Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Perspective:

Computing-aware network (CAN): a systematic design of computing and network convergence*

Xiaoyun WANG[†1], Xiaodong DUAN[2], Kehan YAO[2], Tao SUN[†‡2], Peng LIU[2], Hongwei YANG[2], Zhiqiang LI[2]

[1]*China Mobile Communications Corporation, Beijing 100032, China*

[2]*China Mobile Research Institute, Beijing 100053, China*

[†]E-mail: wangxiaoyun@chinamobile.com; suntao@chinamobile.com

Received Feb. 9, 2024; Revision accepted Mar. 17, 2024; Crosschecked Apr. 26, 2024

https://doi.org/10.1631/FITEE.2400098

The coverage of network resources is increasingly extensive, and computing resources have likewise gradually become fundamental infrastructures, providing ubiquitous computing services. However, in wide area networks (WANs), the underlying network and computing resources are not closely investigated or co-designed, and there are still problems reflected in slow computing service scheduling, inflexible data distribution, and inefficient data transmission. This paper proposes the architectural design of a computing-aware network (CAN), with the core contribution of introducing the awareness plane to collect, manage, and synthesize computing and network information. In this way, the awareness plane, control plane, and data plane are formed as a closed-loop control system to improve the overall system's awareness capability, decision-making capability, and data forwarding functionality. To enable the CAN architecture, three key technologies are proposed as follows: computing-aware traffic steering (CATS), elastic broadcast, and wide-area high-throughput transmission. The paper takes artificial intelligence (AI) model training, inference, and offline parameter transmission as examples to show the applicability of CAN and identifies some future research directions.

1 Introduction

Computing services are gradually evolving from centralized to ubiquitous. Cloud (Armbrust et al., 2010) and edge (Mao et al., 2017) infrastructures are widely deployed to provide on-demand and flexible computing service access. In the meantime, networks are evolving from simple data connection to offering information services and data processing capabilities, becoming platforms that carry diverse services. However, system design goals have become more diverse, such as including more critical service requirements, high resource utilization, and low energy consumption. The underlying network needs to be closely co-designed with the computing infrastructure to optimize routing decisions and computing resource selection.

Computing and network convergence gradually takes place in data centers. Data centers are evolving from providing general-purpose computing services to hosting more high-performance computing and AI computing services (Su et al., 2022). Service providers like Google redesign data center computing infrastructure with the underlying data center network from the aspects of protocol stack innovation, network topology, packet switch mode, as well as software and

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 62032003) and the 2023 Beijing Outstanding Young Engineers Innovation Studio, China

ORCID: Xiaoyun WANG, https://orcid.org/0000-0002-3574-9746; Tao SUN, https://orcid.org/0009-0003-3491-8813

hardware implementation (Gibson et al., 2022). Rather than adopting the commonly used Clos-based architecture (Clos, 1953), Google's solution uses cell switching, dragonfly network topology, and software remote memory access (RMA) operations (Singhvi et al., 2020), to meet the requirements for low latency and high interaction, which are raised by computing services. The full-stack innovation can improve AI services to a large extent.

However, there are only a few studies on the computing and network convergence in terms of WANs. Sky computing (Stoica and Shenker, 2021) is one of the few representatives. It solves mainly the problem of resource sharing and task decomposition across multi-clouds by building a compatibility layer on top of cloud infrastructure and using a unified application programming interface (API) to abstract the differences in hardware and software between multi-clouds. Since it has not touched the design of the underlying network, the over-the-top solutions are not enough to satisfy some WAN-specific service requirements. The computing and network convergence in WANs still needs further research. Major problems still exist as described in the following:

1. How can timely computing service scheduling be achieved? Scheduling a large number of service requests across multiple edge nodes usually faces the problem that the closest edge node may not be the best node for processing the current service request.

2. How can flexible data distribution be realized? There are several communication patterns by which compute nodes deliver data. Point-to-point data delivery across compute nodes will introduce a lot of bandwidth occupancy, data movement, and data copies, which is not suitable for all communication patterns.

3. How can effective throughput in ultra-distant lossy networks be increased? Transferring bulk data between large data centers over long distances is not efficient, and it still generally depends on physical hard disk transport, which is inefficient, high-cost, and not safe.

Solving the above problems is not easy, and will involve the following technical challenges:

1. It is difficult to achieve efficient and low-overhead propagation of computing information within the network when scheduling across multiple edge nodes. Additionally, quick as well as accurate decision-making is challenging based on multi-dimensional computing and network information.

2. The realization of flexible one-to-many communication in WANs is faced with high costs in terms of forwarding status maintenance. Existing solutions do not fit well either because of single functionality or the overhead in tree state management.

3. The throughput of ultra-distant data transmission drops dramatically when the packet loss rate increases. Traditional transmission control protocol (TCP) or user datagram protocol (UDP) transmission cannot guarantee the throughput requirements.

To achieve these goals, this paper presents the architecture of CAN and proposes three key technologies for addressing specific problems. The core design of the CAN is the introduction of an awareness plane to collect, manage, and synthesize computing as well as network information, and forming a closed-loop control system with the collaboration of the awareness plane, control plane, and data plane. The three proposed technologies are CATS, elastic broadcast, and wide-area high-throughput transmission.

The major contributions of this paper can be summarized as follows:

1. The architecture design of the CAN, with a newly added awareness plane, as well as an enhanced control plane and data plane;

2. Three key technologies to support and prove the applicability of the CAN;

3. Some preliminary experimental results on one of the key technologies, wide-area high-throughput transmission, to show the performance improvement it could bring to CAN systems.

2 Architecture design

2.1 Concept and architecture

In this paper, CAN is defined as the integrated interconnection, joint awareness, and hybrid control of computing and network resources, so as to achieve timely computing service scheduling across compute nodes, flexible data pre-processing and distribution, and highly effective throughput transmission.

Fig. 1 shows the architectural design of CAN. There are three functional planes in the CAN, i.e., the

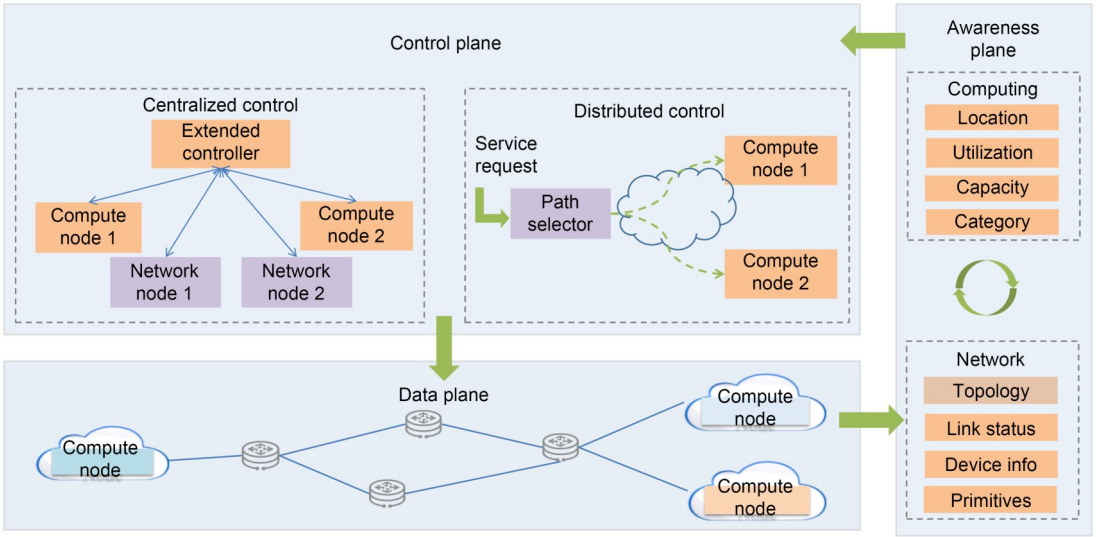

Fig. 1 Computing-aware network (CAN) architecture

awareness plane, control plane, and data plane. The awareness plane is newly introduced and is the core functional module of the architecture. The information generated from the awareness plane is a prerequisite for the enhancement of control and data plane functionality. The three planes mutually depend on one another to form a closed-loop control system.

2.1.1 Awareness plane

The awareness plane is designed to manage and synthesize the data collected from both computing resources and network resources. Some existing work also designs awareness planes (Kind et al., 2008; Yao HP et al., 2019) for network monitoring through a series of traffic measurement methods such as deep packet inspection and active measurement. However, they do not collect data from computing resources. The awareness plane proposed in this paper is the first that could gather computing as well as network information. It has two major functions, as follows:

1. Data management: Computing-related information includes the geographical location of cloud or edge compute instances, total computing capacity, the real-time utilization, and the category, such as central processing units (CPUs) for general-purpose processing, or graphical processing units (GPUs) for specific tasks like AI training, image rendering, etc. Network-related information includes network topology, link status, network device information, and primitives. Primitives describe the behaviors of network nodes, such as data aggregation, tailoring, and stitching, which are beyond simple forwarding.

2. Data synthesis: Not all the data collected from the network or compute nodes are useful for making decisions. Additionally, flooding too much data directly into the network might incur much overhead in terms of bandwidth occupancy and slow routing convergence. The decision-maker might also not be powerful enough to process the full-dimensional data to generate policies. The awareness plane can pre-process and filter the data. For example, compute nodes may report many metrics, as described in the data management part, but the awareness plane might generate only the most representative metric, i.e., the processing delay, and propagate it to the decision-maker.

2.1.2 Enhanced control plane and data plane

The control plane defines two control modes for running algorithms and generating policies for computing scheduling and data transmission. Centralized control is realized by extending the common network controller to communicate with compute nodes; thus, the controller can decompose and distribute tasks to the corresponding cloud, edge, or network nodes. Awareness functions can be integrated within the centralized controller. Distributed control is triggered by specific computing services. Path selectors are deployed distributedly within the network, and the awareness of computing as well as network information can be routed to these path selectors. When a client initiates

a computing service request, the path selector that has received this request will make decisions based on the information imported from the awareness plane.

The data plane contains the underlying network nodes that forward and pre-process data packets, and the compute nodes that provide computing services. Network nodes are enhanced to take more work beyond packet forwarding, like data aggregation, tailoring, and stitching. For different service requirements, different protocols are selected for data transmission on the data plane. For example, wide-area high-throughput transmission depends on modified remote direct memory access (RDMA, Koop et al., 2007). Flexible one-to-many data delivery is realized by extending multicast protocols, and details will be provided in Section 3.

2.2 Comparison with other architectures

There is some other work ongoing related to the CAN. Dynamic-anycast in compute first networking (CFN-dyncast) was proposed by Liu et al. (2021) to solve the problem of service timeout caused by load imbalance across edge servers. This work focuses mainly on service scheduling in multi-access edge computing environments. The corresponding scenario of the work is one of the CAN's target problems; in addition, the CAN considers flexible one-to-many data distribution and wide-area high-throughput transmission. Compared with CFN-dyncast, the CAN has a larger problem space and the architecture is more systematic. A recommendation (ITU-T, 2021) has published the functional architecture of a computing power network. Relative key technologies and application scenarios were proposed by Tang et al. (2021). The computing power network was primarily designed for the next-generation telecommunication network. It introduces a computing power trading platform to realize efficient computing scheduling between providers and consumers. A comprehensive comparison between different architectures is presented in Table 1.

3 Key technologies of CAN

This section lists three key technologies that are crucial for building a CAN system and are coordinated to fulfill the CAN's core functionalities, i.e., CATS, elastic broadcast, and wide-area high-throughput transmission. The awareness functions play a pivotal role in

Table 1 Comparison of computing-aware network (CAN) with related architectures

Perspective for comparison	Computing-aware network	Computing power network	Dynamic-anycast in compute first networking (CFN-dyncast)
Scenarios	Internet	Next-generation telecommunication network	Internet
Structures and layers	Infrastructure layer	Infrastructure layer, management and orchestration layer, and operation and service layer	Infrastructure layer
Technologies and protocols	Design specific network and transport layer protocols for three technologies, including BGP (Rekhter et al., 2006) extension for computing-aware traffic steering, BIER (Dolganow et al., 2017) extension for elastic broadcast, and modified RoCEv2 (InfiniBand Trade Association, 2014) for wide-area high-throughput transmission	Present an overall functional architecture, but lack specific and detailed protocol design	Establish segment routing over IPv6 tunnel between ingress and egress routers to forward network traffic to appropriate service instances
Core ideas	Introduce an awareness plane to improve the interdependence of network and computing resources. Form a closed-loop system to improve the overall system performance	Introduce a computing power trading platform to realize efficient computing power scheduling	Focus on multi-access edge computing. Realize efficient edge service scheduling through computing awareness and dynamic anycast

BGP: border gateway protocol; BIER: bit index explicit replication; RoCEv2: remote direct memory access over converged Ethernet version 2

CATS and elastic broadcast, which is reflected in leveraging the computing as well as network information to generate routing policies. The corresponding scenarios are that CATS follows a distributed manner to select a path towards an appropriate compute instance, while elastic broadcast follows centralized control for data distribution. Wide-area high-throughput transmission, as the basis for large volumes of data transmission, is critical for enhancing wide-area data transmission capabilities, especially for applications that need reliable bulk data transfer. This technology extensively improves the applicability of CAN.

3.1 Computing-aware traffic steering

CATS is a technical scheme for computing service scheduling across multiple compute instances deployed at different edge sites. The core function is to find an optimal path by comprehensively analyzing computing capabilities and network status, which can be mapped into the awareness plane of the architecture. This paper proposes a novel solution scheme for CATS. There is an ongoing working group with the same name of CATS in Internet Engineering Task Force (IETF, https://datatracker.ietf.org/wg/cats/about/). The working group currently focuses mainly on progressing a consensus on the problem statement, use cases, and requirements (Yao KH et al., 2024). This paper proposes the first clear technical solution for CATS.

Scheduling a large number of service requests across multiple edge nodes usually faces the problem that the closest edge node may not be the best node for processing the service request. Since the computing resources deployed at a single edge site are constrained, long queueing latency often exists at one edge site when service scheduling is busy (Ali-Eldin et al., 2021). In addition, link congestion may be a root cause of load imbalance. In this case, the optimal results cannot be obtained by only one-sided scheduling of network or computing resources. It is necessary to further consider how to better match the end-to-end network and computing resources with the service requirements.

There are two potential ways to realize this, i.e., an application-based method and a network-based method. For the former, applications need to obtain the network status. However, network service providers may not expose the detailed network information; thus, applications need to detect the network information by themselves, e.g., by using ping (https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/ping). Moreover, even if applications could obtain the accurate network information, they would use a controller to gather both the computing and network information, and then calculate and choose the best service site and the corresponding network path, which is difficult and inefficient. For the latter, the network needs to obtain the computing status. It may not use a controller to perform the calculation; instead, it can use network elements to distributedly collect both the network and computing information, and then steers the traffic when it receives the service request.

The key challenge of CATS is to balance the control accuracy and delay overhead for different service requirements. In this case, the convergence time of table lookup for packet forwarding should be considered. Distributed control has better processing efficiency than centralized methods, and it is more suitable for the dynamic scheduling of delay-sensitive services. Distributed control adopts the “on-path” mechanism; that is, the edge router directly steers the service request to the target compute node without obtaining the destination address from an external component, i.e., the domain name system (DNS) server or the load balancer. In this way, the extra signaling overhead of querying the destination address is eliminated. Especially for multi-target content acquisition services, long application layer redirection and database query time can be saved.

The distributed model is shown in Fig. 2. The potential risk is the overload of the network by signaling the computing resource status, so the rate of signaling and the kind of granularity of the computing

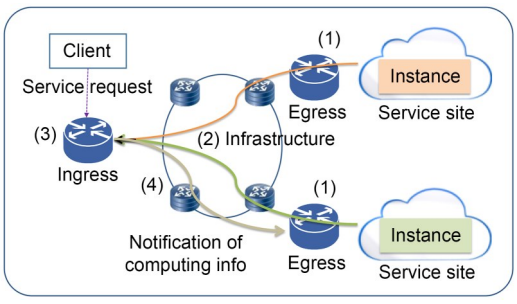

Fig. 2 Computing-aware traffic steering based on distributed control

information should be evaluated. It is expected that only the edge router will need to be updated to support the computing status collection and distribution.

In the first step, edge egress routers collect the computing information of the connected compute nodes. Then, they send the message to the edge ingress router connected to the client. The edge ingress router maintains the routing information table, which contains the service ID (anycast address), host ID (unicast address), and computing information received from edge egress routers. In addition, the edge ingress router collects real-time network status information. After receiving the service request, the edge ingress router makes a comprehensive decision about the optimal path and compute node selection, according to the information collected about computing resources and network status. The message is forwarded to the target compute node in the common routing mode.

There are several methods for distributing the computing status from the edge sites to every edge router. Resource joint awareness encapsulates computing status information in routing protocols and distributes it to service demanders along with the network path. Here, we use the border gateway protocol (BGP, Rekhter et al., 2006) as an example.

BGP update messages are used for inter-domain routing, and they can notify about network information changes, including topology changes. At present, there is related work proposing to use the internal border gateway protocol (iBGP) extension to announce "reckoned computing information" (Dunbar et al., 2024). The computing load information can be directly added to the path attributes of update messages to extend the mechanism for the notification of computing status changes. Fig. 3 shows the protocol extension, which works in the single network domain. The CAN is proposed to work in both a single domain and multi-domains, so more enhanced work should be carried out.

The first is to distribute the computing status in a specific domain group, which is to divide the notification domain, announce information, narrow down the scope of invalid notification, and reduce invalid notification information, including notification domain division, notification domain identification, and notification domain adjustment. The second is to consider the adaptive computing status announcement based on notification domain settings, considering factors such as the frequency of updates on computing status and service popularity to achieve accurate notifications. The third is to specify the address family. Not all the applications need the CATS service. For the services that need it, their corresponding service address can be divided into a CATS address family by the supported edge network node. Thus, the compute status needs only to be distributed to the address family.

As a supplement, the extension of interior gateway protocol (IGP) (Savage et al., 2016) would affect all routers within the domain, which could lead to unavoidable route flapping and is more complex. Further research is needed to solve this issue.

3.2 Elastic broadcast

Elastic broadcast is designed mainly to accommodate the one-to-many collective communication (Chan et al., 2007) pattern that is primarily used in parallel computing. The technology needs to take into account the network device information, primitives, network topology, and compute instances, so as to build a specific compute group and assign tasks. It can be used for AI model training and inference across data centers. There has been no previous work on flexible

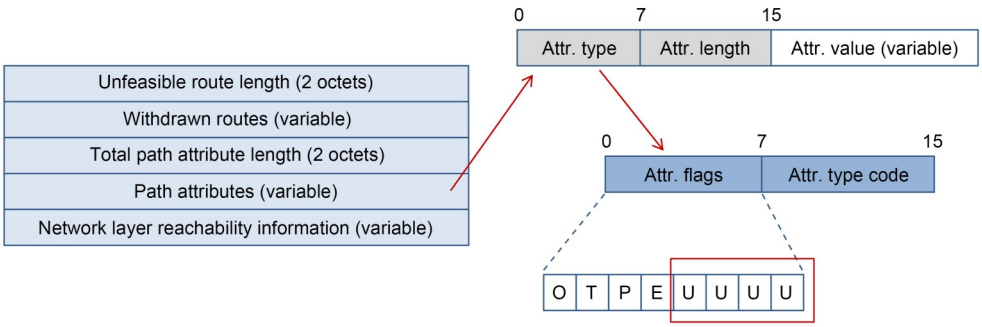

Fig. 3 Border gateway protocol (BGP) extension (Attr.: attribute)

one-to-many data delivery in WANs. This paper provides a novel stack and protocol design for implementing elastic broadcast.

AI model training and inference are generally currently employed in data centers. Since there is a controversial issue where the total size of models is still growing while the power consumption of a single data center will meet a bottleneck, there will be geographically distributed deployment of these applications. The major problem is that the point-to-point implementation of collective communication across multiple compute nodes to transmit a large number of model parameters will introduce a lot of bandwidth occupancy, data movement, and data copies, which will inevitably generate huge overhead. This problem can be amplified in a one-to-many data delivery scenario, since one-to-many collective operations are frequently performed in AI model training systems (Chunduri et al., 2018). There are similar issues that happen in data centers. A previous solution (Li et al., 2024) for implementing one-to-many collective communication is based mainly on extending the multicast forwarding tree. However, this is unacceptable in wide-area scenarios, since it will introduce unacceptable overhead when maintaining forwarding status in every tree node. Optimizing the issue in WANs is more difficult because it is hard to balance the cost of forwarding status maintenance and function flexibility.

The state-of-the-art research on AI model training and inference in WANs focuses only on the optimization of computing patterns. Recent research (Yuan et al., 2022) proposes only optimization strategies on parallel policy and scheduling algorithms, without considering how parallel computing patterns can be co-designed with the underlying network. There is still large overhead in the underlying communication. As for one-to-many communication patterns, existing multicast protocols are not enough for extension because of huge overhead in forwarding status maintenance and single functionality. To the best of our knowledge, this is the first work that optimizes WAN-grade one-to-many collective communication by redesigning specific network protocols.

Fig. 4 shows the deployment of the solution to realize elastic broadcast. Since most of the implementations of collective communication are pre-configured, we extend the network controller to build the actual compute group. Compute instances at endpoints can talk to the network controller by setting up a worker stub locally. The network controller grabs both the computing and network information, including network topology, and network devices' information and their compute capabilities, like aggregation, tailoring, stitching, and duplication.

The illustration of data scattering, one of one-to-many collective communication modes, is shown in Fig. 4. Scattering means that the sender sends different data pieces to different receivers separately. If the collective operation is broadcasting, the sender would send the same data copies to multiple receivers. In Fig. 4, compute instance 0 sends out one piece of data called data_ori, and compute instances 1, 2, and

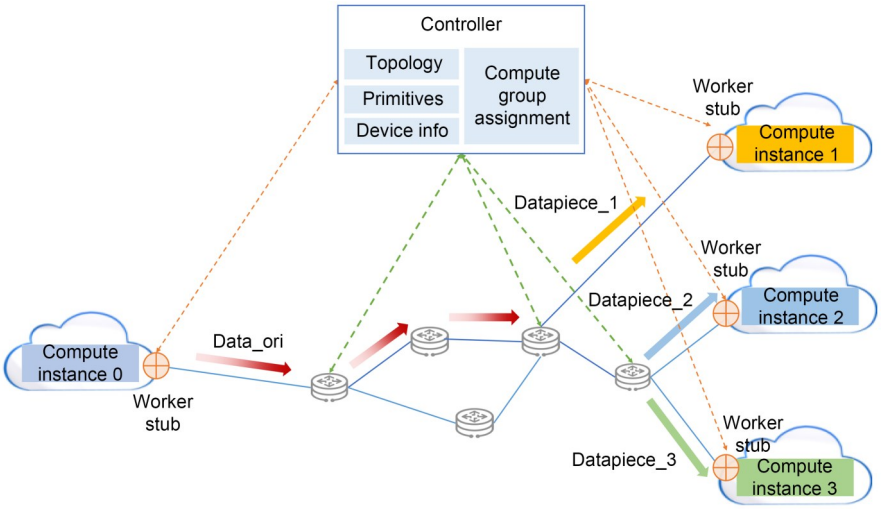

Fig. 4 Illustration of elastic broadcast

3 receive datapiece_1, datapiece_2, and datapiece_3, respectively. Traditionally, if we want to realize scattering, compute instance 0 should send the data piece three times. Our solution is to extend bit index explicit replication (BIER) (Dolganow et al., 2017) to fulfill the job. A major advantage of BIER is that it is stateless, and it does not need to maintain much multicast state information in the forwarding table. However, the standard BIER protocol supports only the broadcasting collective operation, which is not flexible. We insert the Payload length and Payload fields after BitString, to mark the specific data piece that belongs to the receiver that the BitString designates. The detailed protocol design is shown in Fig. 5.

In summary, the proposed solution is flexible for implementing multiple one-to-many collective operations, by copying or tailoring data inside the network forwarding node, according to the network device's capabilities. The advantages are straightforward. A lot of bandwidth occupancy can be saved because of data deduplication during transmission. In addition, the optimization can reduce many of the data copies at endpoints. Previous unicast transmission mode needs N data copies to send N pieces of data to different receivers; our solution needs only one copy to finish the job. This will release additional computing resources for other computational tasks or simply save energy consumption.

3.3 Wide-area high-throughput transmission

Wide-area high-throughput transmission is critical for building high-performance data plane functionalities and extending the CAN's applicability. Take e-commerce applications as an example. The collected data need to first be transmitted to the remote cloud for user recommendation model training, and then the model is deployed to the front-end for inference so as to realize the customized recommendation. Wide-area high-throughput data transmission is an important guarantee. This paper proposes a design framework of a transport protocol for improving wide-area transmission throughput.

There are two open issues in terms of implementing high-throughput data transmission:

1. Wide-area congestion control algorithms based on packet loss and round-trip time (RTT) to adjust the sending rate lead to low bandwidth utilization and inaccurate rate adjustment. Both Tahoe (Kaj and Olsén, 2001) and CUBIC (Ha et al., 2008) adjust the transmission rate based on packet loss, but the high packet loss rate limits the realization of high throughput. If the wide-area data throughput rate is to reach 10 Gb/s, the packet loss rate should be lower than 10^{-10} (Kurose, 2001), which is not realistic. Congestion control algorithms based on RTT can achieve higher throughput, such as BBR (Cardwell et al., 2016) and Copa (Arun and Balakrishnan, 2018), but due to the delay asymmetry of WAN's bidirectional links, using the sum of bidirectional link delay to adjust the one-way data transmission rate will lead to an inaccurate result, which is difficult to maximize the use of link bandwidth.

2. The processing efficiency of server-side CPU limits the data transmission rate. Traditional data transmission mechanisms based on socket (TCP/IP stack) require CPU kernels to parse and encapsulate packets, which consumes a large amount of CPU resources. RDMA allows applications to directly read and write data to the network interface card, realizing kernel bypass, greatly reducing CPU consumption, and improving transmission bandwidth. RDMA over converged Ethernet version 2 (RoCEv2) has been widely used in data center networks, but its use in WANs is faced with the following technical challenges:

The native RDMA packet loss re-transmission mechanism (Go-Back-N) is sensitive to packet loss, and a small amount of packet loss will cause a significant decrease in throughput, which cannot be adapted to WAN scenarios.

Existing wide-area congestion control algorithms cannot achieve high-throughput data transmission, which makes it difficult to match the RDMA transmission rate.

To solve the above problems, this paper designs a wide-area high-throughput transmission scheme

Fig. 5 Bit index explicit replication (BIER) extension to support data scattering

based on RoCEv2, which realizes end-to-end high-throughput data transmission through the network and end collaboration. The new congestion control algorithm can improve the throughput at the network side, while the RDMA protocol can improve the data transmission throughput at the end side. Both of them are independent and can also be realized together.

Fig. 6 shows the three key designs of the new protocol, which are as follows:

1. Fast packet loss recovery. When the packet loss rate is low, the forward error correction mechanism (Baldantoni et al., 2004) is introduced to rectify the packet loss and improve the transmission throughput. For example, Reed–Solomon code (Xiao et al., 2013) is used to group K data packets and generate R redundant data packets according to K data packets, which together form a packet with length N. In this packet, packet recovery can be realized when the number of lost packets is smaller than or equal to R. The values of K and R must take into account the bandwidth and delay product, receiver side buffer, and link transmission rate.

2. Precise re-transmission of lost packets. When a large number of packets are lost continuously on a link, the packet loss recovery algorithm cannot be used to recover packets. Therefore, data integrity must be ensured by re-transmission. The precise re-transmission of lost packets is an improvement on the Go-Back-N mechanism. The receiver uses the packet sequence number to accurately locate the sequence number of late packets and combines the packet loss timer and link bandwidth delay product to determine whether packets are lost. After packet loss is confirmed, the packet sequence number of the lost packet is sent back to the sender by a negative acknowledgement packet message, realizing precise re-transmission of packet loss. The precise packet loss re-transmission reduces the flow completion time and improves the link throughput rate.

3. Optimized congestion control algorithm based on one-way delay. According to the one-way delay and available bandwidth in the same direction of data transmission, the congestion status is determined comprehensively, and the data transmission rate is dynamically adjusted. Compared with the congestion control algorithm based on RTT, the control is more accurate and the optimization of the link bandwidth is better.

The wide-area high-throughput transmission scheme is proposed to achieve high throughput on the network side and save the computing resources of end devices at the same time, which provides a guarantee of high-performance interconnection for CAN.

4 Simulations and applications

4.1 Preliminary test of wide-area high-throughput transmission

The CAN is composed of three key technologies to solve three major problems, to improve the quality of computing services over WANs. We conducted some preliminary simulations to show how to improve the effective throughput.

The simulation of wide-area transmission was based on a network emulator. We measured the throughput of standard TCP with iperf3 (https://iperf.fr/) testing tools, and tested our solution with field programmable

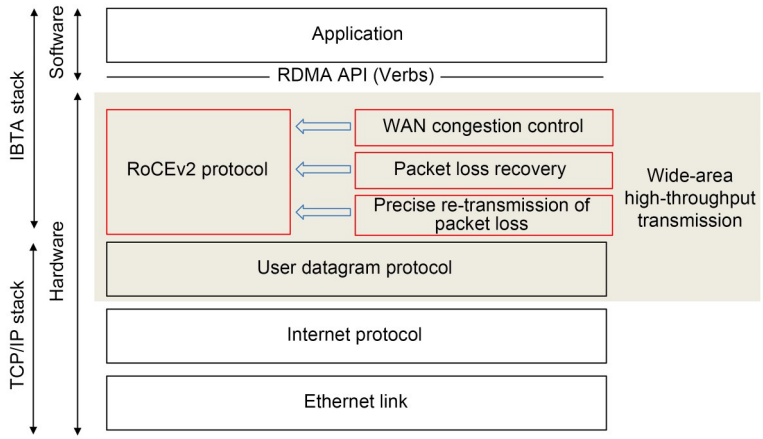

Fig. 6 Wide-area high-throughput transmission protocol stack

gate array (FPGA), under multiple test portfolios, by changing the packet loss rate and RTT. The packet loss rate was set to 0%, 0.05%, 0.10%, 0.15%, 1%, and 10%. RTT was set to 0.02, 1, 5, 20, and 50 ms. The throughput test results of different technologies are shown in Fig. 7. With the increase of WAN delay determined by transmission distance and packet loss rate, the throughput of standard TCP decreases significantly, as shown in Fig. 7a. Our solution is relatively stable, as shown in Fig. 7b. For example, when the RTT is 20 ms (meaning that the transmission distance is around 2000 km) and the packet loss rate is around 0.15%, the throughput of our solution is 8.09 Gb/s, while that of the TCP-based solution is only 0.008 74 Gb/s; the throughput performance of our solution is 925.6 times that of TCP's performance. These preliminary simulation results clearly indicate the improvement of the performance of highly effective throughput transmission.

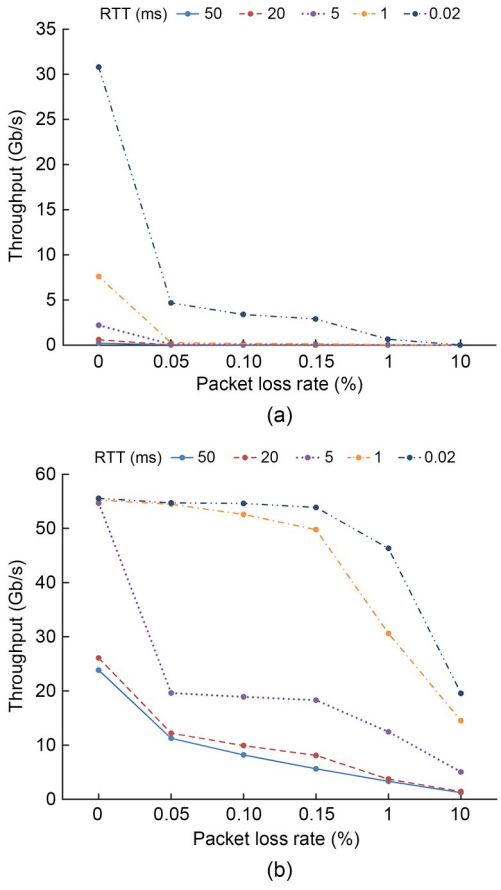

Fig. 7 Performance of transmission control protocol (TCP) based solution (a) and performance of wide-area high-throughput transmission (b)

4.2 Applying CAN to AI model training, inference, and offline transmission

AI foundation models are rapidly growing, and the total number of parameters of generative pre-trained Transformer 3 (GPT-3) (Zong and Krishnamachari, 2022) is reported to be at the level of hundreds of billions, which has very high requirements for large-scale computing capabilities. The deployment of AI model training and inference infrastructure will be geographically distributed in WANs in the future, due to the bottleneck of single data center power consumption and some specific requirements raised by cross-entity collaborative AI training or inference.

CAN's key technologies fit quite well to optimize AI services in WANs. The lifecycle of AI services is shown in Fig. 8. The centralized task scheduler preconfigures the model training. After the training phase, well-trained models are deployed into each edge node for processing different inference service requests. At the last stage, inference information will be fed back for the next round of model training.

Elastic broadcast can optimize model training. During model training, the primary collective operations are Allreduce, Broadcast, All-to-all, and Scatter, covering one-to-many and many-to-one communication patterns. One-to-many operations can be optimized accordingly.

CATS is used for model inference. Model inference is latency-sensitive, especially for high-mobility terminals. There is a need to dynamically schedule optimal edge nodes for processing. Rather than being initiated by a centralized task scheduler, inference service requests are initiated by end devices and are notified to the routing decision-maker to select the current optimal path towards a specific edge node. The decision is based on the awareness of edge nodes' computing information and network link status.

Wide-area high-throughput transmission is used for offline model deployment and parameter updates. In the offline mode, a large number of model parameters need to be sent across distant cloud sites and edge sites, which incurs a lot of latency overhead and bandwidth occupancy. Highly effective throughput transmission could help ease this situation by ensuring a high-performance bulk data transfer while maintaining reliable data transmission.

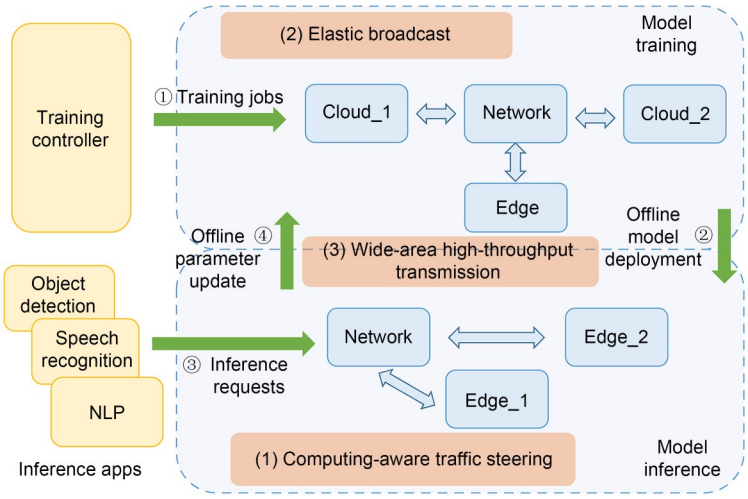

Fig. 8 Use case: centralized model training and distributed model inference based on CAN (CAN: computing-aware network; NLP: natural language processing)

The application of AI large model training and inference demonstrates that, under the framework of the CAN, three main stages within the lifecycle of AI services can be optimized by the three key technologies of the CAN, respectively. This is a clear indication of the potential value and applicability of the CAN.

5 Future work

This paper proposes the CAN architecture, a novel systematic design of computing and network convergence, to satisfy more critical service requirements. Besides the three enabling technologies, there are some directions in which the work can be extended, as follows:

1. Energy efficiency of computing and network convergence. This paper focuses mainly on improving service performance by proposing three key technologies to solve the three main problems that arise when providing computing services over WANs. Besides performance, it is worthwhile to investigate how to reduce energy consumption. According to the work of the International Energy Agency (IEA, 2024), data centers' total electricity consumption could reach more than 1000 TWh in 2026 and will keep increasing in the future. The total number of AI foundation model parameters will reach the trillion level and even more, which is indeed a threat to the power consumption of a single data center. It is necessary to explore scheduling strategies of computing and network resources from a global perspective to achieve the global optimal energy efficiency.

2. The convergence of computing, network, and applications. The methodology of co-designing computing and network resources is built on the condition that computing and network service providers are willing to share information mutually. To evolve sustainably, the convergence should involve applications for maximum system optimization. There are some ongoing attempts in terms of the collaboration between network and applications (Arkko et al., 2023), and also between computing and network in the IETF CATS working group. In the future, we will investigate how to systematically leverage all three aspects; for example, CATS can be extended to support application awareness, and more accurate routing policies can be generated when the decision node can integrate network status, service requirements, and computing information.

Contributors

Xiaoyun WANG, Xiaodong DUAN, and Tao SUN designed the research and the system architecture. Kehan YAO, Peng LIU, and Hongwei YANG drafted the paper. Hongwei YANG and Zhiqiang LI designed the simulations and processed the data. Tao SUN helped organize the paper. Xiaoyun WANG, Xiaodong DUAN, and Tao SUN revised and finalized the paper.

Conflict of interest

Xiaoyun WANG is the editor-in-chief of this special feature and Tao SUN is an executive lead editor of this special feature; they were not involved with the peer review process of this paper. All the authors declare that they have no conflict of interest.

References

Ali-Eldin A, Wang B, Shenoy P, 2021. The hidden cost of the edge: a performance comparison of edge and cloud latencies. Proc Int Conf for High Performance Computing, Networking, Storage and Analysis, Article 23. https://doi.org/10.1145/3458817.3476142

Arkko J, Hardie T, Pauly T, et al., 2023. Considerations on Application-Network Collaboration Using Path Signals. RFC 9419, RFC.

Armbrust M, Fox A, Griffith R, et al., 2010. A view of cloud computing. *Commun ACM*, 53(4):50-58. https://doi.org/10.1145/1721654.1721672

Arun V, Balakrishnan H, 2018. Copa: practical delay-based congestion control for the Internet. Proc 15th USENIX Symp on Networked Systems Design and Implementation, p.329-342. https://doi.org/10.1145/3232755.3232783

Baldantoni L, Lundqvist H, Karlsson G, 2004. Adaptive end-to-end FEC for improving TCP performance over wireless links. Proc IEEE Int Conf on Communications, p.4023-4027. https://doi.org/10.1109/ICC.2004.1313306

Cardwell N, Cheng YC, Gunn CS, et al., 2016. BBR: congestion-based congestion control: measuring bottleneck bandwidth and round-trip propagation time. *Queue*, 14(5):20-53. https://doi.org/10.1145/3012426.3022184

Chan E, Heimlich M, Purkayastha A, et al., 2007. Collective communication: theory, practice, and experience. *Concurr Comp Pract Exper*, 19(13):1749-1783. https://doi.org/10.1002/cpe.1206

Chunduri S, Parker S, Balaji P, et al., 2018. Characterization of MPI usage on a production supercomputer. Proc Int Conf for High Performance Computing, Networking, Storage and Analysis, p.386-400. https://doi.org/10.1109/sc.2018.00033

Clos C, 1953. A study of non-blocking switching networks. *Bell Syst Tech J*, 32(2):406-424. https://doi.org/10.1002/j.1538-7305.1953.tb01433.x

Dolganow A, Przygienda T, Aldrin S, et al., 2017. Multicast Using Bit Index Explicit Replication (BIER). RFC 8279, RFC.

Dunbar L, Malis A, Jacquenet C, et al., 2024. Dynamic Networks to Hybrid Cloud DCs: Problems and Mitigation Practices-Draft-Ietf-Rtgwg-Net2cloud-Problem-Statement-37. IETF.

Gibson D, Hariharan H, Lance E, et al., 2022. Aquila: a unified, low-latency fabric for datacenter networks. Proc 19th USENIX Symp on Networked Systems Design and Implementation.

Ha S, Rhee I, Xu LS, 2008. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Oper Syst Rev*, 42(5): 64-74. https://doi.org/10.1145/1400097.1400105

IEA, 2024. Electricity 2024: Analysis and Forecast to 2026. Available from https://www.iea.org/reports/electricity [Accessed on Feb. 5, 2024].

InfiniBand Trade Association, 2014. Supplement to InfiniBand Architecture Specification Volume 1 Release 1.2.2 Annex A17: RoCEv2 (IP Routable RoCE).

ITU-T, 2021. Y.2501: Framework and Architecture of Computing Power Network. Draft Recommendation ITU-T. Available from https://handle.itu.int/11.1002/1000/14768 [Accessed on Feb. 5, 2024].

Kaj I, Olsén J, 2001. Throughput modeling and simulation for single connection TCP-Tahoe. *Teletraffic Sci Eng*, 4:705-718. https://doi.org/10.1016/S1388-3437(01)80163-3

Kind A, Dimitropoulos X, Denazis S, et al., 2008. Advanced network monitoring brings life to the awareness plane. *IEEE Commun Mag*, 46(10):140-146. https://doi.org/10.1109/mcom.2008.4644132

Koop MJ, Jones T, Panda DK, 2007. Reducing connection memory requirements of MPI for InfiniBand clusters: a message coalescing approach. Proc 7th IEEE Int Symp on Cluster Computing and the Grid, p.495-504. https://doi.org/10.1109/CCGRID.2007.92

Kurose JF, 2001. Computer Networking: a Top-Down Approach. Pearson, UK.

Li WX, Zhang JY, Liu YF, et al., 2024. Cepheus: accelerating datacenter applications with high-performance RoCE-capable multicast. Proc IEEE Int Symp on High-Performance Computer Architecture.

Liu B, Mao JW, Xu L, et al., 2021. CFN-dyncast: load balancing the edges via the network. Proc IEEE Wireless Communications and Networking Conf Workshops, p.1-6. https://doi.org/10.1109/WCNCW49093.2021.9420028

Mao YY, You CS, Zhang J, et al., 2017. A survey on mobile edge computing: the communication perspective. *IEEE Commun Surv Tutor*, 19(4):2322-2358. https://doi.org/10.1109/COMST.2017.2745201

Rekhter Y, Li T, Hares S, 2006. A Border Gateway Protocol 4 (BGP-4). RFC-4271, RFC.

Savage D, Ng J, Moore S, et al., 2016. Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP). RFC 7868, RFC.

Singhvi A, Akella A, Gibson D, et al., 2020. 1RMA: re-envisioning remote memory access for multi-tenant datacenters. Proc Annual Conf of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, p.708-721. https://doi.org/10.1145/3387514.3405897

Stoica I, Shenker S, 2021. From cloud computing to sky computing. Proc Workshop on Hot Topics in Operating Systems, p.26-32. https://doi.org/10.1145/3458336.3465301

Su JS, Zhao BK, Dai Y, et al., 2022. Technology trends in large-scale high-efficiency network computing. *Front Inform Technol Electron Eng*, 23(12):1733-1746. https://doi.org/10.1631/FITEE.2200217

Tang XY, Cao C, Wang YX, et al., 2021. Computing power network: the architecture of convergence of computing and networking towards 6G requirement. *China Commun*, 18(2): 175-185. https://doi.org/10.23919/jcc.2021.02.011

Xiao JM, Tillo T, Zhao Y, 2013. Real-time video streaming using randomized expanding Reed–Solomon code. *IEEE Trans Circ Syst Video Technol*, 23(11):1825-1836. https://doi.org/10.1109/TCSVT.2013.2248235

Yao HP, Mai TL, Jiang CX, et al., 2019. AI routers & network mind: a hybrid machine learning paradigm for packet routing. *IEEE Comput Intell Mag*, 14(4):21-30. https://doi.org/10.1109/mci.2019.2937609

Yao KH, Trossen D, Boucadair M, et al., 2024. Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements: Draft-Ietf-Cats-Usecases-Requirements-02. IETF.

Yuan BH, He YJ, Davis J, et al., 2022. Decentralized training of foundation models in heterogeneous environments. Proc 36th Int Conf on Neural Information Processing Systems.

Zong MY, Krishnamachari B, 2022. A survey on GPT-3. https://arxiv.org/abs/2212.00857