Data Center Network. These all serve the National Supercomputing Internet, which is a project that explore how to combine high-performance computing (supercomputing) with wide-area computer networks.

High-speed Network Measurement

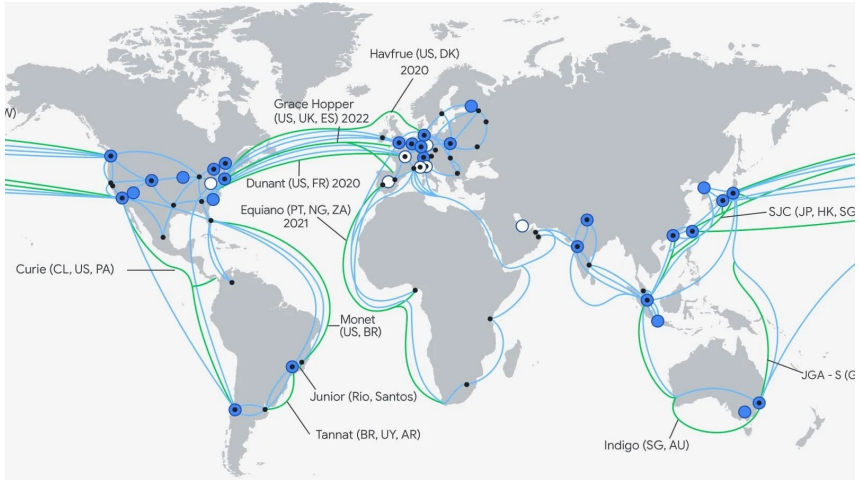
Proposed an SRv6 based active network telemetry architecture and implementation strategy (2nd Internet Architecture Academic Conference, China Information and Communications Conference, 2019). Integrated the segment routing label stack into custom probes by encoding it as IPv6 extension headers, and enabled flexible orchestration of telemetry instructions and paths with an improved breadth first search algorithm. This is among the earliest research efforts to combine SRv6 with INT in the field. Centered on two system goals, improving telemetry freshness and reducing telemetry intrusiveness, proposed a multi objective optimization method for selecting telemetry carrying flows (CN Patent 202010609191.7, 2020; APNOMS, 2021), and developed and open sourced an orchestration solver based on NSGA (Non dominated Sorting Genetic Algorithm). Proposed, early in the community, a hybrid in band telemetry task orchestration algorithm that fuses active and passive telemetry (APNOMS, 2022). Published a survey on in

1 About of SCSC (NSCCJN)

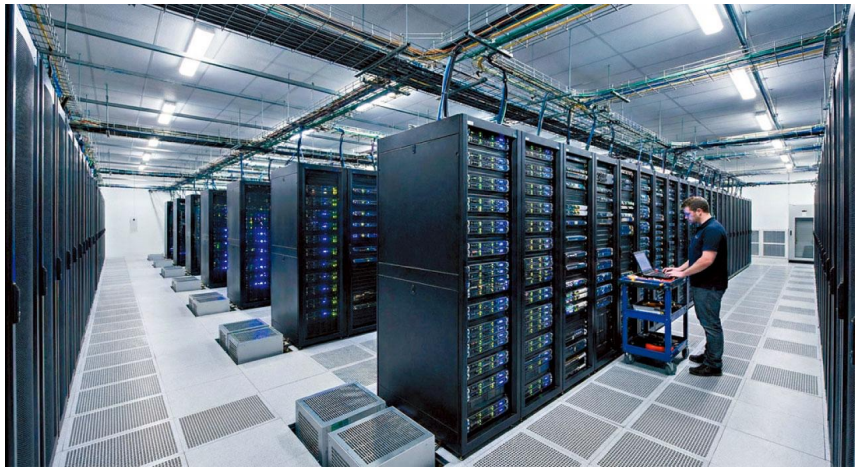
2 **RDMA/RoCEv2**

3 ByteTuning

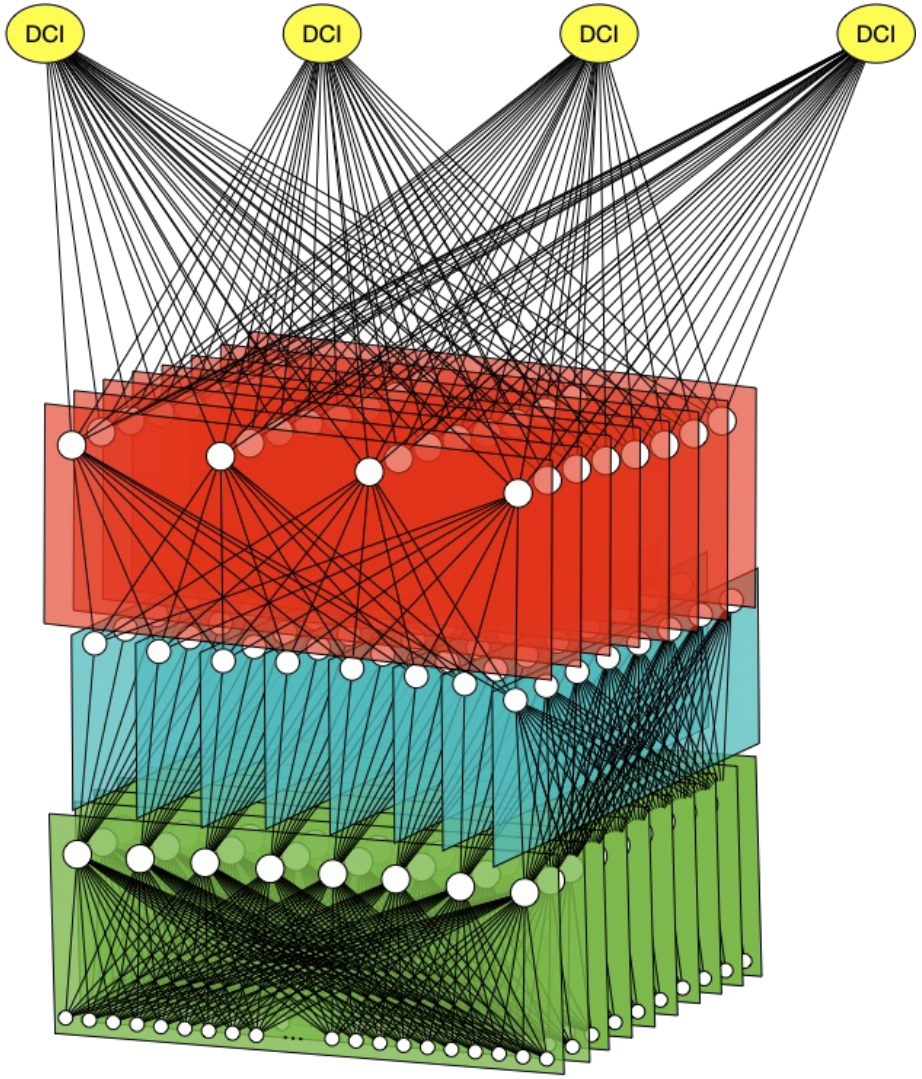
Data Center and Data Center Network



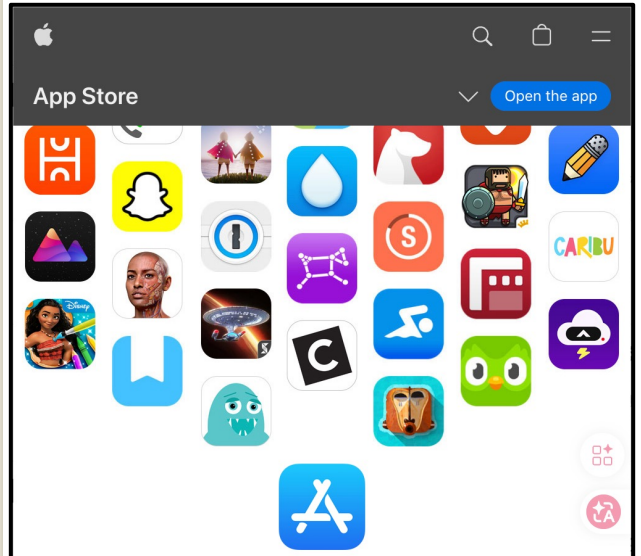
The Data Center of Google



The physical image of Data Center



The network topology of Data Center

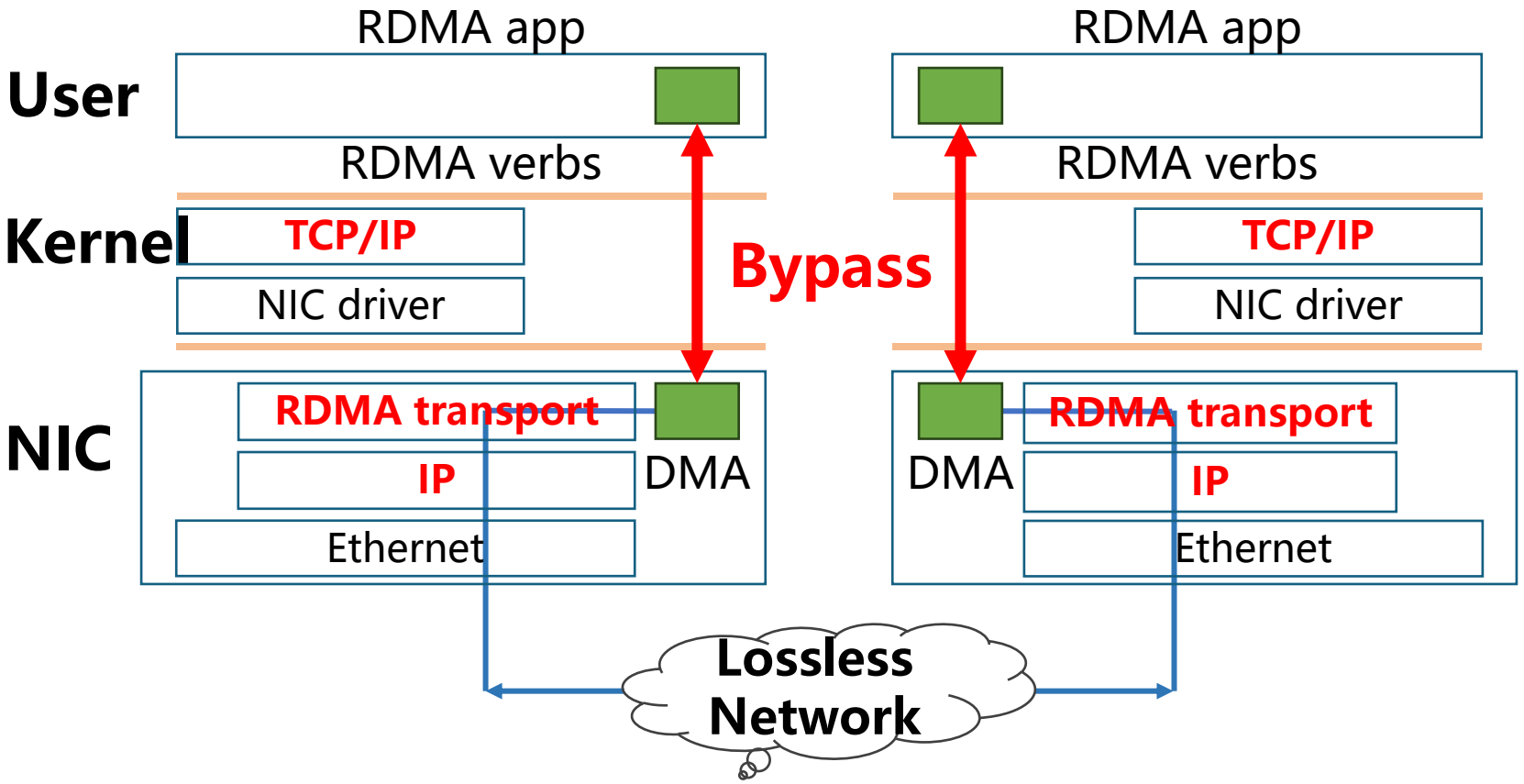


Almost all application services run in data centers.

TCP and RDMA

Traditional TCP Protocol
1-10Gbps, ms-level latency

RDMA Protocol
100-800Gbps, us-level latency

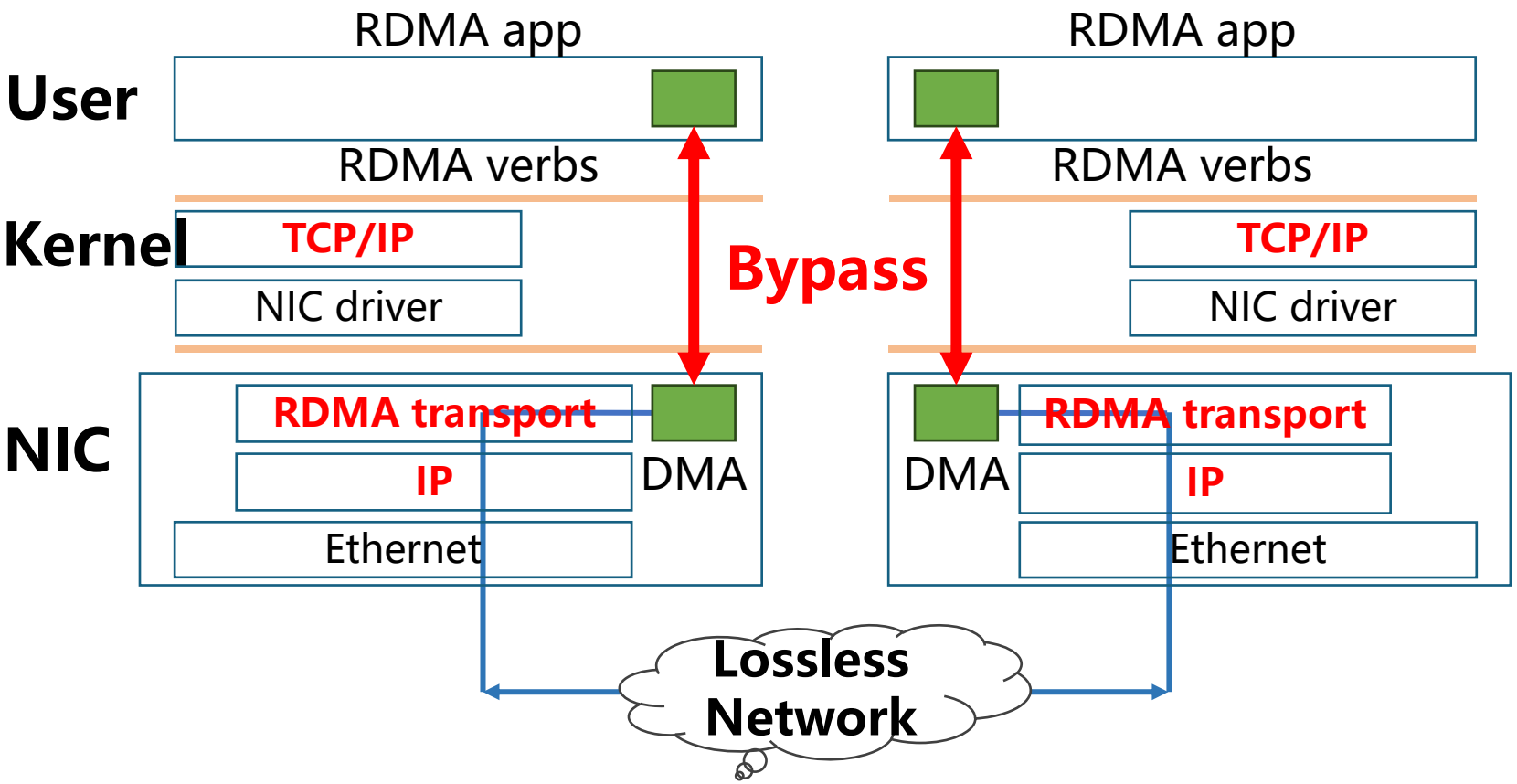


TCP and RDMA

Traditional TCP Protocol
1-10Gbps, ms-level latency

→

RDMA Protocol
100-800Gbps, us-level latency

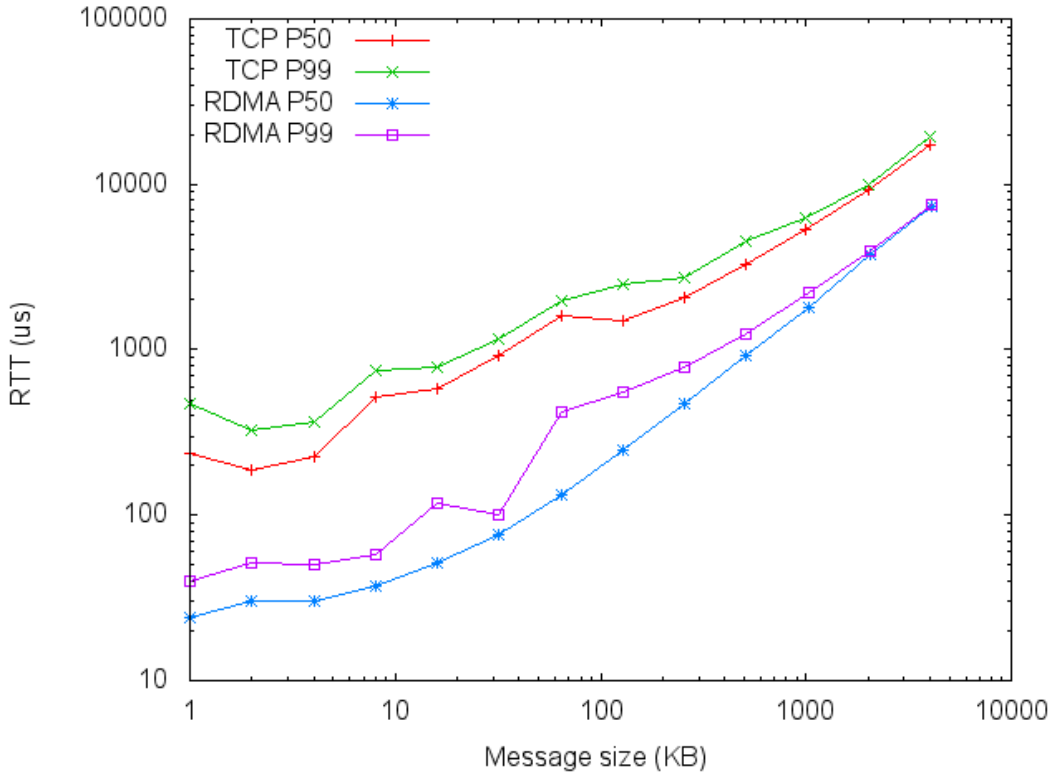


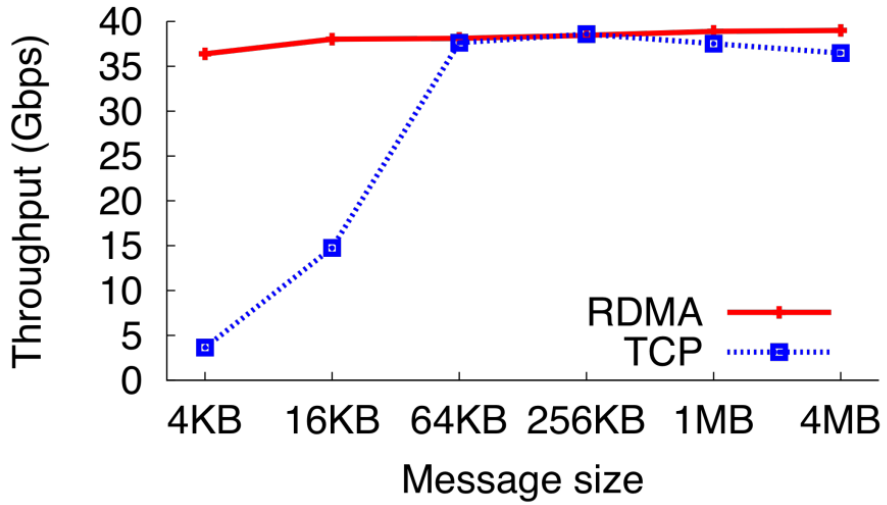
- **Remote**
  data transfers between nodes in a network
- **Direct**
  no Operating System Kernel involvement in transfers, everything about a transfer offloaded onto Interface Card
- **Memory**
  transfers between user space application virtual memory, no extra copying or buffering
- **Access**
  send, receive, read, write, atomic operations

TCP and RDMA

Traditional TCP Protocol
1-10Gbps, ms-level latency

RDMA Protocol
100-800Gbps, us-level latency





- For small msgs (<32KB), OS processing latency matters
- For large msgs (100KB+), speed matters

TCP and RDMA

Traditional TCP Protocol
1-10Gbps, ms-level latency

```
76 recv rate: 30012423744
77 recv rate: 29785573024
78 recv rate: 32537844160
79 recv rate: 34104756640
80 recv rate: 32433151744
81 recv rate: 47936439424
82 recv rate: 47009696672
83 recv rate: 45762704800
84 recv rate: 36870458528
85 recv rate: 44697521312
86 recv rate: 47689360512
87 recv rate: 52305616256
88 recv rate: 36145854304
89 recv rate: 43588678304
90 recv rate: 50147339616
91 recv rate: 38479512416
92 recv rate: 49721405056
93 recv rate: 50277126944
94 recv rate: 46907554080
95 recv rate: 45712658208
96 recv rate: 50521826912
97 recv rate: 47425490240
```

```
top - 06:36:10 up 5 days,  1:59,  0 users,  load average: 1.44, 0.80, 0.65
Tasks:   9 total,   1 running,   8 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.4 us,  4.5 sy,  0.0 ni, 92.6 id,  0.0 wa,  0.4 hi,  2.1 si,  0.0 st
KiB Mem:  52827267+total, 18587868 used, 50968480+free,   147664 buffers
KiB Swap: 67108860 total,        0 used, 67108860 free. 14185556 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1657 root      20   0 1212108  13460   2708 S  4.3  0.0   2:00.42 pserver
    1 root      20   0   18200   3276   2776 S  0.0  0.0   0:00.01 bash
  114 root      20   0   18192   3336   2840 S  0.0  0.0   0:00.00 bash
 1620 root      20   0   24244   2572   2320 S  0.0  0.0   0:00.00 tmux
 1622 root      20   0   24504   3080   2456 S  0.0  0.0   0:00.06 tmux
 1623 root      20   0   18228   3468   2940 S  0.0  0.0   0:00.00 bash
 1634 root      20   0   18228   3416   2888 S  0.0  0.0   0:00.00 bash
 1645 root      20   0   18228   3420   2892 S  0.0  0.0   0:00.00 bash
 1675 root      20   0   21956   2500   2176 R  0.0  0.0   0:00.01 top
```

TCP: Eight connections, 30-50Gb/s,
Client: 2.6%, Server: 4.3% CPU

RDMA Protocol
100-800Gbps, us-level latency

```
OnWrite rate = 88551 mbps
OnWrite rate = 87825 mbps
OnWrite rate = 88364 mbps
OnWrite rate = 87896 mbps
OnWrite rate = 87437 mbps
OnWrite rate = 87527 mbps
OnWrite rate = 86992 mbps
OnWrite rate = 87257 mbps
OnWrite rate = 87884 mbps
OnWrite rate = 87851 mbps
OnWrite rate = 88063 mbps
OnWrite rate = 87444 mbps
OnWrite rate = 88320 mbps
OnWrite rate = 87506 mbps
OnWrite rate = 87827 mbps
```

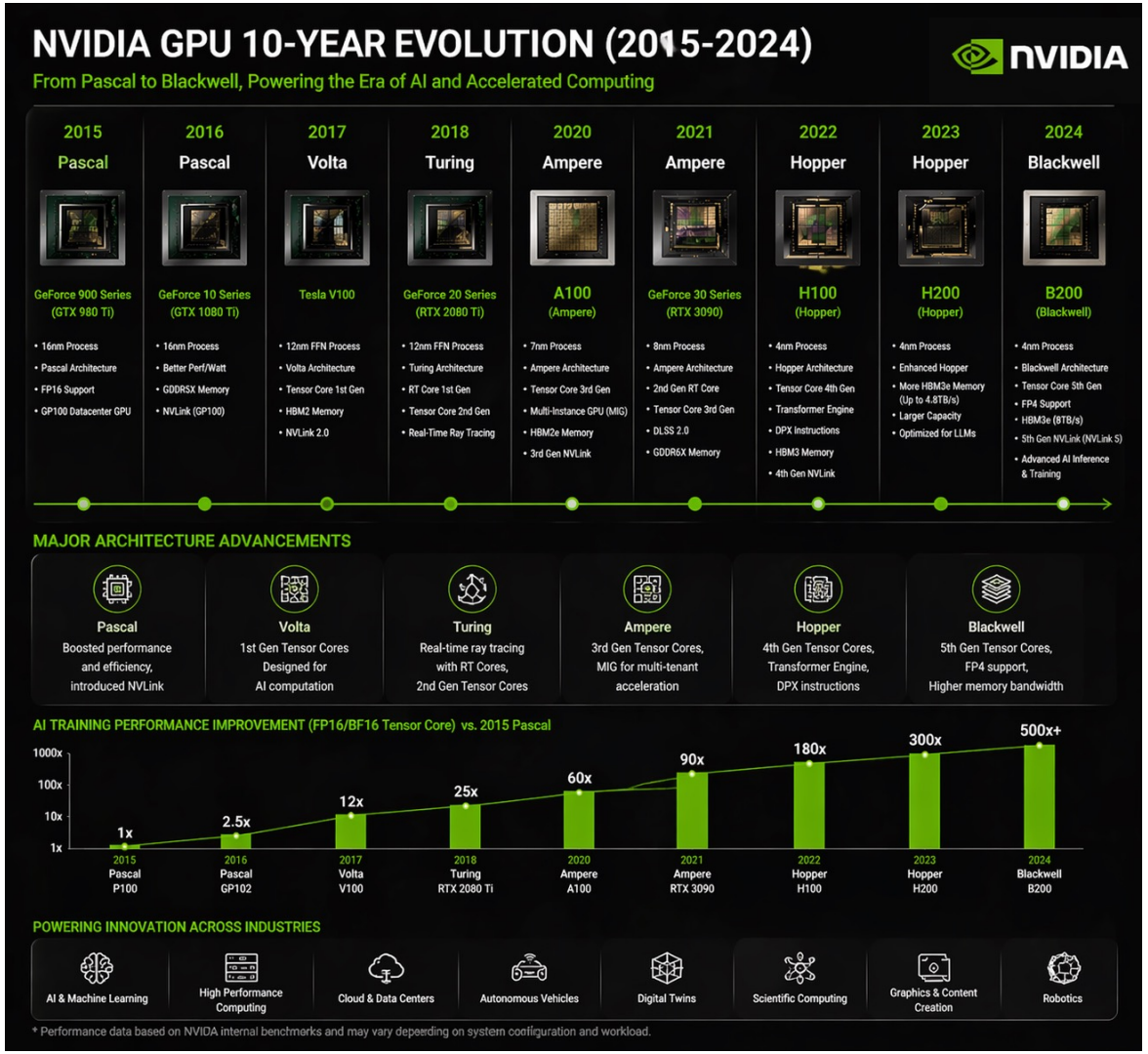
```
top - 18:16:47 up 21 days, 12:44,  0 users,  load average: 5.38, 3.46, 3.37
Tasks:   8 total,   1 running,   7 sleeping,   0 stopped,   0 zombie
%Cpu(s): 15.0 us,  3.0 sy,  0.0 ni, 81.1 id,  0.0 wa,  0.3 hi,  0.7 si,  0.0 st
KiB Mem:  52827267+total, 64229080 used, 46404358+free,   496340 buffers
KiB Swap: 67108860 total,     2940 used, 67105920 free. 50241336 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1888 root      20   0  178048   4504   3528 S  1.7  0.0   0:31.01 rfwork
    1 root      20   0   18200   3340   2836 S  0.0  0.0   0:00.02 bash
 1785 root      20   0   24244   2572   2320 S  0.0  0.0   0:00.00 tmux
 1787 root      20   0   24768   3532   2676 S  0.0  0.0   0:00.23 tmux
 1788 root      20   0   18228   3464   2936 S  0.0  0.0   0:00.00 bash
 1799 root      20   0   18228   3460   2932 S  0.0  0.0   0:00.00 bash
 1813 root      20   0   18228   3436   2908 S  0.0  0.0   0:00.00 bash
 1824 root      20   0   21956   2512   2180 R  0.0  0.0   0:00.09 top
```

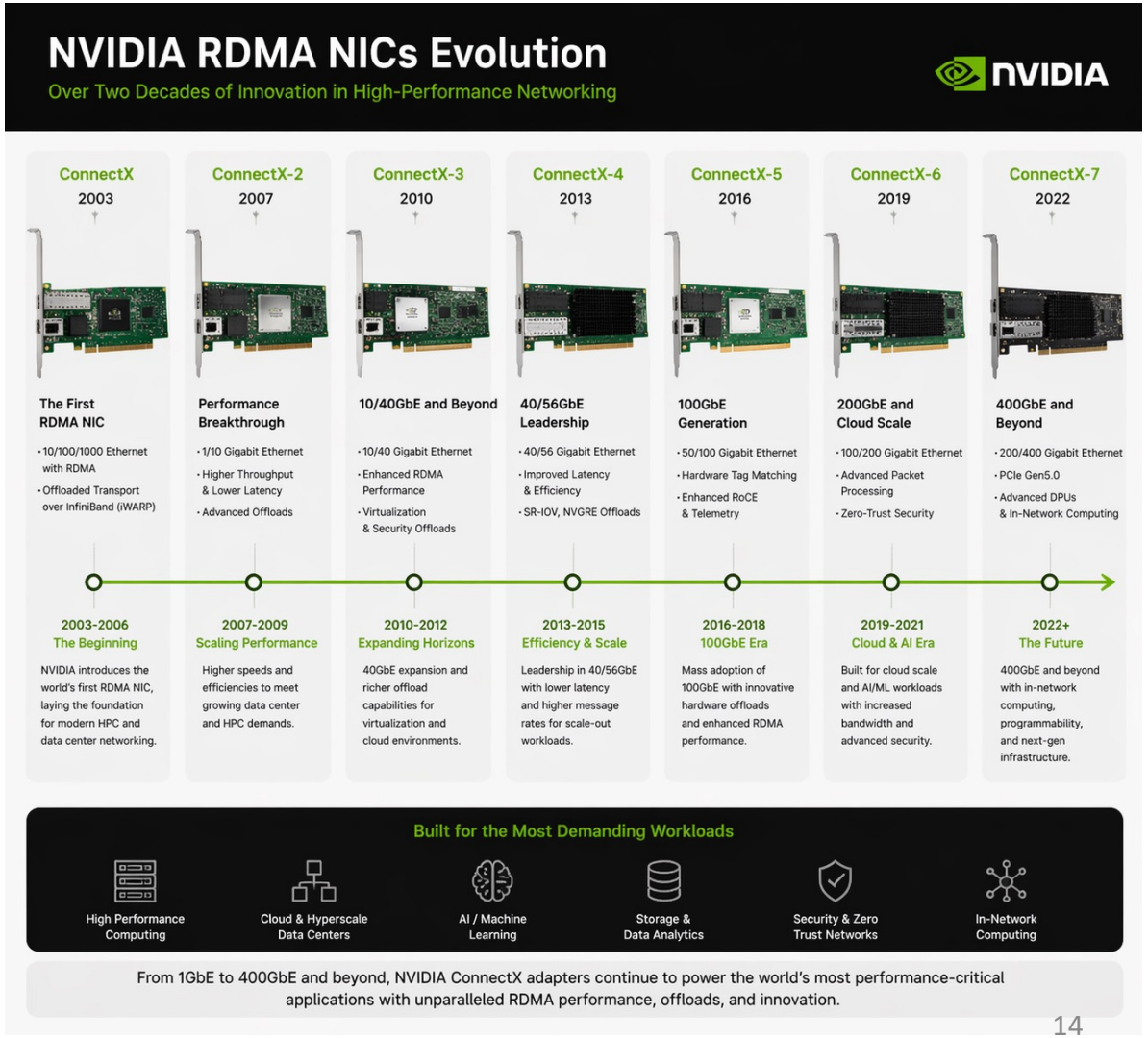
RDMA: Single QP, 88 Gb/s, 1.7% CPU

*Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz, two sockets 28 cores

RDMA NIC

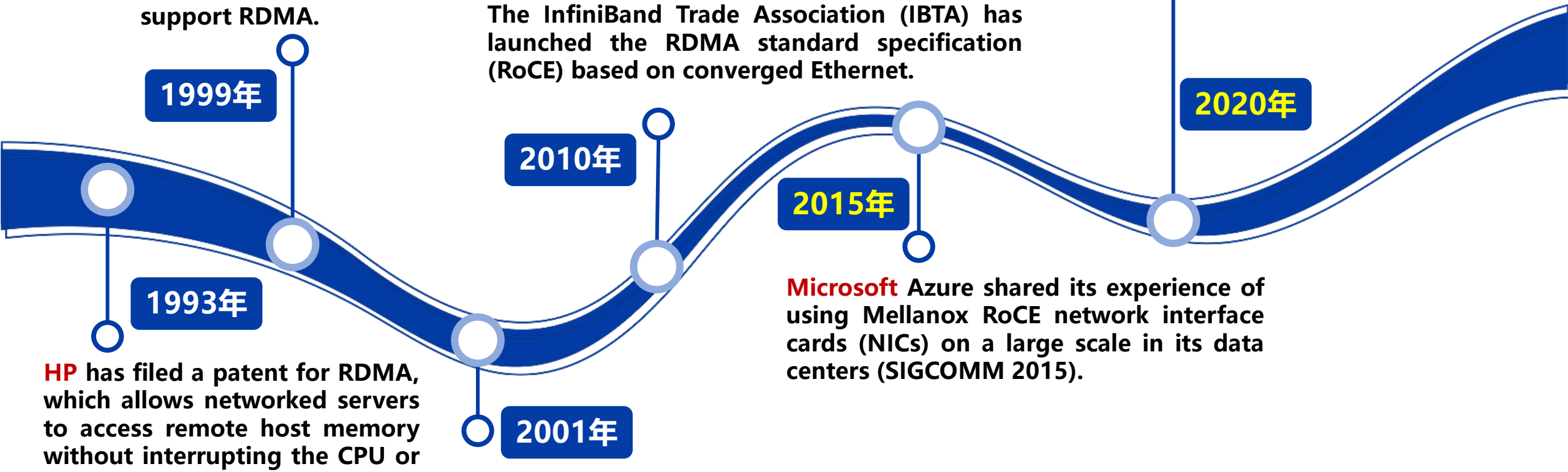
Nvidia is not only the inventor of GPUs, but also the leading manufacturer of high-performance network interface cards.





RDMA NIC

Nvidia is not only the inventor of GPUs, but also the leading manufacturer of high-performance network interface cards.



1993年

HP has filed a patent for RDMA, which allows networked servers to access remote host memory without interrupting the CPU or operating system, marking the birth of RDMA technology.

1999年

The Israeli company **Mellanox** was founded, and its main products are network cards that support RDMA.

2001年

Mellanox has released an RDMA NIC based on the Infiniband protocol standard.

2010年

The InfiniBand Trade Association (IBTA) has launched the RDMA standard specification (RoCE) based on converged Ethernet.

2015年

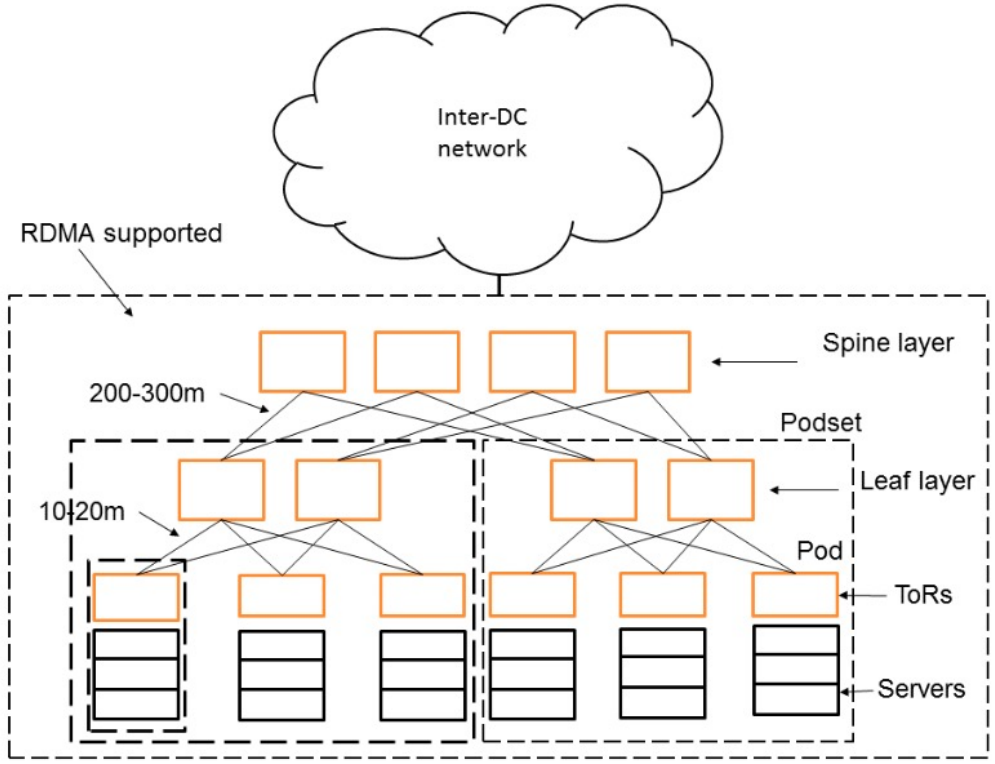
Microsoft Azure shared its experience of using Mellanox RoCE network interface cards (NICs) on a large scale in its data centers (SIGCOMM 2015).

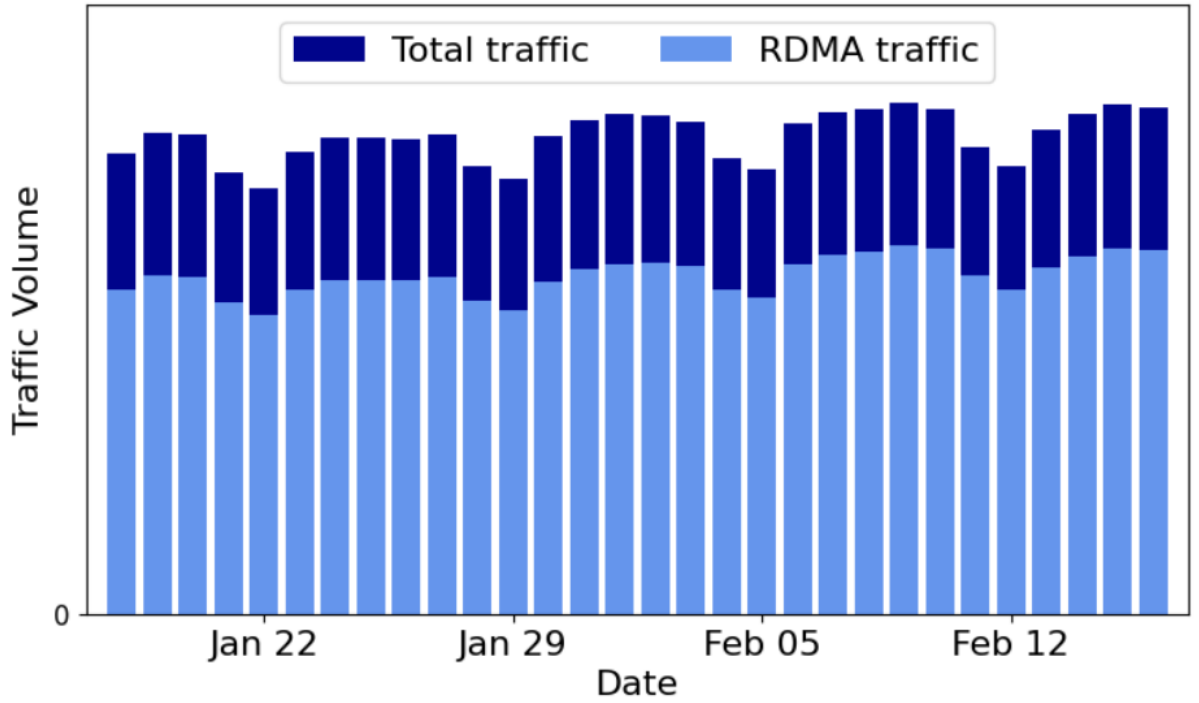
2020年

Nvidia (Computing) completes its acquisition of **Mellanox** (Networking) for $6.9 billion.

RDMA Application

- *Microsoft: The first major company to deploy RDMA on a large scale, and the designer of the RDMA congestion control algorithm DCQCN, which is the most widely used RDMA congestion control algorithm in the industry.*
- *To date, RDMA traffic accounts for over 70% of traffic in Microsoft Azure data centers.*

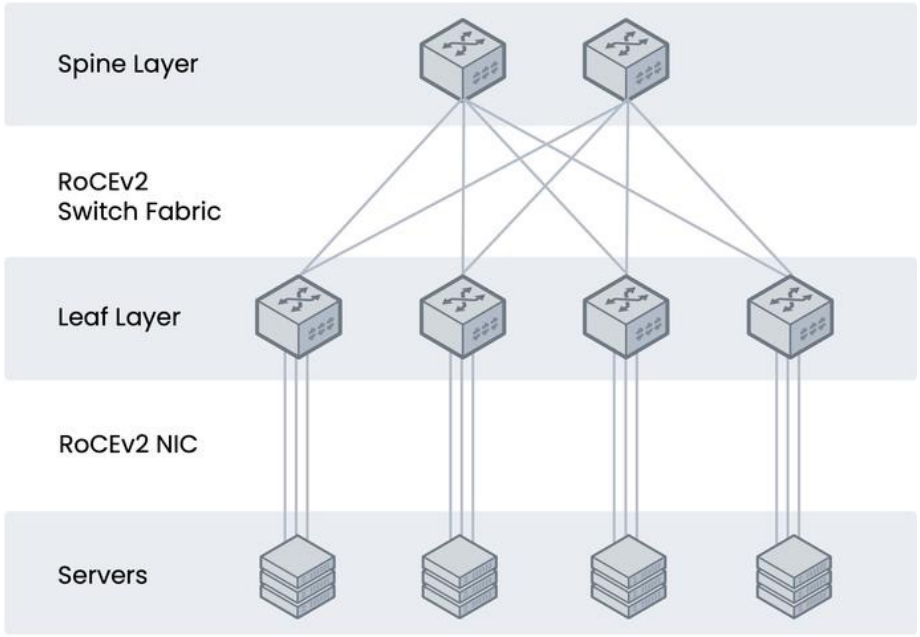




RDMA over Commodity Ethernet at Scale (SIGCOMM' 15) **Empowering Azure Storage with RDMA (NSDI' 23)**

RoCEv2 Network

RDMA over Converged Ethernet version 2 (RoCEv2) is an evolution of the RoCE protocol, allowing for the direct memory access between computers over Ethernet.



S6820-32H
32*100G

S9820-64H
64*100G

S9820-8C
128*100G or 32*400G

S9820-8C
128*100G or 32*400G

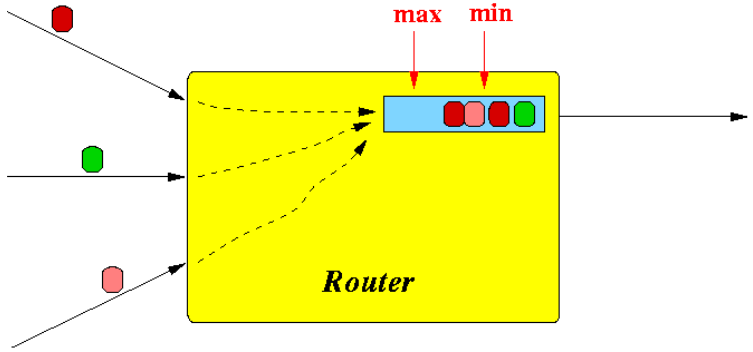
ConnectX-5 25GbE Dual-port
SFP28, PCIe Gen 3.0 x 8

ConnectX-6 100GbE Dual-port
QSFP56, PCIe 4.0 x16

ConnectX-7 400GbE Single-port
OSFP, PCIe 5.0 x16

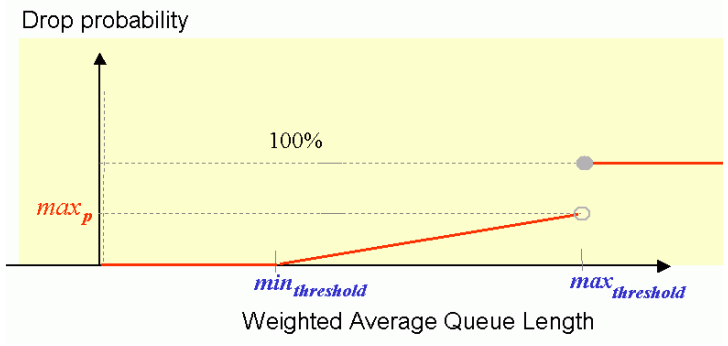
ECN

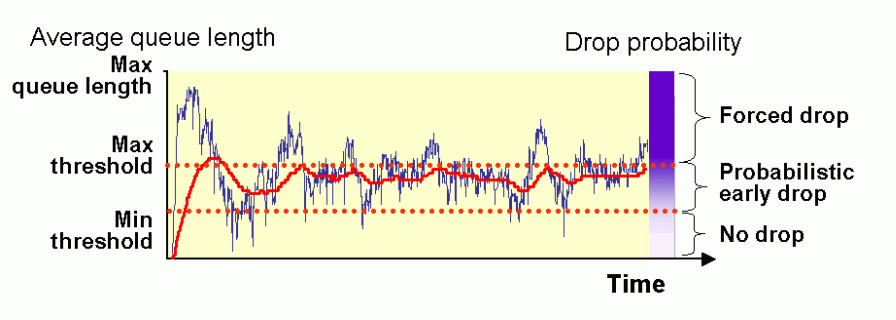
- ***Random Early Detection (RED) is a proactive congestion avoidance mechanism used in computer networks, primarily within routers, to manage and mitigate network congestion before it reaches critical levels.***
- *ECN reuses RED, but instead of proactively dropping packets, it only marks packets, avoiding the impact of packet loss on transmission.*



Random Early Detection

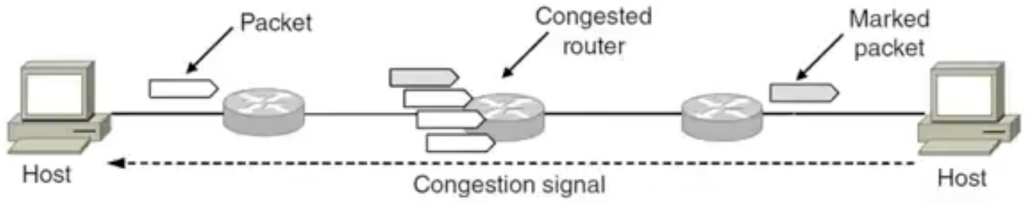
```
compute average queue length;

    if ( avg. queue length < min_threshold )
    {
       admit new packet to output queue;
    }
    else ( min_threshold < avg. queue length < max_threshold )
    {
       mark new packet with probability Prob(queue length);
    }
    else
    {
       always mark new packet;
    }
```





ECN

- *Random Early Detection (RED) is a proactive congestion avoidance mechanism used in computer networks, primarily within routers, to manage and mitigate network congestion before it reaches critical levels.*
- ***Explicit Congestion Notification (ECN) reuses RED, but instead of proactively dropping packets, it only marks packets, avoiding the impact of packet loss on transmission.***



ECN Bits (Code)	Meaning
00	Non-ECT—Packet is marked as not ECN-capable
01	ECT(1)—Endpoints of the transport protocol are ECN-capable
10	ECT(0)—Endpoints of the transport protocol are ECN-capable
11	CE—Congestion experienced

With ECN, routeres/switches deliver clear signal to hosts, and congestion is detected arly no loss.
Congestion Signal: Loss -> RTT -> ECN

- *ECN alone is not sufficient to achieve zero packet loss.*
- *Priority-based flow control (PFC) is Hop-by-hop flow control.*

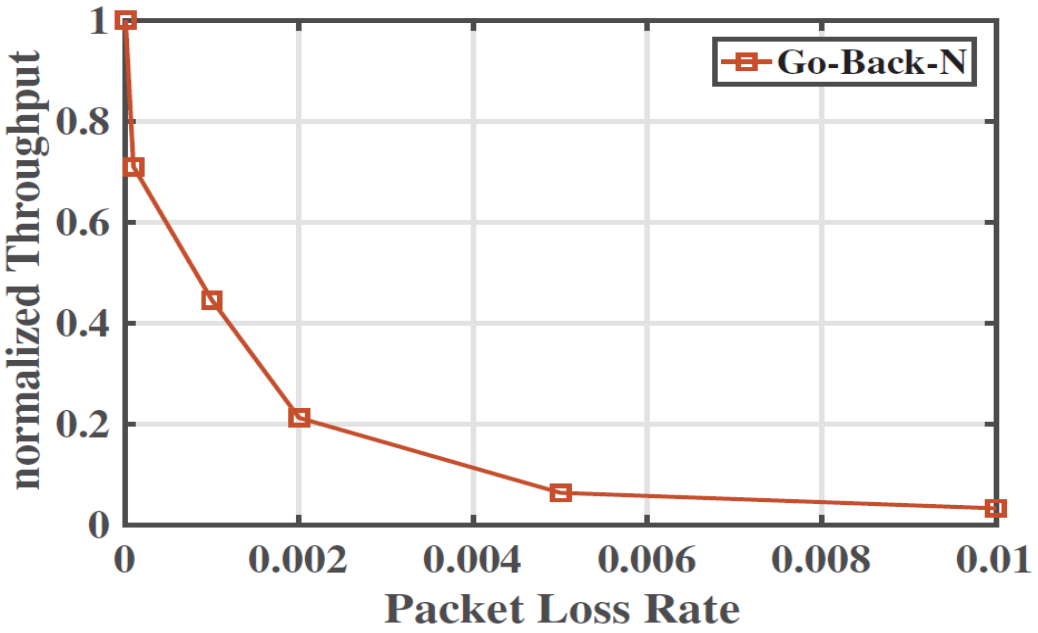
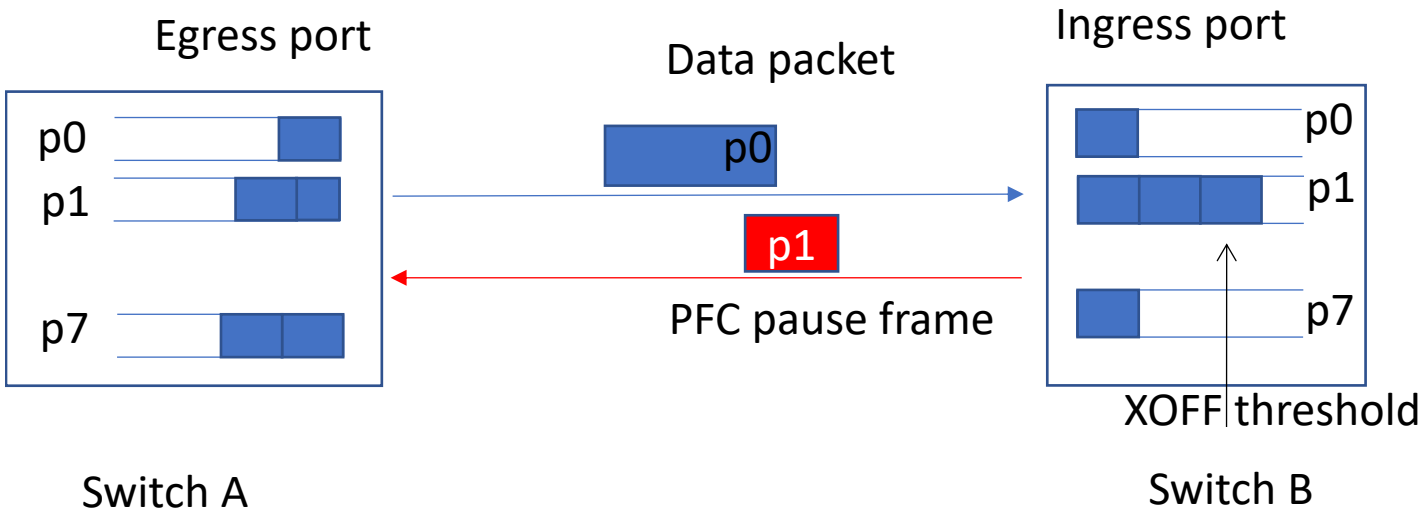
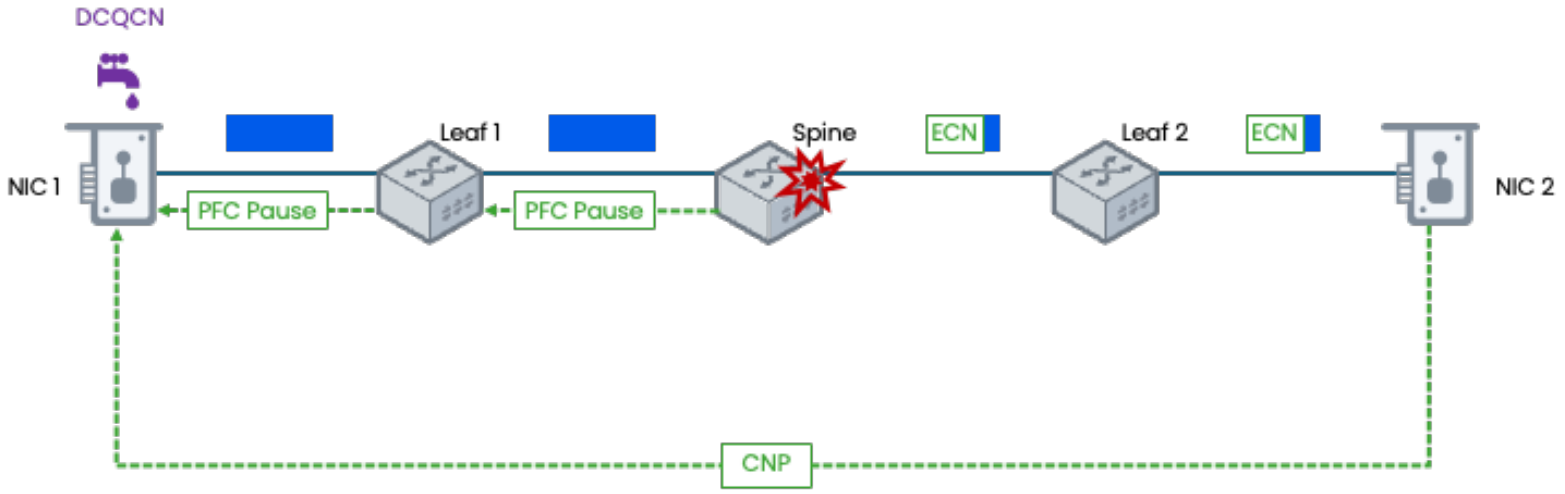


Fig. 2: Impact of packet loss rate on throughput



DCQCN

- *By combining ECN and PFC, Microsoft proposed DCQCN in 2015, which is currently the most widely used congestion control mechanism for RDMA networks.*



1 About of SCSC (NSCCJN)

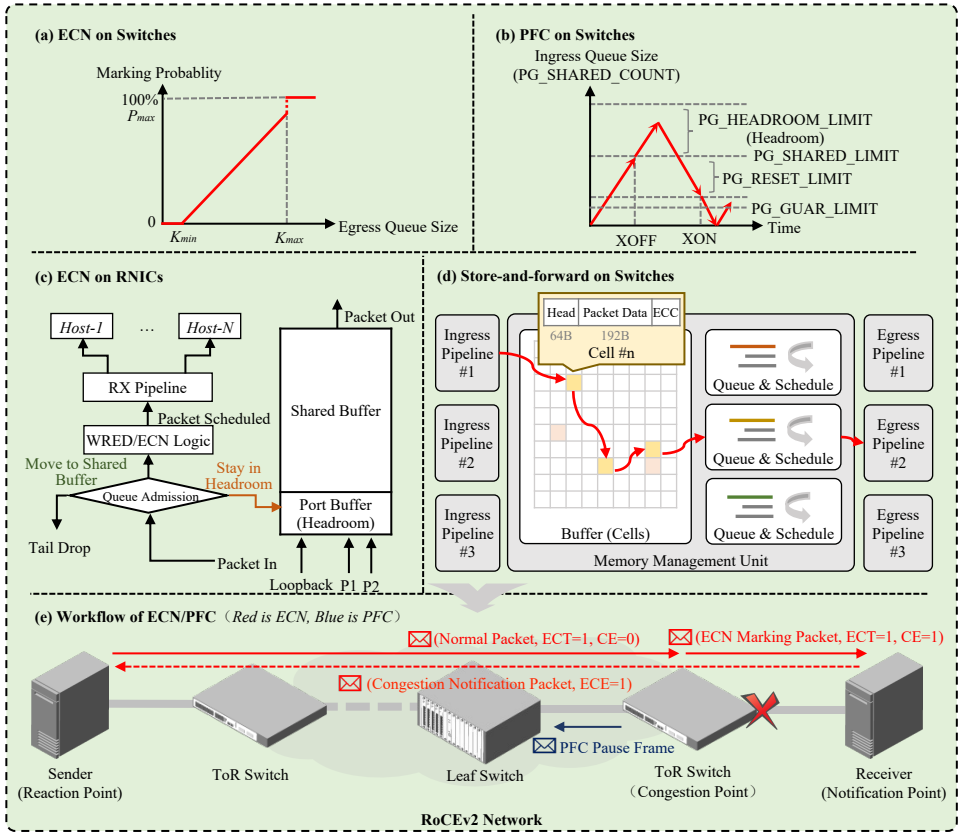
2 RDMA/RoCEv2

3 **ByteTuning**

Tan L, Jiang Z, Liu K, et al. ByteTuning: Watermark Tuning for RoCEv2[J]. IEEE Transactions on Cloud Computing, 2025, 13(1): 303-320.

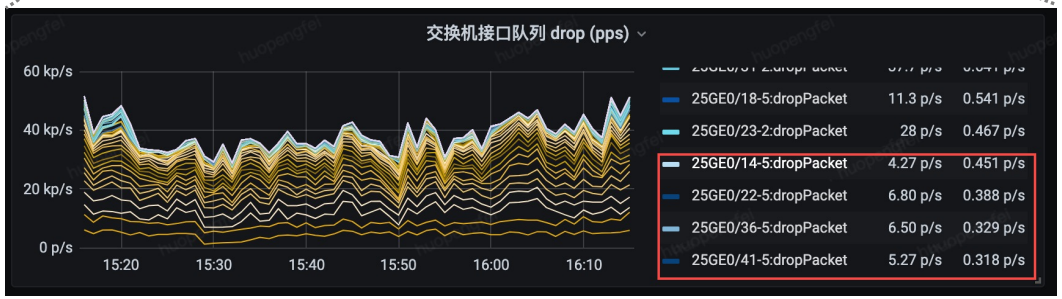
Background

- ***RDMA networks have become the most popular architecture for carrying 100G+ network in data center .***
- ***RoCEv2 is the most mainstream and cost-effective implementation of RDMA. Its key networking protocols include DCQCN on the server side, and ECN/PFC on the switch side.***



There are many ECN and PFC parameters in RoCEv2. How to optimally configure them is a key engineering challenge.

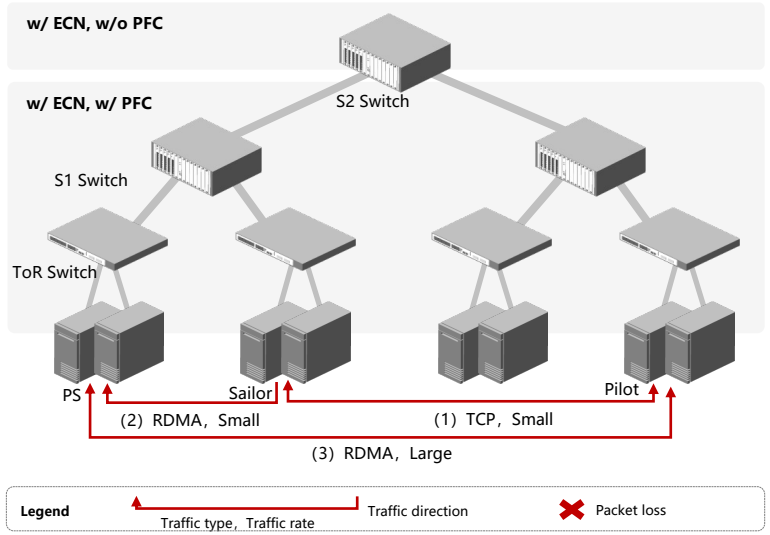
```
SXDTLQ_D3_201-18-02-51.S0-2-160#show queue-buffer interface tf0/41
Interface TFGigabitEthernet 0/41:
Slice 1:
Type       Queue  Used cells  Available cells  Usage  Usage warn limit  Usage warn count  Peaked cells  Service pool
Unicast    0      0           98319            0%     0%                0                 0             0(SP-0)
Unicast    1      2245        96074            2%     0%                0                 102196        0(SP-0)
Unicast    2      6           98313            0%     0%                0                 102182        0(SP-0)
Unicast    3      0           98319            0%     0%                0                 12985         0(SP-0)
Unicast    4      0           98319            0%     0%                0                 0             0(SP-0)
Unicast    5      1           98318            0%     0%                0                 37984         0(SP-0)
Unicast    6      0           98319            0%     0%                0                 0             0(SP-0)
Unicast    7      0           98319            0%     0%                0                 0             0(SP-0)
Multicast  0      0           12296            0%     0%                0                 0             0(SP-0)
Multicast  1      0           12296            0%     0%                0                 0             0(SP-0)
Multicast  2      0           12296            0%     0%                0                 2             0(SP-0)
Multicast  3      0           12296            0%     0%                0                 0             0(SP-0)
Multicast  4      0           12296            0%     0%                0                 0             0(SP-0)
Multicast  5      0           12296            0%     0%                0                 0             0(SP-0)
Multicast  6      0           12296            0%     0%                0                 0             0(SP-0)
Multicast  7      0           12296            0%     0%                0                 0             0(SP-0)
Service pool  Global shared cells  Usage  Usage warn limit  Usage warn count  Used cells  Peaked cells
0(SP-0)       110600               72%    0%                0                 80729       114578
1(SP-1)       0                    0%     0%                0                 0           0
2(SP-2)       0                    0%     0%                0                 0           0
3(SP-3)       0                    0%     0%                0                 0           0

Slot  Slice  PortGroup  Total cells  Total usage  Usage warn limit  Static used cells  Global shared cells  Available shared cells
0     1      1          131072       61%          0%                84                 110600              29871

SXDTLQ_D3_201-18-02-51.S0-2-160#
```



In many-to-one RDMA incast scenarios, switches can overwhelm shared buffers, causing output packet loss.

Traversal results

- ***Experimental Setup:*** *The testbed consists of 24 servers, two 25G access switches, and two 100G aggregation switches, equipped with Mellanox CX5 25G NICs. The congestion control algorithm is DCQCN, using Mellanox-recommended fixed parameter settings.*
- ***Methodology:*** *We select commonly used switch-side ECN/PFC configuration parameters, including K_{min}, K_{max}, P_{max}, Alpha, and Headroom, and exhaustively evaluate different parameter combinations to study their impact on throughput, queue occupancy, and flow completion time. The search space contains a total of 3,300 parameter combinations. Traffic is generated using Perftest (ib_write_bw).*

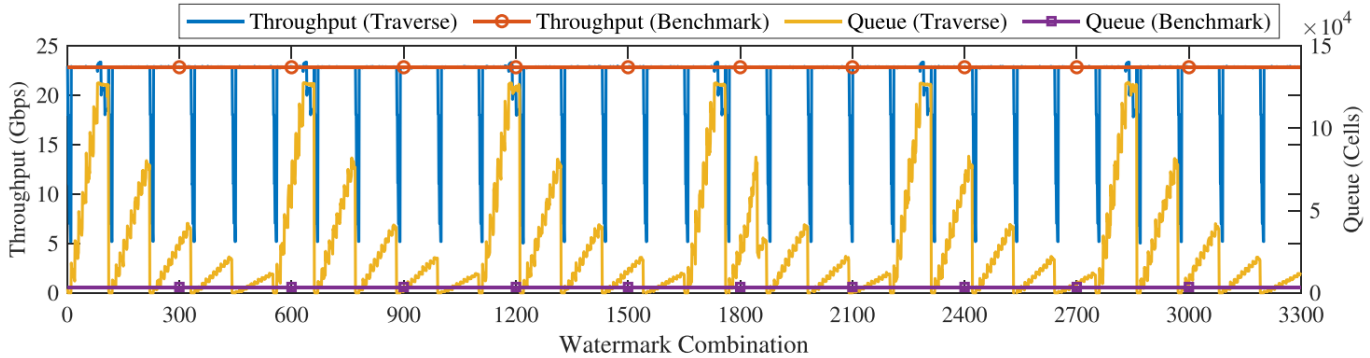


ECN/PFC threshold parameter search space

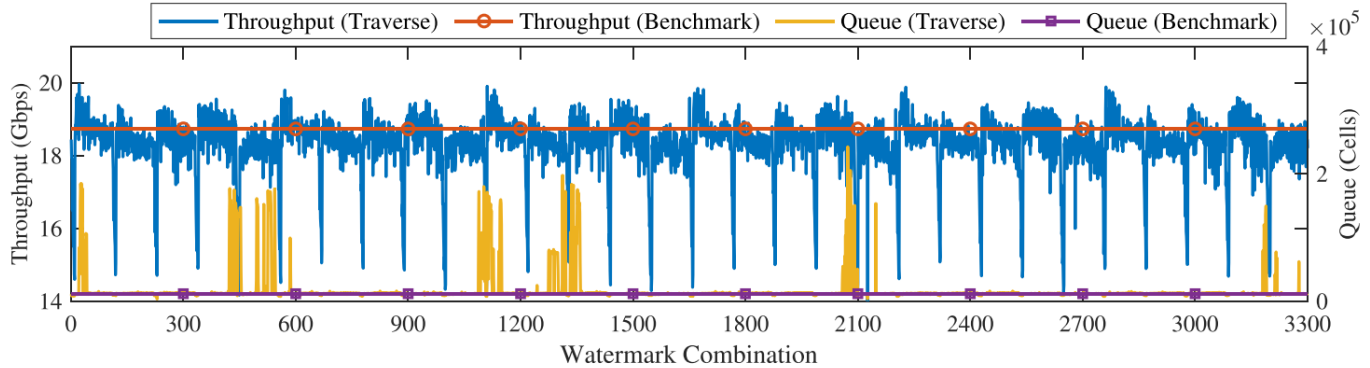
Parameter	Search Space
K_min	0, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000
K_max	K_min+500
P_max	1, 10, 20, 30, 40, 50, 60, 70, 80, 90
$Alpha$	1, 3, 5, 11, 20
$Heamroom$	100, 200, 300, 400, 500, 600

Traversal results

- ***Experimental Setup:*** *The testbed consists of 24 servers, two 25G access switches, and two 100G aggregation switches, equipped with Mellanox CX5 25G NICs. The congestion control algorithm is DCQCN, using Mellanox-recommended fixed parameter settings.*
- ***Methodology:*** *We select commonly used switch-side ECN/PFC configuration parameters, including K_{min}, K_{max}, P_{max}, Alpha, and Headroom, and exhaustively evaluate different parameter combinations to study their impact on throughput, queue occupancy, and flow completion time. The search space contains a total of 3,300 parameter combinations. Traffic is generated using Perftest (ib_write_bw).*



(a) Throughput and Queue when 2-to-1



(b) Throughput and Queue when 24-to-1 RDMA Incast

Experience parameters are not optimal.

Traversal results

- ***Experimental Setup:*** *The testbed consists of 24 servers, two 25G access switches, and two 100G aggregation switches, equipped with Mellanox CX5 25G NICs. The congestion control algorithm is DCQCN, using Mellanox-recommended fixed parameter settings.*
- ***Methodology:*** *We select commonly used switch-side ECN/PFC configuration parameters, including K_{min}, K_{max}, P_{max}, Alpha, and Headroom, and exhaustively evaluate different parameter combinations to study their impact on throughput, queue occupancy, and flow completion time. The search space contains a total of 3,300 parameter combinations. Traffic is generated using Perftest (ib_write_bw).*

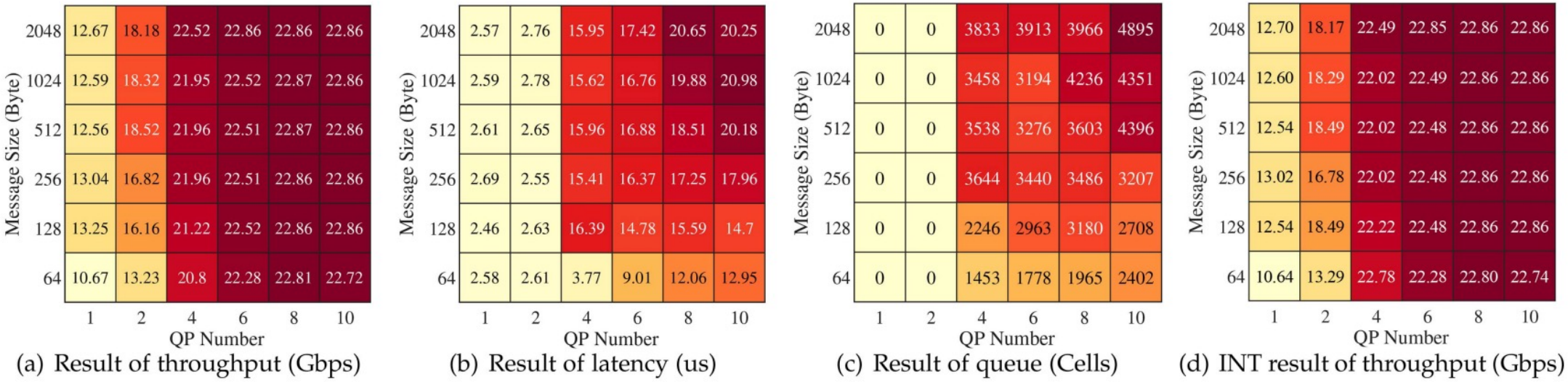
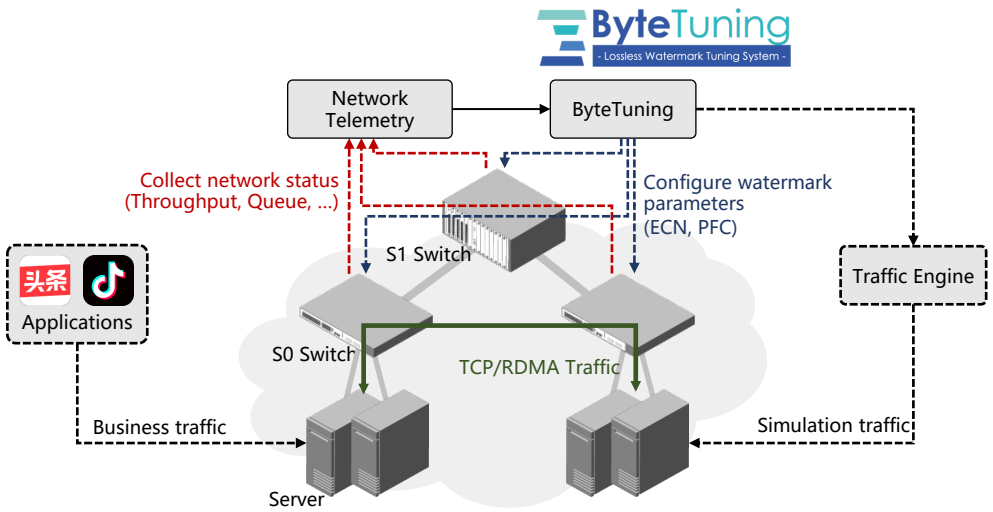


Fig. 10. Measurement results of throughput, latency, and queue when 2-to-1 with different traffic patterns.

We need to tune the ECN/PFC paremeters.

- *We designed a watermark tuning approach based on simulated annealing: **ByteTuning**.*



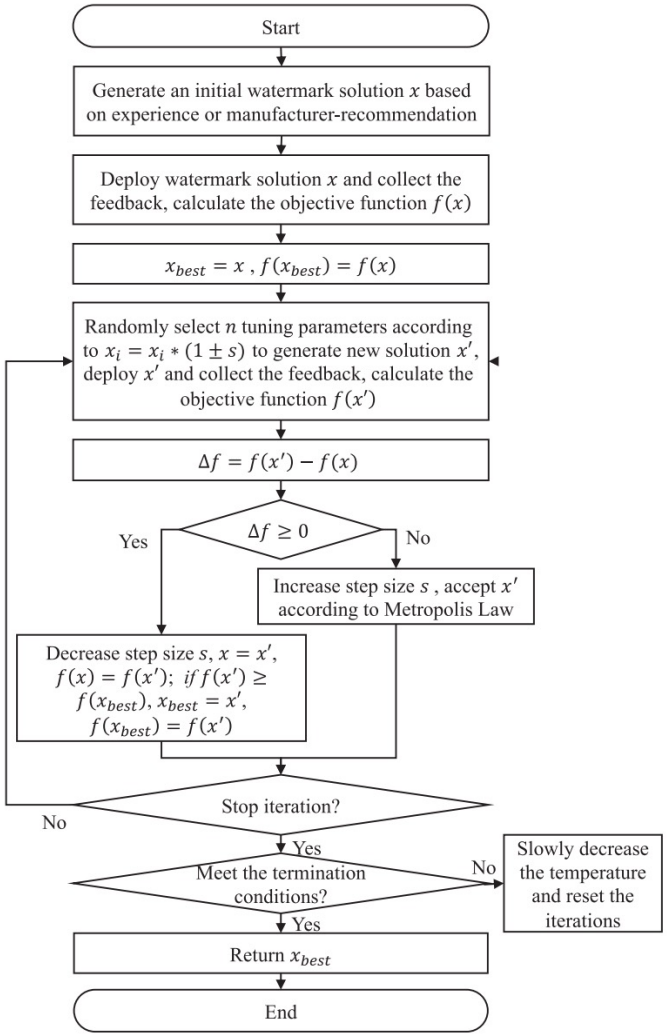
Optimization Objectives

Throughput-sensitive

$$f = \text{maximize} \sum_{i=1}^{n} [\beta \frac{Throughput_{now}(switch_i)}{Throughput_{max}(switch_i)} + (1-\beta) \frac{Queue_{min}(switch_i)}{Queue_{now}(switch_i)}]$$

Latency-sensitive

$$f = \text{maximize}\{\frac{\sum_{i=1}^{n} [\beta \frac{Throughput_{now}(flow_i)}{Throughput_{max}(flow_i)} + (1-\beta) \frac{Queue_{min}(flow_i)}{Queue_{now}(flow_i)}]}{\gamma \sum_{i=1}^{n} [Throughput_{now}(flow_i) - \overline{Throughput}_{now}(flow_i)]^2}\}$$



Experimental Result

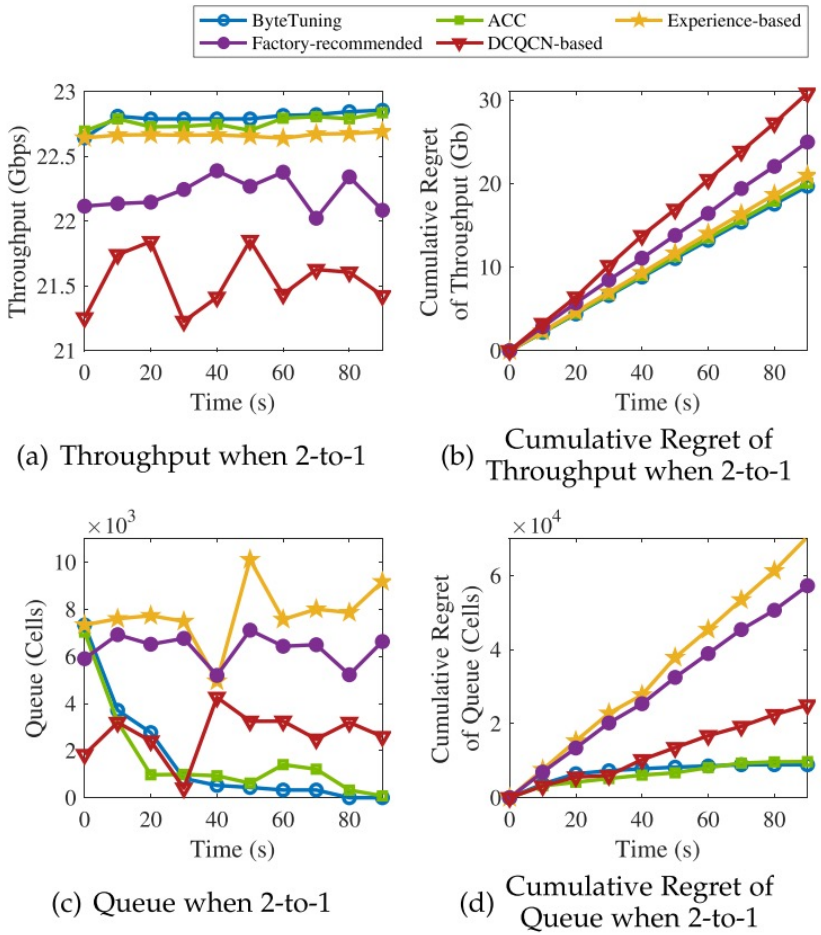


Fig. 11. The tuning result when 2-to-1.

TABLE V
THE TUNING RESULT OF REDIS STORAGE

	Redis over DCTCP	Redis over RDMA (Experience-based)	Redis over RDMA (Manufacturer-recommended)	Redis over RDMA (ByteTuning)
Throughput (Gbps)/IOPS (K)	14.41 / 25.51	20.73 / 39.11	21.41 / 41.03	**22.59 / 43.20**
FCT (ms)	15.60	11.23	9.86	**9.16**
Queue (Cells)	1.16 M	8.23 K	4.81 K	**2.53 K**
ECN-marked packets	**3.79 K**	12.80 K	6.72 K	4.97 K

1 About of SCSC (NSCCJN)

2 RDMA/RoCEv2

3 ByteTuning

Thanks for listening.