Enhancing SD-WAN Performance

Overcoming the challenges limiting connection aggregation in traditional SD-WAN solutions

Dejero

Table of Contents

Executive Summary 3

Introduction 5

The Rise of SD-WAN 5

SD-WAN in Operation 6

Practical Limitations 7

Building A Better Solution 7

Traditional SD-WAN Connection Aggregation 8

Aggregating Connections at Branch Offices 8

Maintaining Static Source Addressing 8

Congestion Control 12

Links with Different Capacity 14

Links with Different Latencies 15

Aggregating Connections at the Head Office 17

Static IPs 17

An Enhancement to Connection Aggregation: Smart Blending 20

Architecture 20

Functional Requirements 20

Agnostic of Connection Symmetry 21

Real-Time Connection Performance Monitoring 21

Flow/Application Acceleration 21

Aggregating Connections 21

Overcoming Traditional SD-WAN Connection Aggregation Challenges 22

Further Optimizations 22

Conclusions 23

Review of Traditional SD-WAN Connection Aggregation 23

Summary of Performance Issues 23

A Superior Alternative: Smart Blending 24

Summary of Key Advantages 24

Executive Summary

SD-WAN is both a response to, and an enabler of, the ongoing adoption of Software-as-a-Service (SaaS) and other cloud-based solutions.

For many organizations, the shift to the cloud has caused branch offices and remote locations to become less dependent on MPLS links to the head office, and instead become more reliant upon high speed connections to the public cloud.

By using connection aggregation technology to combine commercially available broadband Internet links, software-defined wide area networks (SD-WAN) reduce—and in some cases completely eliminate—dependence upon MPLS links. In doing so, SD-WAN contributes to simplifying operations, lowering costs, and improving reliability, all without sacrificing performance.

But SD-WAN does more than just change how branch offices connect to important services—it unlocks new possibilities for remote and nomadic locations dependent upon wireless Internet services.

Faced with significant technical hurdles, traditional SD-WAN connection aggregation keeps each flow 'sticky' to a single link. However, this approach trades performance for engineering simplicity—a minimum threshold of capabilities is reliably delivered, but never the full potential of all the connections—leading to suboptimal performance and an inefficient use of resources.

In stark contrast, a solution that measures dynamic connection characteristics in real time and intelligently splits flows across multiple connections can deliver vastly superior results. This approach—termed 'smart blending'—is technically feasible if certain requirements are met.

This whitepaper explores the aggregation approach used by most SD-WAN solutions; in doing so, it illustrates the technical hurdles that must be overcome to realize the full potential of SD-WAN.

Then, it introduces smart blending, and shows how this technique delivers important advantages.

Dejero

Introduction

As organizations rapidly adopt cloud-based, software-as-a-service (SaaS) offerings, their branch offices have a shrinking dependence on dedicated (and expensive) multiprotocol label switching (MPLS) links to the head office—since the majority of services to which they need access are now hosted in the public cloud.

> SD-WAN is both a response to, and an enabler of, the shift to the cloud

Additionally, increased SaaS makes nomadic branch locations much more viable, since LTE and Wi-Fi connections are becoming sufficiently fast and ubiquitous.

But relying on cloud-based compute, storage, and networking services to power a wide range of crucial functions—from communication, collaboration, and productivity tools, to artificial intelligence applications and mission-critical services—means that all of an organization's offices are increasingly dependent upon reliable connectivity to the public Internet, particularly over the first and last mile.

The Rise of SD-WAN

In response to—and further enabling—these shifts is the rise of software-defined wide area networks (SD-WAN), in which WAN management and operation are simplified by separating the network's hardware from the software controlling it. SD-WAN also lets organizations leverage lower-cost Internet and cloud connectivity, and sometimes even completely replace dedicated, private WAN technologies, while still meeting demanding quality of service (QoS) needs.

> With SaaS, branch offices are less dependent on MPLS links to the head office, but instead become more reliant on reliable, high speed connections to the public cloud

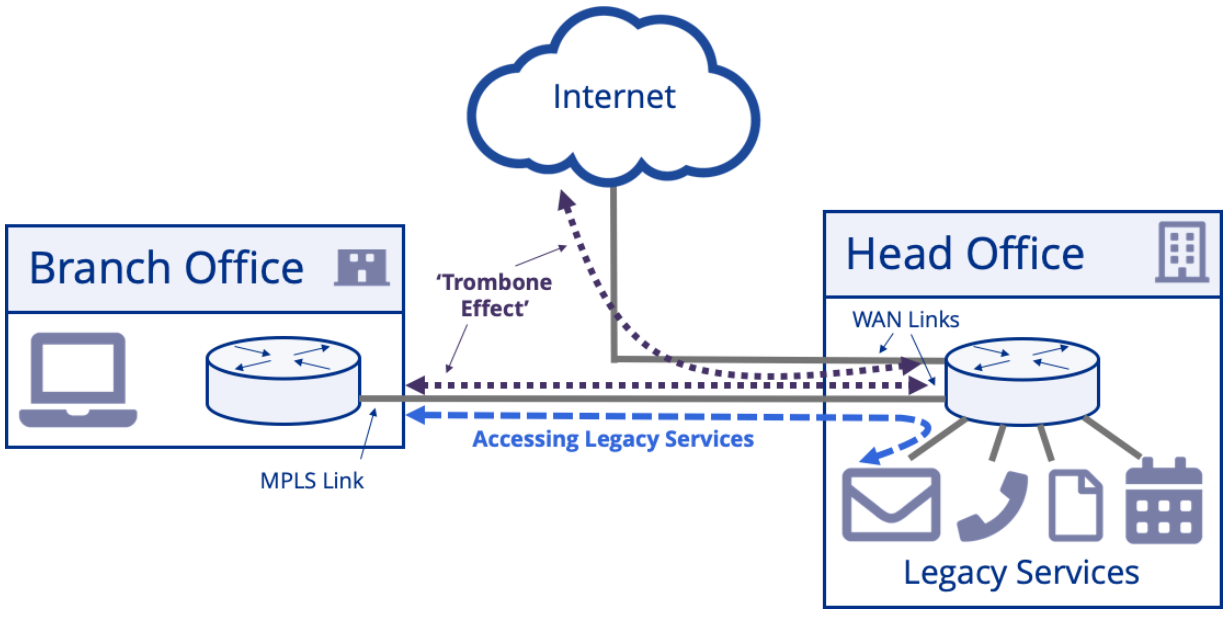

Figure 1 — Traditional WAN model, in which the Branch Office has a dedicated MPLS connection to the Head Office

SD-WAN in Operation

When organizations hosted their business-critical services locally (for instance, in the head office), each branch office relied upon a dedicated WAN connection (typically an MPLS link) to the head office to access those services—this scenario is illustrated by the dashed blue line in Figure 1.

But SaaS and cloud hosting changes the traffic distribution, with the bulk now going to the public Internet. Organizations that continue to use only an MPLS link between the branch office and the head office suffer from the 'trombone effect', in which branch traffic traverses the WAN twice—introducing unnecessary latency and consuming expensive (relative to the public Internet) MPLS resources. This effect is illustrated by the dotted purple line in Figure 1.

> Organizations that use the cloud extensively exchange much more traffic with the public Internet than with the head office

By aggregating commercially available broadband Internet links (such as cable or DSL), SD-WAN reduces—and in some cases completely eliminates—dependence upon the MPLS link.

In fact, because reliance upon a single network connection introduces tremendous risk and vulnerability, a major driver of SD-WAN adoption is its ability to aggregate multiple network connections (for instance, public Internet and MPLS) and route traffic more efficiently—so, at the branch office, public Internet traffic goes directly over all available broadband links (the dotted purple line in Figure 2), and traffic destined for the head office goes over the MPLS link and VPN tunnels (the dashed blue line).

> "SD-WAN is moving users away from MPLS to hybrid and internet-only WANs. Infrastructure and operations leaders responsible for WAN design should leverage SD-WAN to improve availability and save costs."
>
> —Gartner

Effectively aggregating multiple separate Internet connections is an important step towards meeting the reliability, performance, efficiency, and QoS demands of today's connected organizations.

> 'Traditional' SD-WAN solutions were built for a world of fixed branch offices using wired broadband technologies

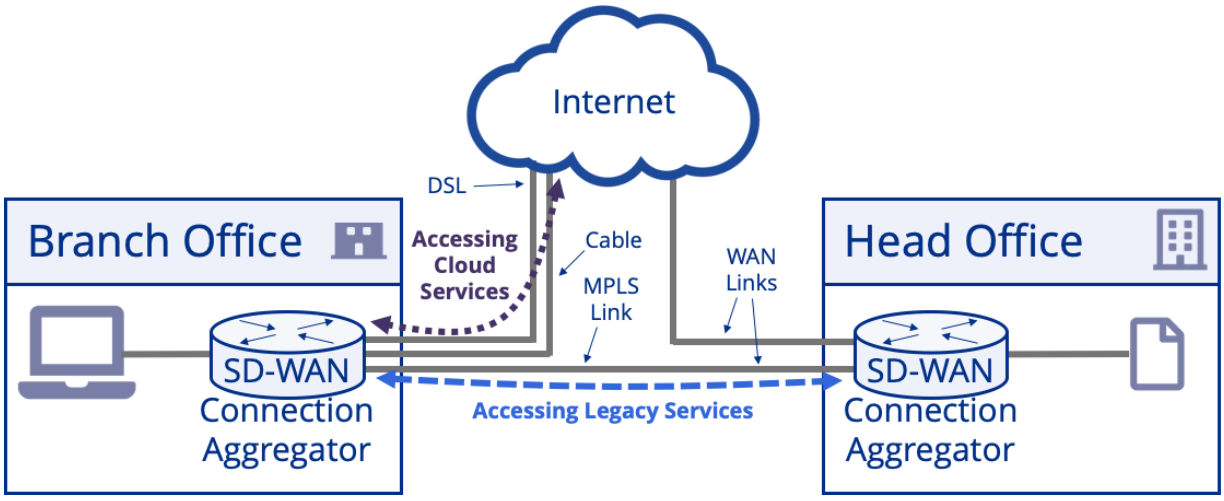

Figure 2 — Aggregating multiple links (for example, cable, DSL, and MPLS) for head office and Internet connectivity

Practical Limitations

Despite sharing a common label, SD-WAN solutions aren't created equal—for instance, some vendors combine MPLS and broadband links into an aggregate connection, while others position SD-WAN as a replacement for MPLS and only combine broadband links—so we must take some care when speaking in generalizations. Nevertheless, as the market matures and SD-WAN technologies gain widespread adoption, some limitations are becoming apparent.

Broadly, 'traditional' SD-WAN solutions can provide sufficient connection aggregation for stationary applications when using wired broadband technologies including fiber, DSL, and cable. However, even in this scenario the aggregation doesn't make efficient use of the available network resources, which negatively impacts connection reliability, speed, and efficiency.

Plus, to meet the ever-increasing reliability and speed demands that are the price of cloud- and SaaS-dependence, enterprises need to add wireless technologies such as Wi-Fi, cellular (for example, 4G LTE, 5G), and satellite to the mix, even for stationary applications and head offices.

> To meet ever-increasing demands, enterprises need to add wireless access technologies to the WAN mix

However, the limitations of traditional SD-WAN solutions are exacerbated by wireless technologies, which often exhibit significantly variable latency and stark differences in bandwidth capacity (which itself varies over time).

And for nomadic sites, the aggregated connection might depend entirely upon these less reliable—when compared to fixed access—and more variable technologies.

In short: connection aggregation techniques created for, and in a world of, stationary applications and wired networks are extending into environments where connections are not as reliable—both over time, and by individual connection. This extension reveals limitations that must be addressed in SD-WAN solutions for organizations to truly and confidently embrace SaaS for all aspects of operation.

> Introducing wireless connections also enables nomadic and truly mobile sites

Building A Better Solution

In this whitepaper, we explore the current state of SD-WAN connection aggregation from the perspective of branch offices and the head office, and in doing so we expose shortcomings of existing solutions as well as some technological challenges.

Then, we present an alternative approach that overcomes these challenges to unlock the maximum potential of SD-WAN.

Traditional SD-WAN Connection Aggregation

To illustrate the challenges facing today's SD-WAN solutions, we'll examine both 'ends' of a WAN connection:

- Branch offices: the term "office" is a bit of a misnomer, as the discussion applies to fixed office locations, nomadic outposts (for example, research labs), mobile units (for example, video production trailers), and even home offices
- Head Office: the location at which particular organization resources (for example, data, technology services) are kept

> Because of significant technical challenges caused by IP behavior, most SD-WAN solutions incorporate connection aggregation techniques that keep each flow 'sticky' to a single link—but this trade-off comes at an enormous performance cost

Aggregating Connections at Branch Offices

Effectively and efficiently distributing traffic from a single flow among multiple WAN links—which likely each have different and variable capacity, as well as different and variable latency—is a significant technical challenge.

If it's attempted naively, without a full understanding of and appreciation for how the TCP, UDP, and application-layer protocols behave and will react, then the result is typically worse performance than can be achieved by the flow using only a single link—an outcome that undermines the performance and efficiency promises of SD-WAN.

Consequently, most SD-WAN solutions incorporate aggregation techniques that keep each flow sticky to a single connection.

Unfortunately, this approach still makes significant performance and efficiency sacrifices; to better understand the reasons why most have chosen this path, and the performance and efficiency sacrifices they've made, we'll begin by exploring three roadblocks that present major challenges to traditional SD-WAN connection aggregation:

- Complications related to source IP addressing
- Aggregating links with different capacity
- Aggregating links with different latency

> Today's branch 'office' might be a traditional office, a nomadic outpost, or an entirely mobile unit

Maintaining Static Source Addressing

Consider a typical SD-WAN setup that has a 30 Mb/s cable connection and a 25 Mb/s DSL line; in this scenario, the maximum combined theoretical aggregate bandwidth is 55 Mb/s.

For obvious reasons, the branch office wants to take full advantage of this combined capacity, even if there's only a single active flow (for instance, an HTTP-based video stream).

Unfortunately, utilizing both links for a single flow poses a significant technical challenge. To understand why requires briefly examining the protocols that underlie all IP-based Internet communication (our intention here isn't to provide a comprehensive technical explanation of these protocols, but to focus on the properties that make effective connection aggregation challenging).

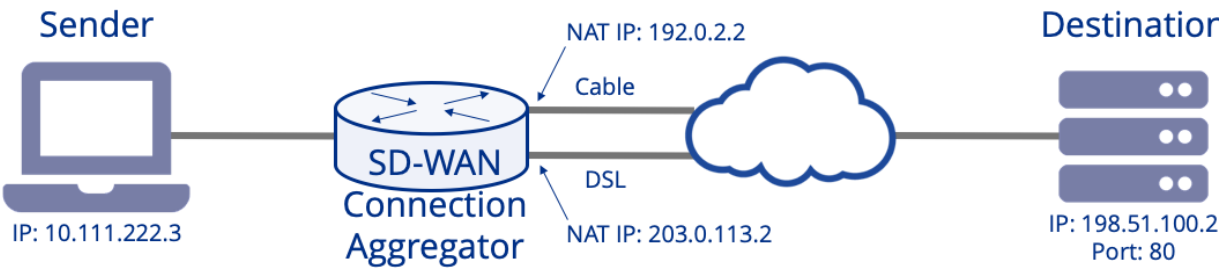

Figure 3 — TCP flow over two simultaneous, NATed connections, resulting in two separate flows

Internet Protocol (IP) is the primary OSI layer-3 protocol used for Internet communications. This protocol is responsible for addressing of hosts, encapsulating data into datagrams (including fragmentation and reassembly), and routing datagrams from a source to a destination host across IP networks.

Of particular note is that a prerequisite for two hosts to have a continuous 'conversation,' their respective IP addresses cannot change during that conversation. If either side experiences an IP change, then all active conversations between those two hosts break and must be re-established.

OSI layer-4 protocols—primarily Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)—extend source and destination IP addressing to include the concept of source and destination ports. Both TCP and UDP use a '4-tuple' to identify a distinct conversation (a 'flow') between two hosts.

```
{Source-IP, Source-Port,
Destination-IP, Destination-Port}
```

This unique 4-tuple extends communication options enormously. As a familiar example, unique 4-tuples make it possible to have two web browser windows open at the same time, concurrently loading web pages from the same site: the source IP, destination IP, and destination port (i.e., 80) are likely the same for those two conversations, but the source port is different, allowing the conversations to remain distinct.

Restricted Per-Flow Bandwidth

Consider an SD-WAN setup with two broadband links, each provided by a different ISP; while the connection type doesn't matter, let's nevertheless say that ISP-A provides a cable link and ISP-B provides a DSL link. The links are assigned their own IP addresses, each of which belongs to the address space owned by the respective ISP; this configuration is illustrated in Figure 3.

> For two hosts to have a continuous conversation, their respective IP addresses cannot change

Let's see what happens when the sender attempts to create a new TCP flow to the destination, with the 4-tuple:

Source IP	Source Port	Destination IP	Destination Port
10.111.222.3	54321	198.51.100.2	80

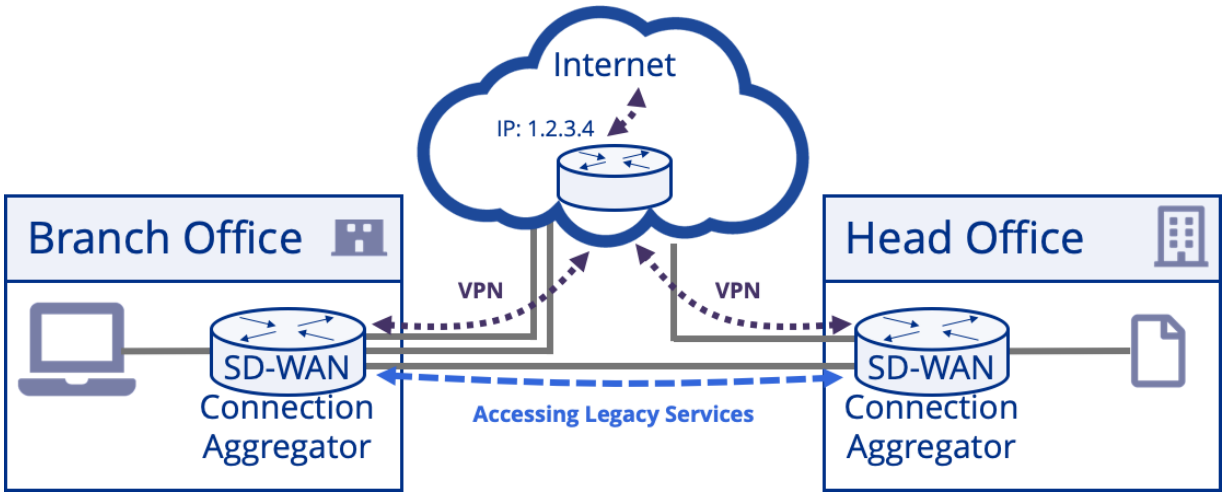

Figure 4 — Cloud-based SD-WAN with VPN tunneling

The SD-WAN connection aggregator cannot send some packets for this flow via ISP-A and others via ISP-B because the outbound network address translation (NAT) engines would create two flows (note that the source ports may or may not be the same, but the source IPs are guaranteed to be different).

Here's what things would look like from a 4-tuple perspective:

> With the connection aggregation approach used by most traditional SD-WAN solutions, the maximum bandwidth achievable by a single flow is limited to the bandwidth of the single largest link—a huge problem for locations dependent on wireless Internet connections

Source IP	Source Port	Destination IP	Destination Port
192.0.2.2 (ISP-A)	54321	198.51.100.2	80
203.0.113.2 (ISP-B)	54321	198.51.100.2	80

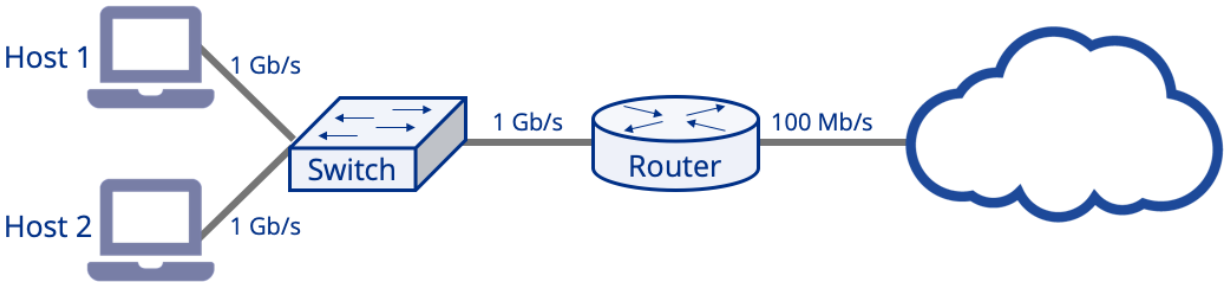

Figure 5 — Two hosts sharing limited bandwidth

If this scenario were permitted, then the receiver would see the traffic as two flows, but the sender would see it as only one, and they would fail to communicate successfully.

The requirement to maintain the consistency of this 4-tuple is what prevents most connection aggregation implementations from being able to use all connections simultaneously for a single flow; faced with this technical hurdle, most SD-WAN vendors elected to keep each flow sticky to a single connection.

> Keeping each flow sticky to a link has another drawback: if a link fails, then all the active flows terminate, time out, and must be reestablished

Unfortunately, the consequence for organizations is that the maximum possible bandwidth achievable by a single flow is limited to that of the largest link bandwidth. While this deficiency might not be a show-stopper for fixed branch offices with several reliable, fast links, it's a severe limitation for remote and mobile 'offices' reliant upon relatively low-bandwidth or highly variable wireless Internet connections.

> The cloud-based SD-WAN router helps with link failover, but still limits the bandwidth available to each flow to that of a single link

Implications for Link Failover

Keeping a flow 'sticky' to one of the links also has consequences for failover behavior; in this scenario, the flow cannot be transparently moved to the other link, because the source IP would change.

Some cloud-based SD-WAN solutions address this problem by pairing an SD-WAN router at the branch office with one in the cloud (in addition to one at the head office), as shown in Figure 4.

In this arrangement, traffic that's destined for the public Internet no longer goes directly to the Internet over the WAN links, but is instead tunneled through the VPN links to the cloud SD-WAN router (the dotted purple lines). The source IP address in the TCP flow's 4-tuple becomes that of the cloud router.

Source IP	Source Port	Destination IP	Destination Port
1.2.3.4	54321	198.51.100.2	80

If the WAN link to ISP-A fails, then the traffic can be transparently moved over to the other VPN tunnel through ISP-B, and the 4-tuple for the conversation with the web server remains unchanged.

This cloud-based setup addresses the failover scenario, but still does not allow both WAN links to be used simultaneously for a single flow to achieve the combined throughput.

To understand why, we'll need a quick backgrounder on congestion control.

Congestion Control

A fundamental challenge of IP-based communication is how to divide shared bandwidth fairly. Consider the simple instance of two hosts who need to engage in a single-flow 'conversation': to use the network efficiently, they need to discover the bandwidth available for the exchange.

Now extend the example to more than two hosts, each with many concurrent flows, and be mindful that no host has any special knowledge of the relative bandwidth capacity of the many different network links the data might take. Efficient network utilization has to be an emergent property, built on very basic principles, and implemented independently on a flow-by-flow basis.

And that's what congestion control algorithms are designed to deliver.

Between any two hosts on the Internet there are numerous network connections—or links, hops, etc.—each with a bandwidth limit that's tied to the physical characteristics of the links and that varies due to a range of factors.

For example, one hop might be a router with a GigE link (i.e., 1,000 Mb/s) on one side and a FastEthernet link (i.e., 100 Mb/s) on the other. At any given time, there will also be numerous flows passing through this hop, each contending for its fair share of bandwidth.

In Figure 5, let's assume the very simple case that:

- The two hosts each have a single flow destined for the Internet
- These two flows comprise the only traffic flowing through the router
- Each flow wants to use as much bandwidth as is available

Under these assumptions, the fair share each flow should receive is 50 Mb/s. But routers don't explicitly communicate capacity or contention information to hosts[1]. Instead, the congestion control algorithm governing each flow's behavior must independently discover and establish a flow's fair share based on observable metrics. Typically, congestion control algorithms consider packet loss, changes in round-trip time (RTT)—also called jitter—and the rate of acknowledgments (ACKs).

Due to their very different characteristics, TCP and UDP manage congestion differently, which has important consequences for application behavior and link aggregation.

> Most of the widely deployed TCP congestion algorithms depend heavily on packet loss as the prime indicator of congestion

TCP Congestion Control

The most widely deployed TCP congestion control algorithms (for example, NewReno, CUBIC) use packet loss as their primary metric. Basically, they continually increase the rate at which they transmit data until they observe packet loss, which they interpret as evidence that they've reached (exceeded) their fair share.

Continuing the example of Figure 5, let's now say the two hosts start their flows at the same time, and their congestion control algorithms are identical and are executing simultaneously[2].

[1] Contention information *can* be communicated with explicit congestion notification (ECN), but it's not widely adopted

[2] This example belies the complexity of a real-world network, with vast numbers of flows starting and stopping at different times, all contending for shared resources, but it's sufficient to illustrate the fundamentals of TCP's congestion control algorithm

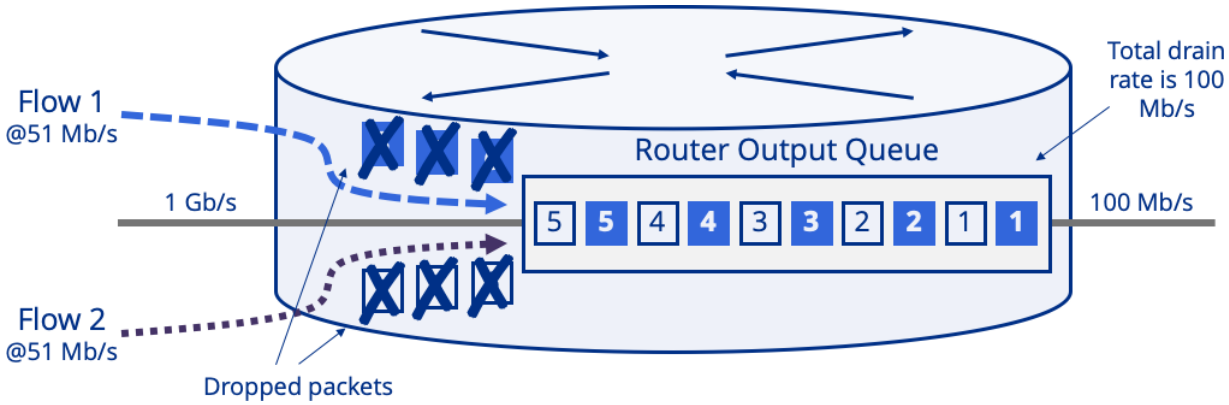

Figure 6 — When the flows try to send more than 100 Mb/s in aggregate, the buffer fills to maximum capacity and packets are dropped

Under these assumptions, here's what will happen:

1. The two hosts will each ramp up their transmission rate; so long as the sum of their rates is below 100 Mb/s, the router immediately transmits the packets to the Internet
2. The hosts will receive positive feedback (ACKs) from their respective destinations, indicating spare capacity; they'll continue to increase their transmission rates
3. The sum of their transmission rates will exceed 100 Mb/s (in Figure 6, each flow sends 51 Mb/s); because packets are arriving at the router faster than it can forward them on, the router will buffer packets in a queue
4. As packets continue to accrue in the queue (buffering), eventually the queue will exceed its packet storage capacity; at this point, the router will drop packets (shown by an "X" over packets in Figure 6) by neither adding them to the queue nor forwarding them along[3]
5. Through a lack of ACKs, the hosts will learn that packets failed to arrive at their respective destinations; the hosts (each, independently) will interpret this packet loss to indicate that they've reached their fair share of the available capacity

UDP Congestion Control

Unlike TCP, UDP provides only a minimal set of services on top of IP—primarily checksums for data integrity and flow multiplexing.

> In this example, packet drops are an accurate indicator that each flow has attained its fair share of available bandwidth

Notably absent is a protocol-level congestion control algorithm. As a result, UDP-reliant applications tend to fall into one of two categories:

1. Those that use **Application-Layer Congestion Control**: Applications that need to adapt to the available bandwidth capacity must implement their own congestion control algorithms. Usually, these algorithms are very similar to those of TCP—they rely on the same metrics and interpret them similarly—but they're implemented by the application rather than by the UDP.
2. Those that '**Spray-and-Pray**': This type of application sends its UDP packets to the destination with no regard for whether it's exceeding the available capacity or its fair share of bandwidth through the intermediate hops. Generally, this approach is successful only for applications with small bandwidth requirements relative to the capacity of the available network connections.

> Bufferbloat can cause jitter and can reduce overall network throughput; when a buffer fills completely, packets get dropped in-transit

[3] Buffering is a normal behavior and allows network resources to gracefully handle bursts of traffic by increasing latency slightly rather than dropping packets; excessive buffering (bufferbloat) can cause high latency as queues fill and, ultimately, packet drops—learn more at https://www.bufferbloat.net and https://en.wikipedia.org/wiki/Bufferbloat

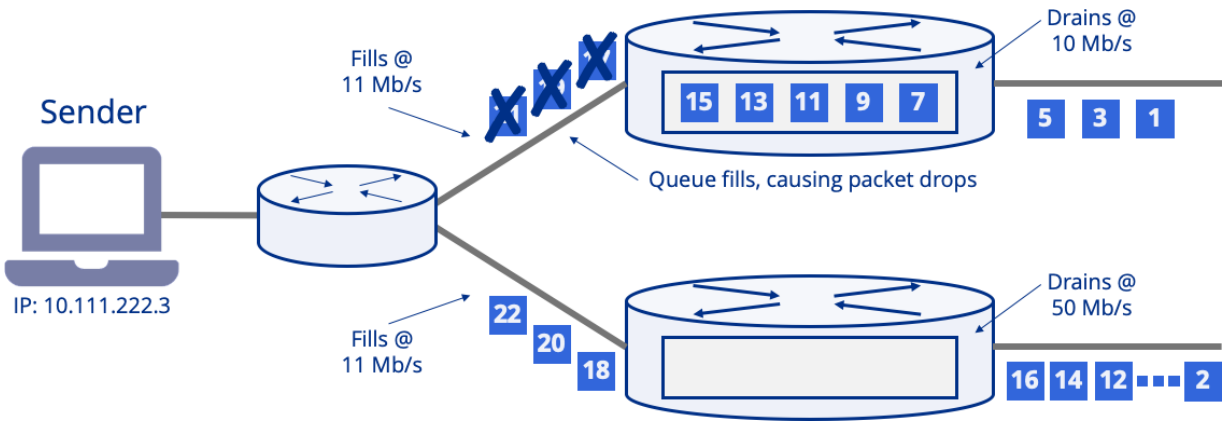

Figure 7 — The round-robin approach leads to underutilized aggregate connection bandwidth

Now we'll explore some real-world challenges facing SD-WAN solution providers:

- Aggregating links with different capacity
- Aggregating links with different latency

Links with Different Capacity

Consider two WAN links with identical latency but different capacity (for example, two DSL lines provided by the same ISP, with one provisioned for 10 Mb/s and the other for 50 Mb/s).

How does TCP traffic and UDP traffic behave when these links are aggregated?

> Application-layer congestion control in UDP-based applications usually mimics TCP's algorithms

Restricted TCP Performance

Let's examine what happens if an SD-WAN connection aggregator round-robins the packets between the two connections: as the TCP flow slowly increases its transmission rate, it eventually exceeds 20 Mb/s, and the state of the two DSL routers will be as shown in Figure 7.

The TCP flow interprets the packets dropped in the 10 Mb/s router as a clear indicator that it has reached the capacity of the aggregated link (keep in mind that

> In this example, with a naïve round-robin approach to routing packets, the packet drops cause the sender to stop at 20 Mb/s, well below the 60 Mb/s available

the flow has no way of knowing where the packets were dropped, only that they were dropped); the flow dutifully keeps its transmission to 20 Mb/s, well below the 60 Mb/s theoretical aggregated capacity.

Once more, the branch office is unable to fully utilize the aggregated link capacity, even though the cloud-based router preserves the source 4-tuple.

> Few WAN links have constant, known capacity

But what if the SD-WAN connection aggregator was told the capacity of the two links? Would the problem be solved if the aggregator distributed packets in a weighted round-robin manner, such that it places five times more packets on the 50 Mb/s link?

Indeed, that would achieve the desired result in this simple example, and in fact that's what some SD-WAN aggregators allow. Unfortunately, the real world rarely cooperates: few WAN links have constant, known capacity. For example, LTE and Wi-Fi are variable, and even some fixed lines have variable capacity due to burstable/best-effort approaches and constraints imposed by the nature of shared network resources.

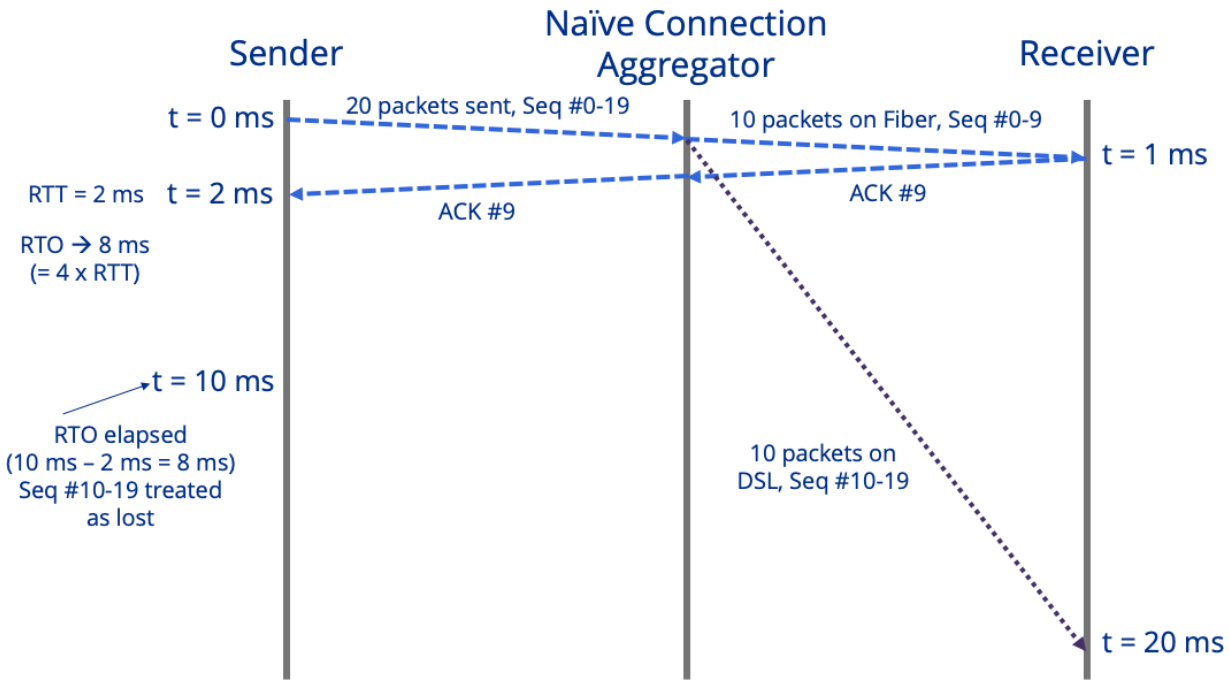

Figure 8 — Differences in connection latency are interpreted as packet loss

> In this example, the different link latencies cause the Sender to incorrectly conclude that packets were dropped, even though they weren't; in response, the Sender backs off and even unnecessarily retransmits the 'dropped' packets

To briefly summarize this section: TCP's reliance on packet loss as a congestion indicator makes it difficult for SD-WAN connection aggregators to use multiple links simultaneously for the same flow.

Again, when faced with this significant technical challenge, most SD-WAN providers chose to keep individual flows sticky to particular links; unfortunately, this choice guarantees that the aggregated connection can't be efficiently utilized.

Restricted UDP Performance

Because application-layer UDP congestion control behaves similarly to TCP congestion control, the lessons above apply to this class of UDP-based application.

Moreover, spray-and-pray UDP applications tend to have relaxed requirements around packet loss, packet ordering, and jitter, so those with low bandwidth requirements are quite tolerant of aggregation across links with different capacity; however, those with larger bandwidth requirements can experience premature loss in the simple round-robin example, even though the other connection still has plenty of capacity.

Links with Different Latencies

Similar issues can occur if the SD-WAN connection aggregator has WAN links with the same capacity, but different latency (for example, a DSL line with 20 ms latency and a fiber-optic link with 1 ms latency, each provisioned for 10 Mb/s capacity).

> TCP's reliance on packet loss as a congestion indicator raises the technical hurdles for SD-WAN vendors

> In the real world, latency doesn't just vary by link, it also varies over time on the same link—even on wired broadband connections

Restricted TCP Performance

To understand why issues arise in this scenario, we need to examine how TCP congestion algorithms determine packet loss. Most algorithms implement some variation of retransmit timeout (RTO), whereby a packet is treated as lost if it isn't acknowledged (ACKed) by the receiver before the RTO expires. In practice, the RTO is typically a statistically filtered version of round-trip times observed previously.

Let's consider what happens if an SD-WAN connection aggregator uses a round-robin algorithm to send bursts of packets down the two links (Figure 8).

In this example, the different connection latencies cause the sender to declare prematurely that packets 10 through 19 were lost; the congestion control algorithm interprets this loss as indication that it has reached link capacity, so it stops increasing the transmission rate—and unnecessarily retransmits packets 10 through 19.

What if the SD-WAN connection aggregator intentionally added a 19 ms delay to all packets transmitted via the fiber link to equalize the latency of the two connections?

While that workaround would help in this scenario, it has the obvious and significant drawbacks of guaranteeing that flows can't take advantage of the lower-latency link, and—making matters even worse—of forcing all flows to experience the highest aggregated latency.

> LTE connections latency can vary from 35 ms to over 1000 ms

In any case, some connection types—including LTE and Wi-Fi—have latency that varies over time across a significant range (for instance, LTE can vary from 35 ms to over 1000 ms), rendering this 'intentional-delay' approach functionally impractical.

Unpredictable UDP Application Experience

Again, there is little difference between the behavior described above for TCP and the behavior of application-layer UDP congestion control.

However, spray-and-pray UDP applications, warrant further explanation.

These applications generally tend to have relaxed requirements around packet loss, packet ordering, and jitter, but this laissez-faire approach extends only to a point: there are thresholds at which application performance suffers. For instance, a live video streaming application needs to put packets in the proper order so they can be decoded and played back sequentially; it will also try as effectively as possible to get all packets to the destination before the playback deadline, to prevent visible decoding artifacts or errors.

> Spray-and-pray UDP applications are very tolerant of packet loss, disordered packets, and jitter—but only up to a point, and only for relatively low-bandwidth demands

An SD-WAN router that naively splits the UDP packets from this type of flow over links with different latencies unintentionally introduces jitter into the packet-stream. If the latencies differ sufficiently among links, then many UDP applications will become unusable—or, at least, will offer a terrible user experience—because the jitter buffer will bloat or overflow.

In Figure 9, a UDP-based video streaming application has a maximum jitter buffer of 150 ms, and naively using both links causes video playback to fail because some packets arrive after their playback time.

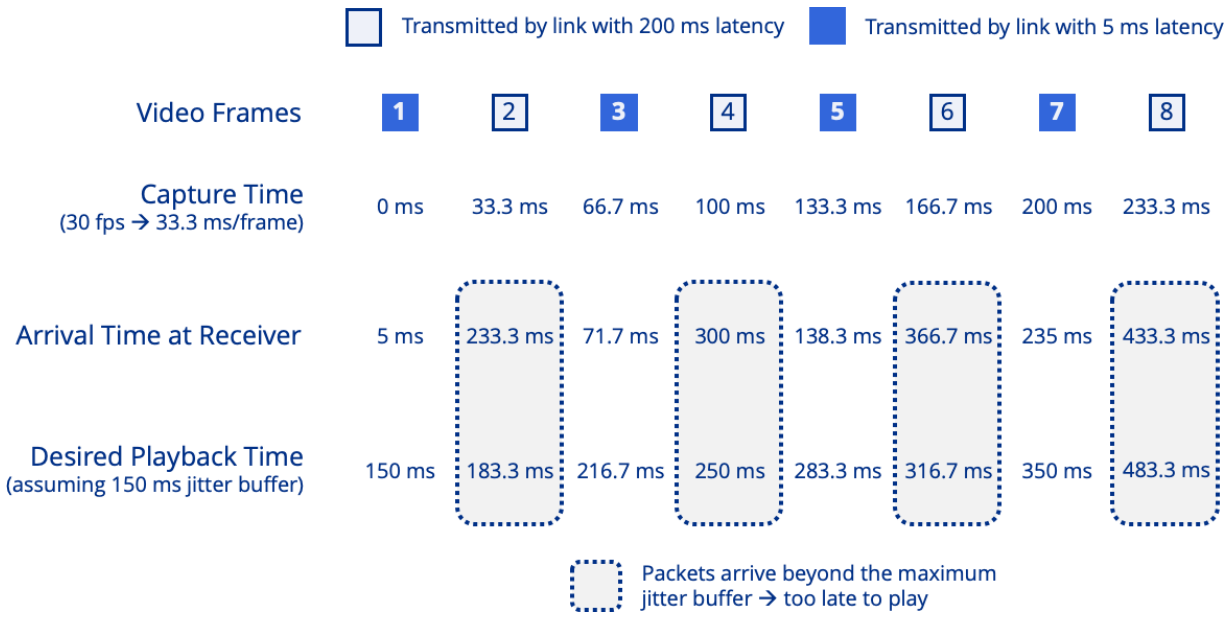

Figure 9 — Variation in latency (i.e., jitter) causes video playback to fail

> In this example, the difference in link latency causes many video frames to arrive after their playback time, causing a terrible quality of experience

Aggregating Connections at the Head Office

So far, we've examined connection aggregation from the perspective of the branch 'office'—whether fixed, mobile, or nomadic—using commodity WAN links in pursuit of connection redundancy and increased performance.

Now let's turn our attention to the head office; even in a SaaS-oriented world, branch offices still need to connect to the head office to access particular services.

> With traditional SD-WAN connection aggregation, branch offices and the head office use very different mechanisms

Static IPs

The head office must be able to accept incoming connections from the branch offices, so the head office must have known, static IP addresses.

Commodity WAN links tend to be assigned dynamic IP addresses, and are sometimes even hidden behind a carrier-grade NAT (CG-NAT) or firewall; while they cannot be used as-is at the head office, there are still a few options.

> The head office must be able to accept incoming connections from the branch offices, so it must have known, static IPs

Exterior Border Gateway Protocol (eBGP)

eBGP is a routing protocol used by ISPs and large organizations to manage routing. In eBGP, each entity is referred to as an Autonomous System (AS); each AS uses eBGP to advertise to its peers the IP subnets that it can reach, and the number of hops required to reach them. Using this information, each AS builds a table of preferred routes over which to send packets, based upon the destination IP in the header of each packet.

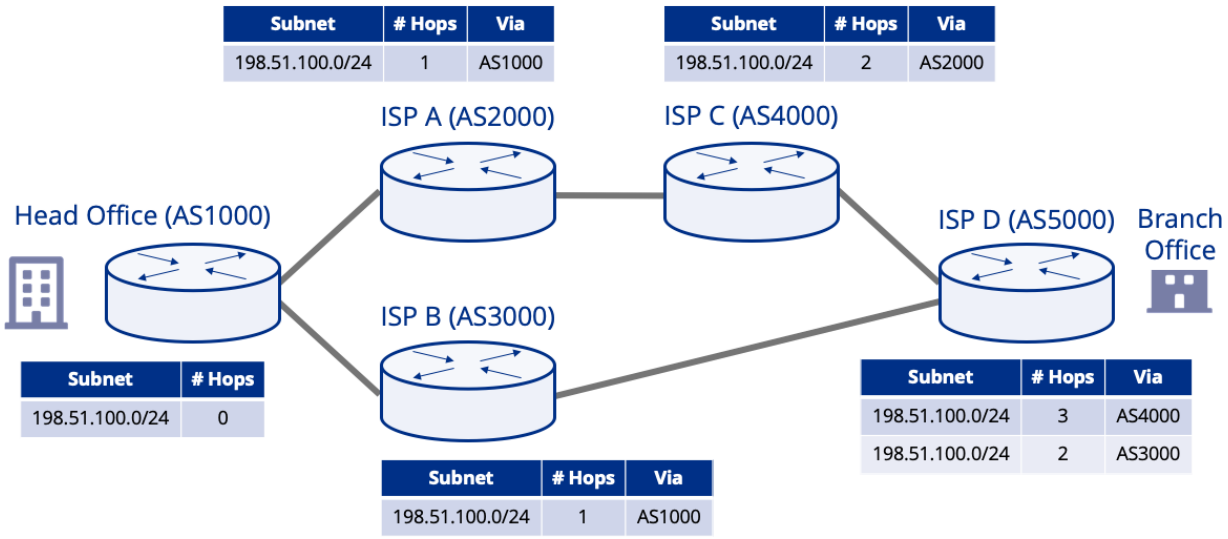

Figure 10 — eBGP routing (the tables indicate the adverstised eBGP routes)

To use eBGP for redundant connectivity, an organization must:

1. Own its own portable IP address space: "portable" means the IPs don't belong to a specific ISP, but to the organization itself, so the organization can bring the IP space with it to any ISP it wishes
2. Find at least two ISPs willing to interconnect: usually, this redundant outcome is achieved by paying a transit fee to each ISP, but if the organization is large enough then it could negotiate a settlement-free peering arrangement

Interconnection requires the ISPs to advertise to all their directly connected peers that they can reach the organization's portable IP space directly (i.e., in a single hop).

Figure 10 shows a simple interconnect arrangement linking a branch office to a head office.

By default, AS5000 prefers to send traffic destined for 198.51.100.0/24 through AS3000, because it requires fewer hops. Ultimately, however, AS5000 can override the default based on some preference: for instance, the link through AS4000 may be cheaper, may have more available capacity, may be lower-latency, etc.

Similarly, AS1000 can elect which path to use to send traffic to the branch office.

> As was the case with branch offices, the organization still isn't able to completely and efficiently utilize the full potential of the aggregated connections

Clearly, it's very easy for the situation to arise in which the traffic in each direction traverses a different path—known as triangular or asymmetric routing. Moreover, individual packets within the same flow may take different paths.

This behavior frequently causes application and service performance issues: for example, the TCP's RTO calculation assumes symmetry.

For this reason, organizations prefer to either:

- Use the links in a failover manner
- Split the portable IP space and advertise only distinct subsets to each AS: continuing the example from Figure 10, AS1000 might advertise 198.51.100.0/25 through AS2000 and 198.51.100.128/25 through AS3000; when a link fails, both smaller subsets are advertised over the remaining link

In both cases, as was the situation when we examined branch offices, the organization is unable to completely and efficiently utilize the full potential of the aggregated connections.

Additionally, one of the drawbacks of eBGP is that the global convergence time for route advertisements is on the order of minutes. This long convergence time means that if a link fails at the head office, then even though the IP/port 4-tuple stays constant, applications will likely time out, terminate their flows, and be forced to reconnect.

> eBGP also suffers from global convergence time on the order of minutes, which causes problems if a head-office link fails

An Enhancement to Connection Aggregation: Smart Blending

In contrast to keeping each flow sticky to a single link, an approach that intelligently splits flows among links—as granularly as on a packet-by-packet basis—is theoretically able to deliver on the full promise of SD-WAN connection aggregation.

This approach—termed 'smart blending'—lets organizations effectively, efficiently, and reliably leverage the complete aggregate capacity of their individual links, even when those links have different and variable capacity and latency.

> A 'smart blending' approach can intelligently split flows among links—while sidestepping the problems already outlined—to deliver the full promise of SD-WAN connection aggregation

Architecture

Similar to cloud-based SD-WAN routers, smart blending at a remote office consists of two main components (see Figure 11):

1. A remote gateway terminal CPE (for example, at a branch office, a mobile site, a nomadic location, etc.)
2. A network service end-point hosted in the cloud

> With smart blending, both the head office and remote locations use the same aggregation techniques—no eBGP required!

The remote terminal supports multiple WAN connections, and creates a tunnel through each to the cloud end-point.

Unlike traditional SD-WAN solutions, aggregating multiple connections at the head office via smart blending is just a mirror of the branch office implementation (see Figure 12): a gateway is installed at the head office, which opens tunneled connections to the cloud-based end-point, which itself is assigned a fixed/static IP address (or larger address space, if required).

Functional Requirements

Achieving high performance results while entirely avoiding the issues outlined previously demands meeting particular requirements.

> Smart blending makes no assumptions about connection performance symmetry—the upstream and downstream can have different capacity

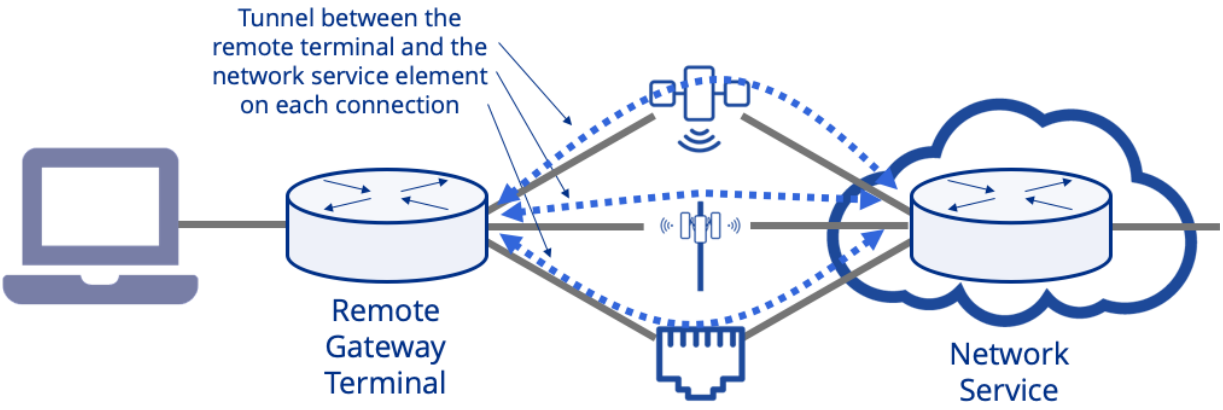

Figure 11 – 'Smart blending' architecture for branch sites

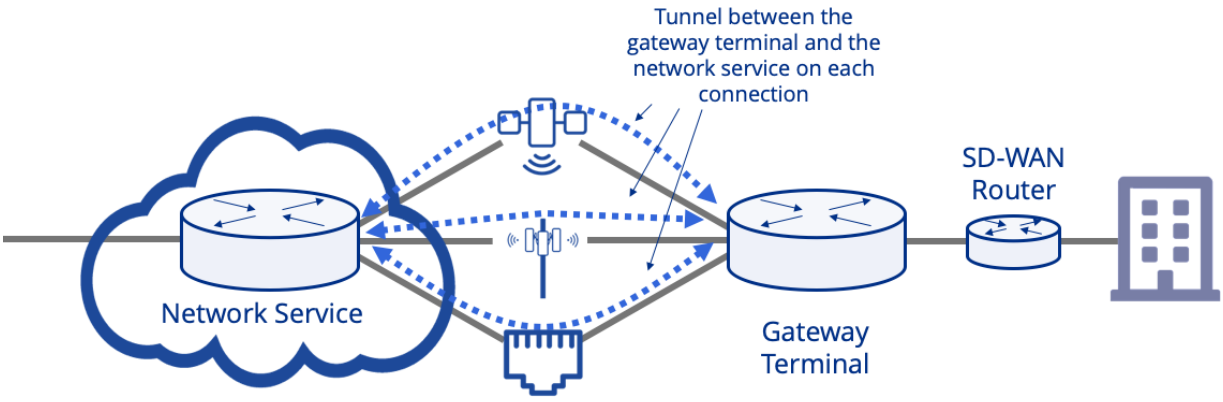

Figure 12 — 'Smart blending' architecture at the head office

Agnostic of Connection Symmetry

To allow blending WAN connections in both the transmit and receive directions, the solution must not make any assumptions about connection performance symmetry.

Instead, smart blending relies on measuring dynamic connection characteristics in real time to inform decision-making.

> Smart blending is only possible if connection characteristics are monitored in real-time

Real-Time Connection Performance Monitoring

To account for dynamic behavior, connection characteristics—including throughput, current latency, and packet loss—must be measured in real time.

Real-time feedback allows the smart blending solution to operate granularly as it determines the best WAN link over which to send a particular packet.

Flow/Application Acceleration

To directly address the complexities introduced by TCP and UDP congestion control algorithms when their flows are split across WAN connections with different properties, smart blending incorporates flow/application accelerators. These accelerators account for the respective protocol's congestion control behavior to ensure that the sender's congestion control algorithm never sees premature or mistaken indications that it has achieved link capacity.

In addition to enabling smart blending to split individual flows across multiple links, these accelerators also enable very specific and demanding applications, like reliable low-latency constant-bitrate video streaming.

> By using flow/application acceleration, smart blending prevents the sender's congestion control algorithm from prematurely slowing down

Aggregating Connections

Smart blending intelligently sends LAN-based flows through tunnels over WAN links, where they're NATed to an IP belonging to the cloud end-point. Like cloud-based SD-WAN, this architecture allows flows to remain unbroken if the WAN links experience failure or IP address change, because the source 4-tuple is preserved.

> Plus, smart blending enables even deeper optimizations, squeezing every last bit of capacity out of the SD-WAN's aggregated connections

Overcoming Traditional SD-WAN Connection Aggregation Challenges

Smart blending overcomes the challenges of traditional SD-WAN connection aggregation by meeting key requirements:

- Individual flows can achieve the maximum aggregated bandwidth of all the WAN links, because a flow can be transparently split across all available links while presenting a consistent 4-tuple
- Failure of a WAN link does not cause flows to terminate, timeout, and reestablish, because the 4-tuple remains unchanged
- Because the solution dynamically accounts for link variability, individual flows can be efficiently split across links with different and variable bandwidth capacity, and across links with different and variable latency

Smart blending overcomes traditional SD-WAN connection aggregation challenges

Further Optimizations

This approach also delivers the operational advantage that new connections can be added dynamically, without needing reconfiguration by an administrator, because connection characteristics are discovered automatically.

Additionally, smart blending can:

- optimize flow assignment based on link characteristics, even dynamically (i.e., as characteristics change over time)
- maximize throughput by determining the optimal packet size of each link
- easily incorporate administrative routing preferences, like setting a target blended bitrate and a priority order of links to use

Conclusions

Relying on cloud-based compute, storage, and networking services to power a wide range of crucial functions means that all of an organization's offices (including mobile and nomadic sites) are increasingly dependent upon reliable connectivity to the public Internet, particularly over the first and last mile.

> Organizations depend on reliable connectivity to the public Internet, from all their offices and locations

SD-WAN lets organizations leverage lower-cost Internet and cloud connectivity, and sometimes even completely replace dedicated, private WAN technologies, while still meeting demanding QoS needs.

SD-WAN's ability to aggregate multiple network links into a single connection is especially important, because it increases reliability.

However, effectively and efficiently distributing traffic from a single flow among multiple WAN links—which likely each have different and variable capacity, and variable latencies—is a significant technical challenge.

Review of Traditional SD-WAN Connection Aggregation

Broadly, traditional SD-WAN solutions divide into two categories:

1. Direct to public Internet
2. Cloud-based router

In the first category, flows go directly to the Internet over the available WAN links, source-NATed with the IP address assigned to each link by the respective ISP.

> Traditional SD-WAN solutions aggregate connections but—due to significant technical hurdles—keep individual flows 'sticky' to a single link

Simpler solutions operate in a failover mode, meaning only one WAN link is active at a time and all flows are sticky to that link. If the active link fails, then all flows are terminated and applications must reopen their flows to be assigned to the new active link. This behavior causes significant problems in some application scenarios.

Most solutions can also operate in a load-balanced mode, in which multiple WAN links are active concurrently, with individual flows assigned to a link based on dynamic algorithms or administrator preference. Once assigned to a link, a flow remains sticky to it for the flow's entire lifetime. If a link fails, then all flows that were assigned to that link are terminated, and the applications must reopen their flows to be assigned to a different active link.

In the second category, the SD-WAN opens a VPN tunnel on each of its WAN links to a cloud-based router, and the flows are sent over these tunnels. The flows are subsequently source-NATed to the public Internet with an IP belonging to the cloud router rather than an IP corresponding to the WAN ISPs.

These solutions can also operate in failover and load-balanced modes, but they have the advantage that when a link fails, the existing flows don't need to be terminated—because the cloud-based IP remains unchanged. Instead, the existing flows are transparently moved to a different VPN tunnel.

It's important to note that the flows still stay sticky to a particular VPN tunnel in steady state, moving only upon a failure event.

Summary of Performance Issues

Because of the technical challenges associated with splitting a flow across multiple links, most SD-WAN solutions keep flows sticky to a single connection.

This approach guarantees mediocre performance: the organization as a whole is able to use the combined connection capacity, but the individual flows aren't; additionally, handling link failure without causing flow timeouts remains a problem for some implementations.

> Keeping each flow sticky to a link guarantees mediocre performance—a reliable minimum, but well below the potential

If the organization is sufficiently large (and therefore can be reasonably expected to have many concurrent flows), then it will experience efficient overall utilization of all links, but individual flows will still be limited to the capacity of only a single link.

A Superior Alternative: Smart Blending

In contrast to keeping flows sticky to connections, an approach that measures real-time link characteristics and intelligently splits flows among links—as granularly as on a packet-by-packet basis—can better deliver the full potential of SD-WAN.

Compared to other connection aggregation techniques, this approach—smart blending—delivers both improved reliability and faster connection speeds, enabling improved connectivity to the Internet, to cloud-based SaaS applications, and to cloud services including storage and compute—especially in mobile and nomadic situations.

> Smart blending delivers improved reliability and faster connection speeds—especially in mobile and nomadic situations

Summary of Key Advantages

As a replacement for traditional, flow-based SD-WAN connection aggregation methods, smart blending is a compelling addition to SD-WANs that delivers significant advantages, including:

1. Achieving high utilization and performance when blending, even with only a single flow, and even with unreliable WAN connections—thereby making SD-WAN viable for mobile and nomadic applications dependent upon unreliable connections or connections of significantly different characteristics
2. Enabling particularly demanding applications, like low-latency constant bitrate video streaming
3. Simplifying operational management and improving failover performance by automatically adapting in real time to the addition or removal (including failure) of WAN connections—additions result in an immediate increase to blended capacity, while failures are transparent (i.e., outstanding flows remain unbroken)
4. Administratively configured connection priorities that dynamically and adaptively use the available links (in priority order) to achieve the target blended bitrates

Dejero

About Dejero

Driven by our vision of reliable connectivity anywhere, Dejero delivers fast and dependable connectivity required for cloud computing, online collaboration, and the secure exchange of video and data.

With our global partners, Dejero supplies the equipment, software, connectivity services, cloud services, and support to provide the uptime and bandwidth critical to the success of today's organizations.

To learn more:

connect@dejero.com

+1 519 772 4824

www.dejero.com