XRON: A Hybrid Elastic Cloud Overlay Network for Video Conferencing at Planetary Scale

Bingyang Wu
Peking University and Alibaba Cloud

Kun Qian
Alibaba Cloud

Bo Li
Alibaba Cloud

Yunfei Ma
Alibaba Cloud

Qi Zhang
Alibaba Cloud

Zhigang Jiang
Alibaba Cloud

Jiayu Zhao
Alibaba Cloud

Dennis Cai
Alibaba Cloud

Ennan Zhai
Alibaba Cloud

Xuanzhe Liu
Peking University

Xin Jin
Peking University

ABSTRACT

Quality and cost are two key considerations for video conferencing services. Service providers face a dilemma when selecting *network tiers* to build their infrastructure—relying on Internet links has poor quality, while using premium links brings excessive cost.

We present XRON, a hybrid elastic cloud overlay network for our planetary-scale video conferencing service. XRON differs from prior overlays with two distinct features. First, XRON is hybrid, i.e., it leverages both Internet and premium links to *simultaneously* achieve high quality and low cost. Second, XRON is elastic, i.e., it exploits elastic cloud resources to adaptively scale its capacity based on realtime demand. The data plane of XRON combines active probing and passive tracking for scalable link state monitoring, uses asymmetric forwarding based on heterogeneous bidirectional link qualities, and quickly reacts to sudden link degradations without the control plane involvement. The control plane of XRON predicts video traffic based on application knowledge, and computes global forwarding paths and reaction plans with scalable algorithms. Large-scale deployment in DingTalk shows that XRON reduces video stall ratio and bad audio fluency by 77% and 65.2%, respectively, compared to using Internet links only, and reduces cost by 79%, compared to using premium links only.

CCS CONCEPTS

• **Networks** → **Overlay and other logical network structures**; **Cloud computing**; • **Applied computing** → **Electronic data interchange**;

KEYWORDS

Video conferencing, overlay network, cloud computing, elasticity

ACM SIGCOMM '23, September 10–14, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0236-5/23/09…$15.00
https://doi.org/10.1145/3603269.3604845

ACM Reference Format:
Bingyang Wu, Kun Qian, Bo Li, Yunfei Ma, Qi Zhang, Zhigang Jiang, Jiayu Zhao, Dennis Cai, Ennan Zhai, Xuanzhe Liu, and Xin Jin. 2023. XRON: A Hybrid Elastic Cloud Overlay Network for Video Conferencing at Planetary Scale. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10–14, 2023, New York, NY, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3603269.3604845

1 INTRODUCTION

Video conferencing is becoming increasingly popular and important in our society [12, 33, 38, 49]. We operate one of the largest video conferencing services in the world with hundreds of millions of users. Most of our top-tier business customers are transnational corporations that rely on our service to host international online meetings for regular corporation management and various business events. This calls for a planetary-scale video conferencing service. We deploy video conferencing clusters as containers globally in different cloud regions of our cooperated cloud provider. The geographically distributed deployment ensures our users have access to the nearby video conferencing service. We interconnect our video conferencing clusters across the globe to form the network infrastructure for delivering video conferencing traffic in the wide area network.

Quality and cost are two key considerations for video conferencing services. Cloud platforms typically provide *network tiers* with different cost and performance characteristics. Video conferencing providers face a dilemma when selecting network tiers to build their services. Internet links are cheap, but directly connecting clusters via Internet links is practically far away from satisfactory under the strict quality requirement of video conferencing. Alternatively, cloud providers typically build their own private inter-datacenter networks or rent dedicated links from ISPs, and expose them as a premium network tier. Using premium links can satisfy the need of video conferencing, but the cost is excessive.

We present XRON, the network infrastructure for our planetary-scale video conferencing service. XRON leverages *overlay paths* that forward video conferencing traffic via intermediate cloud regions to achieve better quality than direct paths. XRON is an overlay network for video conferencing with two distinct features—*hybrid* and *elastic*.

First, XRON is a *hybrid* overlay network using both Internet links and premium links. The key problem of Internet links is instability. The latency and loss rate of Internet links can spike significantly in a short period (§2.2). While such sudden spikes are not a problem for throughput-intensive services like bulk data transfers, they can seriously compromise user experience for latency-sensitive video conference services. XRON relies on Internet links most of the time for low cost, and quickly switches Internet links to premium links or other high-quality Internet links during temporary performance degradations for consistent high quality.

Second, XRON leverages *elastic* cloud resources to dynamically scale its capacity based on the traffic demand. The traffic demand of video conferencing changes over time (§2.3). Over-provisioning the overlay network introduces unnecessary cost, while under-provisioning cannot meet the demand at peak hours. XRON adopts the idea of resource elasticity in serverless computing [46]. It alleviates the burden of resource provisioning from the overlay operator. The capacity of the overlay network can be dynamically scaled by adding or removing containers in the cloud. While cloud platforms provide reactive auto-scaling based on resource usage, it is too slow for latency-sensitive video conference services (§2.3). We leverage application knowledge to build a prediction model that can accurately predict future traffic demand of our video conference service. XRON *proactively* scales the resources to avoid service quality degradation due to slow scaling.

XRON exploits hybrid links and elastic cloud resources to achieve *both* high quality and low cost. XRON's principled design consists of a data plane (§4) and a control plane (§5). The data plane includes a set of XRON gateways distributed in the cloud regions to carry video conferencing traffic. The gateways combine sampling-based active probing and passive tracking to achieve *scalable* link state monitoring for the large overlay network (§4.1). Traffic forwarding in the data plane is *asymmetric* in that the two directions of a video stream may use different paths, to exploit heterogeneous bidirectional link qualities (§4.2). The data plane uses a distributed reaction mechanism to quickly update overlay paths locally, in face of sudden link quality degradations (§4.3).

The control plane is a logically centralized controller that decides resource scaling and forwarding paths for the overlay network. The controller leverages a domain-specific prediction model to accurately predict future video traffic demand (§5.1). Based on the global knowledge of the overlay network, the controller uses a scalable two-step control algorithm to compute the number of gateways and their forwarding tables for each region (§5.3). The controller also computes the backup paths for fast data plane reaction (§5.4).

XRON is a production system that has been deployed to support DingTalk video conferencing service since August 2022. The overlay nodes of XRON are deployed in eleven Alibaba Cloud regions across the globe. Online production statistics show that XRON reduces video stall ratio and bad audio fluency by 77% and 65.2%, respectively, compared to using public Internet links only. XRON reduces the cost by up to 79%, compared to using premium links only.

Overlay networking is a canonical topic studied by our community for decades [6, 21, 24, 33, 37, 42, 43, 51, 56, 57]. We remark that besides our engineering contributions in building and deploying a planetary-scale system in the real world, the key technical contribution of XRON is the design of a *hybrid* and *elastic* overlay, which makes XRON different from prior overlay networks. Early overlays such as RON [6] employ *only* Internet links for resiliency. This is not sufficient for fast reaction to sudden performance degradations. Naively updating the overlay path based on the global view is an order of magnitude slower than what video conferencing requires. XRON uses *hybrid* network resources to fast react to performance degradations. In terms of elasticity, while there are recent cloud overlays (e.g., Skyplane [29]) leveraging elastic cloud resources, the objective is different: XRON is designed for latency-sensitive video conferencing services, while Skyplane is for throughput-intensive bulk data transfers. This introduces a different set of design constraints and solutions, including prediction-based proactive resource scaling, scalable real-time link state monitoring, and fast distributed data plane reaction.

This work does not raise any ethical issues. Data is anonymized and approved by our institute's review board. It was collected as part of routine operations, and only used for the performance optimization of our video conferencing system.

2 BACKGROUND AND MOTIVATION

In this section, we first provide background on our video conferencing service, DingTalk (§2.1). Then, we introduce the choices of heterogeneous network resources, and their *pros* and *cons* (§2.2). Finally, we describe the dynamic demand of video conferencing traffic (§2.3).

2.1 Cloud-Native Video Conferencing

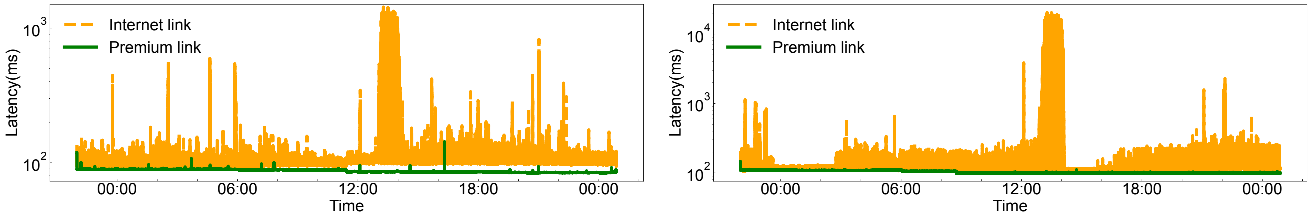
Building and deploying software services on cloud platforms has become a widely-adopted industry practice. The cloud-based approach is particularly appealing for planetary-scale services, as it enables these services to expand to a global scale without the frustrations of building and operating hardware infrastructure in each nation. To leverage these benefits, we work closely with a large public cloud provider, and build our video conferencing service entirely on the public cloud.

We deploy video conferencing clusters globally in multiple cloud regions of the public cloud to ensure service proximity to users. Each cluster consists of a set of containers that perform video transcoding, audio mixing, and video transmission. Users install our video conferencing client software on their devices like mobile phones, laptops, and desktops. The functionalities of the client software include video encoding/decoding, denoising, image enhancing, and video stream transmission. A client is connected to its closest cluster for participating in a video conference. For cross-region video conferences, multiple clusters are involved, and they transmit video streams with each other.

2.2 Heterogeneous Network Resources

There are two choices to interconnect our video conferencing clusters across different cloud regions: Internet links and premium links. Below we compare the performance and cost of the two types of links and describe the challenge.

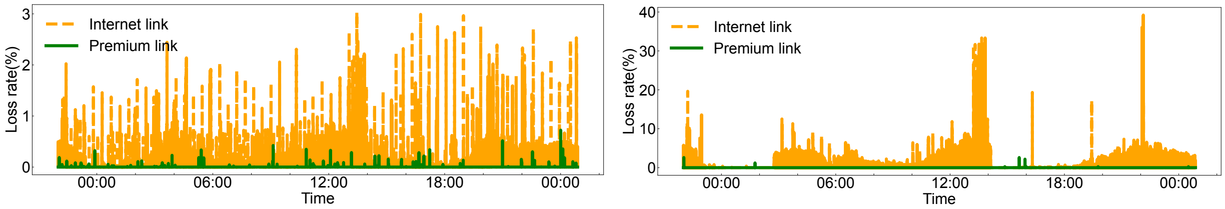
Performance comparison. We conduct a measurement to characterize the performance of the two link types. We measure pairwise latency and loss rate among eleven regions of the cloud, on which



(a) Average latency of all pairs.

(b) Latency of one example pair.

Figure 1: Comparison of latency between Internet links and premium links.



(a) Average loss rate of all pairs.

(b) Loss rate of one example pair.

Figure 2: Comparison of loss rate between Internet links and premium links.

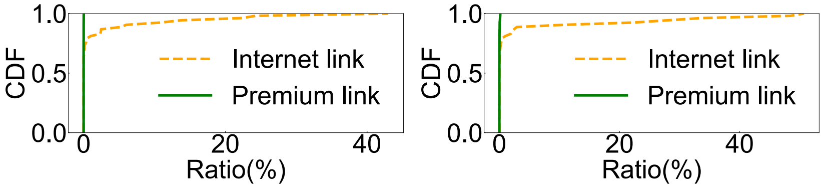
our video conferencing service operates. We probe each pair of regions every second for one day. Figure 1a and Figure 1b show the average latency of all region pairs and the latency of one example pair over time, respectively. Premium links have lower average latency than Internet links. More importantly, the latency of Internet links fluctuates significantly over time, while premium links are more stable. The fluctuation of an individual link latency is more significant than the average of all links. The latency of the example pair goes up to as high as 20518 ms. Stability is particularly important to real-time applications like video conferencing, as a sudden increase in latency can significantly impact user experience.

Figure 2 shows the loss rate results. Similar to the latency results, premium links have a lower average loss rate and less fluctuation than Internet links. Compared to the average, the difference between premium links and Internet links is more significant at the individual link level. The maximum average loss rate in Figure 2a is 3.3%, while the maximum loss rate of the example pair in Figure 2b is 39.2%. The loss rate significantly impacts the fluency of video conferencing applications. When the loss rate is high, lost packets cannot be recovered by error correction codes and it would take a few round-trip times (RTTs) for retransmission, causing video stalls on the user side.

To further compare their performance, we calculate the fraction of time with high latency and high loss rate for each link, and Figure 3 shows the CDF of all links. Based on our operational experience with our video conferencing service, we use the following thresholds: latency > 400ms, loss rate > 0.5%. Note that we do not argue these numbers are the best thresholds to determine whether the latency or loss rate is too high for video conferencing; they are just empirical values used here to illustrate the performance differences between Internet links and premium links. Results show that almost all premium links have a very small fraction of time with latency > 400ms or loss rate > 0.5%. In comparison, Internet links have a very long tail. 20% of Internet links have more than 10% of time with high latency and more than 22% of time with high loss rate.

Cost comparison. Network usage on cloud platforms is priced based on the product of egress data volume and unit egress fee. The unit egress fee varies according to different source regions for Internet links and different source-destination region pairs for premium links. Figure 4 shows the normalized price distribution of Internet links and premium links in our used cloud, respectively. Each price is normalized to the unit egress fee of the most expensive Internet link. The unit egress fees of premium links are significantly higher than those of Internet links: the median cost difference is 7.6× and the maximum is 11.4×.

Challenge. Neither Internet links nor premium links can simultaneously provide high quality and low cost. As a video conferencing service provider, the bandwidth expense contributes to a significant portion (> 60%) of our operating cost. While premium links provide high quality, we cannot afford to use them for cross-region traffic of all users. Earlier versions of our service use premium links for a small portion of users that pay extra subscription fees; other users are served with Internet links. Due to the high cost of premium links, the pricing is not attractive for many users, and the majority of our users experience bad quality for cross-region conferences. This affects the overall user experience and inhibits further user growth of our service.



(a) Latency.

(b) Loss rate.

Figure 3: Comparison of fraction of time with high latency (> 400ms) and high loss rate (> 0.5%) between Internet links and premium links.

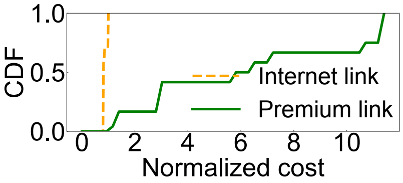
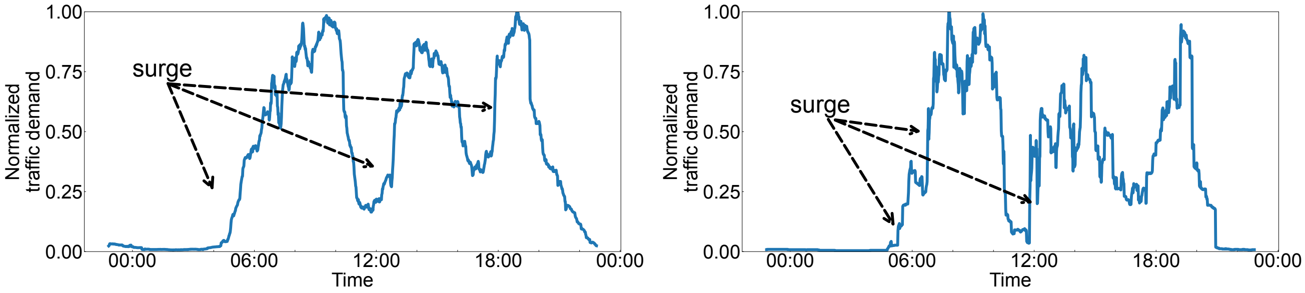


Figure 4: Comparison of cost between Internet links and premium links.



(a) All cross-region traffic.

(b) Traffic of one example pair.

Figure 5: Dynamic video conferencing demand.

2.3 Dynamic Video Conferencing Demand

The traffic demand of our video conferencing service is highly dynamic. Figure 5a shows the total cross-region traffic demand of our service over time for one day. The traffic demand is normalized to the maximum demand of the day. The peak demand is more than 145× compared with the bottom demand. The demand may increase dramatically in a short period. For example, the normalized demand increases from 0.31 at 13:57 to 0.46 at 14:02, which is a 48% increase in five minutes. Next, we zoom in to show the traffic demand of individual region pairs. As different pairs have similar characteristics, we show the demand of one representative pair over a day in Figure 5b. The peak demand is 247× of the bottom demand, and the demand increases by 3.4× in five minutes from 6:27 to 6:32. This indicates that the demand of an individual pair may fluctuate more significantly.

Challenge. Traditionally, service providers build their own physical infrastructure, and face the common problem of over-provisioning or under-provisioning. Our service is built with a containerized, cloud-native approach, and can leverage elastic cloud resources to dynamically scale based on traffic demand. Cloud platforms offer native auto-scaling mechanisms based on CPU, memory, or bandwidth utilization for containers. While starting a container only takes a few seconds, there are a few overheads imposed by cloud platforms, which would enlarge the startup time to minutes. First, preparing a container instance takes tens of seconds, due to the complex orchestration stack of cloud platforms. Second, upon a cache miss when loading the container image, it needs extra time to pull the image from the raw image repository. Third, several common procedures (e.g., IP allocation) shared by the entire cloud platform can be slow during high load of the cloud. Fourth, a series of checking procedures for software and hardware states have to be done before a container is finally ready. As a result, the native reactive auto-scaling mechanisms are too slow for latency-sensitive video conferencing services during demand spikes.

3 XRON OVERVIEW

XRON is a hybrid elastic cloud overlay network for our video conferencing service that achieves *both* high quality and low cost. XRON is *hybrid*, i.e., it uses both Internet links and premium links to build the overlay network. XRON is *elastic*, i.e., it leverages elastic cloud resources to dynamically scale its capacity based on the traffic demand. Figure 6 shows a high-level overview of XRON, which includes a data plane and a control plane. XRON is a containerized system. All components are packaged and deployed as containers and are managed by container orchestration tools based on Kubernetes [2] provided by the cloud platform.

XRON data plane. The data plane consists of XRON gateways, which are responsible for cross-region video stream transmission. Each gateway runs as a container managed by a Kubernetes cluster. Each gateway has a monitoring module that monitors the states of adjacent links in terms of latency and loss rate. Given the scale of XRON, the data plane combines active probing and passive tracking for scalable real-time monitoring (§4.1). The link states are reported to the control plane for forwarding decisions.

The gateways receive forwarding table updates from the control plane. The forward tables specify which is the next hop and whether

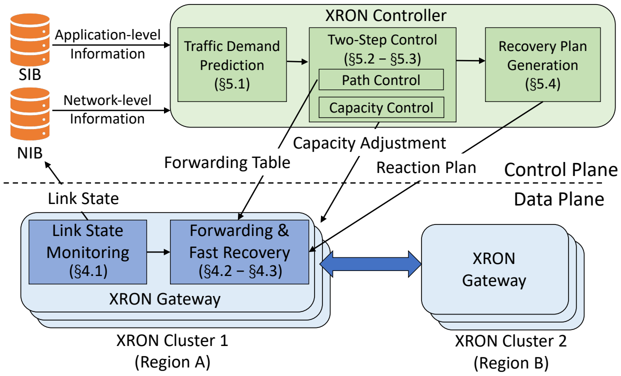

XRON Controller
Application-level Information
SIB
Network-level Information
NIB
Traffic Demand Prediction (§5.1)
Two-Step Control (§5.2 – §5.3)
Path Control
Capacity Control
Recovery Plan Generation (§5.4)
Capacity Adjustment
Forwarding Table
Reaction Plan
Control Plane
Data Plane
Link State
Link State Monitoring (§4.1)
Forwarding & Fast Recovery (§4.2 – §4.3)
XRON Gateway
XRON Gateway
XRON Cluster 1 (Region A)
XRON Cluster 2 (Region B)

Figure 6: XRON overview.

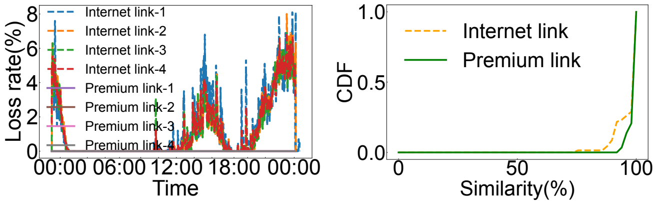
to use the Internet link or the premium link for each video stream. As XRON is an overlay network, each gateway not only transmits streams originating from its own cloud region but also acts as *relays* for streams from other regions. The forwarding in XRON is *asymmetric*, i.e., the two directions of a stream may use different overlay paths (§4.2). This is based on our observation that an overlay link may have different performance in its two directions.

XRON uses a fast distributed reaction mechanism to handle sudden link performance degradations locally (§4.3). When detecting a performance drop of an outgoing link, a gateway switches to a backup link, which is pre-calculated by the controller to avoid capacity overloads and congestions. Compared to the minute-level global control loop, the distributed local reactions can be done in seconds, ensuring consistent high quality for video conferencing sessions.

XRON control plane. The control plane is a logically centralized controller that orchestrates capacity scaling and stream transmission of the overlay network. The control plane contains a network information base (NIB) and a stream information base (SIB). The NIB stores network-level information, including link states reported by XRON gateways and link pricing information fetched from the cloud platform. The SIB stores application-level information for video conferencing streams, including the source, destination and bitrate of each stream and the quality requirements. The controller uses a prediction model based on application knowledge to predict future traffic demand (§5.1). With a global view of network information and stream information, the controller models the objective and the constraints (§5.2), and uses a scalable two-step control algorithm to compute the number of gateways and their forwarding tables for each region (§5.3). The controller also computes the backup paths for fast data plane reaction (§5.4). The controller periodically runs the computation and issues updates to the data plane.

4 XRON DATA PLANE

The data plane of XRON consists of multiple gateways for cross-region video stream transmission. These XRON gateways ensure low-latency, stable transmission via techniques including scalable link state monitoring, (§4.1), asymmetric forwarding (§4.2), and fast distributed reaction (§4.3).



(a) Different links of one pair. **(b) Overall similarity of all pairs.**

Figure 7: Different links in the same region pair usually share similar network conditions.

4.1 Scalable Link State Monitoring

Basic monitoring mechanisms. Each XRON gateway is equipped with a monitoring module that monitors the states of overlay links with other gateways. Link states, in the XRON context, mean latency and loss rate. The module uses a combination of *active probing* and *passive tracking*.

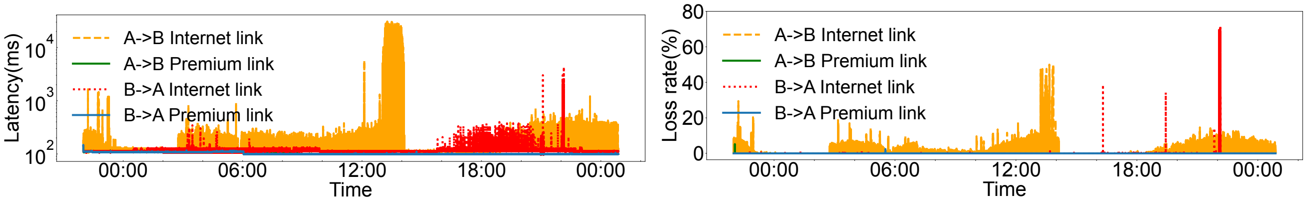
Active probing. To monitor real-time link states, the monitoring module uses pseudo packets to proactively probe the network at a high frequency: about 400 milliseconds per probing burst, and each probing burst contains fifteen 1.5KB packets. Suppose two gateways (G_A and G_B) check the link state (i.e., latency and loss rate) between them via probing. The latency is the transmission time of a probe. A probe is judged as a loss when the following conditions happen: (i) more than twenty succeeding responses are received or (ii) the response does not arrive after three RTTs.

Passive tracking. XRON uses a regular technique (similar to previous efforts [58]) that passively tracks the video conferencing packets to monitor the link states. Only passive tracking is not sufficient for links with no or low traffic. Passive tracking works together with active probing for comprehensive link state monitoring.

Scalable group-based active probing. While our above probing design can detect link degradations timely, its frequency and overhead are, nevertheless, too *aggressive*. Suppose we have N regions and each region has M gateways. The above design needs to probe $N(N-1)M^2$ links. Worse, the overhead increases quadratically with our service capacity M, which seriously affects the scalability of our system.

Based on our experience, we observed that different links in the same region pair share similar network conditions with a high possibility. Figure 7a shows the loss rate of different links from an example region pair during one day. Although each link may encounter bursty loss at a different time and the loss conditions during the bad quality time might be different, all links share a similar variation tendency. Furthermore, we compute the similarity of links' qualities in each region pair, respectively. The similarity is defined as the time proportion where different links share the same quality. As shown in Figure 7b, for all region pairs, links have the same quality in more than 77% of the time. Furthermore, the similarity of links in 80% region pairs is more than 90%.

Driven by the above observation, we group the XRON gateways in a region into a *probing group*, and select R representatives for each region pair to execute full link state probing. The probing results are then aggregated to the XRON controller in a group granularity



(a) Latency

(b) Loss rate

Figure 8: Different link performance in two directions.

for later decision making. The overall detecting and reporting cost, therefore, decrease from $O(N(N-1)M^2)$ to $O(N(N-1)R)$. Due to this grouping-based design, the overall probing cost is negligible in practice.

4.2 Asymmetric Forwarding

Observation: heterogeneous bidirectional link states. Existing global overlay network solutions [6, 29, 33] simply assume the two directions of an overlay link have the same performance. However, asymmetric underlay routing is quite common in practice [35, 50]. Figure 8 shows the link states of two directions of an example region pair. In more than 60% of time, two directions of the Internet links have different states. If we simply use the round-trip performance of links to compute the forwarding plan, the solution space is limited and the forwarding decisions are sub-optimal.

Asymmetric forwarding. At runtime, the XRON gateways forward video conferencing traffic according to their forwarding tables received from the controller. The forwarding tables specify which is the next hop and whether to use the Internet link or premium link for each video stream. Based on the above observation, the monitoring module monitors and reports the states of both directions of each link to the controller, and the controller computes asymmetric forwarding decisions, i.e., the two directions of a video stream may use different forwarding paths and different types of links. For example, suppose we have three regions A, B and C, if only $A \longrightarrow C$ encounters high loss rate, video traffic can be transmitted from A to C through path $A \longrightarrow B \longrightarrow C$, while ACKs can be forwarded back through $C \longrightarrow A$ directly.

4.3 Fast Reaction to Link Degradations

Observation: temporary link degradations are common. Figure 9 shows the duration of performance degradations (latency > 400 ms or loss rate > 0.5%) on all region pairs. Note that the link failure is just an extreme case of performance degradation. The number of short-term (< 30s) performance degradation cases is two orders of magnitude higher than long-term performance degradation cases. Such temporary degradations may result in unavailability for hundreds of RTTs in the worst case, which is unacceptable for video conferencing. When a network degradation occurs, the transient loss rate is too high to be recovered by video redundant coding methods, such as forwarding error correction (FEC) [13].

Fast distributed reaction. A strawman solution is to call the controller to compute new paths when a link degradation is detected by monitoring. The gateways, however, are typically far away from the controller—they can be on different continents. The communication between the controller and the gateways cannot be fast enough to handle the degradation quickly. Given the scale of XRON, frequently reporting degradation events to the controller and waiting for forwarding table updates would introduce unaffordable overhead.

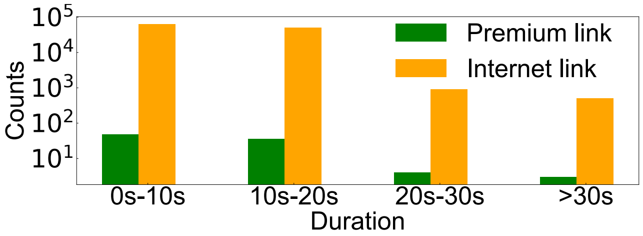
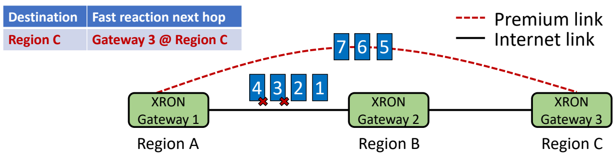


Figure 9: Temporary (≤30s) network performance degradations are common.



Destination	Fast reaction next hop
Region C	Gateway 3 @ Region C

Figure 10: Fast reaction in the data plane.

We propose a fast distributed reaction mechanism to handle temporary link degradations. Every time when the XRON controller computes forwarding tables, it also computes fast reaction plans for each region. The fast reaction plans find the best combination of premium links under capacity constraints (§5.4). As shown in Figure 10, when a temporary degradation of the Internet link $A \longrightarrow B$ occurs, XRON gateways forward the traffic via the premium link $A \longrightarrow C$, without reporting the failure event to the controller. Since the XRON controller is not involved in this control loop, short-term link degradations can be handled within seconds. Because the pricing is based on the transmitted data volume rather than the bandwidth, the cost of this mechanism is cheap enough as it only uses premium links during link degradations, which we also empirically show in §6.3.

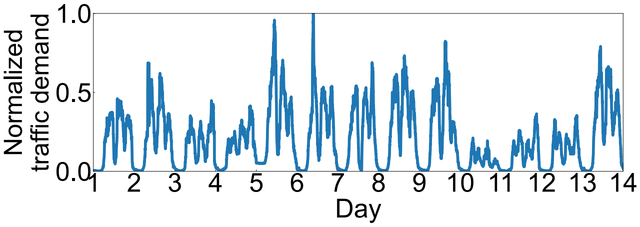


Figure 11: Traffic demand in a region over two weeks.

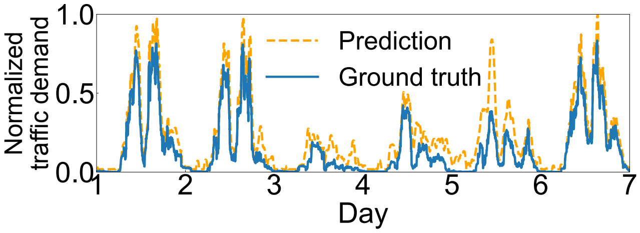


Figure 12: Predicted traffic and actual traffic.

5 XRON CONTROL PLANE

XRON control plane is a logically centralized controller that decides resource scaling and forwarding paths. The controller relies on traffic demand prediction (§5.1) and a set of modeled objective and constraints (§5.2) to compute two results: the number of gateways and their forwarding tables (§5.3), and backup paths for fast data plane reaction (§5.4).

5.1 Traffic Demand Prediction

As revealed in §2.3, handling burst network usage of the video conferencing service is hard, since cloud-native container autoscaling mechanisms cannot meet such fast requirement changes. Depending on our operational experience, reserving five minutes is enough for covering the uncertainty introduced by container provisioning. Therefore, this part focuses on providing precise short-term predictions in the future five minutes.

Observation: our traffic demand has a three-peak pattern. To achieve accurate prediction, we first study the traffic demand pattern. Figure 11 shows a typical traffic pattern of our video conferencing service for a region pair over two weeks. We observe that the traffic has a three-peak pattern every day. With deep diving in Figure 5b, the overall traffic increases greatly after 6:00 and the first peak appears usually at 10:00, where the traffic surges from nearly zero to the highest in a day. Later at 12:00, the traffic falls back, rises again around 14:00, and achieves the second peak at about 16:00. The last phase of increase starts at 18:00 and the peak appears at 20:00. This three-peak pattern also matches our prior knowledge about video conferencing: users tend to have a regular daily routine, and often communicate with others at their work time or study time, which is the main reason for the three-peak pattern. Daily or weekly meetings in the company and scheduled online classes in the school also strengthen the periodicity of traffic demand.

Symbol	Description
V	The set of regions.
N_i	The set of containers in the region i.
N_{max}	The container limit in each region.
$P_{i,j}$	A forwarding path from region i to region j.
$Lat_{i,j}^{t}$	Latency of link type t from region i to region j.
$RTT\_Limit_{i,j}$	Latency limit of path $P_{i,j}$.
$Loss_{i,j}^{t}$	Loss rate of link type t from region i to region j.
$Loss\_Limit_{i,j}$	Loss rate limit of path $P_{i,j}$.
$T_{i,j}$	Traffic demand from region i to region j.
$S_{i,j}^{t}$	$S_{i,j}^{t} = \{(m,n) \mid (i,j,t) \in P_{m,n}\}$
S_i	$S_i = \bigcup_{t \in \{I,p\}} \bigcup_{j \in V} (S_{i,j}^{t} \cup S_{j,i}^{t})$
B_c	Processing capacity of a container.
B_I^{i}	Internet link bandwidth of region i.
C_c	Cost of a container.
$C_I(i)$	Cost of the Internet link per byte of region i.
$C_p(i,j)$	Cost of the premium link per byte from region i to j.

Table 1: Key notations in the problem formulation.

Traffic prediction model. The aforementioned observation shows a strong time sequential law of traffic demand. Driven by the such phenomenon, we propose a Discrete-Time Fourier Transform (DTFT) approach to predict traffic demand. Specifically, DTFT first transforms the time sequence of traffic demand from the time domain to the frequency domain. We choose one hundred most prominent harmonics to fit the pattern and filter out random jitters. Then, we transform the traffic demand sequence back from the frequency domain to the time domain, and the results in the future timestamp are the prediction results. The input data for calculating traffic demand comes from application-level information collected by SIB[1], including video type, traffic source and destination, frame rate, resolution, and bitrate. This DTFT prediction can roughly predict the traffic pattern, we add an empirical rule based on our experience to further improve the accuracy: the predicted demand should be no smaller than the actual demand in the last turn, to reduce the risk of excessive scale-down. In the production environment, there would be rare corner cases that cannot be fully predicted with historical data. The above empirical rule makes XRON tightly track the actual demand. Figure 12 visualizes a real case of our predicted results, which shows that our prediction can efficiently cover the real demand.

5.2 Modeling Objective and Constraints

We model the objective and constraints below. The key notations of the problem formulation are in Table 1.

Network performance constraints. Based on our experience, latency and loss rate are two key factors affecting video conferencing quality, which is also confirmed by previous work [15, 33]. Available bandwidth can also be estimated by using latency and loss rate [6]. We have the following constraints for the end-to-end latency limit $Lat\_Limit_{m,n}$ and loss rate limit $Loss\_Limit_{m,n}$ for path $P_{m,n}$ from region m to region n, respectively. These limits can be configured as the video conferencing service requires, and is well-studied [33].

[1] Because XRON is an overlay network designed and operated by the video conferencing service provider, it can use statistics of video conferencing sessions without harming user privacy or leaking sensitive data.

$$\forall (m,n) \in V^2, Lat(P_{m,n}) = \sum_{(i,j,t)\in P_{m,n}} Lat^t_{i,j} \le Lat\_Limit_{m,n}$$

$$\forall (m,n) \in V^2, 1 - \prod_{(i,j,t)\in P_{m,n}} (1 - Loss^t_{i,j}) \le Loss\_Limit_{m,n}$$

Gateway capacity constraints. First, a XRON gateway is deployed as a container, which has limited processing capacity B_c. The total traffic in region i cannot exceed the total processing capacity of its containers $B_c \cdot N_i$. Second, the total traffic in region i over Internet links cannot exceed the bandwidth limit B^i_I imposed by region i. Third, premium links are charged based on both source and destination regions, and their bandwidth limits are also imposed per region pair, instead of per region. Thus, the constraint becomes that the total traffic over a premium link from region i to region j cannot exceed the bandwidth limit of link $B^{i,j}_d$. Fourth, the number of containers in each region cannot exceed the container quota N_{max} imposed by the cloud platform.

$$\forall i \in V, \sum_{(m,n)\in S_i} T_{m,n} \le B_c \cdot N_i$$

$$\forall i \in V, Thpt_I(i) = \sum_{j\in V} \sum_{(m,n)\in S^I_{i,j}} T_{m,n} \le B^i_I$$

$$\forall (i,j) \in V^2, Thpt_p(i,j) = \sum_{(m,n)\in S^d_{i,j}} T_{m,n} \le B^{i,j}_d$$

$$\forall i \in V, N_i \le N_{max}$$

Objective function. The objective is to minimize the weighted sum of path latencies and resource costs, including network links and containers. The weights can be adjusted based on operational requirements.

$$Util_{Lat} = \sum_{m,n} \frac{Lat(P_{m,n})}{Lat\_Limit_{m,n}}$$

$$Util_{Cost} = C_c \cdot N + \sum_i C_I(i) \cdot Thpt_I(i) + \sum_{i,j} C_p(i,j) \cdot Thpt_p(i,j)$$

$$Minimize \quad w_{Lat} \cdot Util_{Lat} + w_{Cost} \cdot Util_{Cost}$$

5.3 Scalable Two-Step Control Algorithm

The optimization problem formulated in §5.2 is NP-hard [3, 17]. Directly solving it does not scale given the size of our system. We propose a scalable two-step control algorithm to solve the problem. This algorithm is a heuristic that decouples path control (i.e., deciding the forwarding tables of each gateway) from capacity control (i.e., deciding the number of gateways in each region) to achieve scalability. Empirically, the algorithm can finish in two seconds for our system.

Step 1: Path control with current topology. The first step is to compute forwarding paths and the corresponding forwarding tables based on the current topology. This step takes into account the network performance constraints and capacity constraints, and does not consider how to add or remove gateways in each region. Empirically, we observe that video streams with long end-to-end latencies are more prone to breaking the minimum video quality bound. Therefore, the key idea of the algorithm is to prioritize the assignment of such streams to good paths. Algorithm 1 shows the pseudocode. It sorts the streams based on latency in descending order (line 8). Then it iterates over the streams and tries to use the shortest path for each stream (lines 9-10). It updates the related data structures after each path assignment (lines 11-21). The algorithm outputs the set of used gateways and their forwarding tables. We denote the result by R_{cur}.

Algorithm 1 Path control on current topology

```
1:  Input: traffic demand $S$, current topology $G$
2:  Output: used gateways $L$, forwarding tables $F$
3:  procedure PathControl
4:      Initialize forwarding tables: $F \leftarrow \emptyset$.
5:      Initialize used capacity list: $L \leftarrow \emptyset$.
6:      while $|S| > 0$ do
7:          Build shortest path graph $P$ on $G$
8:          Sort $S$ based on latency in descending order
9:          for $s \in S$ do
10:             $p \leftarrow P.shortestPath(s.src, s.dst)$
11:             if $p.canBeBuilt()$ then
12:                 $c \leftarrow \min(s.throughput, p.capacity)$
13:                 $F \leftarrow F \cup p.addPath(s, c)$
14:                 $L.update(p, c)$
15:                 $s.demand \leftarrow s.demand - c$
16:                 if $s.demand == 0$ then
17:                     $S \leftarrow S - s$
18:                 $p.capacity \leftarrow p.capacity - c$
19:                 if $p.capacity == 0$ then
20:                     $P \leftarrow P - p$
21:                     break
```

Step 2: Capacity control for next topology. The second step is to change the number of gateways (i.e., containers) in each region. The controller runs Algorithm 1 without the gateway capacity constraints. We denote the result by R_{next}. For each region, if the number of used gateways exceeds the number of currently available gateways, then the difference is the number of new gateways to add for the next epoch. On the other hand, if the numbers of used gateways in both R_{cur} and R_{next} are smaller than the number of currently available gateways, then the difference between the maximum of the former two and the latter is the number of gateways to remove for the next epoch.

5.4 Reaction Plan Generation

Algorithm 2 computes the backup paths for fast data plane reaction. It first traverses all paths according to the forwarding tables F computed by Algorithm 1 (line 5). For each region in the path, it uses the direct premium links as the default backup path (lines 10) and searches whether using a subset of current relay nodes with premium links can provide better performance (lines 11-13). If so, it updates the corresponding backup paths (line 13). The primary concern behind this algorithm is whether all constraints can still be satisfied when the gateways use backup paths without the global view of the network. To address this concern, Algorithm 2 has the following two properties.

Property 1. *The backup paths computed by Algorithm 2 are always better than the paths in F.*

Algorithm 2 Reaction plan generation

```
Input: Forwarding tables $F$ computed by Algorithm 1
Output: Reaction plan $all\_rec\_plans$
procedure RecoveryPlanComputation
    $all\_rec\_plans \leftarrow \emptyset$
    for $(s, p) \in F$ do
        // Fetch regions used by original path $p$ for stream $s$.
        $(r_1, r_2, ..., r_d) \leftarrow p.getRegions()$
        // Generate reaction plans in reverse order
        for $r_i \leftarrow (r_d, r_{d-2}, ..., r_1)$ do
            $rec\_plan[r_i] = \{r_d\}$
            for $r_j \leftarrow (r_{i+1}, r_{i+2}, ..., r_d)$ do
                if $rec\_plan[r_j] \cup \{r_j\}$ is better then
                    $rec\_plan[r_i] \leftarrow rec\_plan[r_j] \cup \{r_j\}$
            $all\_rec\_plans[s, p, r_i] \leftarrow rec\_plan[r_i]$
```

For a forwarding path p_F in F, the path p_{naive} that replaces all Internet links in p_F with premium links is better than p_F, in terms of latency and loss rate. Algorithm 2 further searches for a path p_{best} that is better than p_{naive} and uses fewer relay regions. Therefore, we have $p_{best} \geq p_{naive} \geq p_{origin}$, meaning that Algorithm 2 can always find better paths than those in F.

Property 2. *The reaction plan computed by Algorithm 2 satisfies all constraints.*

The backup paths computed by Algorithm 2 use a *subset* of regions on the paths in F. This means that the ingress and egress bandwidth of each region consumed by the reaction plan is no greater than that of F. Therefore, the capacity constraints are satisfied. Property 1 indicates that the backup paths are better in terms of latency and loss rate. Thus, the network performance constraints are also satisfied.

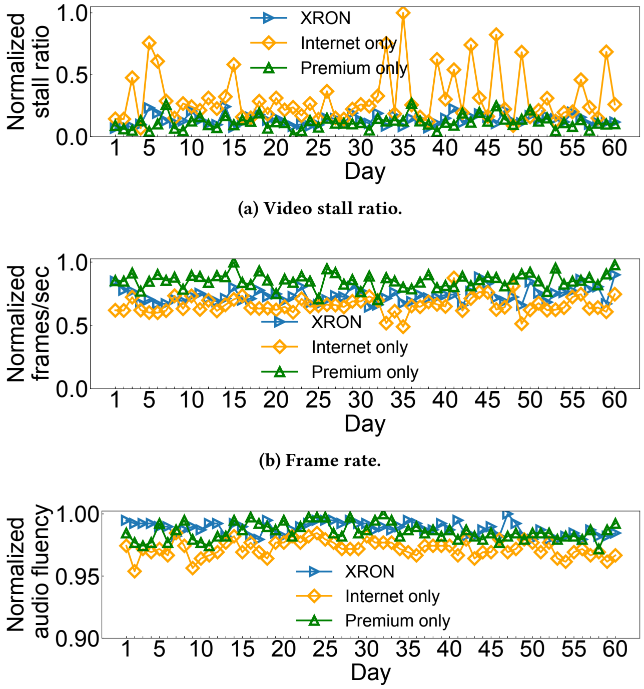
6 EVALUATION

In this section, we present the evaluation results of XRON. We first show the end-to-end application performance results of XRON from its large-scale production deployment (§6.1). Then we evaluate the network-level performance of XRON (§6.2). After that, we analyze the cost of XRON (§6.3). Finally, we conduct an ablation study to show the benefits of key mechanisms proposed in XRON (§6.4).

6.1 End-to-End Application Performance

XRON has been deployed in eleven cloud regions across four continents to support our video conferencing service, DingTalk, for more than six months since August 2022. We compare XRON with two previous versions of our video conferencing service: (*i*) *Internet only version*, which only uses Internet links, and (*ii*) *Premium only version*, which only use premium links. We are currently in the process of gradually migrating users to XRON. At the time of submission, 10% of sessions are randomly scheduled to use XRON. During this process, all three versions run in parallel, which allows us to compare the three systems at the application level, i.e., the end-to-end video conferencing performance.

Overall user experience. We present core user-experience metrics reported by our video conferencing service in two months. Due to confidentiality, all reported metrics are normalized against the largest value in the dataset. Figure 13 shows the normalized average video stall ratio, frame rate, and audio fluency each day.



(a) Video stall ratio.

(b) Frame rate.

(c) Audio fluency.

Figure 13: The overall statistics of different versions of our video conferencing service in the deployment.

Figure 13a shows the video stall ratio of different versions in our production. With real-time monitoring, XRON leverages hybrid network resources to avoid bad network links, and thus achieves a low video stall ratio. The Internet only version has the highest and most unpredictable video stall ratio due to the best-effort nature of Internet links. Compared with the Internet only version, XRON reduces the average video stall ratio by 77%. More importantly, XRON keeps the variance of video stall ratio low, which is critical for delivering a consistent user experience. Furthermore, the performance of XRON is close to the premium only version, which verifies the efficiency of our global routing and fast reaction mechanisms.

Figure 13b shows the frame rate of different versions. The frame rate of XRON is slightly lower than that of the premium only version. XRON improves the frame rate by 12% compared to the Internet only version, which is significant given the scale of our service.

Figure 13c shows the audio fluency of different versions. Audio fluency is measured using an improved version of the E-model [1]. It considers the comprehensive impact of various factors, e.g., overall loudness, signal-to-noise ratio, talker/listener echo, end-to-end latency, etc. The overall audio fluency is scored from one to five (higher is better). XRON outperforms the Internet only version by 1.58% and achieves similar scores as the premium only version.

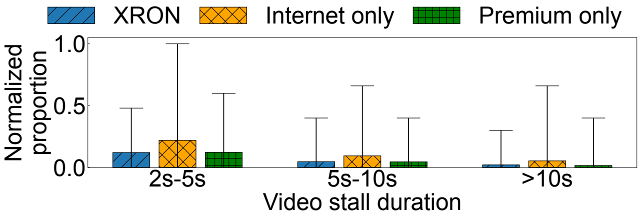


Figure 14: Comparison of video stall durations.

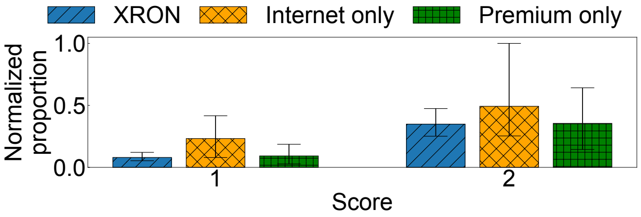


Figure 15: Comparison of low scores of audio fluency.

Reducing bad cases. Furthermore, we present the statistics of severe video and audio performance degradations. The real-time link state monitoring and fast reaction mechanisms in XRON play an important role in preventing severe performance degradations. Figure 14 shows statistics about long video stalls (i.e., ≥ 2s). XRON decreases long video stalls by 49.1% compared to the Internet only version. Figure 15 shows the percentage of low audio fluency scores (≤ 2) under different versions. In the daily operation of our conferencing service, the audio fluency score = 1 is defined as a bad audio experience. XRON reduces the cases of bad audio experiences by 65.2% compared to the Internet only version.

6.2 Network-Level Performance

To further investigate the performance of XRON, we evaluate the network-level performance between different video conferencing clusters.

Overall network-level performance. We create full-mesh video conferencing sessions between all regions for all three versions (XRON, Internet only and premium only). Table 2 and Table 3 show the results of latency and loss rate monitored on all video conferencing clusters, respectively. Latency is measured as the delay between sending a message and receiving its ACK at the sender side. Because XRON can find low-latency paths and quickly recover from network degradations, XRON reduces 99% and 99.9% latency by 1.9× and 9× compared to the Internet only version, respectively. Besides, the 99.9% loss rate is reduced by 263× compared with the Internet only version. Both latency and loss rate results of XRON are close to the premium only version, which is consistent with our expectations.

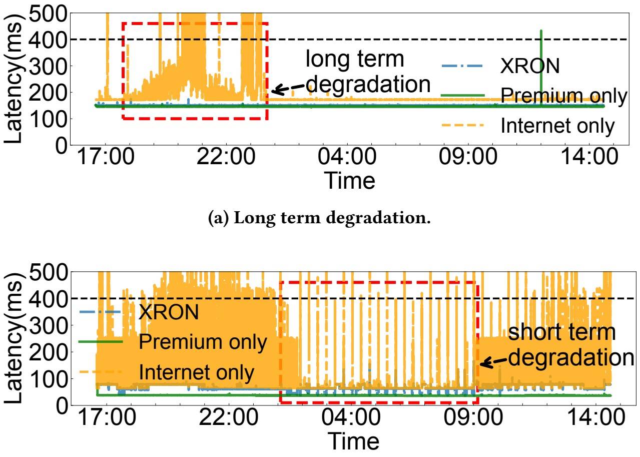
Case studies. Besides aggregated statistics of full-mesh video conferencing sessions, we also trace typical region pairs as case studies. These region pairs are popular for our users and represent different network degradation patterns. Figure 16a shows a long-term network degradation case (highlighted with a box in the figure).

Service	Average	95%	99%	99.9%
Premium	121	268	272	278
XRON	126	262	274	285
Internet	205	347	786	2762

Table 2: Comparison of latency (ms) between different video conferencing clusters.

Service	Average	95%	99%	99.9%
Premium	0.0022	0	0	0.037
XRON	0.0034	0	0	0.052
Internet	0.182	0.71	5.88	13.21

Table 3: Comparison of loss rate (%) between different video conferencing clusters.

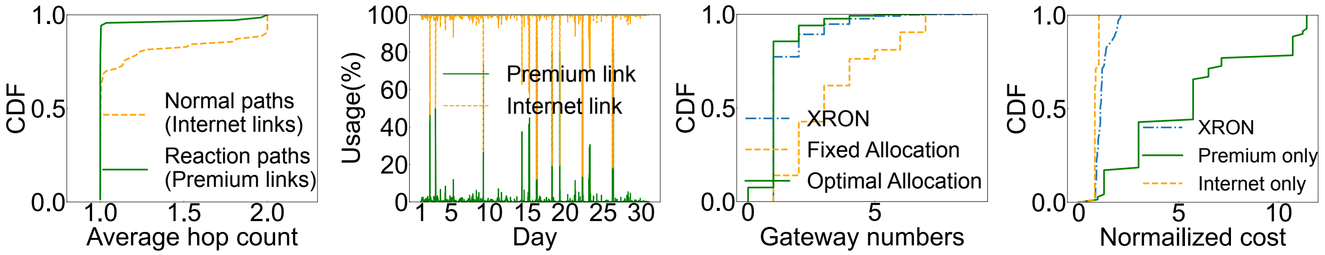


(a) Long term degradation.

(b) Short term degradation.

Figure 16: Network degradation case studies.

From 17:42 to 23:37, the Internet links encounter long latency due to packet loss and retransmissions. It leads to great latency increase in the Internet only version. In this case, XRON can use alternative cheap Internet links to reroute and keep the transmission latency steady, while maintaining the operation cost of XRON low. Besides the long-term degradation case, Figure 16b further shows the short-term but frequent network degradation case (from 00:13 to 09:04), which is also highlighted with a box in the figure. For this region pair, the direct Internet link is the shortest path but drops packets from time to time. With the fast reaction mechanism, XRON can utilize this direct Internet link when the performance is good while leveraging rerouting and fast recovery to avoid unexpected performance drops. In both cases, XRON decreases the maximum latency of video streams by more than 184×, compared to the Internet only version, and consistently provides users with good network performance.



(a) Hop count distribution.

(b) Usage of different link types.

(c) Used gateway distribution.

(d) Overall cost distribution.

Figure 17: Comprehensive cost analysis of XRON.

6.3 Cost

Cost is an important aspect of a large-scale video conferencing service. We first analyze (*i*) overlay path length (i.e., relay hop count), (*ii*) premium link usage, and (*iii*) container usage (for XRON gateways), which are all critical variables to the overall cost. Then we show the overall cost comparison between XRON and the alternatives.

Overlay path length. We calculate the hop count of all overlay paths. Figure 17a shows the CDF of the hop count. The average hop count for normal paths using Internet links is 1.19. The average hop count for fast reaction paths using premium links is only 1.04. 94% of the overlay paths are no more than two hops. It means that XRON delivers qualified cross-region transmission with a small traffic overhead.

Premium link usage. Premium links are used in our fast reaction mechanism. Since premium links are more expensive than Internet links, the usage of premium links is another critical metric for the overall cost. As shown in Figure 17b, the usage of premium links in XRON is negligible compared to that of Internet links. Only 3% of network traffic is transmitted on the premium links, making the cost of using premium links low. This result also indicates that the fast reaction mechanism runs efficiently.

Container usage. Besides network bandwidth fees, another part of the cost for XRON is the expenses on the containers to run XRON gateways. XRON currently serves 10% of overall cross-region traffic. To estimate the performance of capacity control under full traffic, we collect two weeks of full-scale traffic demand between all regions in the production environment and conduct an emulation. We compare XRON with two baselines: *Optimal Allocation* and *Fixed Allocation*. Optimal Allocation always maintains the best number of containers assuming perfect future demand predictions. Fixed Allocation provisions a fixed number of containers according to the traffic demand in the peak hours during the last week. Figure 17c shows the CDF of the required number of containers in all regions. Compared to Fixed Allocation, XRON reduces the number of used containers by 57%. Furthermore, the result of XRON is close to that of Optimal Allocation. XRON reserves a bit more containers as headrooms, which is affordable in practice.

Overall cost. This part provides an overall cost comparison of different versions under the same full-scale traffic demand. Figure 17d shows the normalized cost of different versions. The cost of any region-pair in XRON is strictly less than that in the premium only version. The overall cost of XRON is 4.73× less than the premium only version and 1.37× higher than the Internet version. Considering the outstanding performance compared with the Internet only version, XRON is a cost-effective solution.

6.4 Ablation Study

We further conduct an ablation study on three core components of XRON, i.e., fast reaction, asymmetric forwarding, and prediction-based proactive scaling.

Fast reaction. To evaluate fast reaction, we create three 24-hour full-mesh video conferencing sessions across all regions. Each session is served by one of the following three variants: XRON-Premium, which always uses the best overlay paths consisting of only premium links; XRON-Basic, which uses all the mechanisms of XRON except fast reaction; and XRON, which uses all the mechanisms of XRON. Figure 18 shows the number of large latency cases (i.e., > 400ms) between two adjacent frames. In this experiment, XRON provides a similar performance as XRON-Premium. Moreover, fast reaction significantly reduces the number of 0.4-1s and 1-2s latency cases by 97.6% and 99.8%, respectively. XRON further eliminates large latency cases which are greater than 2 seconds. This experiment indicates the fast reaction mechanism is crucial to achieving better end-to-end performance.

Asymmetric forwarding. We compare the latency of overlay paths calculated by two controller versions: (*i*) only considering symmetric forwarding and (*ii*) considering asymmetric forwarding. At the end of each scheduling period, we record the overlay paths calculated by different controllers, and calculate the latency of overlay paths and the corresponding latency speedup ratio. Figure 19 presents the CDF of the latency speedup ratio. As shown in the figure, nearly 40% overlay paths can achieve better performance while considering asymmetric forwarding. More importantly, based on our observations in production, some overlay paths accelerated by asymmetric forwarding carry a large number of video streams. These overlay paths deliver a large amount of video streams with higher performance and lower cost.

Prediction-based proactive scaling. Finally, we compare our prediction-based proactive scaling mechanism with the reactive scaling mechanism. Figure 20 shows the error rate of these two mechanisms. The error rate is calculated as the ratio of capacity

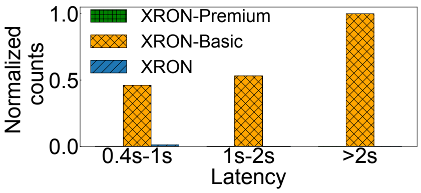


Figure 18: Benefits of fast reaction.

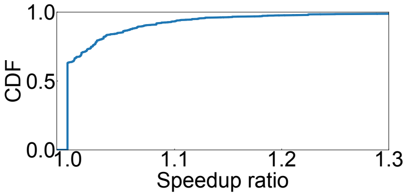


Figure 19: Benefits of asymmetric forwarding.

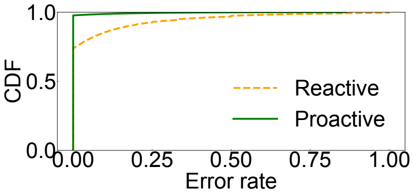


Figure 20: Benefits of prediction-based proactive scaling.

under-provisioning. Our prediction-based proactive scaling mechanism prevents 97.7% duration from under-provisioning and significantly reduces the error rate by 91% compared to the reactive scaling mechanism.

7 RELATED WORK

Overlay networks. RON [6] is an early overlay network. It monitors link quality and dynamically routes traffic to achieve resiliency. Succeeding works [9, 33, 57] enhance routing under homogeneous network resources. XRON further explores the larger design space introduced by heterogeneous network resources. Skyplane [29] is a recent multi-cloud overlay network specifically designed for throughput-intensive bulk data transfers. In comparison, XRON focuses on serving latency-sensitive video conferencing.

There is a lot of work on Content Delivery Networks (CDNs) [10, 21, 24, 37, 42, 43, 51, 56]. LiveNet [37] is a recent work that builds an overlay network on a CDN infrastructure for live video streaming, which is fully based on premium links (loss rate < 0.175%). Generally, CDN providers own dedicated physical resources (servers, clusters, and networks), and leverage resource scheduling mechanisms to optimize transmission performance. The main difference is that XRON handles the challenge of managing heterogeneous network resources and elastic capacity in a cloud-native scenario.

Traffic engineering. Many centralized traffic engineering solutions have been proposed to optimize network performance [3, 4, 8, 18, 25, 26, 30–32, 47, 48, 54, 61]. These solutions serve self-built/rented dedicated networks by corresponding cloud providers, while XRON optimizes transmission through mixing multiple network tires. Furthermore, XRON elastically subscribes cloud resources based on the domain knowledge of video conferencing.

Network monitoring. Cloudcast [7] proposes a measurement system for public cloud connectivity. Pingmesh [23] is a production system for network monitoring in datacenters. Many other works [11, 16, 19, 34, 63, 64] focus on large-scale network failure detections. TSLP [15] leverages active probing to measure underlay link congestion, while XRON focuses on overlay cross-region transmission. These works focus on general network measurement and analysis but do not target degradations influencing real-time video streams.

Video stream transmission. A lot of research efforts have been devoted to optimizing the performance of real-time video streaming, including encoding/decoding algorithms [14, 20], adaptive bitrate streaming algorithms [5, 27, 39, 53, 55], video transport solutions [20, 28, 40, 41, 44, 45, 52, 60, 62] and other video quality enhancements [22, 36, 38, 59]. They are complementary to XRON and can be employed together with XRON. In our production deployment, we have deployed some of these enhancements. With comprehensive tests and online monitorings, they work well with XRON.

8 CONCLUSION

XRON is a hybrid elastic cloud overlay network for our planetary-scale video conferencing service. XRON leverages heterogeneous cross-region network resources provided by the cloud platform to build the overlay network, in order to satisfy the quality requirement of video conferencing and achieve low cost at the same time. In the data plane, we leverage the asymmetric features of cross-region links to optimize performance. We also design an efficient fast reaction mechanism for overcoming temporary performance degradations. In the control plane, we leverage application-level runtime information to precisely predict traffic demands, and efficiently calculate the forwarding tables, the capacity and the fast reaction plan for each region. XRON has been deployed in eleven cloud regions across the globe to support our video conferencing service since August 2022. Experimental results show that compared to the Internet only version, XRON reduces video stall ratio and bad audio fluency by 77% and 65.2%, respectively, and compared with the premium only version, XRON cuts the cost by 79%.

ACKNOWLEDGMENTS.

This work was supported by the National Key Research and Development Program of China under the grant number 2022YFB4500700, and Alibaba Research Intern Program.

We sincerely thank the anonymous reviewers for their valuable feedback. Xin Jin is the corresponding author. Bingyang Wu, Xuanzhe Liu and Xin Jin are also with the Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education. We also thank Jianfeng Liao, Chenda Huang, Xingjin Zhou, Xinyang Dong, Feng Li, Jichao Shi, Yiyuan Huang, Ruixin Li, Liping Mao, Lei Zhang, Jianmin Liu, Yu Lan, Bei Li, Zhuyu Zhao, Hailong Qiu, Xuan Zeng, Weifeng Shen for their sincere cooperation on the implementing, testing, deploying and operating XRON on the DingTalk video conferencing system.

REFERENCES

[1] 2019. G.107.1 : Wideband E-model. https://www.itu.int/rec/T-REC-G.107.1.
[2] 2023. Kubernetes. https://kubernetes.io/.
[3] Firas Abuzaid, Srikanth Kandula, Behnaz Arzani, Ishai Menache, Matei Zaharia, and Peter Bailis. 2021. Contracting Wide-area Network Topologies to Solve Flow Problems Quickly.. In *USENIX NSDI*.

[4] Satyajeet Singh Ahuja, Varun Gupta, Vinayak Dangui, Soshant Bali, Abishek Gopalan, Hao Zhong, Petr Lapukhov, Yiting Xia, and Ying Zhang. 2021. Capacity-Efficient and Uncertainty-Resilient Backbone Network Planning with Hose. In *ACM SIGCOMM*.

[5] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2018. Oboe: Auto-Tuning Video ABR Algorithms to Network Conditions. In *ACM SIGCOMM*.

[6] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. 2002. Resilient Overlay Networks. *ACM SOSP* (2002).

[7] Yaniv Ben-Itzhak, Aran Bergman, Israel Cidon, Igor Golikov, Alex Markuze, Noga Rotman, and Eyal Zohar. 2022. Cloudcast: Characterizing public clouds connectivity. *arXiv preprint arXiv:2201.06989* (2022).

[8] Jeremy Bogle, Nikhil Bhatia, Manya Ghobadi, Ishai Menache, Nikolaj Bjørner, Asaf Valadarsky, and Michael Schapira. 2019. TEAVAR: Striking the Right Utilization-Availability Balance in WAN Traffic Engineering. In *ACM SIGCOMM*.

[9] Chris X. Cai, Franck Le, Xin Sun, Geoffrey G. Xie, Hani Jamjoom, and Roy H. Campbell. 2016. CRONets: Cloud-Routed Overlay Networks. In *IEEE ICDCS*.

[10] Matt Calder, Ryan Gao, Manuel Schröder, Ryan Stewart, Jitendra Padhye, Ratul Mahajan, Ganesh Ananthanarayanan, and Ethan Katz-Bassett. 2018. Odin: Microsoft's Scalable Fault-Tolerant CDN Measurement System. In *USENIX NSDI*.

[11] Yiyang Chang, Chuan Jiang, Ashish Chandra, Sanjay Rao, and Mohit Tawarmalani. 2019. Lancet: Better Network Resilience by Designing for Pruned Failure Sets. *Proc. ACM Meas. Anal. Comput. Syst.* (2019).

[12] Marco Chiesa, Gábor Rétvári, and Michael Schapira. 2018. Oblivious Routing in IP Networks. *IEEE/ACM Transactions on Networking* (2018).

[13] Martin P. Clark. 2000. *Wireless Access Networks - Fixed Wireless Access and WLL Networks.* John Wiley & Sons, Inc.

[14] Mallesham Dasari, Kumara Kahatapitiya, Samir R. Das, Aruna Balasubramanian, and Dimitris Samaras. 2022. Swift: Adaptive Video Streaming with Layered Neural Codecs. In *USENIX NSDI*.

[15] Amogh Dhamdhere, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and Kc Claffy. 2018. Inferring Persistent Interdomain Congestion. In *ACM SIGCOMM*.

[16] Asma Enayet and John Heidemann. 2022. Internet Outage Detection Using Passive Analysis. In *ACM SIGCOMM Conference on Internet Measurement Conference.*

[17] S. Even, A. Itai, and A. Shamir. 1976. On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM J. Comput.* (1976).

[18] Andrew D. Ferguson, Steven D. Gribble, Chi-Yao Hong, Charles Edwin Killian, Waqar Mohsin, Henrik Mühe, Joon Suan Ong, Leonid B. Poutievski, Arjun Singh, Lorenzo Vicisano, Richard Alimi, Shawn Shuoshuo Chen, Michael Conley, Subhasree Mandal, Karthik Nagaraj, Kondapa Naidu Bollineni, Amr Sabaa, Shidong Zhang, Min Zhu, and Amin Vahdat. 2021. Orion: Google's Software-Defined Networking Control Plane. In *USENIX NSDI*.

[19] Rodrigo Fonseca, Tianrong Zhang, Karl Deng, and Lihua Yuan. 2019. dShark: A general, easy to program and scalable framework for analyzing in-network packet traces. *USENIX NSDI* (2019).

[20] Sadjad Fouladi, John Emmons, Emre Orbay, Catherine Wu, Riad S. Wahby, and Keith Winstein. 2018. Salsify: Low-Latency Network Video through Tighter Integration between a Video Codec and a Transport Protocol. In *USENIX NSDI*.

[21] Michael J. Freedman. 2010. Experiences with CoralCDN: A Five-Year Operational View. In *USENIX NSDI*.

[22] Ehab Ghabashneh and Sanjay Rao. 2020. Exploring the interplay between CDN caching and video streaming performance. In *IEEE INFOCOM*.

[23] Chuanxiong Guo, Lihua Yuan, Dong Xiang, Yingnong Dang, Ray Huang, Dave Maltz, Zhaoyi Liu, Vin Wang, Bin Pang, Hua Chen, et al. 2015. Pingmesh: A large-scale system for data center network latency measurement and analysis. In *ACM SIGCOMM*.

[24] Dongsu Han, David Andersen, Michael Kaminsky, Dina Papagiannaki, and Srinivasan Seshan. 2011. Hulu in the neighborhood. In *International Conference on Communication Systems and Networks*.

[25] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. 2013. Achieving High Utilization with Software-Driven WAN. In *ACM SIGCOMM*.

[26] Chi-Yao Hong, Subhasree Mandal, Mohammad Al-Fares, Min Zhu, Richard Alimi, Kondapa Naidu B., Chandan Bhagat, Sourabh Jain, Jay Kaimal, Shiyu Liang, Kirill Mendelev, Steve Padgett, Faro Rabe, Saikat Ray, Malveeka Tewari, Matt Tierney, Monika Zahn, Jonathan Zolla, Joon Ong, and Amin Vahdat. 2018. B4 and after: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-Defined WAN. In *ACM SIGCOMM*.

[27] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. *SIGCOMM CCR* (2014).

[28] Per Hurtig, Karl-Johan Grinnemo, Anna Brunstrom, Simone Ferlin, Özgü Alay, and Nicolas Kuhn. 2019. Low-Latency Scheduling in MPTCP. *IEEE/ACM Transactions on Networking* (2019).

[29] Paras Jain, Sam Kumar, Sarah Wooders, Shishir G Patil, Joseph E Gonzalez, and Ion Stoica. 2023. Skyplane: Optimizing Transfer Cost and Throughput Using Cloud-Aware Overlays. *USENIX NSDI* (2023).

[30] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2013. B4: Experience with a Globally-Deployed Software Defined Wan. In *ACM SIGCOMM*.

[31] Chuan Jiang, Zixuan Li, Sanjay Rao, and Mohit Tawarmalani. 2022. Flexile: Meeting Bandwidth Objectives Almost Always. In *ACM CoNEXT*.

[32] Chuan Jiang, Sanjay Rao, and Mohit Tawarmalani. 2020. PCF: Provably Resilient Flexible Routing. In *ACM SIGCOMM*.

[33] Junchen Jiang, Rajdeep Das, Ganesh Ananthanarayanan, Philip A. Chou, Venkata Padmanabhan, Vyas Sekar, Esbjorn Dominique, Marcin Goliszewski, Dalibor Kukoleca, Renat Vafin, and Hui Zhang. 2016. Via: Improving Internet Telephony Call Quality Using Predictive Relay Selection *(ACM SIGCOMM)*.

[34] Yuchen Jin, Sundararajan Renganathan, Ganesh Ananthanarayanan, Junchen Jiang, Venkata N. Padmanabhan, Manuel Schroder, Matt Calder, and Arvind Krishnamurthy. 2019. Zooming in on Wide-Area Latencies to a Global Cloud Provider. In *ACM SIGCOMM*.

[35] Ethan Katz-Bassett, Harsha V. Madhyastha, Vijay Kumar Adhikari, Colin Scott, Justine Sherry, Peter Van Wesep, Thomas Anderson, and Arvind Krishnamurthy. 2010. Reverse Traceroute. In *USENIX NSDI*.

[36] Jaehong Kim, Youngmok Jung, Hyunho Yeo, Juncheol Ye, and Dongsu Han. 2020. Neural-Enhanced Live Streaming: Improving Live Video Ingest via Online Learning. In *ACM SIGCOMM*.

[37] Jinyang Li, Zhenyu Li, Ri Lu, Kai Xiao, Songlin Li, Jufeng Chen, Jingyu Yang, Chunli Zong, Aiyun Chen, Qinghua Wu, Chen Sun, Gareth Tyson, and Hongqiang Harry Liu. 2022. LiveNet: A Low-Latency Video Transport Network for Large-Scale Live Streaming. In *ACM SIGCOMM*.

[38] Xianshang Lin, Yunfei Ma, Junshao Zhang, Yao Cui, Jing Li, Shi Bai, Ziyue Zhang, Dennis Cai, Hongqiang Harry Liu, and Ming Zhang. 2022. GSO-Simulcast: Global Stream Orchestration in Simulcast Video Conferencing Systems. In *ACM SIGCOMM*.

[39] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *ACM SIGCOMM*.

[40] Zili Meng, Yaning Guo, Chen Sun, Bo Wang, Justine Sherry, Hongqiang Harry Liu, and Mingwei Xu. 2022. Achieving Consistent Low Latency for Wireless Real-Time Communications with the Shortest Control Loop. In *ACM SIGCOMM*.

[41] Vikram Nathan, Vibhaalakshmi Sivaraman, Ravichandra Addanki, Mehrdad Khani, Prateesh Goyal, and Mohammad Alizadeh. 2019. End-to-End Transport for Video QoE Fairness. In *ACM SIGCOMM*.

[42] Erik Nygren, Ramesh K. Sitaraman, and Jennifer Sun. 2010. The Akamai Network: A Platform for High-Performance Internet Applications. *SIGOPS Operating Systems Review* (2010).

[43] Chunyi Peng, Minkyong Kim, Zhe Zhang, and Hui Lei. 2012. VDN: Virtual machine image distribution network for cloud data centers. In *IEEE INFOCOM*.

[44] Alexander Rabitsch, Per Hurtig, and Anna Brunstrom. 2018. A Stream-Aware Multipath QUIC Scheduler for Heterogeneous Paths. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*.

[45] Devdeep Ray, Jack Kosaian, K. V. Rashmi, and Srinivasan Seshan. 2019. Vantage: Optimizing Video Upload for Time-Shifted Viewing of Social Live Streams. In *ACM SIGCOMM*.

[46] Johann Schleier-Smith, Vikram Sreekanti, Anurag Khandelwal, João Carreira, Neeraja Jayant Yadwadkar, Raluca Ada Popa, Joseph E. Gonzalez, Ion Stoica, and David A. Patterson. 2021. What serverless computing is and should become: the next phase of cloud computing. *Commun. ACM* (2021).

[47] Rachee Singh, Sharad Agarwal, Matt Calder, and Paramvir Bahl. 2021. Cost-effective Cloud Edge Traffic Engineering with Cascara.. In *USENIX NSDI*.

[48] Rachee Singh, Nikolaj Bjorner, Sharon Shoham, Yawei Yin, John Arnold, and Jamie Gaudette. 2021. Cost-Effective Capacity Provisioning in Wide Area Networks with Shoofly. In *ACM SIGCOMM*.

[49] Alisha Ukani, Ariana Mirian, and Alex C. Snoeren. 2021. Locked-in during Lock-down: Undergraduate Life on the Internet in a Pandemic. In *ACM SIGCOMM Conference on Internet Measurement Conference*.

[50] Kevin Vermeulen, Ege Gurmericliler, Italo Cunha, David Choffnes, and Ethan Katz-Bassett. 2022. Internet Scale Reverse Traceroute. In *ACM SIGCOMM Conference on Internet Measurement Conference*.

[51] Limin Wang, Kyoung Soo Park, Ruoming Pang, Vivek Pai, and Larry Peterson. 2004. Reliability and Security in the CoDeeN Content Distribution Network. In *USENIX ATC*.

[52] Hongjia Wu, Özgü Alay, Anna Brunstrom, Simone Ferlin, and Giuseppe Caso. 2020. Peekaboo: Learning-Based Multipath Scheduling for Dynamic Heterogeneous Environments. *IEEE Journal on Selected Areas in Communications* (2020).

[53] Francis Y. Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein. 2020. Learning in Situ: A Randomized Experiment in Video Streaming. In *USENIX NSDI*.

[54] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, Ankur Jain, Victor Lin, Colin Rice, Brian Rogan, Arjun Singh, Bert Tanaka, Manish Verma, Puneet Sood, Mukarram Tariq, Matt Tierney, Dzevad Trumic, Vytautas Valancius,

Calvin Ying, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2017. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *ACM SIGCOMM*.

[55] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. *SIGCOMM CCR* (2015).

[56] Minlan Yu, Wenjie Jiang, Haoyuan Li, and Ion Stoica. 2012. Tradeoffs in CDN Designs for Throughput Oriented Traffic. In *ACM CoNEXT*.

[57] Diman Zad Tootaghaj, Faraz Ahmed, Puneet Sharma, and Mihalis Yannakakis. 2020. Homa: An Efficient Topology and Route Management Approach in SD-WAN Overlays. In *IEEE INFOCOM*.

[58] Ming Zhang, Chi Zhang, Vivek Pai, Larry Peterson, and Randy Wang. 2004. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *USENIX OSDI*.

[59] Xu Zhang, Yiyang Ou, Siddhartha Sen, and Junchen Jiang. 2021. SENSEI: Aligning Video Streaming Quality with Dynamic User Sensitivity.. In *USENIX NSDI*.

[60] Zhilong Zheng, Yunfei Ma, Yanmei Liu, Furong Yang, Zhenyu Li, Yuanbo Zhang, Jiuhai Zhang, Wei Shi, Wentao Chen, Ding Li, Qing An, Hai Hong, Hongqiang Harry Liu, and Ming Zhang. 2021. XLINK: QoE-Driven Multi-Path QUIC Transport in Large-Scale Video Services. In *ACM SIGCOMM*.

[61] Zhizhen Zhong, Manya Ghobadi, Alaa Khaddaj, Jonathan Leach, Yiting Xia, and Ying Zhang. 2021. ARROW: Restoration-Aware Traffic Engineering. In *ACM SIGCOMM*.

[62] Fan Zhou, David Choffnes, and Kaushik Chowdhury. 2019. Janus: A Multi-TCP Framework for Application-Aware Optimization in Mobile Networks. *IEEE Transactions on Mobile Computing* (2019).

[63] Shunmin Zhu, Jianyuan Lu, Biao Lyu, Tian Pan, Chenhao Jia, Xin Cheng, Daxiang Kang, Yilong Lv, Fukun Yang, Xiaobo Xue, et al. 2022. Zoonet: a proactive telemetry system for large-scale cloud networks. In *ACM CoNEXT*.

[64] Yibo Zhu, Nanxi Kang, Jiaxin Cao, Albert Greenberg, Guohan Lu, Ratul Mahajan, Dave Maltz, Lihua Yuan, Ming Zhang, Ben Y Zhao, et al. 2015. Packet-level telemetry in large datacenter networks. In *ACM SIGCOMM*.