Research on transmission reliability of RoCEv2 protocol in WAN

Wenyi Wang, *Student, NJUPT*

***Abstract*—Modern data center networks utilize the RDMA (Remote Memory Access) protocol to efficiently reduce end-side host CPU utilization and achieve their high throughput and low latency requirements for communication. In recent years, the RoCEv2 protocol, based on RDMA technology, has been widely deployed and used in large data center LANs, with an efficient congestion control protocol at its core. However, RoCEv2 cannot be transmitted in WANs across data centers, mainly because WANs cannot meet the requirements of RDMA lossless transmission, resulting in a dramatic degradation of the transmission performance of the RoCEv2 protocol in WANs. This paper first starts with the mechanism of the RoCEv2 protocol and the characteristics of WAN transmission, studies and analyzes the main congestion control techniques in data center networks and WANs, and investigates the performance, advantages, and disadvantages of various types of packet loss handling mechanisms of the RoCEv2 protocol under different conditions of network delay, packet loss rate, delay jitter, etc. The main work of this paper is to propose a Go-Back-ONE packet loss recovery algorithm based on the original RDMA packet loss retransmission algorithm: when a packet loss is detected, the receiver generates a packet loss notification signal RACK and uses a bitmap to cache the disordered packets, and the sender retransmits the dropped packets with a high priority after receiving the RACK. Small-scale experiments are conducted on the NS3 network simulation platform to simulate network transmission across the WAN, demonstrating that the Go-Back-ONE algorithm can effectively reduce flow completion time and improve link bandwidth utilization.**

***Index Terms*—WAN;RDMA;PFC;Congestion Control;Packet loss Retransmission**

I. Introduction

IN recent years, with the continuous advancement of modern communication technology, computer networks have become an indispensable component of human society. As a crucial element of computer networks, wide area networks (WANs) are able to connect local area networks (LANs) distributed across different geographical locations, facilitating communication and resource sharing between remote users and local hosts. This provides robust support for data exchange and sharing in various fields. However, the traditional TCP/IP protocol is unable to address issues such as high latency, low bandwidth, and instability in WANs, resulting in significant performance degradation for existing applications, particularly those sensitive to latency.

Since the traditional TCP/IP network transport protocol's packet loss retransmission mechanism and congestion control algorithm can not solve a series of problems in the existing network environment, a high-performance network transport technology RDMA (Remote direct memory access) is designed to improve the efficiency of network transmission and reduce the overhead of the end host. RDMA over Converged Ethernet version 2 (RoCEv2) also uses UDP and IP as the network layer and encapsulates packets into Ethernet packets for receiving and sending. Thanks to its compatibility with existing Ethernet and network devices that use Ethernet, RoCEv2 has been deployed on a large scale in data center level Lans, and its performance in this scenario has been widely verified. However, the deployment of RoCEv2 lies at the heart of the need to provide strict lossless transmission in the network environment, which makes its application in the wide area network need to be further explored.

Packet loss in a wide area network can have a significant impact on an application. (1) For some delay-sensitive applications, such as video live broadcasting and high-frequency trading, packet loss will lead to picture stalling and economic losses caused by transaction delay. (2) Once packet loss occurs, the existing transport protocol will trigger the packet loss retransmission mechanism. For example, TCP responds to packet loss by returning to slow start (RTO detection), stopping window growth (during fast retransmission of unparsed segments), or reducing congestion Windows (fast recovery), which causes a sharp deterioration of the 99

II. Background

A. RDMA Protocol

Through the investigation of the world's largest cloud providers - Ali, Amazon and Microsoft, two key requirements are identified in the development of high-performance transmission networks represented by data center networks today: ultra-low latency and high bandwidth. In addition, recent studies[7] also point out the following two trends in the development of high-speed networks:

(1) Resource decomposition and heterogeneous computing of the data center network: In resource

Junchang Wang was with the professor of Computer science, CN, e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

decomposition, the CPU needs to network with remote resources such as GPU, memory and disk at high speed, requiring 3-5 microseconds of network latency and 40-100Gbps of network bandwidth to maintain good application-level performance. In a heterogeneous computing environment, different computing chips, such as cpus, FPgas, and Gpus, also require high-speed interconnections, and the lower the latency, the better.

(2) New applications: such as storage on high I/O speed media, such as fast non-volatile memory and large-scale machine learning training on high-speed computing devices, such as Gpus and ASics. These applications need to transfer large amounts of data periodically because their storage and computation speeds are very fast, resulting in performance bottlenecks in the network. Given that the traditional software-based network stack in the host is no longer able to meet key latency and bandwidth metrics, offloading the network stack to hardware is an inevitable direction for high-speed networks.

Therefore, RDMA technology is gradually studied and plays an irreplaceable role in the development process of high-speed network. Remote Direct Memory Access (RDMA) is a high-performance, low-latency network communication technology in which cpus at both ends rarely participate in the data transmission process. The local NIC copies data from the memory to the internal storage space, assembles packets of each layer through the hardware, and sends the packets to the peer NIC over the physical link. After receiving data, the peer RDMA network card strips each layer header and parity code, and copies the data directly to the memory. Figure 1 more intuitively introduces the working principle of RDMA.

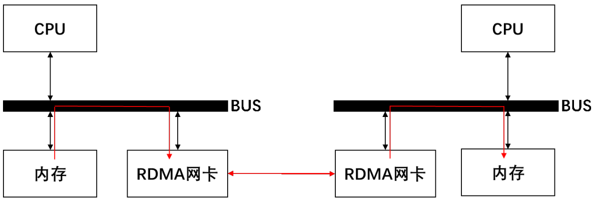

Fig. 1: How RDMA works

Specifically, RDMA technology has three characteristics: Feature 1: Kernel bypass. When data transmission is needed, the sender writes the data into the RDMA buffer, and then sends the data directly to the RDMA buffer of the receiver through the network card, and the receiver reads the data from the RDMA buffer. This process does not require the intervention of the operating system or complex protocol processing, so it can achieve very low latency and high bandwidth.

Feature two: RDMA technology adopts zero-copy technology, that is, no additional data copy operation is needed in the process of data transmission. In traditional network communication, data needs to be copied from the user process to the kernel buffer, and then from the kernel buffer to the network card buffer before being sent out. RDMA technology can avoid these data copy operations by directly accessing physical memory and network card buffer, thus reducing the delay and consumption in the process of data transmission and further improving the communication performance.

Feature three: CPU offloading. RDMA technology can read and write the memory of the remote node without the CPU participating in the communication (but it needs to hold the key to access the remote memory), which makes the CPU computing resources that are occupied can be used for more meaningful work. Figure 2 shows the processing difference between RDMA programming and traditional TCP/IP socket programming.

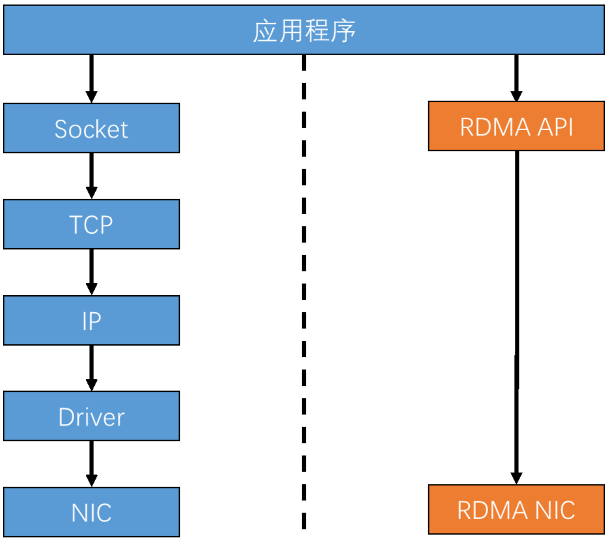

Fig. 2: Socket and RDMA Process

Because of the above characteristics, compared with the traditional Ethernet, RDMA technology achieves higher bandwidth and lower delay at the same time, which is more suitable for the development of high-performance network in today's society.

B. WAN Protocol

A wide area network (WAN) is a network that connects computers and devices in different geographic locations. Due to the wide coverage area of the WAN, its transmission speed and stability are crucial. At present, WAN provides relatively reliable data transmission and data security through the following technologies:

(1) IP network: As one of the most commonly used technologies in the wide area network, IP network is based on the TCP/IP protocol, using the physical equipment and transmission media of the Internet to provide data communication functions. IP networks can enable data transmission across geographical areas, but the disadvantage is that they have limitations in terms of data transmission speed, security and latency.

(2) MPLS network: MPLS is a label-based data transmission technology, which can achieve high-speed data transmission and optimized routing, thereby improving the data transmission speed and quality of the WAN. However, MPLS networks require dedicated hardware and software support, and the cost of deployment is relatively high.

(3) VPN: VPN is a virtual private network technology that can use the public network to establish a secure data channel, so as to achieve remote access and data transmission. VPN technology can ensure the security of data transmission through encryption and authentication, but its disadvantage is that there are certain limitations in data transmission speed and delay.

(4) Optical fiber network: Optical fiber network is a high-speed data transmission technology, which can transmit data through optical fiber to achieve high-speed, stable and secure data transmission. Fiber optic network has great advantages in terms of transmission speed and quality, but it is expensive and requires professional equipment and technical support.

(5) SD-WAN: SD-WAN is a software-defined wide area network (WAN) technology that optimizes the quality and speed of data transmission over the WAN through technologies such as network virtualization and flow control. SD-WAN technology can reduce costs, improve efficiency, and also improve the security and reliability of the wide area network.

III. Related Works

A. DSCP-Based PFC

In order to meet the RoCEv2 protocol deployment to achieve a reliable and efficient high-performance network, previous work has shown that RoCEv2 requires PFC to implement a lossless Ethernet architecture. PFC To prevent overflow of the buffer of Ethernet switches and nics, you can trace the information about incoming queues on the switch or nics. The core working principle of PFC is as follows: When the queue exceeds a certain threshold, the upstream device sends a PAUSE message, and the upstream device immediately stops sending data until it receives the RESUME message.

The following describes the working principle of the DSCP-based PFC mechanism [3] according to Figure 3. First, the Ethernet switch sets one or more priorities for each port. The DSCP field in the packet header is used to classify packets. Generally, eight priorities are set. When the buffer of the switch is congested, a traffic control frame marked with PAUSE is sent to the sending port to pause the transmission of low-priority data streams on the port. At this point, the high-priority data stream can continue to be transmitted. Depending on the DSCP field, the switch can assign packets to different queues, each with an independent PAUSE control. Each PFC queue can be mapped to a threshold Xoff. When the number of packets in a queue exceeds the threshold, the switch sends PAUSE control frames to the sending port to pause the transmission of data streams in the queue. After investigation, the existing switches and ASIC nics are flexible enough and provide the configuration of relevant functions, which greatly supports the deployment of PFC in the commercial environment, and provides a realistic basis for the application of RDMA in the WAN in the future.

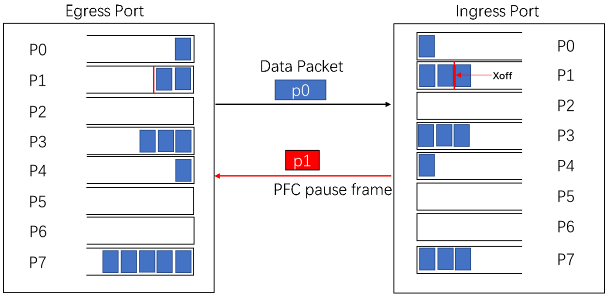

Fig. 3: DSCP-based PFC process

B. ECN-Based CC

Explicit congestion notification (ECN), by modifying the service type field of the network datagram header, divides two bits as special ECN flag bits, which are used to carry congestion status information to realize efficient congestion control algorithm. In the traditional congestion control scheme, once the congestion causes the buffer overflow of the receiver, the solution of packet loss retransmission can only be adopted, which will lead to greater transmission delay and worse network performance. However, the emergence of the ECN flag allows network devices to signal other devices when the network is congested so that they can slow down the data transmission speed, thereby avoiding packet loss caused by buffer overflow to some extent.

Most of the existing commercial switches provide the ECN marking function, so the congestion control algorithm based on ECN marking represented by DCTCP [1] and DCQCN [12] has been widely used in data center networks. The principle of congestion control algorithm based on ECN is as follows: (1) Firstly, set an ECN marker threshold and confirm that the packet sent by the sender supports ECN marker; (2) When the queue usage of the switch or other network devices is greater than this threshold (that is, congestion occurs), the CE bit in the ECN mark of subsequent packets is modified; (3) When receiving an IP packet with CE code point, the receiver will modify the corresponding ECE bit in the returned ACK message. In some schemes, there are many forms of this congestion notification, such as DCQCN to use the form of CNP message for congestion feedback; (4) After receiving the ACK, the sender takes corresponding congestion control measures according to the information in the ACK, reducing the sending rate or rerouting to avoid congestion. Congestion control scheme based on ECN can effectively reduce network congestion without losing packets, and can detect congestion faster and take corresponding measures. It is one of the important congestion control technologies in modern networks, which can improve the efficiency and reliability of network transmission.

In 2010, Alizadeh M [1] and her colleagues proposed DCTCP (Datacenter TCP). As the first dedicated congestion control protocol created specifically for data centers, DCTCP believes that the key to reducing transmission latency is that the sender can provide the appropriate number of packets that the receiver can receive according to the actual network conditions. DCTCP greatly reduces transmission latency because a significant amount of data does not need to be stored in the switch and timeout retransmission does not occur due to transmission errors. By measuring the length of the switch queue to measure congestion and notifying the receiver with an explicit congestion alert, DCTCP establishes a regulatory mechanism between the switch and the receiver and sender. After receiving the ECN label, the receiver sends an ACK and ECE label to the sender. In order to avoid congestion, the sender modifies the sending window as required. DCTCP does not receive ECN-flagged congestion signals, but uses packet loss as a congestion signal.

In 2015, Yibo Zhu[12] et al. proposed DCQCN, which is a rate-based end-to-end congestion control protocol mainly controlled in network cards. DCQCN believes that the RoCEv2 transport technology stack based on Remote Direct Memory access (RDMA) can replace the traditional TCP protocol stack because it can cope with high bandwidth, low CPU overhead and ultra-low latency environments. In order to solve this problem, a congestion control strategy similar to QCN for streams is needed, but QCN cannot be directly applied to the third layer of the network. Therefore, DCQCN combines the control strategy of DCTCP to improve QCN and realize congestion control.

However, the existing congestion control schemes based on ECN markers have several major defects: the first is the setting of the ECN marker threshold. In order to avoid excessive congestion or excessive reduction of transmission rates, ECN thresholds need to be constantly adjusted, which leads to the need to provide accurate end-to-end congestion information in real time. Secondly, this mechanism using the feedback of the receiver will lead to a long path cycle, and there will be a problem that the feedback information has not reached the sender but the congestion has been removed. These are also the core problems in the follow-up research of congestion control.

C. RTT-Based CC

Congestion control algorithms hope to obtain fine-grained congestion signals dynamically and real-time to achieve accurate rate control or window size control, the core is the selection of congestion signals. Delay-based congestion control scheme is an algorithm to dynamically adjust the sending rate. It uses Round Trip Time(RTT) to measure congestion and supports multiple traffic levels[2].

The following is the basic working principle of the delay-based congestion control scheme. In the delay-based congestion control scheme, network delay is the key parameter. Control schemes represented by Timely[7] use Round Trip Time(RTT) to measure latency, which can be achieved by adding time stamps to packets or using ICMP packets. Through delay measurement, the sender can obtain the delay information of the current network and adjust the sending rate dynamically. The delay-based congestion control scheme controls the sending rate by adjusting the congestion window. The congestion window is a parameter used to limit the size of the data sent by the sender. According to the received delay information, the sender dynamically adjusts the size of the congestion window to control the sending rate. Congestion control algorithm is needed to adjust the congestion window size, which is affected by different transport protocols.

The work flow of the delay-based congestion control scheme is as follows: First, when sending data, the sender records the sending time and sends the packet to the receiver. After receiving the packet, the receiver sends a confirmation message, and the sender records the time when the confirmation message is received. From these two timestamps, the delay information of the current packet can be calculated. After receiving the delay information, the sender uses the congestion control algorithm to dynamically adjust the congestion window size to adapt to the changes of the network. Adjusting the size of the congestion window affects the sending rate, thereby avoiding network congestion and resource waste.

In 2015, a congestion control protocol named TIMELY was proposed by R.Maital [6] team. The protocol uses RDMA-based technology to realize end-to-end RTT measurement and rate control by using the method of transmission rate control at the sender end, so as to achieve the purpose of network congestion control in data center. TIMELY contains three modules: RTT measurement, rate calculation, and rate control, which work together to achieve congestion control. Compared with switch-based congestion control schemes, the TIMELY use of end-to-end measurement methods reduces the dependence on switch devices, and rate-based control is more suitable for low-latency data center networks than window-based control. However, it is difficult to measure RTT accurately due to TIMELY modification of network cards, high requirements on hardware performance, and excessive sensitivity to RTT changes in practice.

The limitation of this type of scheme is that although the RTT signal is a valuable congestion signal, the RTT measurement can cause all queues in both directions of the network path to merge, which may not only cause congestion on the reverse ACK path, but also confuse it with the forward path congestion experienced by the data packet. Therefore, it is necessary to consider how to distinguish RTT in various traffic scenarios[10].

IV. Design

A. Why lossless

With the continuous development of the big data society, the demand for long distance and low delay applications is increasing. Typical applications include: emerging real-time services: automatic driving, Cloud VR, machine vision and a large number of real-time services appear. Terminals generate a large amount of data and upload it to computing nodes in the edge and cloud for processing, and require the results to be sent back to terminals in real time. High-performance distributed and parallel computing: Distributed high-performance computing requires memory data to be copied across networks, and network performance must be the same as memory access. If the network for such applications is not strictly lossless, performance losses such as delay jitter, congestion and packet loss can be introduced, leaving the processor idle waiting for data, and dragging down overall parallel computing performance.

The deployment of RDMA in the data center mainly relies on the reliable transport environment provided by the RoCEV2 protocol. The reliability of transmission is the commitment of the network transport layer to the upper-layer applications and the guarantee of stable and low latency for data center applications. In the event of a packet loss, the transport layer usually follows the rules of a Multiplicative-decrease in send rate and retransmits the dropped packets, causing the user to experience a burst of performance degradation. The core of reliable transmission is how to deal with packet loss. As shown in Figure 4, on a 10Gbps link, the packet loss rate of 0.0046% will lead to a sharp decline in throughput. It can be found that the throughput of a link in a lossless environment is much higher than that in an environment with packet loss.

The default retransmission mechanism of packet loss in the existing RDMA transport protocol is Go-Back-N, which detects whether packet loss occurs by detecting the sequence number of received packets. Assume that the ACK of packet 5 is missing when packet 10 is being transmitted. The receiver discards all the answered packets after packet 5 and notifies the sender to resend packets starting from packet 5. According to the Go-Back-N model, the throughput formula is derived as follows:

$$\text{Goodput} = \sum\nolimits_k \frac{(1-\text{ERROR\_RATE})^{k-1}\cdot\text{ERROR\_RATE}\cdot((k-1)\cdot\text{MTU})}{(k-1)^\star\cdot\text{MTU}+B^\star\cdot\text{RTT}} \tag{1}$$

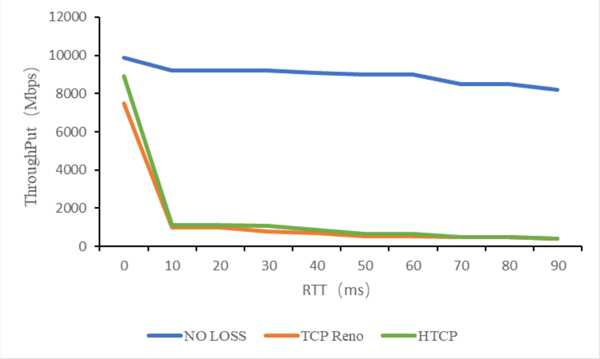

Fig. 4: Loss Rate via BDP

ERROR_RATE indicates the inherent packet loss rate generated by the link hardware. It can be seen that the bandwidth utilization of a link decreases continuously with the increase of RTT and packet loss rate. This makes the transmission reliability of RoCEV2 greatly affected in the scenario where the average RTT is at the millisecond level and the network transmission path is complicated, resulting in poor bandwidth utilization and extremely high stream completion time. To solve this problem, this paper designs the RDMA version of selective retransmission: Go-Back-ONE. The hardware cost is reduced by sacrificing software cost, and the existing packet loss recovery mechanism is optimized. The congestion control algorithm still adopts DCQCN and DCTCP, so that it can be integrated into the existing RDMA network card, and it is tested and analyzed in Section 5.

B. Go-Back-One

In this paper, a Go-Back-ONE packet loss recovery algorithm based on RDMA protocol is designed for packet loss caused by link error. Its core process is that each packet is connected by a strict serial number, when the receiver receives the packet that is not the expected serial number, the receiver will detect the out-of-order, at this time the receiver does not take the operation of directly discarding the out-of-order packet, but cached in its own buffer. At the same time, the receiver will send a packet loss signal to inform the sender, and the sender will resend such packets with high priority after receiving the packet loss signal. After confirming that the lost packets are successfully received, the sender and the receiver resume the normal transmission process. Figure 5 summarizes the core ideas of the Go-Back-ONE algorithm.

The Go-Back-ONE algorithm is designed for packet loss recovery when the link is damaged. The whole algorithm is implemented on the network card of the Sender host and the Receiver host. Figure 6 combines specific network devices, uses two servers and a router to simulate a WAN transmission environment initially, and describes the core flow of the Go-Back-ONE algorithm in the actual environment. If a packet in a stream

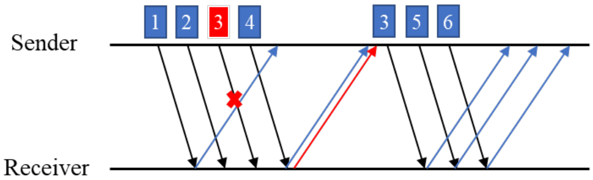

Fig. 5: Go-Back-One

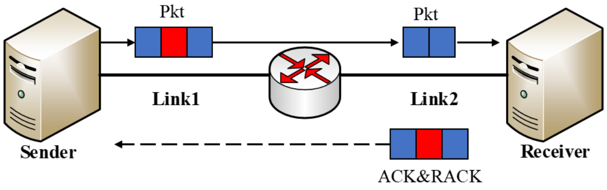

Fig. 6: Go-Back-One Framework

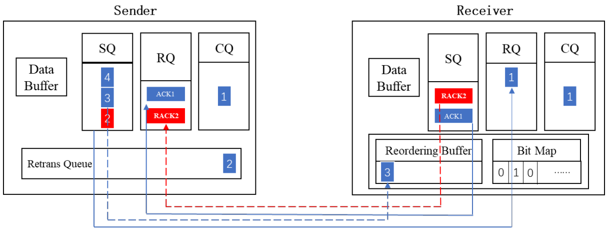

Fig. 7: Go-Back-One Process

is accidentally lost due to a hardware or software error, the Go-Back-ONE algorithm implements packet loss recovery by building a new packet loss notification signal RACK. The Go-Back-ONE algorithm consists of the sender algorithm and the receiver algorithm.

Sender algorithm: The sender mainly involves two important modules. (1) Check whether packet loss notification signal RACK is received. The sender distinguishes a normal ACK from a packet loss notification signal RACK by judging the protocol number field in the received reply packet header. (2) Confirm the sequence number of the next packet to be sent. If a normal ACK is received, the minimum transmission unit is added to the serial number of the sent packet according to the normal sending process. If RACK is received, the system switches to the retransmission process directly. The serial numbers of the packets that are not received are confirmed by the serial numbers of the RACK, and these packets are placed in the retransmission queue. Then the packets in the queue are retransmitted with a high priority. When RACK is no longer received, the sender returns to the normal packet sending process.

Receiver algorithm: Receiver mainly involves two important modules. (1) Confirm whether the received serial number is out of order or packet loss. After receiving a UDP packet, the receiver compares the sequence number of the packet with the sequence number of the packet it wants to receive. Once two values are found to be unequal, it is determined that there is an out-of-order, and it is temporarily stored in the receiver's local buffer through the bitmap data structure rather than directly discarded. (2) Once a packet loss is discovered, a packet loss notification RACK is generated. By assembling a user-defined packet header that carries the sequence number of the currently received packet and the sequence number of the expected packet, the sender can be informed which packets are not received successfully.

Figure 7 illustrates the Go-Back-ONE algorithm with a concrete example of packet transmission in a real network card: the sender sends a stream containing packets 1, 2, 3, and 4, packet 1 is normally sent, and is normally accepted by the receiver, so the receiver sends back the regular reply signal ACK1. At this time, packet 2 is discarded due to a link error, and the receiver directly receives packet 3. At this time, the receiver detects that the order is out of order, caches packet 3, generates packet discard notification signal RACK2 to inform the sender, and marks the corresponding position in the bitmap. After receiving RACK2, the sender pushes packet 2 into the retransmission queue and resends packet 2 with high priority.

The Go-Back-ONE algorithm mainly consists of three core modules to realize the whole packet loss recovery process: receiver serial number confirmation module, packet loss notification generation module and packet loss retransmission module. Figure 8 is a flow diagram of the entire Go-Back-ONE algorithm. The receiver serial number confirmation module is used to establish the packet transmission chain to ensure that the out-of-order situation in the whole transmission process can be detected in the first time. The packet loss notification generation module is used to establish a fast notification mechanism, reduce the performance loss caused by packet loss, and reduce the occupation time of out-of-order packets in the receiver buffer. The packet loss retransmission module sets a high-priority queue to support rapid response after receiving a packet loss notification signal and complete the whole packet loss recovery process.

C. Core Module

1) Receiver serial number confirmation module: The work of this paper is based on the default Go-Back-N algorithm of HPCC[4]. First, the receiver should determine whether to send a normal reply signal according to the serial number of the packet after receiving the packet. Two variables are set: seq, the sequence number of the currently received packet, and ReceiverNextExpectedSeq, the next sequence number expected. Compare the values of these two variables to determine whether packet loss occurs. If the values are equal, the function updates ReceiverNextExpectedSeq with the expected sequence number and data packet, indicating that the data packet has been received, and updates the next expected sequence number, returning

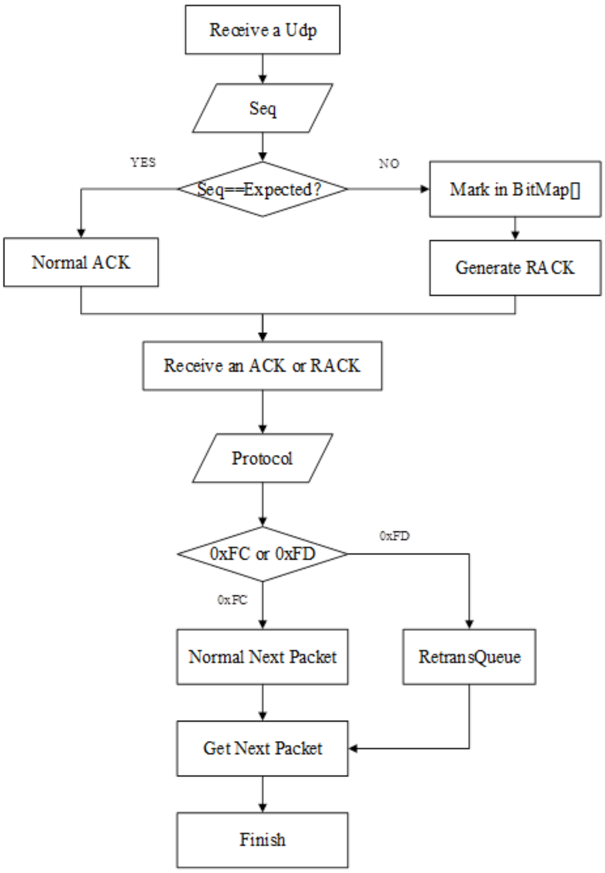

Fig. 8: Flow chart of Go-Back-ONE algorithm

the signal value generating a normal ACK. If the incoming seq is less than the expected packet sequence number, it indicates that a rearrival packet has been received, and the rearrival signal value is returned. If the sequence number of the incoming packet is greater than the expected sequence number of the received packet, it indicates that a packet arrived out of order. In this case, the bitmap caches the packet and records the location of the packet in the array.

In order to solve the problem of not caching out-of-order packages but writing them directly to user memory, we need to consider the problem of WQE matching: the application initiates the RDMA operation by constructing a WQE, and WQE records the description information of the operation. The internal implementation of RDMA assumes that each time a complete packet is received, it corresponds to the next WQE to be processed. The packet is out of order so that the next WQE is not expected to be needed. In this case, each WQE must have its own serial number, and the packet header must carry serial number information for matching. Go-Back-ONE uses a bitmap as a data structure to record missing data serial numbers. The receiver uses an array bitmap of size max_ooo_psn to record the out-of-order packet information. The array subscript indicates the location of the out-of-order packet in the receiver's cache. When the receiver receives a new packet, and its sequence number is greater than the next sequence number expected to receive, it calculates the offset of the current sequence number relative to the received next sequence number, and stores the data size in the corresponding position in the bitmap and returns a specified value, which is the result of serial number verification.

2) Packet loss notification generation module: After the receiver determines the signal value returned by the module through the serial number, if the signal value returned is the signal value of out-of-order and packet loss, a custom response packet RACK is constructed. A user-defined packet header is constructed first, and the value of the protocol field is set to 0xFD, so that the sender can distinguish ACK and RACK. Then create a new packet, newp2, and add seqh2 as the header. Finally, it is added to the high-priority send queue, triggering the network interface to transmit.

3) Sender lost packet retransmission module: After receiving an ACK or RACK, the sender parses the information in the packet header to obtain their related fields: queue index number (qIndex), port (port), serial number (seq), and CNP flag (cnp). The type of response signal received is determined based on the protocol number of Layer 3 forwarding. 0xFC indicates ACK and 0xFD indicates RACK. If a RACK is received, the sender senses that a packet loss has occurred and passes the unreceived packets to the retransQueue. Algorithm 1 shows the whole process after receiving the reply signal.

After the retransmission packet is pushed into the retransQueue, the function that obtains the next packet to be sent must be modified synchronously, and the packets in the retransQueue are preferentially sent. First, two variables, seqToSend and payLoadSizeToSend, are initialized to store the sequence number and load size of the packet to be sent. Then, the sequence number of the sender qp of the RDMA transmission is changed to point to the sequence number of the next packet to be sent to achieve high-priority retransmission. Algorithm 2 explains this process in detail.

D. Analysis

In order to support the existing hardware, the current scheme has only been modified on the RDMA network card [13] side, and is theoretically compatible with existing network devices. In addition to the improvements on the network card side, IRN[5] also pointed out that the upper layer application needs to solve the overwrite write problem: if two RDMA writes occur in the same memory area, the first write loses packets, the second write arrives first, and the first write arrives after the retransmission, the old data may overwrite the new data.

According to IRN research, RDMA selective retransmission can perform effective packet loss recovery only in the scenario where multiple packets are

Algorithm 1 ReceiveACK

```
Require: packet index
Ensure: reflash
$qIndex \leftarrow ch.ack.pg$
$port \leftarrow ch.ack.dport$
$seq \leftarrow ch.ack.seq$
$cnp \leftarrow (ch.ack.flags >> qbbHeader :: FLAG\_CNP)\&1$
$qp \leftarrow$ GetQp$(ch.sip, port, qIndex)$
if $ch.l3Prot == 0xFC$ then
  { ACK}if $\neg m\_backto0$ then
    if $qp$.Acknowledge$(seq)$ then
      $qp.m\_lastAckTime \leftarrow$ Simulator.Now()
      $qp.m\_retransEvent$.Cancel()
      $qp$.ReScheduleTimeoutTimer()
    else
      print "duplicate acks."
    end if
  else
    $goback\_seq \leftarrow seq/m\_chunk * m\_chunk$
    if $qp$.Acknowledge$(goback\_seq)$ then
      $qp.m\_lastAckTime \leftarrow$ Simulator.Now()
      $qp.m\_retransEvent$.Cancel()
      $qp$.ReScheduleTimeoutTimer()
    else
      print "duplicate acks."
    end if
  end if
  if $qp$.IsFinished() then
    QpComplete$(qp)$
  end if
end if
if $ch.l3Prot == 0xFD$ then
  { RACK}if $qp.m\_lastRACK < qp.snd\_una$ then
    $qp.m\_lastRACK \leftarrow qp.snd\_una$
  end if
  if $qp.m\_crtRACK < qp.snd\_una$ then
    $qp.m\_crtRACK \leftarrow qp.snd\_una$
  end if
  if $seq > qp.m\_crtRACK$ then
    $qp.m\_lastRACK \leftarrow qp.m\_crtRACK$
    $qp.m\_crtRACK \leftarrow seq$
    if $qp.m\_crtRACK > (qp.m\_lastRACK + m\_mtu)$ then
      $retransQueueSeq \leftarrow qp.m\_lastRACK + m\_mtu$
      while $retransQueueSeq < qp.m\_crtRACK$ do
        $qp.retransQueue$.push$(retransQueueSeq)$
        $retransQueueSeq \leftarrow retransQueueSeq + m\_mtu$
      end while
    end if
  end if
end if
```

Algorithm 2 GetNextPacket

```
Require: RDMA QP
Ensure:
  function GetNxtPacket(qp):
  seqToSend ← 0
  payLoadSizeToSend ← 0
  if qp.snd_nxt == qp.snd_una then
      {seqToSend}seqToSend ← qp.snd_nxt
      payLoadSizeToSend ← qp.GetBytesLeft() if m_mtu ¡ payLoadSizeToSend then
      payLoadSizeToSend ← m_mtu
    end if
    qp.snd_nxt ← qp.snd_nxt + m_mtu
  else
      {}if not qp.retransQueue.empty() then
      if qp.snd_nxt == qp.snd_una then
        seqOfPkg ← qp.retransQueue.front()
        qp.retransQueue.pop()
        while seqOfPkg ¡ qp.snd_una and not qp.retransQueue.empty() do
          seqOfPkg ← qp.retransQueue.front()
          qp.retransQueue.pop()
        end while
        if seqOfPkg ¡ qp.snd_una then
          seqToSend ← qp.snd_nxt
          payLoadSizeToSend ← qp.GetBytesLeft()
          if m_mtu ¡ payLoadSizeToSend then
            payLoadSizeToSend ← m_mtu
          end if
          qp.snd_nxt ← qp.snd_nxt + m_mtu
        else
          seqToSend ← seqOfPkg
          payLoadSizeToSend ← m_mtu
        end if
      else
        seqToSend ← qp.snd_nxt
        payLoadSizeToSend ← qp.GetBytesLeft()
        if m_mtu ¡ payLoadSizeToSend then
          payLoadSizeToSend ← m_mtu
        end if
        qp.snd_nxt ← qp.snd_nxt + m_mtu
      end if
    end if
  end if
  p ← CreatePacket(payLoadSizeToSend)
  seqTs ← CreateSeqTsHeader()
  seqTs.SetSeq(seqToSend)
  seqTs.SetPG(qp.m_pg)
  p.AddHeader(seqTs)
  udpHeader ← CreateUdpHeader()
  udpHeader.SetDestinationPort(qp.dport)
  udpHeader.SetSourcePort(qp.sport)
  p.AddHeader(udpHeader)
```

transmitted. For example, in the scenario where a single packet message is transmitted, triggering selective retransmission will lead to extremely high tail delay of short messages. Based on this, the scenario mainly studied in this paper is a complex transmission environment with long distance across wide area networks, RTT at the millisecond level and inherent packet loss rate of links. Meanwhile, the research and analysis on transmission reliability of long streams are more focused, and the subsequent research can be further in-depth.

Testing the transmission reliability of RoCEV2 protocol in WAN mainly involves the parameter adjustment of two modules. The first is the congestion control module, which integrates several existing congestion control algorithms in the simulation environment. The first is DCQCN, which mainly uses the intermediate switch to check the current congestion situation. When the queue depth exceeds the threshold, the packet is marked by RED or ECN and forwarded to the next hop. Using DCQCN involves the setting of queue depth threshold and congestion point congestion degree coefficient. The second is DCTCP. The receiver will reply with an ECN-Echo ACK for each packet marked with ECN. The sender quantifies the degree of congestion according to the number of packets arriving to be marked and adjusts the congestion window proportionately. Like DCQCN, it also needs to set a corresponding queue depth threshold and a proportional coefficient. The third is TIMELY. The algorithm does not involve requiring the receiving network card to reply an ACK for each received packet but does not involve the intermediate switch. It measures the RTT of each packet, and when the RTT rises, it indicates that congestion is getting worse; Instead, congestion is easing. The TIMELY use is mainly to set the rate control coefficient and the theoretical lower bound of RTT. In Section 5, each parameter is set to explore the effects of different control granularities.

The second is the packet loss retransmission module, which is also the focus of this paper. The selective retransmission algorithm used in this paper mainly involves the receiver's maintenance of the bitmap, and the array needs to be set an upper limit to prevent unrestricted buffer allocation. Secondly, it is necessary to set two Pointers to complete the serial number of the sender and the receiver. At the same time, NS3's monitoring function for the channel is used to rewrite a tracking algorithm to obtain the link bandwidth size within a period of time for comparison Throughput.

V. Conclusion

A. Parameter

1) Topolofy: The network topology of NS3 small network simulation experiment selected a simple tree structure. The test bench topology simulated the small RDMA PoD in production. The testbed consists of a core switch and two servers connected via two 40Gbps or 100Gbps links. Since the transmission scenario targeted in this paper is mainly in the WAN, and the RTT of the WAN is generally at the millisecond level, this paper assumes that two hosts communicate across the WAN at a distance of 400km and 2000km, that is, each link is set to a transmission delay of 2ms or 10ms at the same time. In addition, to verify whether the packet loss retransmission algorithm works, set the corresponding inherent packet loss rate for the two links. The packet loss rate is mainly caused by hardware or software errors. The default link packet loss rate is 10%. The entire network is a single RDMA domain.

2) Traffic model: A widely accepted and publicly available data center traffic model is used in both experiments and simulations and the experimental traffic is generated by corresponding cumulative distribution functions. In the NS3 small simulation experiment, these two classic traffic loads are mainly used: WebSearch[8] and FB_Hadoop[11], based on the cumulative distribution function summarized in other studies, this paper specified the source address and destination address, set the upper limit of link bandwidth to 100Gbps, adjusted the traffic generation rate, and set the average link load to 30% and 50% respectively. In this way, the flow file required for the simulation experiment is generated. Four traffic model files were initially generated in the experiment: WebSearch30% load, WebSearch50% load, FB Hadoop30% load and FB Hadoop50% load, all of which were sent from node 0 to node 1. Table 1 shows the specific traffic distribution models of these two traffic modes.

TABLE I: Flow size distribution in traffic modeByte

	10^1	**10^2**	**10^3**	**10^4**	**10^5**	**10^6**	**10^7**
FB_Hadoop	1%	2%	60%	71%	88%	97%	100%
WebSearch	4%	7%	10%	15%	60%	75%	100%

B. Experiment

The main performance indicators tested are FCT_Slowdown and real-time link bandwidth utilization. Suppose Tidea represents the flow completion time of a stream in an ideal case, and Treal represents the flow completion time of a stream in a real case. FCT_Slowdown=Treal/Tidea. The closer the result is to 1, the better the network performance can be optimized under the cooperation of congestion control and retransmission mechanism algorithms. Since the FCT of a small stream is normally smaller than that of a long stream, it is not convincing to directly compare the flow completion time of all streams. This paper mainly adopts the following data processing methods: The completion time of all streams is sorted forward, divided into 20 groups on average, and the experimental effect is compared by comparing the median time of the same group or 95% of the completion time of the stream, which is also the standard for measuring the tail delay.

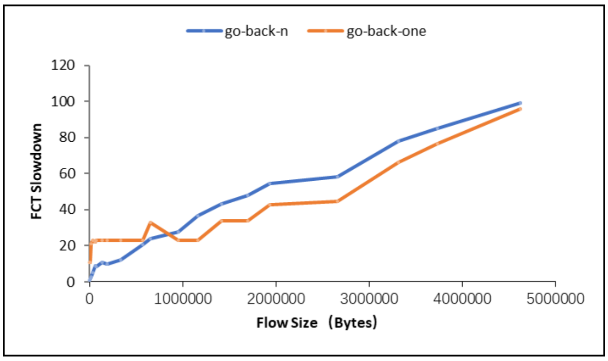

Fig. 9: FCT comparison graph of WebSearch load 30%

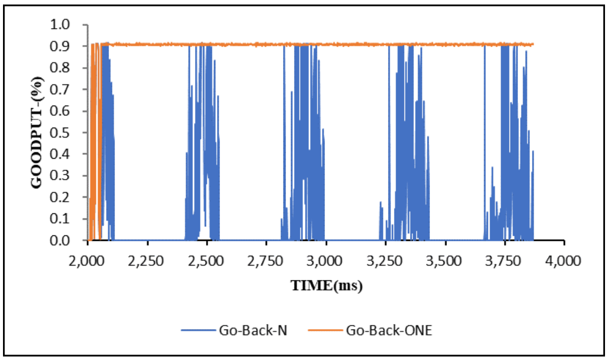

Fig. 10: Goodput comparison of two 100Gbps packet loss retransmission algorithms

The parameters of the first simulation experiment are as follows: the traffic model adopts WebSearch, the congestion control algorithm is set to DCQCN, the bandwidth of each link is set to 100Gbps, the delay is set to 2ms, and the entire transmission environment is set to 8ms, simulating the point-to-point data transmission across the wide area network of 400km. Figure 9 shows the FCT Slowdown at the 95th percentile of the two algorithms at 30% network load. Under the experimental setting, it can be seen that when the average stream size is less than 1MB, the Go-Back-N algorithm has a better FCT deceleration, and when the average stream size is greater than 1MB, the Go-Back-ONE algorithm has a better FCT deceleration, which can be optimized by about 10% on average.

Inspired by the research work of SRNIC[9], it may be due to the occurrence of continuous packet loss that the received sequence number and expected value have too large deviation, which requires polling from the free array subscript of the bitmap one by one until all the lost packets are resent, which may lead to further deterioration of the actual stream completion time. This problem also provides ideas for the subsequent research work.

Figure 10 shows the impact of the two algorithms on FCT after switching to DCTCP. It can be seen that the overall performance of FCT deceleration in long stream is not much different from the previous results. Go-Back-ONE algorithm still has better transmission performance on long stream. The two algorithms are more similar in the performance of small-stream transmission, and Go-Back-N is only slightly better than Go-Back-ONE.

Due to the accumulation of queues in the sender, there will be obvious deviation in the bandwidth utilization test at the source end. The experiment chooses to obtain real-time Throughput by monitoring the receiver channel. The simulation start time set in this paper is 2 seconds, the end time is 4 seconds, the parameter Scale is set for sampling to calculate the number of packets passed within 1/scale millisecond, and the instantaneous link bandwidth is calculated, and then divided by the total link bandwidth set during the test 100Gbps to obtain the effective throughput Goodput.

Figure 10 shows the comparison of the link bandwidth utilization of the two algorithms using two different packet loss retransmission mechanisms in the experimental environment where the link transmission delay is 10 ms, the packet loss rate is 10%, and the congestion control algorithm is DCQCN. 5.6 (a) uses a link bandwidth of 100Gbps, As can be seen from the figure, since the number of packets that need to be retransmitted by the Go-Back-ONE algorithm is much smaller than that of Go-Back-N, the receive bandwidth utilization rate of the receiver using the selective retransmission mode converges to 90%. In terms of value, there is a fluctuation of about 5Gbps. However, the Go-Back-N algorithm will discard all the out-of-order packets, resulting in repeated packet transmission, and the average effective bandwidth utilization in the high-delay WAN environment is significantly worse than that of the Go-Back-ONE algorithm.

VI. Future Work

In this paper, a selective retransmission algorithm based on RoCEV2 protocol, Go-Back-ONE, is proposed based on the research of high performance transmission network and the characteristics of existing WAN technology. This scheme uses the selective retransmission algorithm of TCP but does not implement the complete TCP protocol stack, and does addition on the basis of the default Go-Back-N algorithm. This paper designs and implements the Go-Back-ONE algorithm of RDMA for packet loss recovery, and analyzes the transmission reliability of RoCEV2 in WAN.

The main features of this graduation project are:

(1) The models and principles of traditional WAN and emerging data center networks are studied and analyzed, and the cutting-edge work and advantages and disadvantages of high-performance transmission networks are introduced.

(2) The advantages and disadvantages of the existing packet loss retransmission algorithm and packet

loss notification mechanism in the WAN scenario are studied and analyzed, and the domestic and foreign technical status and application difficulties of various technologies supporting RoCEV2 in the cross-WAN transmission environment are specifically summarized.

(3) Based on the original Go-Back-N packet loss recovery algorithm, an improved selective retransmission algorithm supporting RDMA, Go-Back-ONE, is proposed, which is one of the key technologies to realize RoCEV2 transmission in WAN.

(4) A series of theoretical analysis and experimental verification are carried out on the two core performance indexes of the proposed packet loss recovery algorithm, namely stream completion time and link bandwidth utilization, which reflect transmission reliability. It can be concluded that the Go-Back-ONE algorithm can effectively improve the transmission reliability of RoCEV2 in WAN.

Due to the limitations of various factors, the Go-Back-ONE packet loss retransmission mechanism implemented in this paper still has some defects. However, due to time constraints, the traffic model selected in this experiment is limited, the network topology is relatively simple, and large-scale network simulation experiments are not carried out. In fact, Go-Back-N algorithm has better performance in short flow. In addition, in the process of research, I found that different packet loss recovery algorithms showed different performance advantages and disadvantages under different circumstances.

However, this paper did not design a multi-angle comparative test to analyze which retransmission algorithm has better performance under which circumstances (it can only prove that the retransmission algorithm in this paper has better reliability in WAN transmission through experiments). Therefore, in the future work, it is still necessary to continuously improve the network transmission under different traffic models and network topologies and other details, and comprehensively analyze what algorithm can be used in what scenarios to obtain better transmission performance.

VII. Knowledge

At this point in writing, foolishness finally. Over the past 20 years, although I am not a person who is tired of learning, I have never thought of embarking on the road of scientific research. The grace of many teachers' education, teaching by words and deeds, not only preaching to me, but also leading me to explore in a wider world, sincerely thank my teacher, the grace of education, small life should live up to expectations. Thank my parents, give me peace and joy of a whole childhood time and worry-free youth, the child is not talented, but willing to learn to achieve, solve its hard work, share its worries, such as the high-flying kite, after all back to the hands of parents.

People can not have youth and the perception of youth at the same time, the final wish is four years, the whole university career like that dream, suddenly came to the final chapter, the autobiography of more than 20 years of youth is near the end, and I will enter the study stage of master's and doctoral students. Thanks to my tutor Professor Xiao Fu, who not only guided me to participate in the study of information security competition during the undergraduate period, but also gave me careful guidance for my future work planning. I would like to thank Mr. Wang Junchang for training me in the new field of computer network and stimulating my interest in scientific research and learning.

If life is young again, one or two gold one or two wind. Perhaps regret is something that runs through life, but what helps me out of regret is the most precious, Du Kang only with Meng De drink, mediocre people quickly raise their heads. These four years may have missed many different scenery, but I have irreplaceable friendship; These four years may have missed a lot of opportunities, but I have also embarked on my own path in life. Feeling in the heart, not in the empty text, I would like to record this thick and colorful and ordinary four years in this article.

References

[1] Mohammad Alizadeh et al. "Data Center TCP (DCTCP)". In: *Computer Communication Review: A Quarterly Publication of the Special Interest Group on Data Communication* (2010).

[2] Yanqing Chen et al. "Swing: Providing long-range lossless rdma via pfc-relay". In: *IEEE Transactions on Parallel and Distributed Systems* 34.1 (2022), pp. 63–75.

[3] Chuanxiong Guo et al. "RDMA over Commodity Ethernet at Scale". In: *the 2016 conference*. 2016.

[4] Yuliang Li et al. "HPCC: high precision congestion control". In: *Proceedings of the ACM Special Interest Group on Data Communication* (2019).

[5] Radhika Mittal et al. "Revisiting network support for RDMA". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 2018, pp. 313–326.

[6] Radhika Mittal et al. "TIMELY: RTT-based congestion control for the datacenter". In: *ACM SIGCOMM Computer Communication Review* 45.4 (2015), pp. 537–550.

[7] Heise Netze. "A Remote Direct Memory Access Protocol Specification". In: *heise zeitschriften verlag* ().

[8] Arjun Roy et al. "Inside the social network's (datacenter) network". In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 2015, pp. 123–137.

[9] Zilong Wang et al. “{SRNIC}: A scalable architecture for {RDMA}{NICs}”. In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 2023, pp. 1–14.

[10] Gaoxiong Zeng et al. “Congestion control for cross-datacenter networks”. In: *IEEE/ACM Transactions on Networking* 30.5 (2022), pp. 2074–2089.

[11] Jiao Zhang et al. “Receiver-driven RDMA congestion control by differentiating congestion types in datacenter networks”. In: *2021 IEEE 29th International Conference on Network Protocols (ICNP)*. IEEE. 2021, pp. 1–12.

[12] Yibo Zhu et al. “Congestion Control for Large-Scale RDMA Deployments”. In: *ACM* (2015).

[13] Et al. “” In: 59.1 (2022), pp. 1–21.