# SMPTE ST2110 & OB truck interconnectivity

## Introduction

Within the Media & Entertainment (M&E) world there are various stakeholders involved in the supply chain and value chain that exists between, say, the live sporting event on the field and the television in your living room used to watch this sporting event. Closer to the action on the field there exists an industry of events production companies in the form of Outside Broadcast (OB) trucks, flypacks, and "rolling racks" used to acquire audio/video content which eventually, among other locations, ends up in your living room.

Within the OB trucks are a skilled team of audio & video experts choreographing many elements of the content displayed on your living room television, your phone, your tablets, and many other electronic viewing devices. In addition to the director, who runs the show, there are audio mixing personnel, color-correction specialists, replay technicians, graphics and sound effects masters, as well as a team of engineers who are assigned to each truck and travel with the trucks as they move from venue to venue.

Over the last several years the industry has adopted the SMPTE ST2110 framework for live production. And as the industry continues to transition from SDI-based workflows to IP-based workflows the presence of IP has enabled the ability to allow for ease of integration when the inevitable need for creating elastic environments suddenly presents itself. What is one to do if there is a premier television sporting event that needs a lot more cameras, replay operators, and other capabilities? Well you bring in more OB trucks, rolling racks, & flypacks to increase the overall capacity needed for pre-game, game, and post-game productions.

Fortunately as a well-informed and enlightened systems architect you had the foresight to choose a COTS-based, open API, standards-based IP solution as the switch fabric within your new IP-based SMPTE ST2110 environment. Using a COTS-based IP & Ethernet switching fabric allows you to quickly expand your fabric using either standards-based computer networking protocols, an SDN controller solution, or both.

We've also all heard about "the cloud" and the ability to "pay as you go" in order to tap into compute, storage, and services/workflows on-demand from many of the well-known cloud service providers. Today it's not unusual for workflows to exist on-the-ground, at various remote broadcast studio locations thousands of miles away, and/or in the aforementioned cloud.

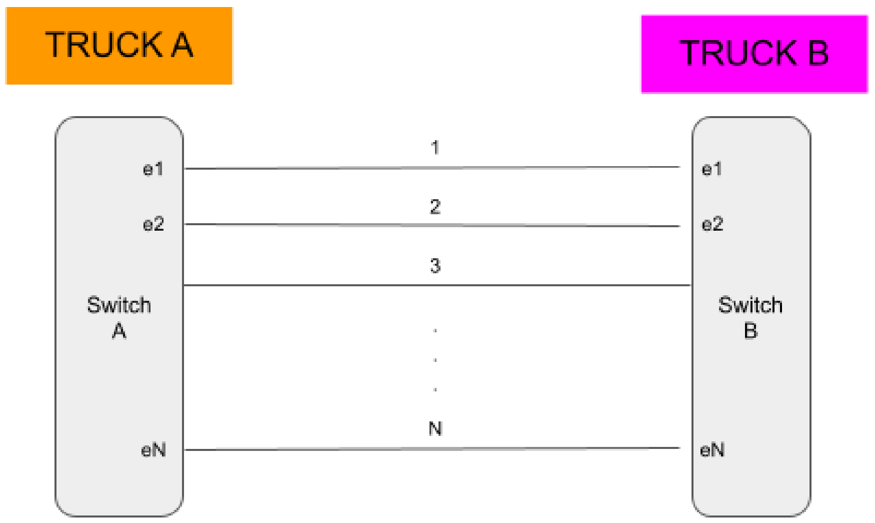
Regardless of whether your event needs more OB trucks on-the-ground, remote workflows that are across the Internet or a leased circuit, or more compute & storage horsepower in the cloud, a COTS-based solution has the ability to make your constantly changing workflow needs more agile, seamless, robust, and manageable in our IP world.

## Creative freedom on the SMPTE ST2110 network

Over the last few years we've had the opportunity to work with some of the top OB truck designers around the world, many of whom have deployed Arista switch fabrics at the heart of their IP designs. As mentioned earlier there are often times when televised events need more than a single OB truck perhaps from the same OB truck manufacturer, as well as situations where many of the competing OB truck manufacturers need to interconnect all their trucks on-site and share SMPTE ST2110 workflows, IP multicast flows, PTP, and more. We are already seeing this shared 2110 IP workflow strategy become more and more frequent and mandatory.

If you've been a voracious reader of Arista's Media & Entertainment deployment and best practices guides you may be familiar with the differences between Layer 2 (L2) and Layer 3 (L3) network designs. There are many advantages in, say, a SMPTE ST2110 environment where we prefer an L3 design for, among other things, increased bandwidth control, packet flooding control, and fault domain control. We've all heard the stories where a technician randomly plugs in an Ethernet switch into another Ethernet switch (because the technician just needed more Ethernet ports to plug in more broadcast devices) and suddenly there are lengthy disruptions to the live broadcast and comm links go down for some brief (yet panic-inducing) and unacceptable period of time. This can occur in L2 environments when there is a Spanning Tree reconvergence or multicast flooding in large L2 domains.

For one of our large OB truck customers, for whom we've designed many trucks over the years, we adhered to SMPTE ST2110 networking best practices. This entailed keeping each truck on their own L3 boundary so that when their trucks need to be connected together (for the reasons mentioned above) having routed interfaces between trucks ("truck-to-truck" or "T2T") makes the most sense. From a very basic IP configuration (there is no consideration, just yet, for IP multicast, routing protocols, PTP, etc) perspective, and from a best practices perspective, let's say that you want N number of links, perhaps as shown below:



Given the physical T2T interconnections above, the basic L3 interconnections would be as follows for 1 of those N links:

**Truck A, Switch A, e1 (basic T2T IP addressing)**

```
interface e1
       description Truck A T2T-1 going to Truck B T2T-1
       no switchport
       ip address 10.0.0.0/31
```

**Truck B, Switch B, e1 (basic T2T IP addressing)**

```
interface e1
       description Truck B T2T-1 going to Truck A T2T-1
       no switchport
       ip address 10.0.0.1/31
```

**NOTE:** Yes, you can have an IP address ending in ".0" and Arista switches support "/31" subnet masks to help conserve IP address space.

So right now we've only configured a single link between Truck A & Truck B (let's call it "T2T-1") connecting the 2 trucks. This, of course, is a single point of failure. What if I want more than a single link? And what if a single link is just not enough bandwidth? As of this writing (year 2023), you can easily have a 400GbE link as T2T-1, with 800GbE and 1.6TbE on the horizon.

As the picture above implies we can have N links between our 2 OB trucks. The more links we provide, the more overall bandwidth between the 2 trucks for our SMPTE ST2110 workflows. And, of course, we'd have to configure (among other things) IP addressing for both Switch A & B's e2, e3, e4, …, eN interfaces, where the e2's in both trucks might look like this (note the differences from e1 in red):

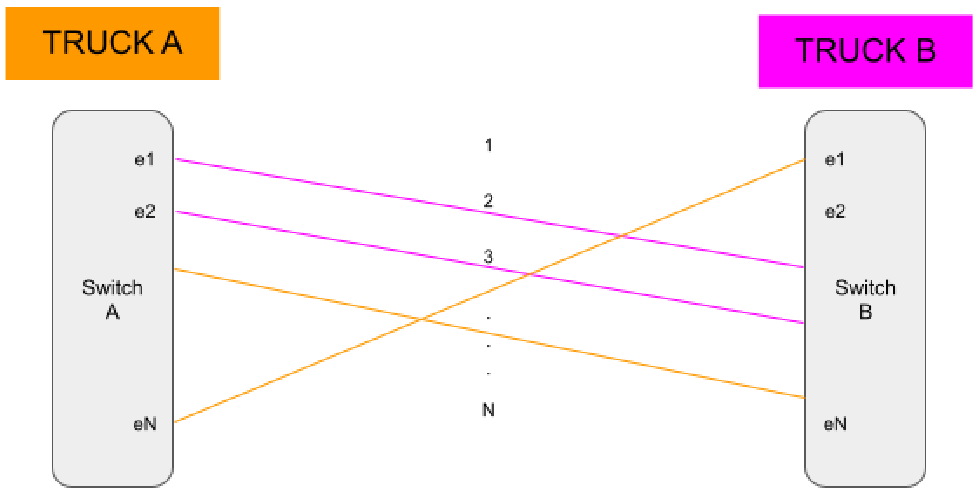
**Truck A, Switch A, e2 (basic T2T IP addressing)**

```
interface e2
      description Truck A T2T-2 going to Truck B T2T-2
      no switchport
      ip address 10.0.0.2/31
```

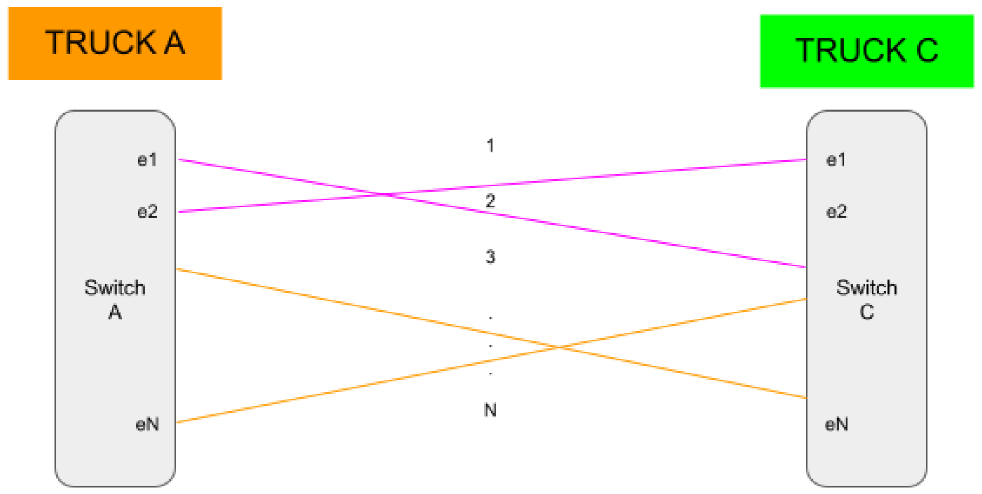
**Truck B, Switch B, e2 (basic T2T IP addressing)**

```
interface e2
      description Truck B T2T-2 going to Truck A T2T-2
      no switchport
      ip address 10.0.0.3/31
```

The keen observer will realize that given the configs above for each of the interfaces (e1, e2, …, eN) that T2T-1 must always go between Truck A's e1 and Truck B's e1, and T2T-2 must always go between Truck A's e2 and Truck B's e2, and so on for all N links. Otherwise the IP addressing must change if, say, someone wants to connect Truck A's e1 to Truck B's e2 - or maybe a technician accidentally connects the wrong T2T links. Wouldn't it be nice if we did not have to worry about e1 connecting to e1, and e2 connecting to e2, etc and just randomly connect the 2 trucks routed interfaces as below:



Or what about when Truck A needs to connect to Truck C for an event next week across the country? Do we now have to re-IP Truck C's IP addresses to mate with Truck A's IP addresses (where Truck C would then have the same IP's as Truck B on all T2T links)? What about when Trucks B & C come together someday further into the future?



Wouldn't it be nice if any OB truck could connect its SMPTE ST2110 network to any other truck's SMPTE ST2110 network easily and just be "plug and play"? There are many ways one could solve this problem, here are just a few:

- Use Arista CloudVision with pre-defined T2T scenarios already built
- Ansible playbooks also with pre-defined scenarios already built
- Use the following IETF RFC compliant (RFC 1812) "pure" networking approach with IP unnumbered

## What is IP unnumbered?

IP unnumbered is a network configuration & design method where multiple interfaces share the same IP address. Instead of assigning a set of dedicated IP addresses from a limited pool to, say, point-to-point links (where there is a matching pair of /31 addresses on each link) and having to manage these IPs and track them, you can reuse the IP address from a loopback (or other) address thereby relieving the design engineer/architect of some responsibility. This allows for more efficient use of IP address space.

Recall from earlier that these T2T interfaces are routed (or L3) interfaces. Essentially they need an IP address assigned at each end, but we want the flexibility of being able to connect any truck (hence, any switch) to any other truck (switch) without having to reconfigure IP addresses on interfaces constantly. The OB trucks are constantly moving around the country/world every week for a different televised event and each new event will require a change in (what is essentially) the network topology. Unlike your typical enterprise or cloud datacenter, the OB truck fabric can grow or shrink as time moves forward. Most datacenters just grow - monotonically increasing in size - they rarely shrink from week to week. The OB trucks are assigned to various events around the globe based on their geographic location at any point in time. Therefore it's highly probable that for events needing more OB trucks that the allocation of trucks needing to be mated together in order to share SMPTE ST2110 workflows needs to be flexible enough to allow for any truck to any truck configurations.

Behind those routed/L3 T2T interfaces exist a bunch of IP subnets within each truck. There may be subnets for cameras, audio devices, replay devices, multiviewers, gateways, and other broadcast devices. How might we inform the other (mated) OB trucks of these IP subnets? We can either statically define routes between the trucks, or use a more agile and robust method via dynamic routing protocols.

## Routing

Let's pretend that within a particular OB truck you have an L3 network on the SMPTE ST2110 fabric and that amongst Ethernet/IP switches within that truck fabric you decide to use OSPF as the routing protocol to advertise IP subnets within the truck. Well when the day comes that requires more than just one truck at an event, connecting two trucks is rather trivial in order to spread workflows around.

If you own both trucks, presumably you trust what's going on in each truck. Hence one could forgo needing to implement extra security and policy measures in order to protect Truck A from Truck B and vice-versa. Note that in heterogeneous environments where the OB trucks being mated together are owned/built by different manufacturers, policy discussions may be beneficial for all stakeholders and fortunately a COTS-based solution (like Arista Networks) allows for flexibility in policy instantiation.

So we want to connect Trucks A & B together and our goal is to make sure that the routed/L3 interfaces don't need an explicit IP address and we also need to run our chosen routing protocol (OSPF) to advertise IP subnets between the two trucks. Let's look at some possible configs now for our two switches from examples earlier:

**Truck A, Switch A (IP unnumbered)**

```
interface lo100
       description OSPF-IP-unnumbered
       ip address 10.99.99.99/32
       ip ospf area 0.0.0.0

interface e1
       description Truck A T2T-1 going to other Truck T2T-1
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0

interface e2
       description Truck A T2T-2 going to other Truck T2T-2
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0
                                 .
                                 .
                                 .
interface eN
       description Truck A T2T-N going to other Truck T2T-N
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0
```

**Truck B, Switch B (IP unnumbered)**

```
interface lo100
       description OSPF-IP-unnumbered
       ip address 10.199.199.199/32
       ip ospf area 0.0.0.0

interface e1
       description Truck B T2T-1 going to other Truck T2T-1
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0
```

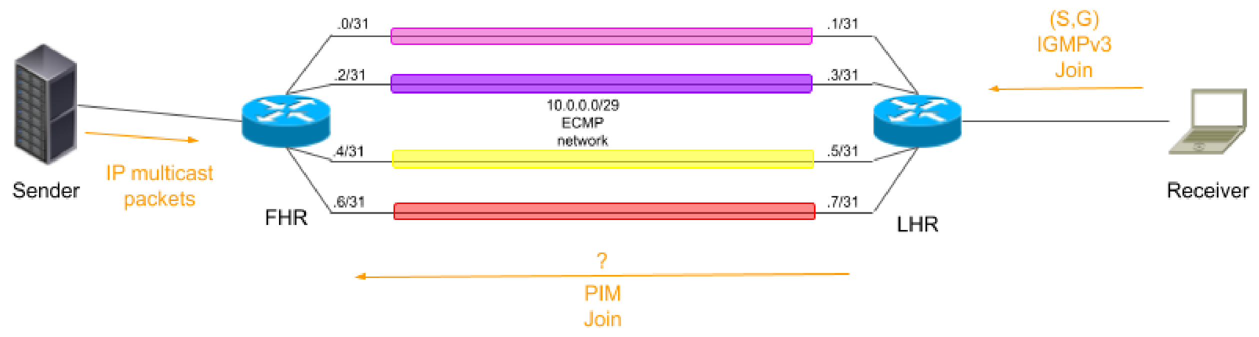
```
interface e2
       description Truck B T2T-2 going to other Truck T2T-2
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0
                           .
                           .
                           .
interface eN
       description Truck B T2T-N going to other Truck T2T-N
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0
```

Interesting to note, the loopback 100 ("lo100") addresses don't need to be on the "same" IP subnet (Truck A has 10.99.99.99/32 and Truck B has 10.199.199.199/32) and also notice that for each T2T interface we're reusing the same loopback address ("lo100") on all N links thereby conserving IP addresses and making the configuration easier. Some of you might stop and wonder to yourself how this might affect flow distribution on the N links for both IP unicast or IP multicast flows.

## ECMP, IP multicast, & IP unnumbered

When you have multiple parallel interfaces between trucks (if not for anything else, for redundancy purposes) there is a selection mechanism in order to determine which of these many paths a packet goes. Let's use the new diagram below as an example.

The FHR (or First Hop Router) is an L3 router running PIM-SM (Protocol Independent Multicast - Sparse Mode) that is closest to the sender. The LHR (or Last Hop Router) is the L3 router running PIM-SM that is closest to the receiver. When a receiver desires to receive a particular IP multicast packet it'll issue an IGMP Join message. Let's pretend that the receiver is IGMPv3 compliant and asks to receive an IP multicast flow ("G") from a particular sender ("S") - in this case it'll issue an IGMPv3 (S,G) Join. When this IGMP Join reaches the LHR, the LHR will issue a PIM Join towards the direction of the sender ("S"). It knows which direction to send that PIM Join towards based on its unicast IP routing table entry for "S" (the path to "S").



In the above case the LHR has to pick which of the 4 colored links to send that PIM Join down. How does it determine that? Before the PIM Join leaves the LHR performs a hash calculation based on 4 attributes (a "4-tuple"): (S, G, nextHopIP, interfaceID). The "S" & "G" are self-explanatory - they are the (S,G) in the original IGMP Join. The "nextHop IP" is the value of the IP address on the other side of the /31 link (on the FHR side, in the above diagram). Of the 4 possible inputs for "nextHop" (one per link), the "nextHopIP" values in each calculation will be "10.0.0.0", "10.0.0.2", "10.0.0.4", and "10.0.0.6", respectively. The "interfaceID" is a unique 32-bit identifier assigned to the interface that the PIM Join will leave the LHR from. Note that this LHR egress interface (that the PIM Join leave from) will then eventually be the IIF (Incoming Interface) for the IP multicast flow originating from the Sender. Likewise, when the multicast routing table on the FHR is programmed, its OIL (Output Interface List) will also contain that same physical link (that was chosen at the prior PIM Join stage for a particular (S,G) or (*,G)) when the multicast tree from the sender to receiver is built.

Speaking of which, what about the situation when the receiver issues an IGMP (*,G) Join? The "S" value that is then chosen for the hash calculation at the LHR will then be the RP (Rendezvous Point) IP address. The hash that is highest across all 4 links will determine which of the 4 links is chosen. If 2 or more hash values are equal, then the link with the highest IP address will be chosen.

You'll also notice that the T2T interfaces in the diagram have unique /31 IP routed interface addressing. How might the hash algorithm change when I use IP unnumbered and I reuse the same loopback IP address on all 4 links? In that case the 4-tuple hash is reduced down to a 3-tuple hash: (S, G, interface ID)

## PIM & PTP

For SMPTE ST2110 we need to configure PIM-SM and PTP with the proper SMPTE 2059-2:2015 PTP profile and message rates on these interfaces. For brevity, let's just look at a single interface on a single truck of a T2T link and put it all together.

**Any truck, any switch (IP unnumbered, PIM, PTP)**

```
interface eN
       description Truck B T2T-N going to other Truck T2T-N
       no switchport
       ip address unnumbered lo100
       ip ospf network point-to-point
       ip ospf area 0.0.0.0
       pim ipv4 sparse-mode
       ptp enable
       ptp sync-message interval -3
       ptp announce interval 0
       ptp delay-req interval -3
```

There are, of course, global configurations (PIM, PTP, and routing protocol) needed to round-out the complete configuration for a fully-operational switch fabric. Assuming that the remainder of the configurations for the switches pertaining to OSPF, PIM, IP routing, and PTP are present and accurate, you are on your way to building a more robust & dynamic T2T fabric.

Let's pretend that each of Truck A & B has their own PTP GMs (because these trucks may also run independently from each other) and that each truck runs on the same PTP domain, assuming that the PTP Priority1 & Priority2 values are appropriately chosen, the PTP BMCA (Best Master Clock Algorithm) will just choose a prevailing GM to synchronize the entire fabric of the combined trucks. Essentially one of the GMs from one of the trucks will become the timing source for both trucks.

## RP design strategy

If we consider the single truck situation, say there only exists Truck A, in order to handle PIM-SM and the case of IGMP (*,G) Joins or devices that are not capable of IGMPv3 (S,G) Joins we need to have an RP within the truck. In fact, each truck needs an RP because each truck can run independently. But what happens when 2 or more trucks need to come together for an event?

In a dynamic PIM-SM environment there are some IETF protocols which can be of assistance here - notably AnyCastRP and MSDP. Both protocols are used to synchronize RP state and tell the other truck(s) about active senders (aka "sources") from within their own truck. In a SMPTE ST2110 environment IP multicast video flows consume many gigabits per second and IP multicast multi-channel audio flows consume many 100's of megabits per second, therefore bandwidth consumption, predictability, & control are important.

In general, we try to design systems that place RPs closer to the IP multicast senders in each truck, and since trucks can run independently from one another there is already an RP in each truck. Having an RP in closer proximity to the sender can reduce the possibility of disruptive & bursty PIM Register traffic needed to build the IP multicast routing tables when the switchover from shared-tree RPT (Root Path Tree) to shortest-tree SPT (Shortest Path Tree) occurs.

From an RP perspective, we essentially need to (1) know where our RP(s) exist and (2) synchronize the necessary state between all of our RPs. Determining the location of RPs can be configured manually or automatically through another IETF protocol called BSR (Bootstrap Router). Since BSR only allows for a single active RP in a PIM domain (while all other RP candidates are just passive backups) we don't recommend using BSR in a SMPTE ST2110 environment due to the large bandwidths and the fact that by definition the active RP would not be the "closest" RP for all active sources but only some sources.

From an IP addressing standpoint every RP is going to have the same IP address - an AnyCastIP address. Well, how can this be? Given that we are using a dynamic routing protocol (OSPF in this case), each RP will advertise this same AnyCastIP address out to the surrounding routers. Every other router (non-RP) in the switch fabric will find its shortest path to its nearest RP that shares this AnyCastIP address. State is then shared via AnyCastRP to the other RPs.

Let's take a look at some AnyCastRP example configurations.

**Truck A, Switch A**

```
interface lo0
   ip address 10.1.0.1/32
   ip address 10.0.0.1/32 secondary (This will be the AnyCastIP address)
ip routing
router multicast
   ipv4
      routing
router pim sparse-mode
   ipv4
      ssm range standard
      rp address 10.0.0.1
      anycast-rp 10.0.0.1 10.1.0.1 (This will be the RP for Truck A, this truck)
      anycast-rp 10.0.0.1 10.2.0.1 (This will be the RP for Truck B)
      anycast-rp 10.0.0.1 10.3.0.1 (This will be the RP for Truck C)
```

**Truck B, Switch B**

```
interface lo0
   ip address 10.2.0.1/32
   ip address 10.0.0.1/32 secondary (This will be a virtual AnyCastIP address)
ip routing
router multicast
   ipv4
      routing
router pim sparse-mode
   ipv4
      ssm range standard
      rp address 10.0.0.1
      anycast-rp 10.0.0.1 10.1.0.1 (This will be the RP for Truck A)
      anycast-rp 10.0.0.1 10.2.0.1 (This will be the RP for Truck B, this truck)
      anycast-rp 10.0.0.1 10.3.0.1 (This will be the RP for Truck C)
```

For any possible truck that, say, Truck A might connect to someday, you can add more `anycast-rp` lines to the config above for each of the other possible RPs in the connected T2T system. Do note that the state synchronization between all RPs is done via a tunneled unicast UDP/IP PIM Register packets and hence can follow a unicast forwarding path out a non-PIM interface. So if a truck is not present and there is a default route in the configuration be careful that you aren't inadvertently sending this UDP traffic into a blackhole (perhaps a perimeter firewall).

For those wishing to safeguard the above situation, you have many options. You could comment out a particular truck RP config (ie. comment out the specific truck anycast-rp config line) that's not physically present and connected. You could use a null0 static route that is summarizable or with a high administrative cost per truck route back to its loopback address used above. When another truck (in the AnyCastRP list) is physically present and dynamically advertising its loopback addresses via an L3 routing protocol the lower administrative distance route (from the dynamic routing protocol) will prevail over an administrative distance of 255.

Note that any other (non-RP) switches in the truck will just need this rp address 10.0.0.1 (which is the AnyCastIP address of the entire switch fabric) config line and not need any of the anycast-rp config line(s). All routers will take the shortest path to the closest AnyCastIP address and will land on its own RP first, assuming its own RP is the point of egress to another truck.

There are certainly various strategies with RPs that one can consider such as placement of RPs, number of RPs, using certain RPs for certain IP multicast ranges, etc. Each design requires different requirements and the real advantage of using a COTS product is the ability to be able to utilize these open-standard based protocols in very flexible, extensible, and agile manners.

## Conclusion

These are just some of the considerations for the various event production companies to consider when bringing together two or more SMPTE ST2110 entities, whether it's two or more OB trucks, two or more rolling racks or flypacks, or trucks mated with rolling racks and flypacks. The approaches can range from a purely "plug and play" open-standards dynamic environment built on open-standard IETF & IEEE networking protocols to an SDN (Software Defined Networking) approach.

A plug and play approach may be an acceptable approach where bandwidth, control, and trust are guaranteed. If all trucks/rolling racks/flypacks are owned and operated by the same entity, in many cases these three attributes are implied. In the latter SDN approach you may consider Arista MCS (Media Control Service) which provides for a robust, resilient, and agile software integration between your broadcast control software and an Arista switch fabric. For those customers requiring more overall workflow control, more workflow & salvo performance, more bandwidth control, and specific architecting of IP multicast flow pathing within arbitrary topologies, we invite you to learn more about Arista MCS on our website.

If competing OB truck (or rolling rack/flypack) vendors must come together to collaborate from a SMPTE ST2110 perspective for a live production, it's perfectly plausible (and already proven in large global sporting events in 2023) for all vendors to come together from an open-standards perspective using standard networking protocols. These protocols can also allow for the addition of policies and rulesets given that the aforementioned implicit trust may not always be present - in much the same way as multiple ISPs come together to share information and traffic flows. One can also build a hybrid SMPTE ST2110 system using both SDN and non-SDN approaches together. See the following link about bringing together both Arista MCS and non-MCS environments.

The flexibility of a COTS, open-standards, open API switch fabric (like Arista Networks) also allows for robust interfacing & peering needs with non-COTS and proprietary legacy-inspired IP-based vendors, helping to give these non-COTS vendors a seat at the SMPTE ST2110 interoperability table, especially in larger multi-vendor multi-OB truck or cloud workflow environments. We will take a look at this topic in a future paper in this series.

In addition, future series papers will dive a bit deeper into the IETF/IEEE protocols mentioned earlier. We will also talk about the complementary network to the SMPTE ST2110 network used to managed everything - the management (or "out of band") network for GUI access, web interface access, software/firmware updates, Internet and VPN connectivity, logging, telemetry, and WiFi for the local & remote VPN users.

Whatever your architectural approach will be, let's not forget that the stakes are high within live television production across many stakeholders. Uptime and MTBF (Mean Time Between Failures) are two of many operational metrics within the hardware electronics world that align with reliability and stability. Bug counts and security vulnerabilities are just two of the many operational metrics that are valued in the software world. Combined together, all of these metrics can and will affect your business.

Before you choose Arista Networks as your SMPTE ST2110 switch fabric know that we prioritize product quality and stability within our own hardware & software designs, as well as within our culture. This is reflected in our annual analyst ratings as leaders in the Ethernet/IP switching infrastructure business and also within our world-class Net Promoter Score (NPS) ratings. The last thing that we want is for our Media & Entertainment customers to worry about the Ethernet/IP switch fabric at the heart of their SMPTE ST2110 environment. In fact, our co-founder, Ken Duda, is so committed to your success and Arista product quality that he hands out his personal cell phone number to every customer that he meets asking them to call him directly if there is ever a problem that seems insurmountable.

**Santa Clara—Corporate Headquarters**
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

**Ireland—International Headquarters**
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

**Vancouver—R&D Office**
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

**San Francisco—R&D and Sales Office 1390**
Market Street, Suite 800
San Francisco, CA 94102

**India—R&D Office**
Global Tech Park, Tower A, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

**Singapore—APAC Administrative Office**
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

**Nashua—R&D Office**
10 Tara Boulevard
Nashua, NH 03062