Saving Private WAN: Using Internet Paths to Offload WAN Traffic in Conferencing Services

BHASKAR KATARIA, PALAK LNU, Microsoft Research, India
RAHUL BOTHRA, UIUC, USA
ROHAN GANDHI,DEBOPAM BHATTACHERJEE,
VENKATA N. PADMANABHAN, Microsoft Research, India
IRENA ATOV, SRIRAAM RAMAKRISHNAN, SOMESH CHATURMOHTA,
CHAKRI KOTIPALLI, RUI LIANG, KEN SUEDA, XIN HE, KEVIN HINTON, Microsoft, USA

Large-scale video conferencing services incur significant network cost while serving surging global demands. Our work systematically explores the opportunity to offload a fraction of this traffic to the Internet, a cheaper routing option offered already by cloud providers, from WAN without drop in application performance. First, with a large-scale latency measurement study with 3.5 million data points per day spanning $241K$ source cities and 21 data centers across the globe, we demonstrate that Internet paths perform comparable to or better than the private WAN for parts of the world (e.g., Europe and North America). Next, we present Titan, a live (12+ months) production system that carefully moves a fraction of the conferencing traffic to the Internet using the above observation. Finally, we propose Titan-Next – a research prototype that jointly assigns the conferencing server and routing option (Internet or WAN) for individual calls. With 5 weeks of production data, we show Titan-Next reduces the sum of peak bandwidth on WAN links that defines the operational network cost by up to 61% compared to state-of-the-art baselines.

CCS Concepts: • **Networks → Network measurement**.

Additional Key Words and Phrases: Internet vs. WAN performance, conferencing services

ACM Reference Format:
Bhaskar Kataria, Palak LNU, Rahul Bothra, Rohan Gandhi,Debopam Bhattacherjee,, Venkata N. Padmanabhan, and Irena Atov, Sriraam Ramakrishnan, Somesh Chaturmohta,, Chakri Kotipalli, Rui Liang, Ken Sueda, Xin He, Kevin Hinton . 2024. Saving Private WAN: Using Internet Paths to Offload WAN Traffic in Conferencing Services. *Proc. ACM Netw.* 2, CoNEXT4, Article 33 (December 2024), 22 pages. https://doi.org/10.1145/3696404

1 Introduction

Conferencing services such as Microsoft Teams[7], Zoom[10], DingTalk[2], and Google Meet[3] have become an indispensable part of our society, especially since the COVID-19 pandemic. However, the skyrocketing growth in demand[8] has also resulted in higher costs incurred by such services[16, 50]. Usually, such large-scale conferencing services use dedicated Media Processor (MP) servers [16, 50]) in cloud data centers (DCs) that receive media streams (audio, video, and screen-share) from users, process, and redistribute them. The cloud providers' private WANs (wide-area networks) have been the default choice [13, 16] to carry traffic between users and the MP servers. With such routing, the conferencing traffic ingresses into and egresses from the WAN

Bhaskar Kataria and Rahul Bothra did the work at Microsoft Research, India.
Contact Information: bk478@cornell.edu, rohangandhi@microsoft.com, debopamb@microsoft.com.

ACM 2834-5509/2024/12-ART33
https://doi.org/10.1145/3696404

closest to the user (not the MP server) thus consuming significant WAN resources and inflating the operational cost for the application. Cloud providers have started providing the *Internet routing* option [5, 6] that allows application traffic to ingress/egress closer to the cloud (MP) server, thus reducing the WAN load. This Internet routing option is significantly cheaper than WAN (by up to 53% [5, 6]) motivating us to explore if a fraction of the conferencing traffic could leverage this routing option, without affecting user experience, thus reducing operational network cost.

First, we shed light on the performance of the Internet versus WAN routing for Microsoft Azure – one of the largest hyperscalars. While both network loss and latency can affect user experience, the former is mitigated to some extent through application layer mechanisms such as [15, 41] while the latter is harder to tackle since it is a more fundamental impediment to interactivity. To understand if we can leverage Internet paths to reduce costs of conferencing services without affecting application performance, we measure latencies using the WAN and the Internet routing options. Our measurements are piggybacked on Microsoft Teams (MS Teams) – a large-scale conferencing service. MS Teams is hosted on Azure; so, we measure latency to Azure. We run 3.5 million measurements/day (average) for 12 months from 241K cities (distinct population centers) across the globe to 21 Azure DCs. Our measurements show that *the Internet offers latency as good as WAN or even better* in parts of the world – especially in Europe (including UK) and North America. In rest of the paper, we use WAN to denote private WAN of Azure, and Internet as public Internet. Such an observation is promising to move some of the WAN traffic to the Internet to reduce costs. We proceed in two phases: (1) We move a fraction of the WAN traffic to the Internet prioritizing *safety* over optimality without changing MP DC assignments to calls. To that extent, we built Titan that has been in production for the last 1 year. (2) We built Titan-Next as a research prototype that further reduces traffic on WAN through joint assignment of MP DCs and Internet routing.

Next, we present Titan – a system that carefully moves a fraction of the total MS Teams traffic to the Internet. A key concern with Internet offload is that MS Teams consumes significant bandwidth, and moving all of its traffic to the Internet can result in network congestion and poor user experience. Measurements through *ping* are too lightweight to gauge capacity on the Internet. Titan moves traffic to the Internet iteratively. In each iteration, it increases traffic on the Internet and measures the latency, loss, jitter and other network and application metrics. The iterations terminate when a subset of the metrics indicates early signs of deteriorating performance. Also, We prioritize *safety* as we stop moving traffic to the Internet after certain point even if there is no deterioration in performance. In this paper, we present the design of Titan and our experiences with moving large amounts of live MS Teams production traffic to the Internet (§4) with Titan for the last 12+ months.

Finally, we present Titan-Next. Titan focuses on choice of routing given the selection of MP servers to calls is fixed. We do even better using Titan-Next by jointly optimizing the MP selection and routing based on the Internet capacity calculated by Titan. Titan-Next is based on three key ideas: (*a*) The MP DC selection and the routing option need to be jointly optimized as the former has bearing on choosing latter. (*b*) Interactivity of the call depends on *maximum end-to-end latency* between participants. Our results show that the participants are sensitive to this metric. Hence, it should be considered while making assignments. (*c*) Titan-Next reduces call migrations using *reduced call configurations*, which is an abstraction for the resource needs of a call (§6.2).

Titan-Next formulates a Linear Program (LP) that uses these key ideas to jointly determine the MP DCs and routing options for calls. Titan-Next is a research prototype that currently works in a *shadow mode* alongside Titan (live in production) and computes the potential savings of doing assignments differently than Titan. We evaluate Titan-Next using 5 weeks (4 weeks of training data + 1 week for evaluation) of call traces from production with $O(10$ million$)$ calls on a weekday. We compare Titan-Next against Titan, Weighted Round Robin (WRR) and Locality First (LF) baselines similar to policies used in production. Our results show that: (*a*) Titan-Next can reduce the sum of peak bandwidth across WAN links by up to 61%, (*b*) Titan-Next achieves end-to-end latency close to LF which specifically optimizes for latency. (*c*) It can cut down the number of call migrations (details in §6.2) across DCs by up to 66%.

This work does not raise any ethical issues.

2 Background

2.1 Primer on conferencing services

MS Teams and other large-scale conferencing services (CS) are known to host large volumes of calls globally generating 100s of Gbps of traffic. Each call is assigned a Media Processor (MP) server hosted on a cloud data center (DC) (called cloud region). Each participant in a call could generate up to 3 distinct streams – audio, video, and screen-share, which are sent to the MP server, which in turn processes and forwards the streams to other participants in the call. For capacity, availability, and performance reasons, a large-scale CS usually has MP servers hosted in multiple DCs globally.

2.2 Problem: MP DC selection and routing

The key problem in MS Teams is to determine the MP server for each call along with routing option for each participant, while balancing the user experience and costs. Such a problem has four aspects: (a) **Resource provisioning:** MS Teams is a *first party CS* with access to compute and network resources from Azure. Due to its scale, high uptime requirements, MS Teams requires vast compute and network resources[1] that cannot be met through uncertainty of *pay-as-you-go* models. MS Teams rather provisions dedicated compute (MP) servers in advance. On the other hand, network bandwidth is more readily shareable with other Azure services. Therefore, though the network is provisioned in advance based on the anticipated need of all services including MS Teams, the billing is done based on the *peak usage* of individual services. (b) **MP DC selection:** based on the resources provisioned, MS Teams needs to assign MP DC for each call. Such an assignment needs to take into account the DC-wise compute provisioning, the geo-distribution of the call participants, and call experience. (c) **Route selection:** while MS Teams uses cloud provider's WAN, as we show in the next sections, the Internet provides performance comparable (or better) than WAN for some parts of the world. Thus, MS Teams needs to determine the routing option (WAN or Internet) for individual participants on a call. (d) **MP selection:** Once the MP DC is selected for a call, MS Teams needs to select the actual MP server in that DC to host the call.

Switchboard [16] efficiently addresses (a) and (b) assuming control of the amount of resources provisioned. In this work, we jointly do (b) and (c) assuming the resources are already provisioned. We use state-of-the-art load balancers for (d).

Metrics of interest: The key metrics of interest for MS Teams are: (a) User experience: user experience is sensitive to network latency, loss, and jitter[47]. MS Teams and conferencing services, in general, tackle jitter to a large extent using jitter buffers [15], and network loss (to certain extent) through error correction and error concealment [15]. (b) Compute and network costs: Compute comprises the MP servers in DCs that are already paid for. MS Teams currently uses Azure's WAN where the network cost depends on the *peak usage* (similar to [16, 30]). Azure is one of the largest cloud provider, and MS Teams is among the top services by traffic rate in Azure. Thus, it is imperative to reduce network cost for MS Teams.

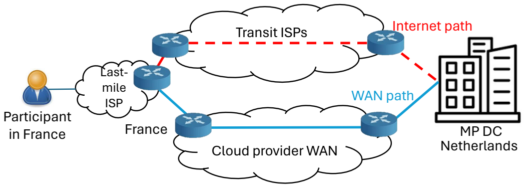
2.3 Internet versus WAN routing

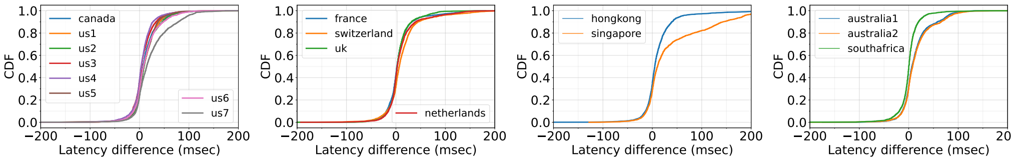
WAN has been a default choice to route the traffic between the cloud VMs and the users in Azure. Large cloud platforms including AWS, Azure, and GCP have started offering an alternate *Internet routing option*, which customers/services could opt for. This is essentially hot-potato routing – egress (likewise, ingress) traffic exits (enters) the Internet closest[2] to the hosted service (Fig.1).

[1] MPs have no/minimal storage requirements. They primarily process and re-distribute streams.

[2] Usually, multiple transit provider options; BGP picks one.

Fig. 1. WAN versus Internet routing. Using WAN routing, the traffic from MP exits the WAN closest to the user (cold-potato). Using Internet routing, the traffic exits the WAN closest to the DC (hot-potato).





(a) North America (b) Europe (c) Asia (d) Australia + Africa

Fig. 2. Comparing latencies for WAN and Internet for 21 Azure DCs in 5 different continents. Negative difference indicates Internet is better. We label latency = RTT. The legends denote the DC locations.

Internet paths are cheaper than WAN up to 53%[5, 6], e.g., GCP charges $0.15 and $0.075 for data transfers per GB[6] using WAN and Internet respectively for Singapore region. This observation motivates us to investigate if a fraction of the MS Teams traffic can be moved to cheaper Internet paths while not compromising on the user experience. Additionally, as MS Teams peak traffic on the WAN is reduced, it provides more bandwidth for other services and reduces long-term capacity provisioning. Lastly, our study makes it evident that the Internet also provides a fall-back option to WAN. We discuss in §4.2 how Internet paths could augment WAN capacity during fiber cuts.

3 Internet paths good enough?

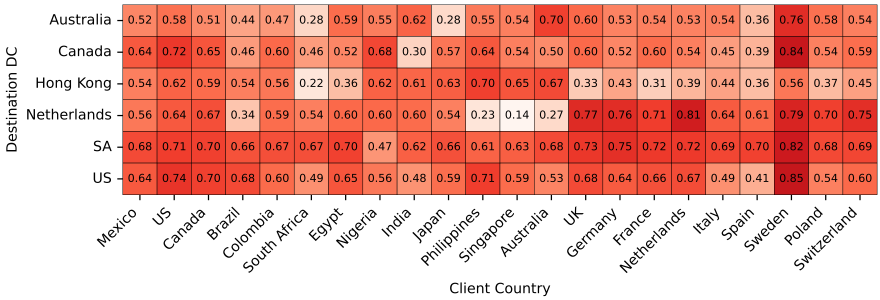
We discuss measurement results contrasting the latency of the Internet and WAN paths.

Methodology: We setup 42 VMs (virtual machines; 2 per DC) in 21 Azure DCs (see Fig.14 Appendix) across the globe. In each DC, one VM uses the Internet path and the other VM uses the WAN path. Both VMs host HTTPS servers that serve a 1×1 image (with some metadata) upon receiving requests from clients. A load-balancer assigns client requests to one of the 42 VMs using round-robin scheduling. MS Teams has multiple 100 million monthly active users. Our latency measurements span 12 months (from June 2023 to June 2024) thus capturing ~1.2 billion measurements. The data is anonymized to remove any Personally Identifiable Information. For each test, the VM (location known) logs the timestamp of the test, /24 masked client IP address, and the request round-trip time (RTT). The client's IP address is translated offline to the client's country, city, and ASN using a proprietary geolocation database having high accuracy. The RTT takes into account only the GET request/response round-trip and disregards any HTTPS connection setup time. *We refer RTT as "latency" for all further analyses.* Table 3 (Appendix) shows the statistics.

Latency analysis: For each hour, we calculate the median latencies between each client country and MP DC pair over Internet and WAN (we also consider finer granularity like ASN instead of country at the end of this section). We take the difference (Internet minus WAN) for these hourly median values from each client country to each MP DC. We choose hourly medians (a) to have sufficient data points, and (b) as median values are well suited for comparison between WAN and the Internet. Also median is a good indicator of expected value in non-normal distributions and resilient to outliers [13, 40]. Fig.2 plots the CDFs of differences for DCs in 5 continents across all client countries for 7 days in June 2024. These results cover source and destination pairs comprehensively around the globe. The key observations are:

(1) In 33.73% cases, Internet is strictly better than WAN.

(2) In 23.98% cases, Internet is worse than WAN by only up to 10 msec.



Destination DC	Mexico	US	Canada	Brazil	Colombia	South Africa	Egypt	Nigeria	India	Japan	Philippines	Singapore	Australia	UK	Germany	France	Netherlands	Italy	Spain	Sweden	Poland	Switzerland
Australia	0.52	0.58	0.51	0.44	0.47	0.28	0.59	0.55	0.62	0.28	0.55	0.54	0.70	0.60	0.53	0.54	0.53	0.54	0.36	0.76	0.58	0.54
Canada	0.64	0.72	0.65	0.46	0.60	0.46	0.52	0.68	0.30	0.57	0.64	0.54	0.50	0.60	0.52	0.60	0.54	0.45	0.39	0.84	0.54	0.59
Hong Kong	0.54	0.62	0.59	0.54	0.56	0.22	0.36	0.62	0.61	0.63	0.70	0.65	0.67	0.33	0.43	0.31	0.39	0.44	0.36	0.56	0.37	0.45
Netherlands	0.56	0.64	0.67	0.34	0.59	0.54	0.60	0.60	0.60	0.54	0.23	0.14	0.27	0.77	0.76	0.71	0.81	0.64	0.61	0.79	0.70	0.75
SA	0.68	0.71	0.70	0.66	0.67	0.67	0.70	0.47	0.62	0.66	0.61	0.63	0.68	0.73	0.75	0.72	0.72	0.69	0.70	0.82	0.68	0.69
US	0.64	0.74	0.70	0.68	0.60	0.49	0.65	0.56	0.48	0.59	0.71	0.59	0.53	0.68	0.64	0.66	0.67	0.49	0.41	0.85	0.54	0.60

Fig. 3. Fraction (F) of times Internet provides better or comparable (within 10 msec) latency compared to WAN. SA denotes South Africa and US denotes the United States. Darker shade means higher F.

(3) In 19.61% cases, Internet is worse than WAN with a latency inflation between 10 and 25 msec.

(4) For the remaining 22.68% cases, WAN latency is better than Internet by more than 25 msec.

As detailed in §5.2, user experience does not degrade substantially with a small increase in latency – we set thresholds to 10 msec and 25 msec accordingly.

Zooming in further: Next, we deep dive into understanding the Internet routing low-latency opportunity for different geographic regions by quantifying the fraction (F) of times (when considering hourly median values for 1 week) Internet paths offer latencies lower than or comparable (≤10 ms inflation) to WAN paths from different client countries to destination DCs. Fig.3 plots the heatmap for paths between 22 countries (spanning 5 different continents; top 20 by call volume and 2 from Africa) and 6 DCs from 5 continents (Orange triangles in Fig.14). The key observations are:

(1) Internet paths often offer lower or comparable latencies in the North America (NA) – Europe corridor. F = 41-85% for Europe to NA and 64-74% for NA to Europe paths.

(2) Europe is well connected (low latency) to European and South Africa DCs over the Internet.

(3) Internet routing between Europe and the DC in Hong Kong performs poorly (F = 31-56%).

Internet routing a viable option: Internet performs well due to the well-provisioned trans-Atlantic fiber connectivity[9] offering similar latency choices to the Internet and WAN. Additionally, Internet provides better performance in some cases due to richer availability of peering points[13].

Stability: We repeat the above experiment with data collected for a week but 6 months in the past (Jan'24). We observe that, in 6 months, the Internet has become slightly better for the NA - Europe corridor, while the broad trends hold true. The figure is available in §A.6 (Fig.19).

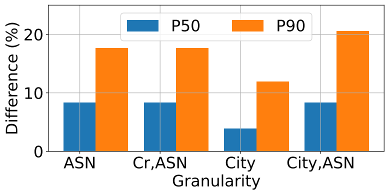


Fig. 4. Difference in F between different granularities and granularity = country. Cr indicates country.

Long-term trends: For both the Internet and WAN paths between the 20 countries (top; by call volume) and all DCs, we measured the weekly median latencies for the weeks separated by 12 months. We found that in 80+% cases latencies have improved for both types of paths. The Internet paths see slightly greater improvements. More details are in §A.4.

Clustering using city and ASN (instead of country): Fig.3 shows the fraction (F) when we cluster the measurements at the country granularity. We do similar analyses at the clustering granularity of cities, ASNs, and city + ASNs. For ASN (similarly other) clustering, we take the weighted difference compared to clustering using country. The weights are the fraction of measurements for each ASN in the country (more details in §A.5). The results (fraction F) do not change significantly compared to the country-level granularity (difference bound to 8% at $P50$) as shown in Fig.4.

In summary, Internet latency is better or comparable to WAN in parts of world. Also, the measurements are stable and the insights hold true even at the granularity of countries thus making us cautiously optimistic about moving a fraction of the MS Teams traffic to the Internet.

4 Titan: Moving calls to Internet

Based on the latency measurements in the previous section and the observation that Internet routing provided sufficiently good latency for MS Teams, we select the client countries and MP DCs in *Europe* as candidates to move some of the traffic to the Internet. In this section, we detail Titan – our system that carefully moves a fraction of the traffic from WAN to the Internet without hampering user experience. Titan has been in production for the last 12 months.

4.1 Determining fraction of traffic to move

The key challenge when moving MS Teams traffic to the Internet is that we do not know the capacity of such paths in advance. Naïvely moving all MS Teams traffic to the Internet could cause network congestion and, hence, a poor user experience. We prioritize *safety* over optimality – we stop moving traffic to the Internet, even if there is no performance degradation. Our design for carefully moving traffic involves the following key elements.

(1) **Granularity:** Titan moves traffic to the Internet at various levels of granularity, from a small number of users, metro, ASN to the country level. As MS Teams is among the top services on Azure with 100s of millions of users, we cautiously start with a fraction of small communities of MS Teams users and move fraction of an entire country if the performance is acceptable.

(2) **Variable traffic allocation:** For each combination of client country and MP DC, we typically increment 1-3% (based on domain knowledge) of the traffic, at a time, to the Internet. After each move, we monitor the performance metrics for a few days for stability and make quick adjustments (detailed next) after observing negative effects. Otherwise, we repeat the process, moving more traffic for the pair to the Internet. We currently stop at 20% based on operational expertise. Each MP DC is connected to Internet via multiple transit providers. When calculating % movement: (a) we consider the minimum capacity available on Azure links peering with the transit providers. (b) we assign different priorities to client countries (based on source traffic volume) and split available (minimum) capacity across client countries depending on their priorities. We observe for a longer duration to maintain and assess the stability of the chosen path. By waiting longer, we can monitor the performance even during changes on Internet outside Titan, and to ensure that MS Teams has sufficiently high performance during such changes.

(3) **Quick reaction to poor performance:** We continually monitor the network metrics as calls progress. We also collect MOS (Mean Opinion Score; user feedback) with a 5-point likert scale at the end of a subset of calls. We decrease/stop moving traffic to the Internet when performance criteria for some of the network/application metrics (including network latency, loss or jitter, or application MOS) are not met. We do so instantaneously. We also reduce *oscillations* when determining traffic to be moved to the Internet by waiting for longer duration. We only move the traffic if Internet provides good performance for longer durations.

While determining points of congestion on the Internet is a challenging task (and a subject for further study), we have a few knobs to react to poor performance quickly: (a) If a large fraction of calls over the Internet in a client country – MP DC mapping shows moderate performance degradation across some of the metrics (e.g., P50 packet loss $\leq$ 1%, latency inflation $\leq$ 10%), we decrement traffic on Internet for that client country – MP DC pair. (b) *emergency breaks:* If there is a severe performance degradation (e.g., P50 packet loss $\geq$ 1%, rare), we reroute traffic over the WAN., (c) If only a few users are facing problems, we move them selectively to WAN as detailed in §6.4. (d) If high degree of unavailability is observed from one MP DC to a transit ASN, network automatically mitigates by failing over to alternative transit peer. Also poor performance might not be because of the traffic offload, it just indicates that the Internet might not be good enough for user experience. If mitigation is not complete due to some constraints, we will detect the anomaly and failover to the WAN.

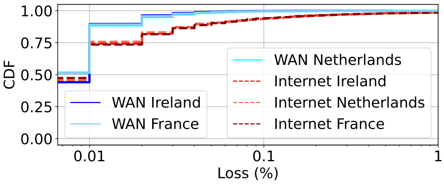


Fig. 5. Packet loss for Internet and WAN.

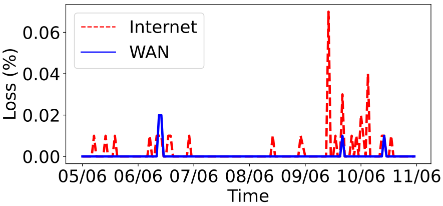


Fig. 6. Packet loss between France and DC in the Netherlands.

(4) **Traffic to be moved:** With Titan, we randomly select the call participants to be assigned the Internet paths, constrained by the fraction chosen for the client country and the target MP DC. Such an assignment easily scales to millions of daily calls. However, as we show in the next section, Titan-Next further improves the cost and performance compared to Titan.

Using Titan, we moved substantial traffic to Internet, saving substantial costs.

Overall, the Titan system involves a combination of A|B testing, performance analysis, rule-based control, and the ability to adapt dynamically based on network conditions.

4.2 Titan production findings

We now detail our production experiences while moving large-scale traffic from WAN to Internet.

(1) The Internet has worse loss (in general): We pick 3 DCs in Europe (Ireland, Netherlands, and France) for which a fraction of the MS Teams traffic is moved to the Internet, and log the average loss reported by RTP [44] (using missing sequence numbers) for each call participant in Europe. For each client country-MP DC pair, we find the hourly median loss for 7 days between 5^{th} and 11^{th} June'24. Fig.5 shows the CDFs of hourly median loss for traffic between the 3 DCs and all client countries in Europe over WAN and the Internet. While it is evident that loss rates are low (≤ 0.01%) for a large fraction of both the Internet (average of 44.9% across DCs) and WAN paths (49.2%, likewise), the Internet paths have higher loss rates at the tail. For ~10% cases, the Internet paths could experience at least 0.1% loss, while such loss over the WAN is almost non-existent.

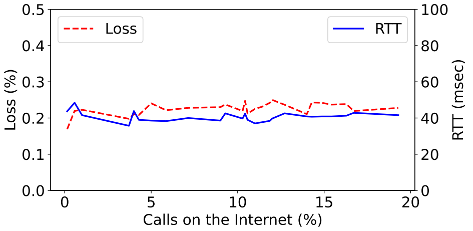


Fig. 7. Loss, RTT vs. fraction of traffic on Internet between the UK users and Netherlands DC.

(2) The Internet has more loss spikes: Fig.6 shows the time series for hourly median loss rates between the Netherlands DC and clients in France. Internet paths have higher (up to 3×) and more frequent loss spikes than WAN, with the peak loss rates for the latter limited to only 0.02%. While MS Teams can mitigate loss to a some degree by using application layer redundancy mechanisms, one should be cautious in moving traffic to the Internet so as not to risk performance degradation due to inflated losses. The trends are similar for other client country – MP DC pairs as shown in Fig.15 (§A.2). We measure the number of 30 minute timeslots over 7 days observing at least 0.1% (and 1%) loss on Internet and WAN for all client countries – MP DC pairs in Europe. Fig.15 in §A.2 shows that Internet has more frequent loss.

(3) Internet has higher jitter: We observed that the Internet has slightly worse jitter than WAN up to 10%. We observed mean jitter of 3.4 msec and 3.52 msec for WAN and Internet paths in North America region. Note that, MS Teams uses jitter buffers[15] and this additional jitter on Internet does not affect performance. In addition to the jitter buffer, MS Teams also has redundancy at multiple levels including the codec, which is why we did not face any issues with network jitter.

(4) The Internet is reasonably elastic: Fig.7 shows the loss and RTT vs. fraction of traffic moved to the Internet between clients in the UK and the MP DC in the Netherlands. Note that, even

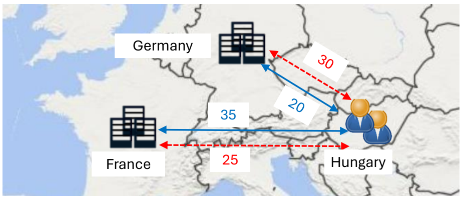


Fig. 8. Benefits of joint optimization. There is a call with users in Hungary with potential MP DCs in France and Germany. Blue and red arrows indicate WAN and the Internet paths; numbers indicate latency (msec).

when 20% of MS Teams traffic is moved to the Internet, neither packet loss rate nor latency shows any systematic inflation. We repeat the same experiment for all client country – MP DC pairs in Europe where we moved 20% traffic. The median change for latency and loss are 3 msec and 0.06% (Fig.16 in §A.3). These results show that Internet is reasonably elastic for a large number of client country – MP DC pairs, demonstrating Titan not adding significantly to the Internet congestion. For some countries though, we could not use Internet due to poor performance even at smaller shifts. Note that at fractions higher than 20% (not tried in production), there is a chance that we congest Internet paths by routing substantial amount of traffic thus inflating loss and latency.

(5) When the Internet is not an option: We have observed Internet paths for some client countries (e.g., Germany, Austria) with high (and unacceptable) loss even when a small amount of traffic was moved. In such cases, we do not use the Internet at all. At the same time, for some MP DCs, we have observed high transient loss from a few client countries, which was affecting user experience. Thus, we stopped moving traffic to Internet for those MP DCs, and relied on the WAN.

(6) Congestion *likely* at the transit ISPs: We detected variations in performance across transit ISPs between DCs and users (Fig.1). We observed higher packet loss along end-to-end paths between an MP DC and multiple ISPs simultaneously. Such one-to-many loss patterns, with no corresponding loss inflation observed at the DC or the WAN, hint at congestion at the transit ISPs. We had to react to performance degradation by steering traffic to alternate transit providers.

(7) Internet as a scaling or fall back option: traffic got moved to Internet for example in instances of surge of traffic for Teams while WAN capacity becomes available, or during capacity crunch on the backbone in case of outage (e.g., fiber cuts).

Lastly, Titan can also be extended to third-party services. Like MS Teams, we could shift traffic for any other service and monitor the performance. However, unlike MS Teams (first-party service), we do not have visibility into application performance for third-party services. We need the applications to report application-level performance. We leave it for the future work.

5 Joint assignment in Titan-Next

As quantified in the previous section, we can move a subset of the MS Teams traffic to the Internet with fairly good performance. At the same time, the cost of using a WAN link depends on the *peak* usage[16, 30]. Therefore, the total MS Teams traffic needs to be split across WAN and Internet to reduce peaks on WAN (and operating costs) while not impacting performance.

Titan assumed that the MP DC assigned to the call is fixed (assigned outside Titan). Titan calculates fraction of traffic on Internet paths, and simply assigns participants to Internet paths *randomly* based on the fractions. However, MP DC assignments done outside Titan may likely be unaware of the Internet offload opportunities and may make sub-optimal assignments (§5.1). We build Titan-Next to make efficient use of the Internet paths by jointly assigning the MP DC and the routing option to individual calls. We next discuss the two key ideas in Titan-Next.

5.1 Joint MP placement and routing

A strawman's approach could be to use Switchboard [16] (based on locality) to first calculate the MP DCs for calls and then move traffic to the Internet if such paths have capacity. However, such a scheme may result in sub-optimal assignment as shown in Fig.8. Imagine a call with 2 users in Hungary and potential MP DCs in France and Germany. The latencies for Internet and WAN

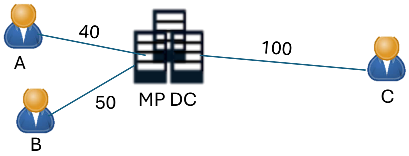


Fig. 9. E2E latency is important in MP DC and routing assignment.

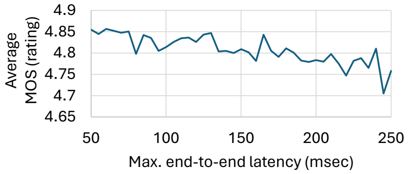


Fig. 10. Impact of max. E2E latency on user experience (5-point likert scale).

paths are shown in the figure. Switchboard, synergistic to locality (nearest DC) may select the DC in Germany by looking at the WAN latencies. Once this DC is picked for MP allocation, we then move the call to the Internet if there is capacity. But this scheme will result in a latency of 30 msec. However, assigning the call to the DC in France with the Internet routing option could have provided lower latency (25 msec) while reducing WAN overhead. Alternately, we may formulate Switchboard to use the Internet paths when calculating MP DCs, and then move calls to use WAN paths, which, similarly, is also sub-optimal. Thus, instead of assigning MP DCs and Internet routing separately, in Titan-Next we *jointly make such assignments* toward minimizing WAN peaks. In doing so, we continue to provide same bandwidth to the users at lower or comparable latency.

5.2 Optimizing for end-to-end latency

The experience of any two users engaged in a conversation depends on the end-to-end (E2E) latency between them (e.g., 50 + 100 = 150 msec for users B and C in Fig.9). Under the assumption that any two users can engage in a conversation during a call, the *maximum* E2E latency across all participants determines the participant engagement and, hence, the user experience in such calls.

To understand the impact of max. E2E latency on user experience for MS Teams, we leverage the MOS (Mean Opinion Score; user feedback) as a measure of the Quality of Experience seen by the user (In contrast, [16, 47] do not show impact of E2E latency on MOS). MS Teams telemetry also collects the latency between MP and the participant. Based on the latency reported for each participant, we calculate the max. E2E latency and group all assigned MOS for 5 msec buckets of max. E2E latency. Fig.10 shows the average MOS for increasing max. E2E latency. We select the range of 50-250 msec as we have significant measurements (at least 1,000 points for each 5 msec bucket). *Note that MOS is collected at the end of a subset of calls and is heavily sampled.* It can be seen that: (a) For max. E2E latency under 75 msec, the impact on MOS is minimal indicating that users are tolerant of E2E latency up to a certain extent. (b) The user experience degrades (mostly linearly) with an increase in the max. E2E latency. Thus, service operators are keen to keep this E2E latency low. We take max. E2E latency into consideration when assigning the MP DCs and routing options to individual calls to bring users (virtually) closer and improve user experience.

6 Titan-Next design

Inputs: Titan-Next needs to assign the MP DC and routing option when the call starts, *i.e.,* when the first user of the call joins[3] based on the location (country) of the first joiner. If needed, The MP DC and routing option could be changed later (§6.2).

The other inputs used by Titan-Next are: (*a*) the numbers of available MPs in individual DCs (fixed; discussed in §2.2), (*b*) rich history (participant's country, media types, time of the call; anonymized) of calls to predict the peaks and make assignments accordingly, (*c*) Internet path capacities for each client country - MP DC pair as recorded by Titan, and (*d*) WAN topology and Internet peering points (MS Teams is a first party conferencing service with access to such details).

[3] We cannot do the assignments when the second or subsequent participants join as there are calls with single participant that use other features of MS Teams (e.g., video recording, transcript) that requires MP assigned.

Call config: We want to assign the MP DC and routing option for each individual call. To do such assignments at scale, we borrow the notion of *call configuration* (call config, for short) from Switchboard [16], which captures the resource requirements of calls through the number of participants and media types of different calls. There are 10s of thousands of call configs in a day – more details and analysis are in [16]. A call config comprises (1) the location (country) of the participants, (2) participant count from each country, and (3) media type (audio, video, or screen-share). A call can have any of the above media streams, but we assign call config using the most resource-hungry media type (audio < screen-share < video). An example call config is *((France*-2, *UK*-1*), Audio)* which represents all audio-only calls with 2 participants from France and 1 from the UK. All calls with the same call config are fungible – they have largely the same resource requirement. The number of call configs is orders of magnitude fewer than the number of calls that helps scale the LP (details in §6.3).

6.1 Titan-Next building blocks

Fig.11 shows the building blocks in Titan-Next. In a nutshell, Titan-Next pre-computes an offline assignment plan, based on the expected (predicted) call demand. It uses such a plan to assign calls at run-time. This way, using prediction, the offline plan can do the assignment to minimize peaks in the network links. Titan-Next has five modules:

(1) **Call records database**: MS Teams records and stores some data (anonymized) for each participant of the call including the start time, media type, time of the call, MP DC country, and the latency experienced by the user (client-to-MP). Titan-Next uses these call records to forecast demands as well as to calculate latencies of call participants.

(2) **Call count prediction**: Titan-Next assigns the MP DC and routing option for (reduced) call configs (explained next). To do so, for each call config, Titan-Next uses Holt-Winters exponential smoothing[4] to forecast the number of calls for the next 1 day at the granularity of 30 min timeslots. Titan-Next predicts for the top 3,000 call configs covering 90+% of all calls (to finish prediction quickly). For each 24 hour prediction with high accuracy (§8.3), we use 4 weeks of training data.

(3) **Call config grouping:** The difference in MP DCs assigned to call config vs. assignment using the country of the first joiner may lead to call migration from one MP DC to another. To reduce such migrations, we transform all calls to their *reduced call config* and group all call configs falling into the same reduced call config (discussed in §6.2).

(4) **Offline precomputed plan:** Using the forecasts, reduced call configs, and network database consisting of WAN topology and Internet path capacity, this module assigns the pairs of MP DC and routing option for each reduced call config for each 30 minute timeslots for next 24 hours. We formulate it as a Linear Program (LP) to minimize the sum of peaks on individual WAN links (§6.3).

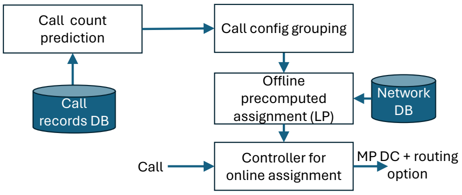


Fig. 11. Building blocks in Titan-Next

(5) **Controller for online assignment:** Given the pre-computed assignment from (4) above, this module assigns the MP DC and routing option to each incoming call when the first participant of a call joins. We use a combination of offline pre-computed plan and the country of the first participant. Moreover, we might need to migrate the call to another DC if the initial assignment is not according to the pre-computed plan. The controller is very fast when assigning the MP DC and routing option and does not cause any performance degradation.

6.2 Reducing call migrations

Titan can make MP DC assignment using LP in Switchboard[16]. Imagine a call where the first joiner is from Germany, and the LP has a pre-computed plan for both (Germany-2, Audio) and

(Germany-3, Audio) call configs. Because the LP makes decisions for individual call configs, such assignments might end up assigning different MP DCs for these two call configs. E.g., (Germany-2, Audio) is assigned to Ireland and (Germany-3, Audio) is assigned to France. When the first user joins, we do not know the call config (we only know the country of the first joiner). Yet, we need to assign the MP DC, either Ireland or France. Let's say we assign the call to Ireland and as the call progresses, we realize that the true call config is rather (Germany-3, Audio) which ought to be assigned to France to adhere to the pre-computed plan. Thus, we need to *migrate the call* from Ireland to France. Such migrations are undesired as they result in user-perceived glitches.

We minimize migrations in Titan-Next using the following mechanism: we first *transform* call configs to factor out the *scale* (number of participants from one country) from the *distribution* (participants across countries in call config). In general, we transform the call config such that the number of participants from each country in the call config has a GCD of 1. For intra-country calls, we transform the call configs to just have 1 participant (e.g., (Germany-2, Audio) is changed to (Germany-1, Audio)). We keep the resource requirement the same. E.g., let's say there are 100 calls with config (Germany-2, Audio). We transform it to 200 calls with config (Germany-1, Audio). We call these new configs *reduced call configs*. We then *group* together all calls based on reduced call configs (e.g., (Germany-2, Audio) and (Germany-3, Audio) are grouped together using the reduced call config (Germany-1, Audio)). We do not group call configs across media types as they have different network and compute requirements. The LP makes decisions at the granularity of this reduced call config. Reduced call config does not affect the correctness of the LP as such configs keep the resource demands unchanged. This way, we significantly reduce call migrations due to differences in assignments for call configs from the same country. However, as detailed in §6.3 and §6.4, we do not eliminate migrations due to differences in media types and for international calls.

Lastly, to migrate the call, we assign a new MP and have the clients connect to that MP. We minimize glitches using redundancy in codec and jitter buffers.

6.3 MP and routing assignment

We now detail our LP to jointly calculate the MP DC and routing option for the reduced call configs. The LP is shown in Fig.20 with notations described in Table 4 in Appendix.

LP objective: The objective of the LP is to minimize the sum of peaks on the WAN links (leveraging Internet paths and MP DC selection). The peaks are calculated for 24 hours period. This directly reduces network costs for MS Teams.

Frequency: We run the LP every 30 min (with fresh estimates) that calculates the assignments for the next 24 hours (to make assignments aware of the daily peak) in 30 min time-slots. This approach: (a) effectively reduces the WAN traffic peak while keeping traffic on the WAN during off-peak hours and (b) by running every 30 min, it adapts the assignments to fresh information about the fraction of traffic on Internet calculated by Titan. The LP does not have details about the run-time conditions (loss and latency) for the participants on the individual calls apriori. Titan-Next adapts to these run-time conditions as detailed in §6.4.

LP variable: The LP variable is $X_{t,c,m,p}$ that indicates the number of calls for reduced call config c in timeslot t assigned to the MP DC m over the path p. We have two options for p for each MP DC – the Internet vs. WAN. When assigned to WAN/Internet, the underlying routing algorithm (outside the scope of Titan-Next) decides the path to the destination.

Constraints: We have five constraints as follows:

(C_1) Total number of calls: For each reduced call config c and timeslot t, we assign MP DC(s) and paths (possibly multiple combinations) to all calls of that config.

(C_2) Compute capacity: For each MP DC, we assign the calls such that the total compute capacity of the MP DC is not exceeded in each timeslot. The $computeUsed()$ function (Fig.20) returns the compute required for c based on its media type and the number of participants. $Cap_{t,m}$ denotes the compute available in the MP DC m in timeslot t.

(C_3) Internet capacity: We assign m and p such that the Internet capacity is not exceeded for any of the Internet paths and timeslots. We calculate Internet capacity as the egress capacity times fraction of traffic moved in Titan. We estimate the Internet path usage by multiplying the fraction of calls to be moved to the Internet, the total number of participants, and the average network usage per participant. The $networkUsed()$ function estimates the bandwidth consumed by c, m, p using its media type and the number of participants. Note that WAN is provisioned to handle *all* MS Teams traffic even if no traffic is assigned to Internet. Thus, we do not have any constraint on the WAN capacity. Going forward, we gradually reduce (or release for other applications) WAN capacity as traffic is moved to Internet. We leave it for the future work.

(C_4) End-to-end (E2E) latency: We do the assignments so that *average* of max. E2E latency across call configs is bounded. $E2Elatency()$ function returns the max. E2E latency given a combination of c, m, and p. We tried putting a bound on maximum of max. E2E latency for each call config, but we observed that such a bound is stretched due to a handful of configs and is not useful for a majority of the configs. Thus, we choose to use a bound on average of max. E2E latency across call configs.

(C_5) Denoting y_l: This constraint ensures that y_l is set to peak link utilization on WAN link l across all timeslots. $isLinkUsed()$ denotes whether l is used.

6.4 Real-time call assignment

Initial assignment: The precomputed plan above assumes full knowledge of the reduced call config. We need to assign the MP DC and routing option when the call starts, *i.e.,* when the first user of the call joins. However, we do not know the full call config when the call starts. We address this challenge as follows: recall that we transform each call config to a reduced version, and a reduced call config can still have different assignments for different media types. For example, (Germany-1, Audio) could be assigned to Ireland, while (Germany-1, Video) is assigned to France. For a new call, we assume it as an intra-country call (such calls are in majority) and pick the *most recently used* reduced call config based on the country of the first joiner. We then use all the counts for each assignment (MP DC and routing option) calculated by the LP for that reduced call config as weights and use weighted random to pick the assignment.

Migration to different MP and routing option: It may happen that even the reduced call config turns out to be incorrect as the call progresses. Consequently, the choice of the MP DC and routing option could also be incorrect. In such cases, we *migrate* the call across MP DCs and routing options. To do so, we wait for 5 minutes (configurable) into the call for the reduced call config to converge (e.g., initial call config = (Germany-1, Audio), while the converged call config = ((Germany-1, France-1), Video). If the MP DC and routing option for reduced call config (after 5 minutes) is different than the initial assignment, we migrate the call to the target assignment.

There are issues in handling surge in calls, or changing paths in real-time as discussed in §D.

7 Ideal: Oracle-based evaluation

We evaluate Titan-Next in this section and the next section. In this section, we assume that we have a ground truth oracle that gives us the start times, participant locations, and media types of all calls. We do this to decouple the impact of the prediction error and carve out the utility of Titan-Next in oracular settings. Next, in §8, we evaluate Titan-Next using prediction output.

7.1 Metrics of interest

We have four metrics of interest: (a) *Sum of peak network bandwidth (BW) on the WAN links*: Peak network BW (in Tbps) on individual WAN links impacts the network capacity to be provisioned on those links to sustain such peaks. Additionally, peak network BW also drives the network costs[16]. Thus, we want to lower the sum of peak WAN BW. (b) *Total traffic on WAN links*: Peak network

BW does not consider traffic during remaining non-peak times. Thus, we consider total traffic on WAN links across time (peak and non-peak). (c) *E2E latency*: user experience depends on the max. E2E latency (§5.2). Hence, we evaluate Titan-Next using such latencies. (d) *Number of call migrations*: As mentioned in §6.4, we may need to migrate the call if the initial assignment of MP DC and routing option is not according to the pre-computed plan. In this section (using a ground truth oracle), the number of migrations is none as the call config is known as apriori. We relax this assumption in the next section, where we assign the MP DC based on the country of the first user.

7.2 Evaluated policies

In addition to Titan-Next, we consider the following three baselines.

Weighted Round Robin (WRR): WRR[16] is easy and practical to implement, and optimizes for compute by balancing calls over multiple MP DCs in the same region. We create buckets for distinct combinations of MP DCs and routing options. Each bucket gets its weight based on its share of compute and the fraction of calls on the Internet. When there are multiple countries in call config, we pick the minimum fraction of calls from its countries. E.g., two DCs have compute capacity as 10K and 20K cores. If the minimum (Internet) fraction for client countries to those DCs are 20% and 15%, then we calculate capacities (weights) as $8K{:}2K{:}17K{:}3K$ for buckets. We select {MP DC, routing option} bucket using such weights.

Locality First (LF): In this policy[16], we assign the MP DC and routing option so as to minimize total latency. We formulate it as a Linear Program (LP). The LP variable is the same as used in §6.3. The objective, rather, is to minimize the total latency. The constraints also match the constraints in §6.3. We do not include end-to-end latency to avoid it affecting the objective (that anyway considers latency). We also consider a variant that optimizes total max. E2E latency.

Titan: Titan selects MP DC through weighted random policy where weights are set in proportion to the number of cores in MP DCs. It then randomly selects calls from each source country – destination MP DC pairs based on the capacity calculated in §4.

Titan-Next **(TN):** We use the LP described in §6.3 to jointly calculate the MP DC and routing option using the oracular ground truth. We use COIN-OR[1] to run the LP.

Switchboard is not a baseline: (a) It provisions resources months in advance, while Titan-Next works within already provisioned limits, and (b) it does not assign routing options or reduce migrations. These are complimentary systems.

7.3 Data sources

(1) **Latency:** We use the latency between participant and destination countries obtained using our measurements in §3, (2) **Capacity of the Internet paths:** We estimate the Internet path capacities as derived in Titan (§4), (3) **Number of calls per call config:** Our telemetry framework logs the number of calls per config. This data is used as the ground truth in this section.

For the evaluation (in this and the next section), we consider all calls that are contained within Europe, *i.e.,* where all participants of the calls are in Europe. Titan has been up and running in Europe for many months with reasonably stable and tested Internet path capacity estimates.

7.4 Comparing WAN bandwidth

Reduced sum of peaks: Fig.12 shows the WAN sum of peaks bandwidth (BW) for each day of a typical week. On weekdays, when the number of calls is high, Titan-Next reduces the WAN BW by 24-28% compared to WRR, and 13-19% compared to LF. LF places the calls to the MP DCs nearest to users, which in turn reduces the number of hops and WAN BW. However, it is not WAN traffic peak aware. In contrast, TN is peak aware and picks the MP DCs and Internet routes for call configs intelligently. The savings in TN are dominated by current limit on Internet offload (max. 20%). In reality, some countries had 5-15% traffic on Internet due to performance deterioration. Moreover, some countries are yet to be flighted to Internet offload in Europe.

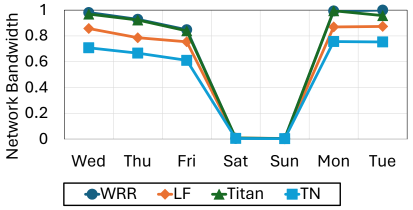


Fig. 12. Sum of peak bandwidth (BW) on individual links in WAN used by WRR, LF, Titan and Titan-Next (TN), calculated each day of the week (§7.4). The BW is normalized to peak BW for WRR.

Total WAN traffic reduction: We see similar savings for total WAN traffic (in PB) in TN with Internet offload calculated by Titan. On weekdays, TN cuts the total traffic by 24-28% and 13.5-18% compared to RR and LF respectively.

Savings with only MP DC placement: Recall that the gains in TN are due to: (a) using Internet paths, and (b) using flexibility in selecting MP DCs. To dissect the gains, in this experiment, we do not assign any calls to Internet for TN only to evaluate the benefits from intelligently assigning MP DCs only. We do so by setting the Internet capacity to 0 in the LP. This way all the calls are forced to use WAN, while still benefiting from flexibility in choosing the MP DCs. Note that we continue to use an objective to minimize sum of peak BW on WAN. On weekdays, savings in TN compared to WRR and LF reduced to 16.7-20% and 3-8%. These savings are purely due to MP DC selection. Remaining savings in TN compared to previous experiment are due to Internet offload.

More savings with more traffic on the Internet: As mentioned in §4, we are so far conservative in calculating traffic on the Internet. In this experiment, we evaluate the savings with TN if we were to hypothetically double the traffic on the Internet. To do so, we double the Internet capacity for each path and rerun the LP. We observe that the savings in TN increase compared to the setting with the original Internet capacities. TN reduces the sum of peak bandwidth by 27-38% and 17-26.5% compared to WRR and LF respectively (weekdays). Interestingly, the savings in TN did not double because we did not have enough traffic on some of the WAN links.

Table 1. Daily average of max. E2E latency across calls (in msec) for WRR, LF, and Titan-Next.

	Mean	Median	P95
WRR	82 - 86	75 - 78	120
LF	71 - 75	70	100 - 103
Titan-Next	74 - 80	70 - 76	103 - 122

LF using E2E latency: We consider a variant of LF minimizing total max. E2E latency across configs. We set the objective in LP accordingly and added a constraint that WAN traffic is less than the capacity on each link (similar constraint on the Internet paths is unchanged). TN reduces peak bandwidth against such a policy by 16-29% (weekdays).

While we discuss results for a specific week here, our broad observations hold true across weeks.

7.5 Comparing end-to-end (E2E) latency

Table 1 shows the daily average of max. E2E latencies across all calls for all three policies. WRR is not optimized for latency. LF is specifically optimized for total latency. In contrast, TN is optimized for network bandwidth (BW) (sum of peak BW on individual links) with a constraint on average of max. E2E latency ($E = 80$ in Fig.20 for weekends and $E = 75$ for weekdays). To provide the best user experience, we keep the max. latency constraint that is feasible (where ILP does not run into infeasibility). Despite TN not being optimized for latency, it achieves latency better than WRR and slightly worse than LF. In TN, the network BW savings were roughly the same for all values above those values of E. This shows that TN can significantly reduce WAN BW compared to LF with a small permissible penalty in E2E latency.

8 Practical: Prediction-based evaluation

In this section, we do not assume ground truth information. We assign the MP DCs and routing option *using country of the first joiner* as detailed in §6.4. The Titan-Next controller makes such assignment using offline precomputed plan using the prediction output.

8.1 Evaluated policies

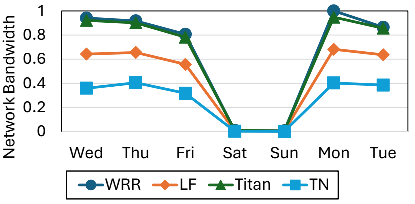


Fig. 13. Sum of peak bandwidth (BW) on individual links in WAN used by WRR, LF, and Titan-Next (TN) calculated for each day of the week (§8.2). The BW is normalized to the peak BW observed for WRR.

We cannot use the WRR, Titan and LF versions from the previous section, as they assume knowledge of the ground truth. Hence, we modify the baselines to select the MP DC and routing option based on the location of the first user. We evaluate: **(1) WRR:** We create the buckets so that each bucket has a distinct combination of MP DC and routing option. We assign the weights to the bucket in the same way as described in §7.2, but they are based on the *country of the first user.* **(2) LF:** We sort the MP DC and routing option buckets in descending order based on the latency from the country of the first joiner and pick the first bucket with enough capacity. **(3) Titan:** We pick the MP DC and routing option bucket using weighted random based on the country of the first joiner. **(4) Titan-Next (TN):** We use the TN controller that performs real-time assignments using a precomputed plan. We train the Holt-Winters time-series prediction model with 4 weeks of data to predict the number of calls for individual call configs for the next 24 hours at the granularity of 30 min. We feed the prediction output to the LP. We assign the MP DC and routing option as described in §6.4. For all the policies, we ensure that traffic does not exceed WAN or Internet bandwidth.

8.2 Comparing WAN bandwidth

Fig.13 shows the sum of peak bandwidth (BW) on WAN. TN reduces such BW by 55-61% and 38-44% on average compared to WRR and LF respectively. The key reason is that LF and WRR no longer have prior knowledge of call configs; they do not know the future call demand. Thus, some of the calls arriving early take the preferred slots while the later calls are assigned far away. TN is peak-aware and uses flexibility in picking the MP DCs and routing options using call history.

8.3 Accuracy of prediction

Fig.18 (Appendix §B) shows the RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) in prediction using the Holt-Winters method. Recall that we predict the number of calls for call configs (not reduced call configs). We measure the error for each call config, normalize it to the peak values, and plot the CDF. This way elephant and mice call configs are treated equally. The median errors are small – 4.9% and 10.6% for MAE and RMSE.

8.4 Reduction in migrations

Table 2. Percentage of calls that need to be migrated.

With call config	With reduced call config
11-34% (average = 31%)	11-19% (average = 15%)

As mentioned in §6.4, we may migrate the call due to differences in the MP DC and routing option assignments few minutes into the call. As discussed in §6.3, we can reduce the number of migrations by using reduced call configs. We compare the volume of call migration needed with and without this approach – in the former case we feed the reduced call config to the offline LP, while in the latter case, we feed the call config as-is. We only consider inter-DC migrations (not due to routing option changes) as they are more damaging. Table 2 shows that reduced call configs cut down the migrations by 38-66% on weekdays (when the number of calls is high). We leave reducing migrations further to future work.

8.5 Titan-Next overhead

The prediction block runs once a day, grouping and LP blocks run every 30 mins, and the controller runs for each call. The prediction building block takes 1.2 – 4.7 seconds per call config. The entire prediction finishes in ~82 min on a single core. The prediction pipeline is embarrassingly parallel and could span multiple cores to scale, as needed. The call config grouping is very fast (finishes in under a minute). The LP takes roughly 1 min. Lastly, the controller is again very fast and assigns the MP DC and routing option within 1 msec per call.

9 Related work

Conferencing: Conferencing services such as Microsoft Teams[7], Zoom[10], Google Meet[3], DingTalk[2], and others have received considerable community attention[18, 19, 31, 37, 50]. Some of the recent work include: (a) resource management[16], (b) network condition based video quality adaptation[35], (c) low latency video transport network[34, 38], and (d) codec and transport collaboration[24, 54]. In contrast, Titan-Next focuses on intelligently reducing costs for these services by leveraging the Internet.

The Internet vs. WAN: Prior works[13, 50] have analyzed performance of Internet and WAN. Compared to/unlike [13], (1) we cover more DCs (11 vs. 21), (2) we have significantly more latency measurements (88K vs. 3.5M) spanning almost the entire globe (241K+ cities), (3) our measurements are piggybacked on MS Teams that report end-to-end latency observed by the individual users ([13] uses only 800 Speedchecker Vantage Points), (4) we measure loss and jitter using a large-scale application (MS Teams) that provide measurements closer to what users *actually* experience, and (5) we find that Internet's performance is better than WAN for significantly larger fraction of measurements. [50] analyzes network performance data for source-destination pairs from 11 DC locations for 1 day. [50] does not reveal any information about the client and cloud location. Our study has 21 DC locations and clients all over the world for close to a year. Unlike [50], we found lower loss on Internet in Europe and North America regions motivating us to move traffic to Internet. [17, 21, 39, 42] shed light on Internet performance. [33] focuses on comparing different cloud providers. Titan-Next is orthogonal – it compares performance metrics for WAN vs. Internet at a significantly larger scale, and uses the insights to select modalities for MS Teams traffic.

Leveraging multiple communication paths: [22, 50] are impressive works that (like Titan-Next) use both WAN and Internet paths. [22] does not handle MPDC placement and [50] keeps switching between Internet and WAN and we prefer not to do it to avoid packet reordering. [28, 51] focus on consuming multiple paths using MPTCP. Such works are complimentary to Titan-Next; they do not consider MP server and routing joint assignment. cISP[14] uses free-space speed-of-light radio connectivity. SCION[48] argues for path-aware routing. Titan-Next leaves it as future work.

Traffic engineering (TE): Many of the TE solutions ([11, 12, 26, 27, 29]) work for WAN. [43, 52] focus on TE for Internet. Such works do not have flexibility of choosing the end-points (MP DCs).

Server (MP) selection: Like MP selection problem in Titan-Next, prior work to study server/DC/replica selection[20, 23, 25, 32, 36, 49, 53]. [45, 46] focus on replica selection to improve the tail latency. In contrast, Titan-Next focuses on joint MP and routing option selection.

10 Conclusion

It is important for large conferencing services like MS Teams to continue offering a good user experience at low costs. In this paper, through large-scale measurements spanning almost the entire globe (241K+ cities), we show that Internet paths provide similar or better latencies in many parts of the world. We present: (a) Titan (running in production) that calculates the MS Teams traffic that could be *safely* offloaded to Internet using latency measurements, and (b) Titan-Next (research prototype) that jointly assigns the server location and routing option to MS Teams calls. Together they cut down the WAN sum-of-peak bandwidth, which determines the network cost, by up to 61%.

Acknowledgements

We thank the reviewers and our shepherd for their helpful feedback that improved the paper.

References

[1] COIN-OR LP solver. https://www.coin-or.org/.

[2] Ding Talk. https://www.dingtalk.com.

[3] Google Meet. https://apps.google.com/meet.

[4] Holt-Winters exponential smoothing. https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html.

[5] Internet path pricing in Azure. https://azure.microsoft.com/en-us/pricing/details/bandwidth/.

[6] Internet path pricing in GCP. https://cloud.google.com/vpc/network-pricing.

[7] Microsoft Teams. https://www.microsoft.com/en-us/microsoft-teams/group-chat-software.

[8] Microsoft Teams user growth. https://www.businessofapps.com/data/microsoft-teams-statistics/.

[9] Submarine cables. https://www.submarinecablemap.com/.

[10] Zoom. https://zoom.us/.

[11] F. Abuzaid, S. Kandula, B. Arzani, I. Menache, M. Zaharia, and P. Bailis. Contracting wide-area network topologies to solve flow problems quickly. In *USENIX NSDI 2021*.

[12] S. S. Ahuja, V. Gupta, V. Dangui, S. Bali, A. Gopalan, H. Zhong, P. Lapukhov, Y. Xia, and Y. Zhang. Capacity-efficient and uncertainty-resilient backbone network planning with hose. In *ACM SIGCOMM 2021*.

[13] T. Arnold, E. Gürmeriçliler, G. Essig, A. Gupta, M. Calder, V. Giotsas, and E. Katz-Bassett. (How Much) Does a Private WAN Improve Cloud Performance? In *IEEE INFOCOM 2020*.

[14] D. Bhattacherjee, W. Aqeel, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, et al. {cISP}: A speed-of-light internet service provider. In *USENIX NSDI 2022*.

[15] BITAG. Latency explained. Technical report, 2022. https://www.bitag.org/documents/BITAG_latency_explained.pdf.

[16] R. Bothra, R. Gandhi, R. Bhagwan, V. N. Padmanabhan, R. Liang, S. Carlson, V. Kamath, S. Acharyya, K. Sueda, S. Chaturmohta, and H. Sharma. Switchboard: Efficient resource management for conferencing services. In *ACM SIGCOMM 2023*.

[17] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla. Why is the internet so slow?! In *Springer PAM 2017*.

[18] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo. Analysis and design of the google congestion control for web real-time communication (WebRTC). In *ACM MMSys*, 2016.

[19] H. Chang, M. Varvello, F. Hao, and S. Mukherjee. Can you see me now? A measurement study of Zoom, Webex, and Meet. In *ACM IMC*, 2021.

[20] D. Chou, T. Xu, K. Veeraraghavan, A. Newell, S. Margulis, L. Xiao, P. M. Ruiz, J. Meza, K. Ha, S. Padmanabha, et al. Taiji: managing global user traffic for large-scale internet services at the edge. In *ACM SOSP*, 2019.

[21] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. Luckie, R. K. P. Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, and K. Claffy. Inferring persistent interdomain congestion. In *ACM SIGCOMM 2018*.

[22] S. Dhawaskar Sathyanarayana, K. Lee, D. Grunwald, and S. Ha. Converge: Qoe-driven multipath video conferencing over webrtc. In *ACM SIGCOMM 2023*.

[23] A. Flavel, P. Mani, D. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev. FastRoute: A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs. In *USENIX NSDI*, 2015.

[24] S. Fouladi, J. Emmons, E. Orbay, C. Wu, R. S. Wahby, and K. Winstein. Salsify: Low-Latency Network Video through Tighter Integration between a Video Codec and a Transport Protocol. In *USENIX NSDI*, 2018.

[25] R. Gandhi, Y. C. Hu, C.-k. Koh, H. H. Liu, and M. Zhang. Rubik: Unlocking the power of locality and end-point flexibility in cloud scale load balancing. In *USENIX ATC 2015*.

[26] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. Achieving high utilization with software-driven wan. In *ACM SIGCOMM 2013*.

[27] C.-Y. Hong, S. Mandal, M. Al-Fares, M. Zhu, R. Alimi, C. Bhagat, S. Jain, J. Kaimal, S. Liang, K. Mendelev, et al. B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined wan. In *ACM SIGCOMM 2018*.

[28] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, Ö. Alay, and N. Kuhn. Low-latency scheduling in mptcp. *IEEE/ACM Transactions on Networking 2018*.

[29] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a globally-deployed software defined wan. In *ACM SIGCOMM 2013*.

[30] V. Jalaparti, I. Bliznets, S. Kandula, B. Lucier, and I. Menache. Dynamic pricing and traffic engineering for timely inter-datacenter transfers. In *ACM SIGCOMM 2016*.

[31] J. Jiang, R. Das, G. Ananthanarayanan, P. A. Chou, V. Padmanabhan, V. Sekar, E. Dominique, M. Goliszewski, D. Kukoleca, R. Vafin, and H. Zhang. Via: Improving Internet Telephony Call Quality Using Predictive Relay Selection. In *ACM SIGCOMM*, 2016.

[32] M. Kwon, Z. Dou, W. Heinzelman, T. Soyata, H. Ba, and J. Shi. Use of Network Latency Profiling and Redundancy for Cloud Server Selection. In *IEEE International Conference on Cloud Computing*, 2014.

[33] A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: Comparing public cloud providers. In *ACM IMC 2010*.

[34] J. Li, Z. Li, R. Lu, K. Xiao, S. Li, J. Chen, J. Yang, C. Zong, A. Chen, Q. Wu, C. Sun, G. Tyson, and H. H. Liu. LiveNet: A Low-Latency Video Transport Network for Large-Scale Live Streaming. In *ACM SIGCOMM 2022*.

[35] X. Lin, Y. Ma, J. Zhang, Y. Cui, J. Li, S. Bai, Z. Zhang, D. Cai, H. H. Liu, and M. Zhang. GSO-simulcast: global stream orchestration in simulcast video conferencing systems. In *ACM SIGCOMM*, 2022.

[36] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew. Greening Geographical Load Balancing. *IEEE/ACM Transactions on Networking*, 2015.

[37] K. MacMillan, T. Mangla, J. Saxon, and N. Feamster. Measuring the performance and network utilization of popular video conferencing applications. In *ACM IMC*, 2021.

[38] Z. Meng, Y. Guo, C. Sun, B. Wang, J. Sherry, H. H. Liu, and M. Xu. Achieving consistent low latency for wireless real-time communications with the shortest control loop. In *ACM SIGCOMM 2022*.

[39] R. K. P. Mok, H. Zou, R. Yang, T. Koch, E. Katz-Bassett, and K. C. Claffy. Measuring the network performance of google cloud platform. In *ACM IMC 2021*.

[40] R. Padmanabhan, P. Owen, A. Schulman, and N. Spring. Timeouts: Beware surprisingly high delay. In *ACM IMC*, 2015.

[41] M. Rudow, F. Y. Yan, A. Kumar, G. Ananthanarayanan, M. Ellis, and K. Rashmi. Tambur: Efficient loss recovery for videoconferencing via streaming codes. In *USENIX NSDI 2023*.

[42] B. Schlinker, I. Cunha, Y.-C. Chiu, S. Sundaresan, and E. Katz-Bassett. Internet performance from facebook's edge. In *ACM IMC 2019*.

[43] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. Engineering egress with edge fabric: Steering oceans of content to the world. In *ACM SIGCOMM 2017*.

[44] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. Technical report, 2003.

[45] S. M. Shithil and M. A. Adnan. A prediction based replica selection strategy for reducing tail latency in distributed systems. In *2020 IEEE CLOUD*, 2020.

[46] L. Suresh, M. Canini, S. Schmid, and A. Feldmann. C3: Cutting tail latency in cloud data stores via adaptive replica selection. In *USENIX NSDI*, 2015.

[47] A. Taneja, R. Bothra, D. Bhattacherjee, R. Gandhi, V. N. Padmanabhan, R. Bhagwan, N. Natarajan, S. Guha, and R. Cutler. Don't forget the user: It's time to rethink network measurements. In *ACM HotNets 2023*.

[48] B. Trammell, J.-P. Smith, and A. Perrig. Adding path awareness to the internet architecture. *IEEE Internet Computing 2018*.

[49] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford. DONAR: Decentralized Server Selection for Cloud Services. In *ACM SIGCOMM*, 2010.

[50] B. Wu, K. Qian, B. Li, Y. Ma, Q. Zhang, Z. Jiang, J. Zhao, D. Cai, E. Zhai, X. Liu, and X. Jin. Xron: A hybrid elastic cloud overlay network for video conferencing at planetary scale. In *ACM SIGCOMM 2023*.

[51] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, J. Liu, and R. Li. A low-latency mptcp scheduler for live video streaming in mobile networks. *IEEE Transactions on Wireless Communications 2021*.

[52] K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, P. Sood, M. Tariq, M. Tierney, D. Trumic, V. Valancius, C. Ying, M. Kallahalla, B. Koley, and A. Vahdat. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *ACM SIGCOMM 2017*.

[53] Q. Zhang, Q. Zhu, M. F. Zhani, R. Boutaba, and J. L. Hellerstein. Dynamic Service Placement in Geographically Distributed Clouds. *IEEE Journal on Selected Areas in Communications*, 2013.

[54] A. Zhou, H. Zhang, G. Su, L. Wu, R. Ma, Z. Meng, X. Zhang, X. Xie, H. Ma, and X. Chen. Learning to coordinate video codec with transport protocol for mobile video telephony. In *ACM International Conference on Mobile Computing and Networking*, 2019.

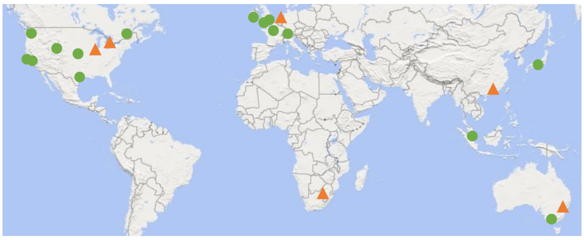


Fig. 14. Locations of the 21 Azure DCs used in the measurements. Orange triangles denote the locations of representative DCs used in Fig.3.

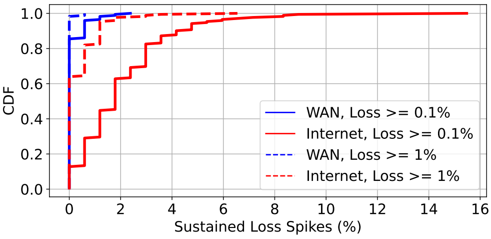


Fig. 15. CDF of packet loss across all client countries - MP DC pairs in Europe.

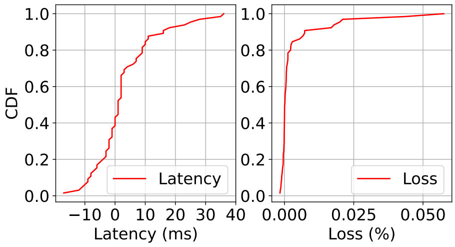


Fig. 16. Elasticity on Internet.

A Internet and WAN performance

A.1 Methodology details

Table 3. Scale of our measurements.

Geography	Unique values
Avg. #measurements/day	3.5 million
Source country	244
Source city	241,777
Source ASN	61,675
IP subnets	4,731,110
Destination DCs	21

Fig.14 shows the locations of the 21 Azure DCs used in measurements. The DCs are in spread across 5 continents. Table 3 shows the statistics of the measurements. We conducted more than 1 billion measurements in a year-long study.

A.2 Loss on Internet and WAN

In this experiment, we measure the number of 30 min time-slots over a 7 day period where the loss on individual paths (Internet or WAN) is at least 0.1%. Fig.15 shows the CDF for all (176) client country - MP DC pairs in Europe. It can be seen that Internet has more frequent loss. 50% source - MP DC pairs suffer loss of at least 0.1% on Internet for at least 2% time-slots. In contrast, 0.1% loss on WAN is rare – the number of time-slots at $P100$ is bound to 2%.

We repeat the same analysis when the loss is minimum 1%. As expected, there are fewer time-slots when the loss on Internet is ≥1% versus when loss is ≥0.1%. However, even in this case, Internet still has more frequent loss compared to WAN.

A.3 Elasticity on Internet

Fig.16 shows the increase in latency and loss between different client country - MP DC pairs in Europe as we increase the fraction of traffic on Internet from 1% to 20%. Note that the changes in latency and loss are impacted by two factors: (a) traffic moved by Titan from 1% to 20%, and (b) underlying infrastructure and routing changes outside Titan. Often, it is not possible to decouple these factors as Titan takes a few months (multiple trial-and-error) to complete movement of traffic to Internet, and it has no visibility into the ISP infrastructure. The negative latency difference is likely because Internet infrastructure improved over a period of time. It can be seen that the latency difference even at $P90$ is under 20 msec. Even for loss, the difference at $P90$ is under 0.01%. Internet providers either have capacity available or are able to quickly add capacity as needed. These results show that Internet is fairly elastic to accommodate increasing traffic from Titan.

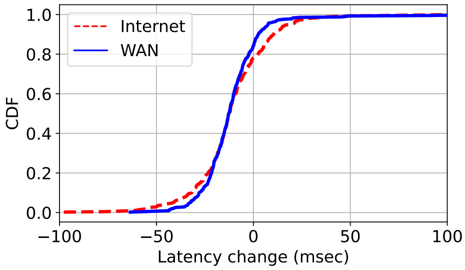


Fig. 17. Latency difference between 2 weeks separated by 12 months. Negative values = improvements.

A.4 Long-term trends:

For both the Internet and WAN paths between the 20 countries (top; by call volume) and all DCs, we measured the weekly median latencies for the weeks that are ~12 months apart. Fig.17 plots the CDFs of changes in latency (new minus old; negative means improvement) for the Internet and WAN paths. While for more than 80% cases latencies have improved for both types of paths, the Internet paths see slightly greater improvements.

A.5 Impact of fine grained granularities

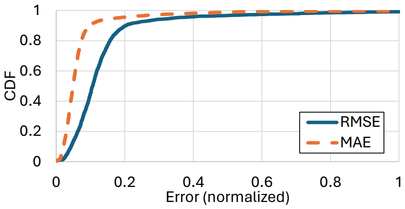


Fig. 18. CDF of error (normalized to max values).

Fig.3 shows the fraction (F) of times (when considering hourly median values for 1 week) Internet paths offer latencies lower than or comparable (≤10 ms inflation) to WAN paths from different source countries to destination DCs. To do such an analysis, we consider the clients at the granularity of a *country*. However, one client country can have different ASNs or cities with potentially different performance (consequently different F). In this section, we detail the difference in F when considering different granularities compared to granularity of country as shown in Fig.4. Let's consider granularity of city + ASN. Let's say one country has N combinations of city + ASN, with fractions F as $\{F_1$ to $F_N\}$. The fraction F for that country is F_c. Similarly, the fractions of number of measurements for individual combination of city + ASN to total number of measurements for that country are $\{w_1$ to $w_N\}$ $(\sum_{i \in N} w_i = 1)$. For each city + ASN combination, we calculate the difference in F compared to granularity of country as follows: The difference (D) is calculated as $D = \frac{\sum_{i \in N} |F_i - F_c| \cdot w_i}{F_c}$. We calculate D for each client country - destination MP country and plot $P50$ and $P90$ in Fig.4. It can be seen that difference D is bound to 11% even at $P90$ for city + ASN. These results show that granularity of country performs similar to more fine grained granularities such as ASN and city + ASN.

A.6 Stability

We perform the same analysis as in Fig.3 using data from a week in December'23 (6 months apart). The results are shown in Fig.19. We find that, in general, the trends are similar as described in Fig.3, while the North America - Europe corridor has improved slightly in 6 months.

B Accuracy of prediction

Fig.18 shows the CDF for accuracy of prediction in terms of RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) across 3,000 call configs. The Holt-Winters based prediction in Titan-Next is fairly accurate with median errors of 4.9% and 10.6% for MAE and RMSE respectively. 95.6% (89.7%) call configs have normalized MAE (RMSE) less than 20%. For a small number of call configs, the errors are relatively large due to unexpected change in the number of calls for such configs.

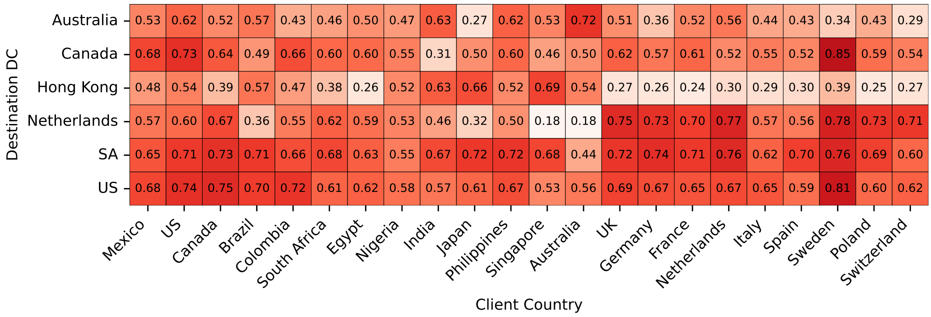


Fig. 19. Fraction of times Internet provides better or comparable (within 10 msec) latency compared to WAN. We show for different source countries and 6 Azure DCs. SA denotes South Africa and US denotes the United States. We use 1-week data from the month of December'23 that is 6 months apart from the dates used in Fig.3.

Table 4. Notations used in the LP.

Notation	Definition
T, M, C	Set of timeslots, set of MP DCs, set of reduced call configs
$Cap_{t,m}$	Compute capacity of the MP DC m for the timeslot t in terms of number of cores
I, W	Set of Internet paths and set of WAN links
P	Set of all Internet and WAN paths. Each MP DC has one Internet and WAN path each
$N_{t,c}, N$	($N_{t,c}$) Number of calls for the call config c for the timeslot t, N: total number of calls
$InternetCap_{t,p}$	Capacity (in Gbps) of the Internet path p in the timeslot t
E	Bound on the average of max. end-to-end latency across reduced call configs
(output) $X_{t,c,m,p}$	Number of calls assigned to c-th call config to the MP DC m and path p for timeslot t
(output) y_l	peak bandwidth used on the WAN link l

LP Variable: $X_{t,c,m,p}$ **Objective:** Minimize $\sum_{l \in W} y_l$

Constraints:

C_1: $\forall t \in T, c \in C, \sum_{m \in M, p \in P} X_{t,c,m,p} = N_{t,c}$

C_2: $\forall t \in T, m \in M, \sum_{c \in C, p \in P} X_{t,c,m,p} \cdot computeUsed(c) \leq Cap_{t,m}$

C_3: $\forall t \in T, p \in I, \sum_{c \in C, m \in M} X_{t,c,m,p} \cdot networkUsed(c, m, p) \leq InternetCap_{t,p}$

C_4: $\frac{1}{N} \cdot \sum_{t \in T, c \in C, m \in M, p \in P} X_{t,c,m,p} \cdot E2Elatency(c, m, p) \leq E$

C_5: $\forall t \in T, l \in W, y_l \geq \sum_{c \in C, m \in M, p \in P} X_{t,c,m,p} \cdot networkUsed(c, m, p) \cdot isLinkUsed(c, m, p, l)$

Fig. 20. LP formulation for joint MP DC and routing option assignment. Notations are in Table 4.

C LP for joint MP DC placement and routing

Fig.20 shows the LP formulation for joint MP DC and routing assignment. The notations are shown in Table 4. The details on the LP formulation are in §6.3. Next we detail the alternate approaches we considered.

What did not work: Majority of the calls today are *intra-country* for MS Teams. We need to assign all calls from the same country to the same MP DC to eliminate call migrations between DCs. We formulated it as an ILP (Integer Linear Program) and the intra-country migrations did fall to zero. However, the network savings also substantially diminished as calls could not be assigned to multiple DCs to save network bandwidth. Consequently, we aim to reduce the number of migrations in Titan-Next instead of eliminating them altogether.

Discussion: As shown in Fig.7 (and Fig.16 in §A.3), packet loss on the Internet does not increase significantly as we increase MS Teams traffic on the Internet up to a certain extent. Also, real packet loss is known only when calls progress. Thus, LP does not consider packet loss during assignments. Secondly, the LP assigns *single* routing option (either WAN or Internet) for all participants of the same call. Without this condition, LP size increased substantially and could not finish in timely manner. Lastly, we don't split traffic from same participant across WAN and Internet links to avoid adverse effects at the receiver, especially for out-of-order packets and the jitter buffer. We leave such traffic splitting for future work.

D Other issues in real-time call assignment

Handling surge in calls: We have rarely witnessed sudden high jump in calls. In such cases, MP servers are to be scaled accordingly. If we witness calls for which LP hasn't assigned capacity, we assign MP DC closest to the first joiner of a call that has enough capacity.

Migration to a different route: Recall that LP based assignment is offline – it does not know the real-time conditions of the Internet paths. It may happen that the performance of the Internet path for a participant on a call is poor due to outages or transient congestion (e.g., frequent packet loss as shown in Fig.6), and we want to react in seconds (cannot wait 30 min for LP to address it – would affect user experience). We monitor the packet loss and latency on the Internet path as the call progresses, and move the user to WAN when the latency and packet loss are above acceptable thresholds: packet loss ≥ 1% and latency threshold is set depending on the physical distance. We observed the median number of users across 2 months with loss on Internet ≥ 1% as 3.96%. In rare events when a large chunk of users experience poor performance, Titan would take charge, adjusting the percentage of traffic on the Internet/WAN paths, and Titan-Next would simply abide. We do not move calls from WAN to Internet as to satisfy the Internet capacity limits.

Received June 2024; revised September 2024; accepted October 2024