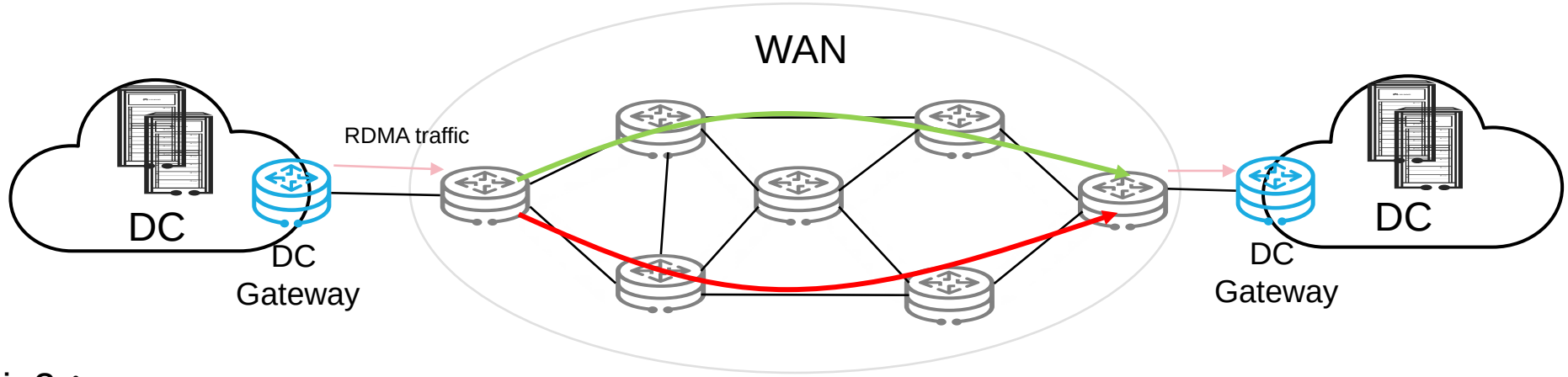
FANTEL for RDMA transmission in WAN

Jiayuan Hu(Hugh)

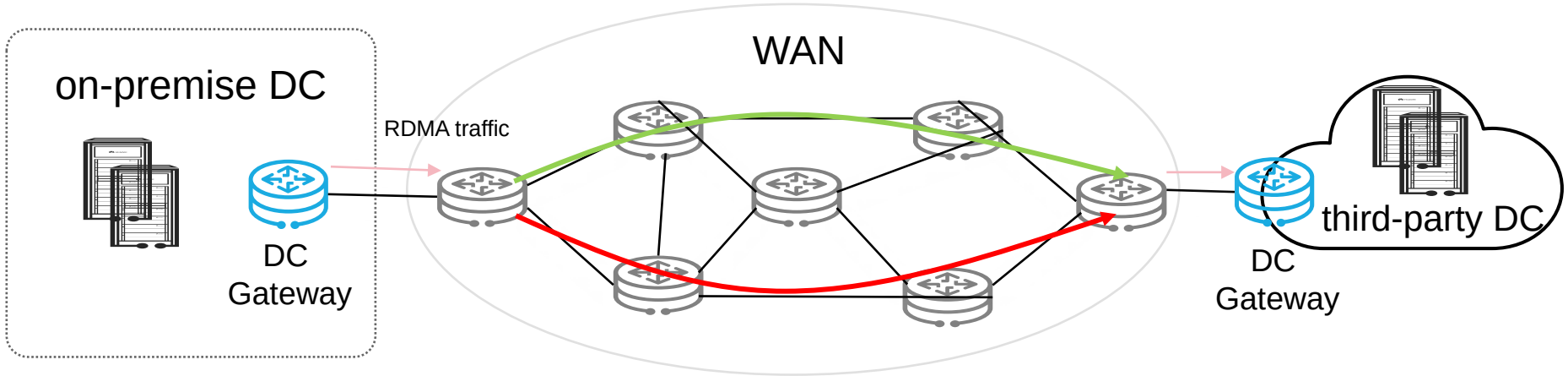
2025.11

Scenario （Why?）

Scenario1:
Distributed model training across DCs： Single DC is limited by space and power supply.



Scenario2：
Distributed model inference between on-premise DC and third-party DC: On-premise DC can‘t meet the requirement of inference concurrency increases.



Problems analysis （Why?）

All scenarios mentioned above have to face a problem, How can we ensure lossless transmission of RDMA traffic in WAN?

- Failure protection: For large-scale and dynamic networks, protection mechanisms need to ensure service continuity in case of failures.
- Congestion control: RDMA traffic is bursty and highly sensitive to packet loss, and WAN requires proactive congestion control mechanism.
- Load balancing for network state changes: Dynamic load balancing based on network state changes can effectively improve network resource utilization.

Fast Notification for tunnel-based lossless RDMA transmission in WAN
draft-hzh-fantel-wan-tunnel-01

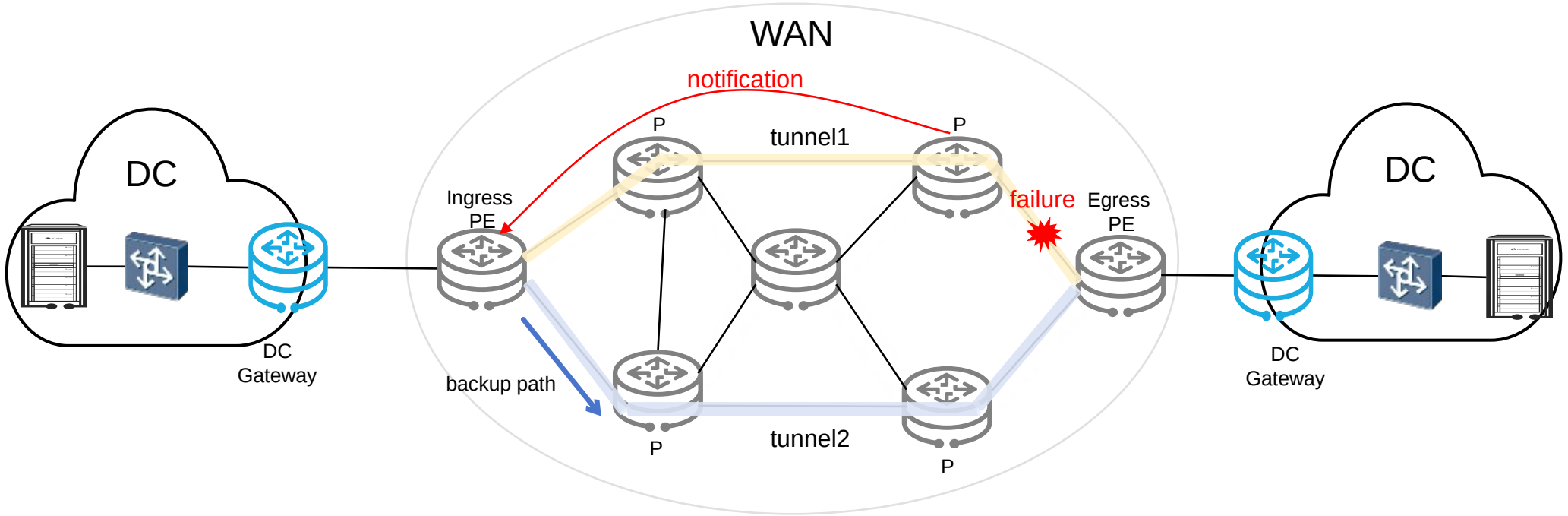
Zehua Hu

Yongqing Zhu

Jiayuan Hu

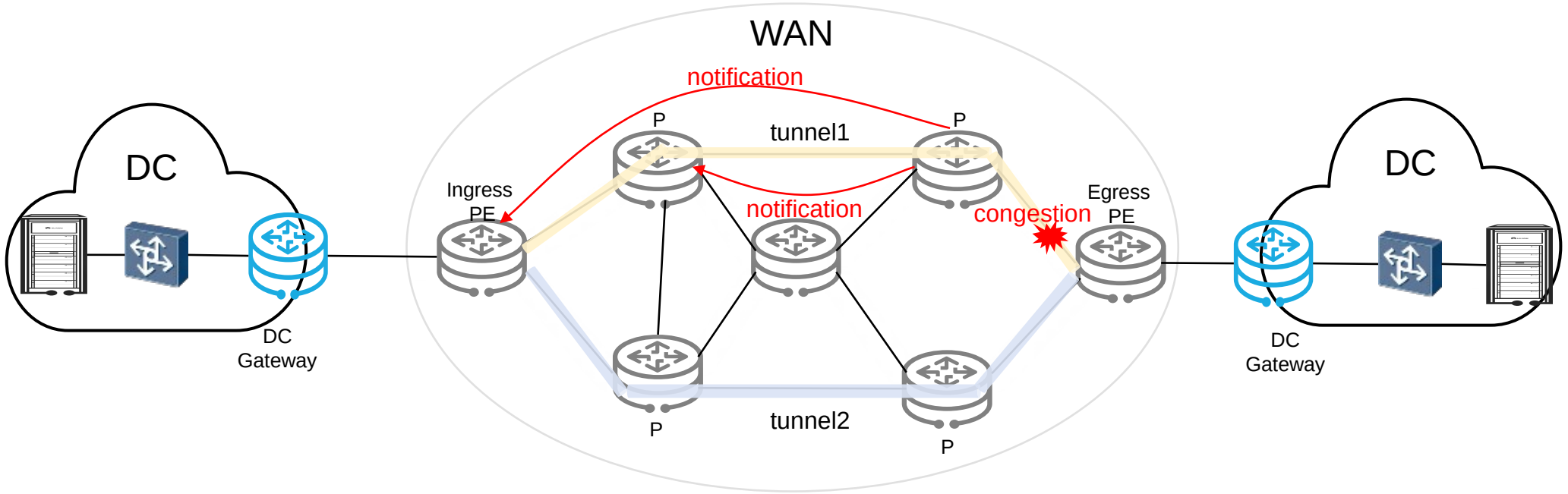
Tanxin Pi

Process analyze: failure protection(How?)



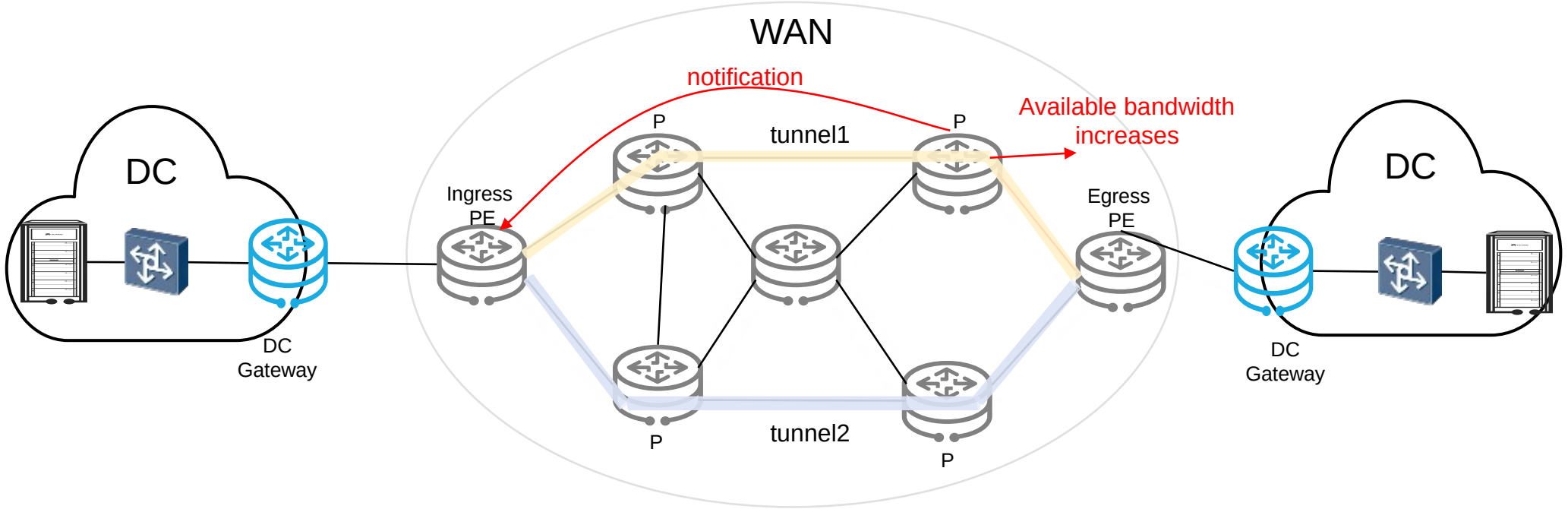
1. When a P node detects a local link/node failure, it collects failure information (In addition to the information of the failed link or node, failure information should also include the information about affected traffic)
2. The P node sends notification to ingress PE with failure information.
3. Ingress PE receives the notification and reroutes the traffic based on its content to exclude the failed link or node:
   - ① *If backup path is available, ingress PE should switch the service traffic to the backup path.
   - ② *If multiple feasible paths exist, ingress PE should updates its load-balancing policy to utilize all available paths.
   - ③ *If no feasible path is available, ingress PE should send error messages to the sender or controller.

Process analyze: congestion control(How?)



1. When a P node detects congestion, it collects congestion information (In addition to the information of the congested link or port, congestion information should also include the information about affected traffic)
2. The P node sends notification to ingress PE and upstream P node with congestion information.
3. The upstream P node receives the notification and reduce the transmission rate of corresponding traffic.
4. Ingress PE receives the notification and reroutes the traffic based on its content to exclude the congested link:
   - ① *If backup path is available, ingress PE should switch the service traffic to the backup path.
   - ② *If multiple feasible paths exist, ingress PE should updates its load-balancing policy to utilize all available paths.
   - ③ *If no feasible path is available, ingress PE should reduce the transmission rate of corresponding traffic, and send error messages to sender or controller

Process analyze: load balancing for network state changes(How?)



1. When a P node detects the network state change, it collects the network state change information, such as link utilization, queue buildup.
2. The P node sends fast notification to the ingress PE with information about the network state change.
3. Ingress PE receives the notification and updates its load-balancing policy to maximize the utilization of network resources.

Packet format for notification(How?)

Solution 1: ICMPv6-based notification

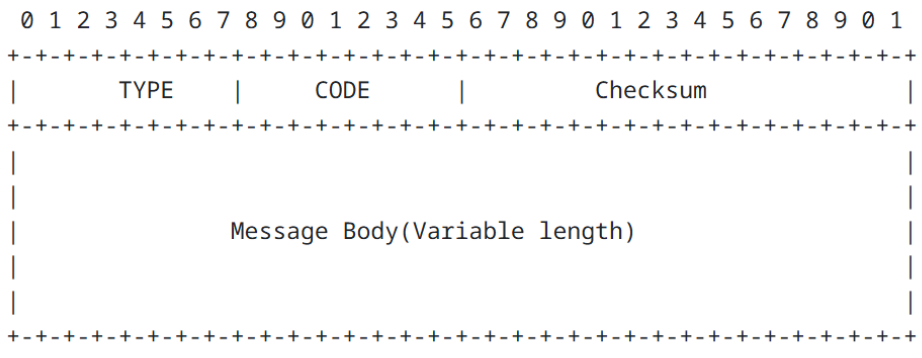


Figure 5: new ICMPv6 message for fast notification

Solution 2: UDP-based notification

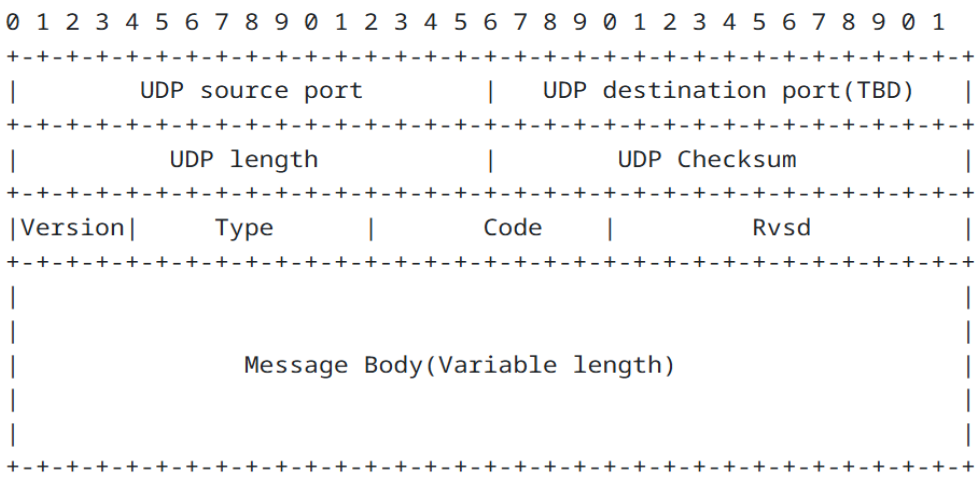


Figure 6: new UDP message for fast notification

1. TYPE: 8-bit identifier for the purposes of notification
2. CODE: 8-bit bitmap that specifies which parameters are included in the message body of the packet
3. Message Body: carries notification information specific to each areas, for failure protection, it should includes path, five-tuple of flow, and failure cause; for congestion control, it should contains path and buffer status; for load balancing, it should comprises link utilization and device load.

The format of message body needs further discussion to ensure both fixed critical information and good extensibility (TLV-based structure may be a suitable approach)

Credit-based Flow Control Based on RSVP for RDMA transmission in WAN
draft-hu-rtgwg-cbfc-rsvp-01

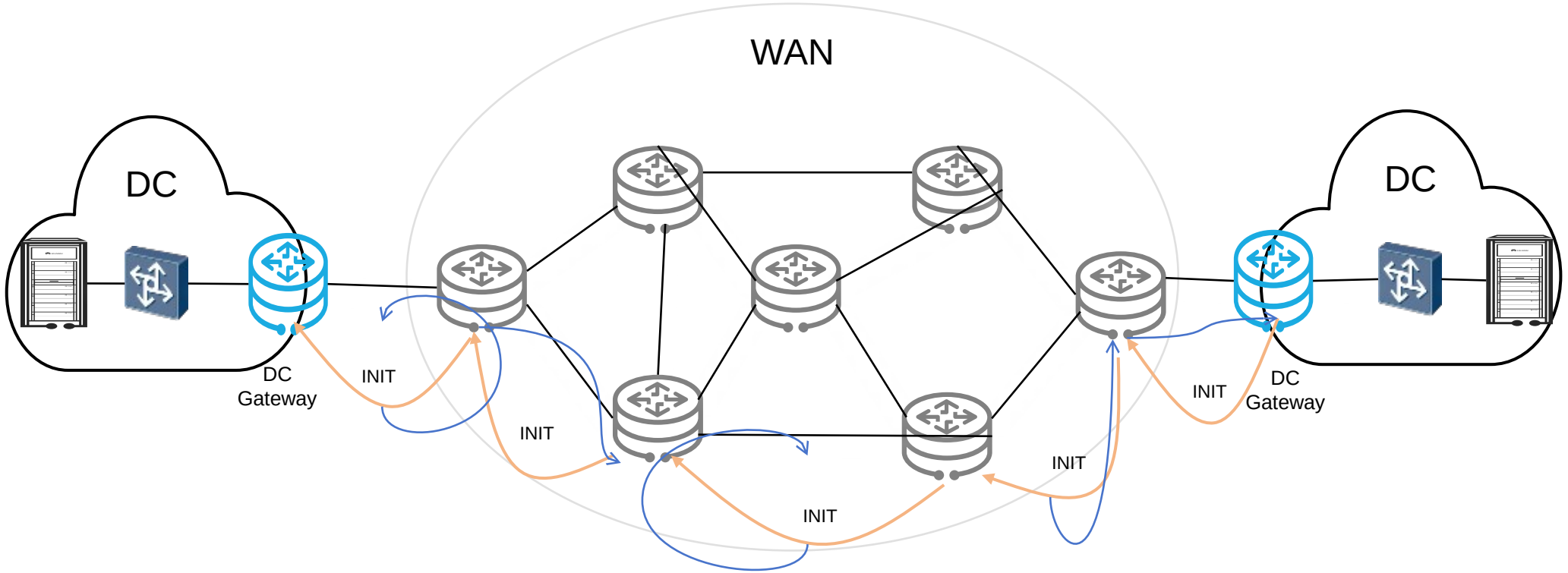
Jiayuan Hu

Zehua Hu

Tanxin Pi

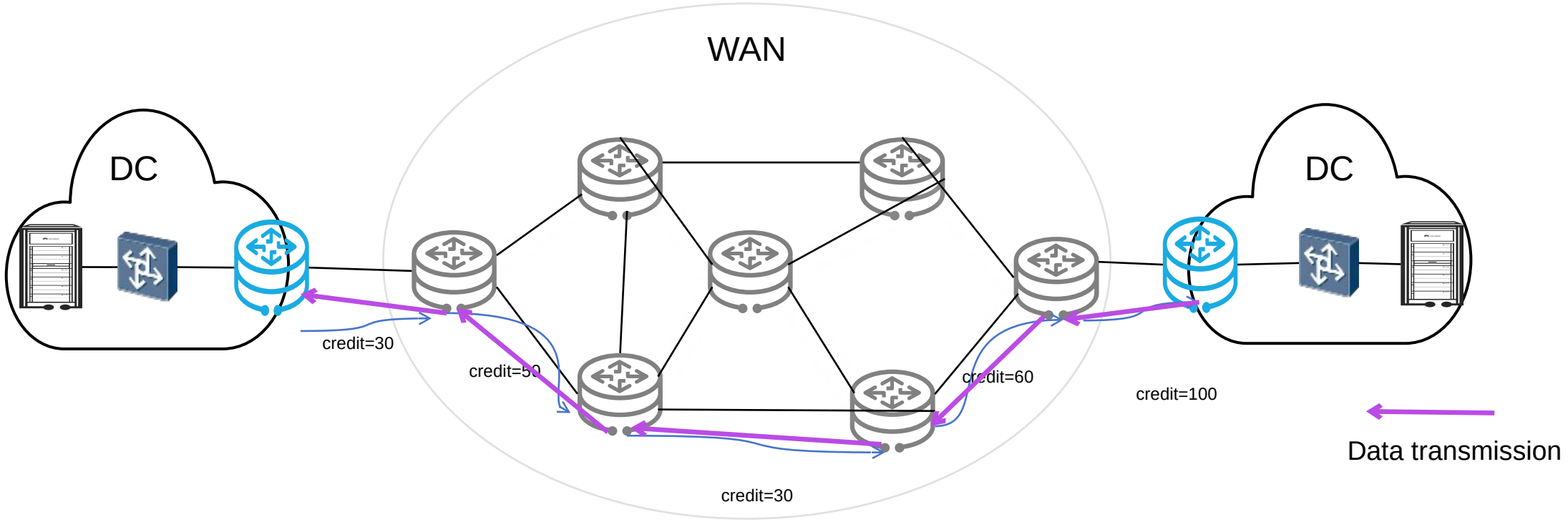
Yongqing Zhu

Credit-based flow control based on RSVP protocol(How?)



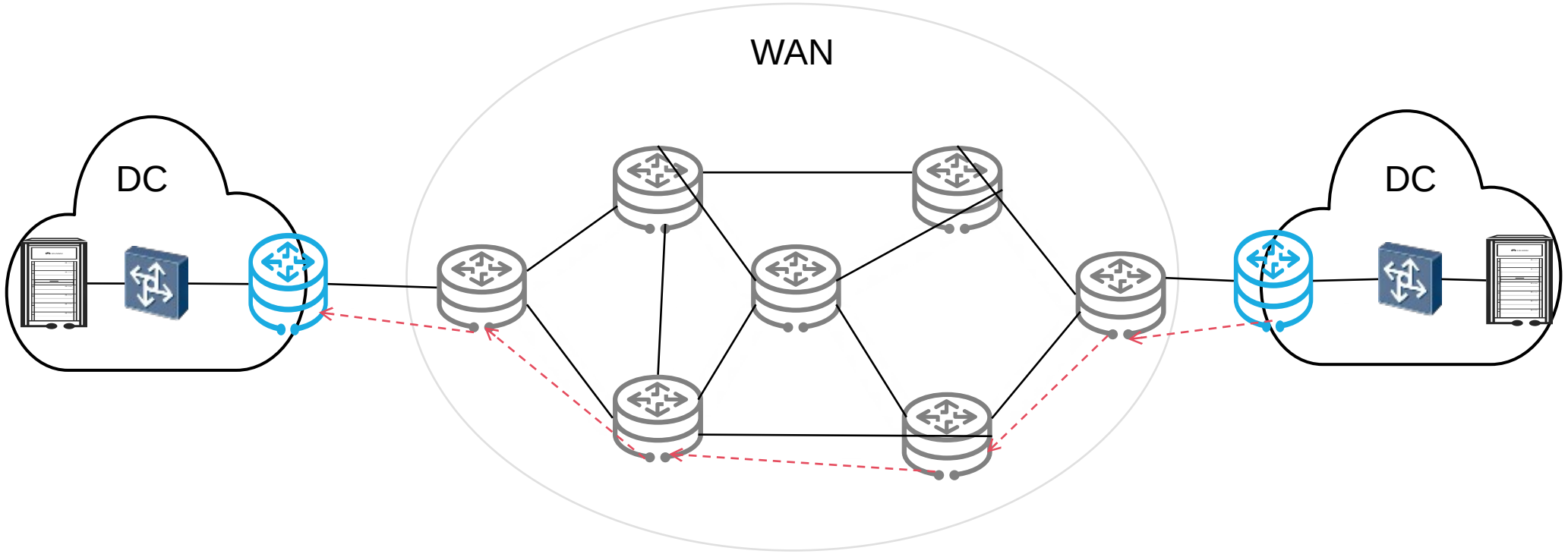
- Devices in WAN need to create transmision initialization and specify the mission ID before mission start
- The devices in the path will send an initial credit value (buffer reserved by the device for mission) to the previous hop device

Credit-based flow control based on RSVP protocol(How?)



- Devices in WAN sends packet to the next hop device based on the size of the received credit
- After the next hop device receives data, it sends a credit value based on the remaining buffer of its own device and returns it to the previous hop. If the cache space has been exhausted, it returns credit=0
- The device receives the credit value returned by the next hop device. If the credit value is not 0, it continues to send data blocks of the corresponding size. If it is 0, it pauses sending

Credit-based flow control based on RSVP protocol(How?)



- After the RDMA traffic transmission is completed, the sender will send a transmission finish message, and the device will release the reserved buffer upon receiving the message.

Credit-based flow control based on RSVP protocol(How?)

Credit-based flow control： Define new message types in the RSVP protocol and adjust transmission rates through real-time feedback of device buffer status

Vers	Flags	Message Type	RSVP Checksum
Send_TTL		Reserved	RSVP Length
Objects			

Length (bytes)	Class Num	C-Type
(Object Content)		

Message Type 27: indicate cbfc message
Class Num27: Transmission initialization
Class Num28: Respond credit message
Class Num29: Transmission finish， no need to respond credit message
object content: Indicate mission ID

Next steps

- Ask for more reviews and comments