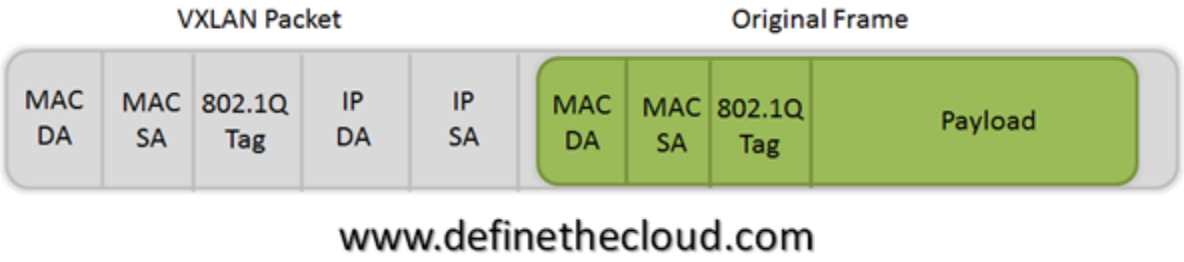
VXLAN DEEP DIVE – PART II

Let's start with the basic concept that VXLAN is an encapsulation technique. Basically the Ethernet frame sent by a VXLAN connected device is encapsulated in an IP/UDP packet. The most important thing here is that it can be carried by any IP capable device. The only time added intelligence is required in a device is at the network bridges known as VXLAN Tunnel End-Points (VTEP) which perform the encapsulation/de-encapsulation. This is not to say that benefit can't be gained by adding VXLAN functionality elsewhere, just that it's not required.



Providing Ethernet Functionality on IP Networks:

As discussed in Part 1, the source and destination IP addresses used for VXLAN are the Source VTEP and destination VTEP. This means that the VTEP must know the destination VTEP in order to encapsulate the frame. One method for this would be a centralized controller/database. That being said VXLAN is implemented in a decentralized fashion, not requiring a controller. There are advantages and drawbacks to this. While utilizing a centralized controller would

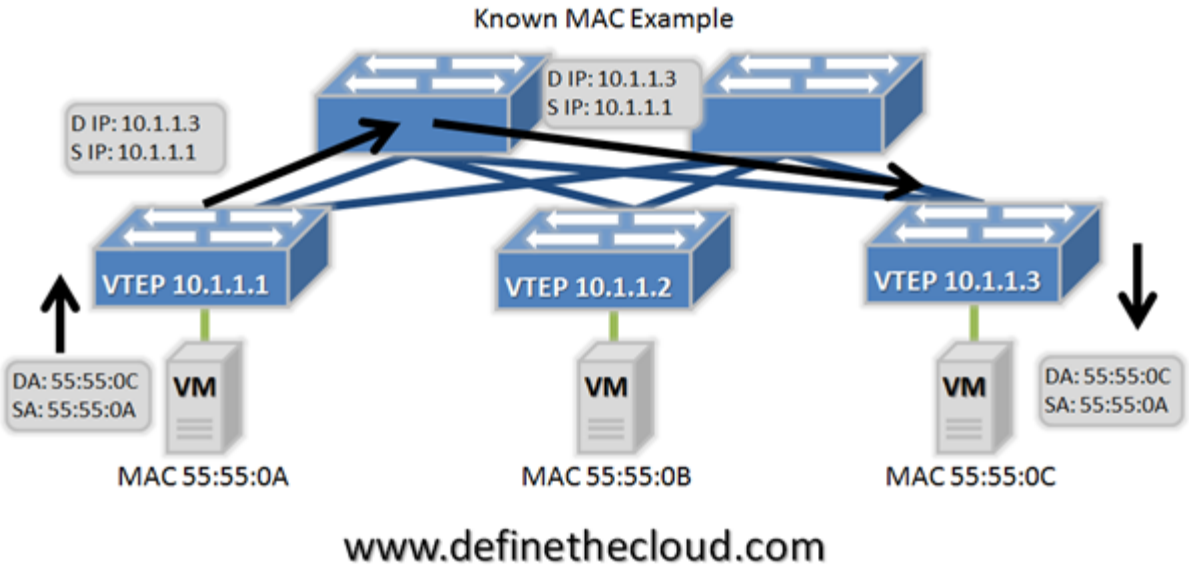
provide methods for address learning and sharing, it would also potentially increase latency, require large software driven mapping tables and add network management points. We will dig deeper into the current decentralized VXLAN deployment model.

VXLAN maintains backward compatibility with traditional Ethernet and therefore must maintain some key Ethernet capabilities. One of these is flooding (broadcast) and 'Flood and Learn behavior.' I cover some of this behavior here (http://www.definethecloud.net/data-center-101-local-area-network-switching) but the summary is that when a switch receives a frame for an unknown destination (MAC not in its table) it will flood the frame to all ports except the one on which it was received. Eventually the frame will get to the intended device and a reply will be sent by the device which will allow the switch to learn of the MACs location. When switches see source MACs that are not in their table they will 'learn' or add them.

VXLAN is encapsulating over IP and IP networks are typically designed for unicast traffic (one-to-one.) This means there is no inherent flood capability. In order to mimic flood and learn on an IP network VXLAN uses IP multi-cast. IP multi-cast provides a method for distributing a packet to a group. This IP multi-cast use can be a contentious point within VXLAN discussions because most

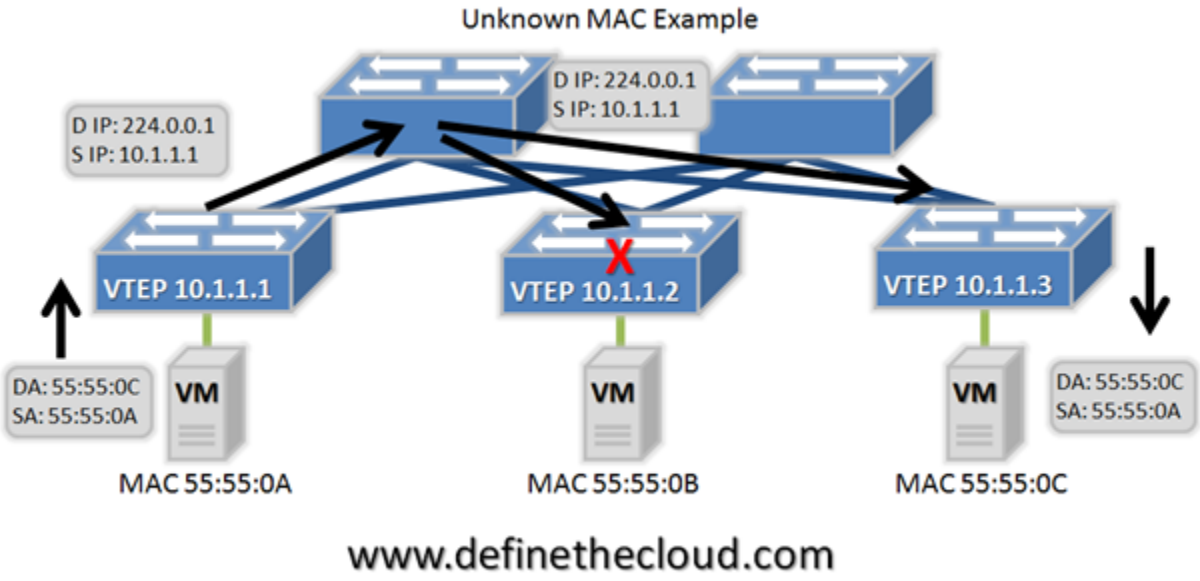
networks aren't designed for IP multi-cast, IP multi-cast support can be limited, and multi-cast itself can be complex dependent on implementation.

Within VXLAN each VXLAN segment ID will be subscribed to a multi-cast group. Multiple VXLAN segments can subscribe to the same ID, this minimizes configuration but increases unneeded network traffic. When a device attaches to a VXLAN on a VTEP that was not previously in use, the VXLAN will join the IP multi-cast group assigned to that segment and start receiving messages.



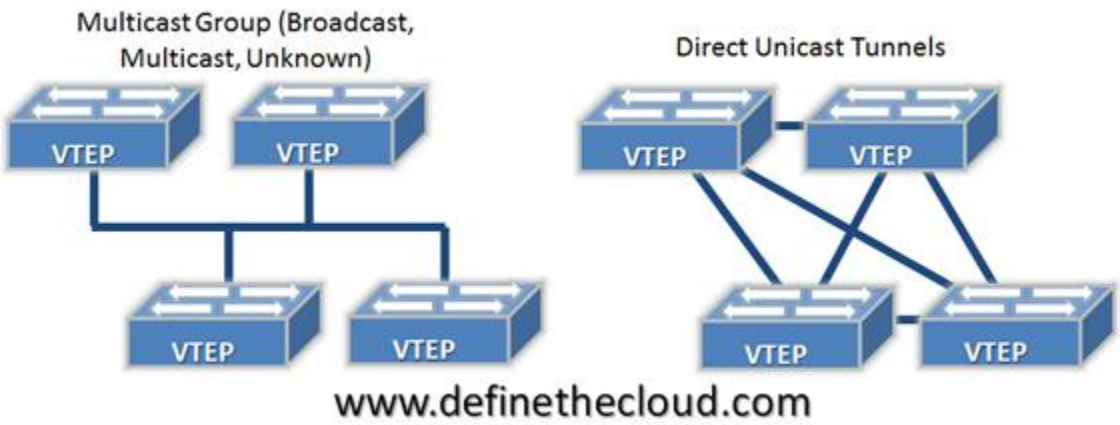
In the diagram above we see the normal operation in which the destination MAC is known and the frame is encapsulated in IP using the source and destination VTEP address. The frame is encapsulated by the source VTEP, de-encapsulated at the destination VTEP and forwarded based on bridging rules from that point.

In this operation only the destination VTEP will receive the frame (with the exception of any devices in the physical path, such as the core IP switch in this example.)

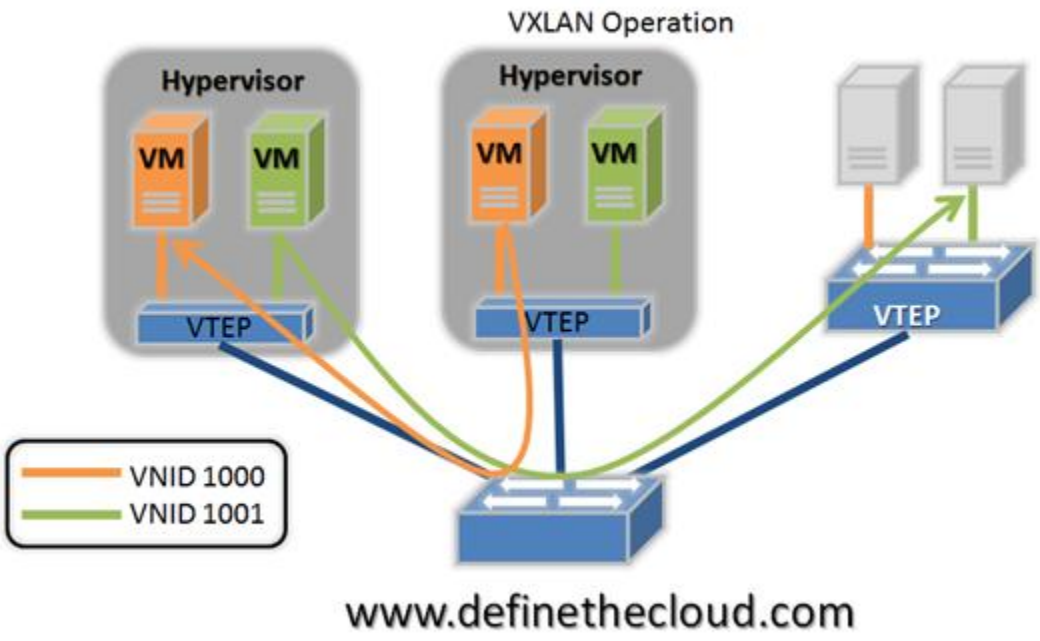


In the example above we see an unknown MAC address (the MAC to VTEP mapping does not exist in the table.) In this case the source VTEP encapsulates the original frame in an IP multi-cast packet with the destination IP of the associated multicast group. This frame will be delivered to all VTEPs participating in the group. VTEPs participating in the group will ideally only be VTEPs with connected devices attached to that VXLAN segment. Because multiple VXLAN segments can use the same IP multicast group this is not always the case. The VTEP with the connected device will de-encapsulate and forward normally, adding the mapping from the source VTEP if required. Any other VTEP that receives the

packet can then learn the source VTEP/MAC mapping if required and discard it. This process will be the same for other traditionally flooded frames such as ARP, etc. The diagram below shows the logical topologies for both traffic types discussed.



As discussed in Part 1 VTEP functionality can be placed in a traditional Ethernet bridge. This is done by placing a logical VTEP construct within the bridge hardware/software. With this in place VXLANs can bridge between virtual and physical devices. This is necessary for physical server connectivity, as well as to add network services provided by physical appliances. Putting it all together the diagram below shows physical servers communicating with virtual servers in a VXLAN environment. The blue links are traditional IP links and the switch shown at the bottom is a standard L3 switch or router. All traffic on these links is encapsulated as IP/UDP and broken out by the VTEPs.



Source: http://www.definethecloud.net/vxlan-deep-divepart-2/