

Two Models for IEEE 802.3ah EPONs

Norman Finn, Cisco Systems

1.0 Introduction

As pointed out in “Spanning Trees and IEEE 802.3ah EPONs”, an EPON operating in its native point-to-multipoint mode is incompatible with the current IEEE 802.1 Spanning Tree Protocols. In addition, routers, hosts, and bridges, insofar as they know about Ethernet LANs, know only about point-to-point and shared LANs. They do not know about point-to-multipoint LANs.

For example, when a router receives an IP multicast packet from an Ethernet interface, it assumes that all of the stations on that interface have also received the packet, and that it does not have to reflect that packet back to that same interface. This is not true if the router’s Ethernet interface is an EPON OLT. Similarly, a file server attached to an OLT, which is not also a router, would prevent an ONU router from reaching any hosts on the other ONUs.

If we assume that the only application for an IEEE 802.3ah EPON is an ISP connected to residential customers, that the ISP can run special software in its OLT-end router/bridge/switch to accommodate an EPON, that ONUs (or the customers’ computers!) are similarly modified, that no second ISP will be required by law to access customers on that same EPON, and that residential customers will never want to run the current standard for IP multicast among each other, or bridge or hub their homes together, or share an 802.11 wireless hub, then all of these bridging and routing issues can be ignored by IEEE 802.3. Of course, this is not the case. An EPON is perfectly applicable to an enterprise campus environment, or to any number of other current and future scenarios. A LAN that cannot fit easily into the existing standards base is clearly handicapped in its bid to become widely accepted.

Therefore IEEE 802.3ah plans, at this writing, to provide a means whereby the OLT and ONUs can emulate a bundle of point-to-point Ethernets. This allows standard Ethernet-compatible devices such as bridges, routers, and hosts to connect to the EPON-based LAN. However, this mode of operation denies to the connected devices one of the great benefits of the EPON: the ability to send one frame from the OLT that is received by all of the ONUs. With the point-to-point emulation model, all downstream broadcasts must be transmitted serially to each receiving ONU.

Another possibility that has been discussed by IEEE 802.3ah (but not adopted at this writing) is to have the OLT, either above or below the MAC layer, automatically reflect all frames received from any ONU back down to all ONUs except the originating ONU. At the OLT end, only one MAC is present, which transmits every frame to all ONUs. In this scenario, the EPON-based LAN emulates perfectly a standard IEEE 802.3 shared LAN. The problem is that, in the service provider world, the ONU customers are not isolated from one another; each receives all of the other ONUs’ point-to-point traffic. Furthermore, all upstream (ONU-OLT) traffic, even that not destined to any other ONU, is reflected down to the other ONUs, a clear waste of bandwidth.

There are thus two opposing views of the utility of an IEEE 802.3ah EPON: a non-standard, but efficient, tool for residential service providers; or a standard, but inefficient, tool for standards-based users. The purpose of this document is to attempt to bring these views together. Section 2.0

on page 2 describes a means whereby an IEEE 802.1D bridge might accomplish the shared LAN emulation required to make an EPON conformant to the expectations of existing standards. Section 3.0 on page 9 describes a means whereby the shared LAN emulation can meet the needs of the residential service provider, without requiring a large number of existing standards, by IEEE, IETF, and others, be revised. In Section 5.0 on page 13, a very different alternative is briefly described, because it has been mentioned on the e-mail exploder and should not be ignored. Closing arguments are in Section 6.0.

Note: Version 4 of this document differs from version 3 in three ways: 1) The term “LMAC”, or “Logical MAC” has been used to differentiate between the full IEEE 802.3 MAC layer and shim sublayers that appear to (sub)layers above or below the LMAC to look like MACs. 2) As suggested by Dolors Sala, the “Advanced Upper-Layer Shared LAN Emulation” of Section 2.5 on page 5 makes use of the point-to-point capability, as well as the point-to-multipoint capability, of the EPON. The latter change, in turn, drove a number of detailed changes to Section 4.0. 3) The need for certain control functions, as described in Section 2.6 on page 8, has been recognized and added.

2.0 Intelligent Shared LAN Emulation

The disadvantages of a “naïve” shared LAN emulation, in which all frames from ONUs are reflected by the OLT, are obvious, especially in an environment where most upstream traffic (ONU to OLT) is not directed towards any other ONU. The appropriate application of bridge-like frame forwarding can avoid needless reflections.

2.1 Previously Discussed Method for Shared LAN Emulation

To date, the only workable plan discussed for emulating a shared LAN is something similar to that shown in Figure 1. Aside from the wastage of bandwidth caused by reflecting all upstream traffic, this figure shows one of the problems of emulating a shared medium: the placement of the shared LAN emulation module which reflects frames back down towards the ONUs.

In Figure 1, if Emulator 1 is responsible for reflecting the frames back down to the ONU, it has problems prioritizing reflected frames and frames transmitted down the stack from higher layers. If Emulator 2 is responsible for reflecting frames, then how is the identification of the transmitting ONU or the selection of destination ONU(s) to be passed through the MAC layer?

2.2 Logical MACs and LMAC IDs

It has not been decided by IEEE 802.3ah whether or not a single ONU can have more than one “logical” MAC. However, in order to accommodate this possibility, and in order to handle cases where an ONU’s MAC and a (clearly) logical emulated LMAC in the OLT are equivalent, we will refer to “Logical MACs” or “LMACs” in this document. Even if an ONU is never allowed to have multiple LMACs, the ONU with a single LMAC is a useful concept in this document.

We will use the term “LMAC ID” in this document to refer to a field, carried with every data frame, which refers to a particular Logical MAC.

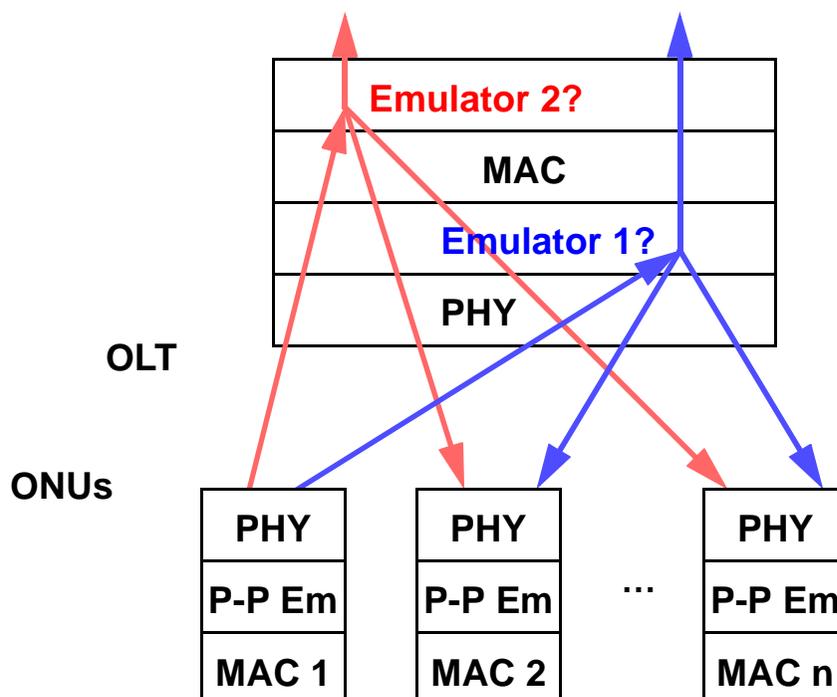


FIGURE 1. Problematical model for Shared LAN Emulation

2.3 MAC Model for Intelligent Shared LAN Emulation

In Figure 2 is the current IEEE 802.3ah model for emulating a bundle of point-to-point LANs, but not a shared medium.

Note that, in Figure 2, the many OLT's LMACs are not separate complete MAC functions; they are virtual, or logical, MACs visible to the higher layers. Presumably, most actual implementations would include a single MAC function. The MAC/Emulation/PHY stack and the upper layers would pass an LMAC selector along with the data frame.

A new model, capable only of emulating a Shared LAN, is shown in Figure 3. In this model, there are two layers of emulation, an upper and a lower. At the lower layer, the OLT LMACs in the point-to-point emulation are inverted; each OLT LMAC reaches all ONU LMACs *except* the corresponding ONU LMAC. One is tempted to call the lower-layer LMACs “anti-LMACs”. In addition, there is an extra LMAC 0 which reaches all ONU LMACs. The upper layer of emulation utilizes the lower-layer MACs and other Layer 2 information to emulate a single shared LAN Emulated LMAC. On the ONU side, each independent shared LAN emulation layer serves what is, to its upper layers, a shared LAN LMAC.

Note that Figure 3 does not emulate a bundle of point-to-point LANs. It does *not* include any provision for any LMAC which is point-to-point in the OLT-ONU direction. This part of the shared LAN emulation is, however, very similar to the functions of the OLT point-to-point LAN emulation shown in Figure 2. In particular, the lower-layer function does *not* reflect any frames back towards the ONUs.

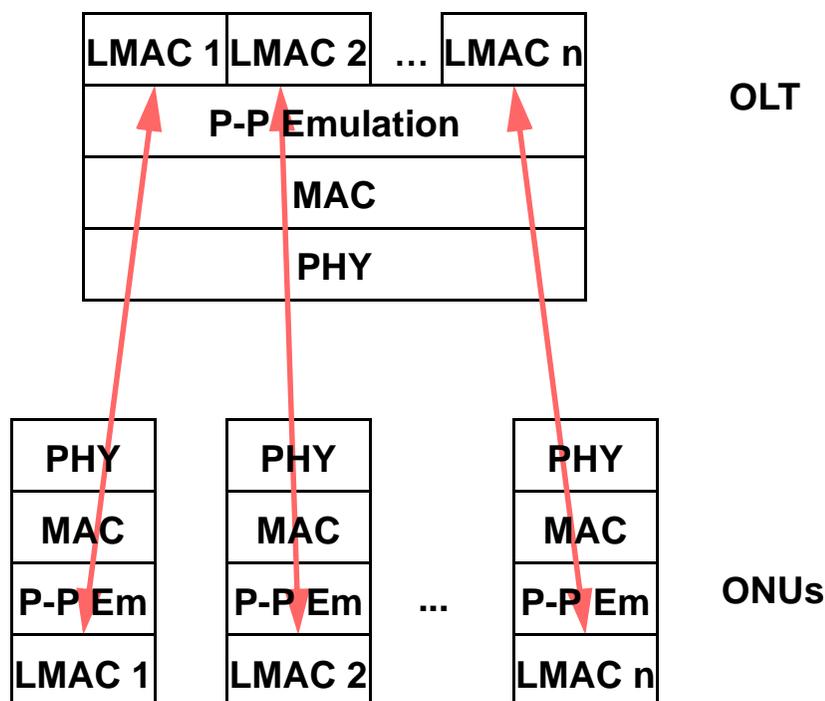


FIGURE 2. MAC model for Point-to-Point LAN Emulation

The lower-layer emulation of Figure 3 needs to perform no more prioritization than does the row of LMACs in the point-to-point emulation of Figure 2. Prioritization of downstream and reflected traffic is performed by the upper-layer emulation. The provision of multiple LMACs at the lower-layer solves the problem of specifying the destination of reflected frames through a single physical MAC; the choice of LMAC specifies the destination set.

The reader will observe, no doubt, that the lower-layer “LMAC 0” is precisely the “Single Packet Broadcast” MAC often discussed. The remaining lower-layer “LMAC x” modules are present for the use of the upper-layer shared LAN emulation module.

Techniques for tagging the upstream and downstream frames are discussed in Section 4.0 on page 11.

2.4 Trivial Upper-Layer Shared LAN Emulation

A trivial implementation for the Upper-Layer Shared LAN Emulation sub-layer is shown in Figure 4. For each lower-layer LMAC, the ULSLE merely reflects any received frame back down to the same lower-layer LMAC, and in addition, passes it up to the Emulated LMAC. No frames are ever received on LMAC 0. All frames passed down from the upper layers are passed to LMAC 0 for transmission to all ONU’s LMACs. Mission accomplished! Of course, this simple emulation is subject to the main objection of shared LAN emulation: bandwidth is wasted if, as is usually the case, most frames from ONUs are not for other ONUs.

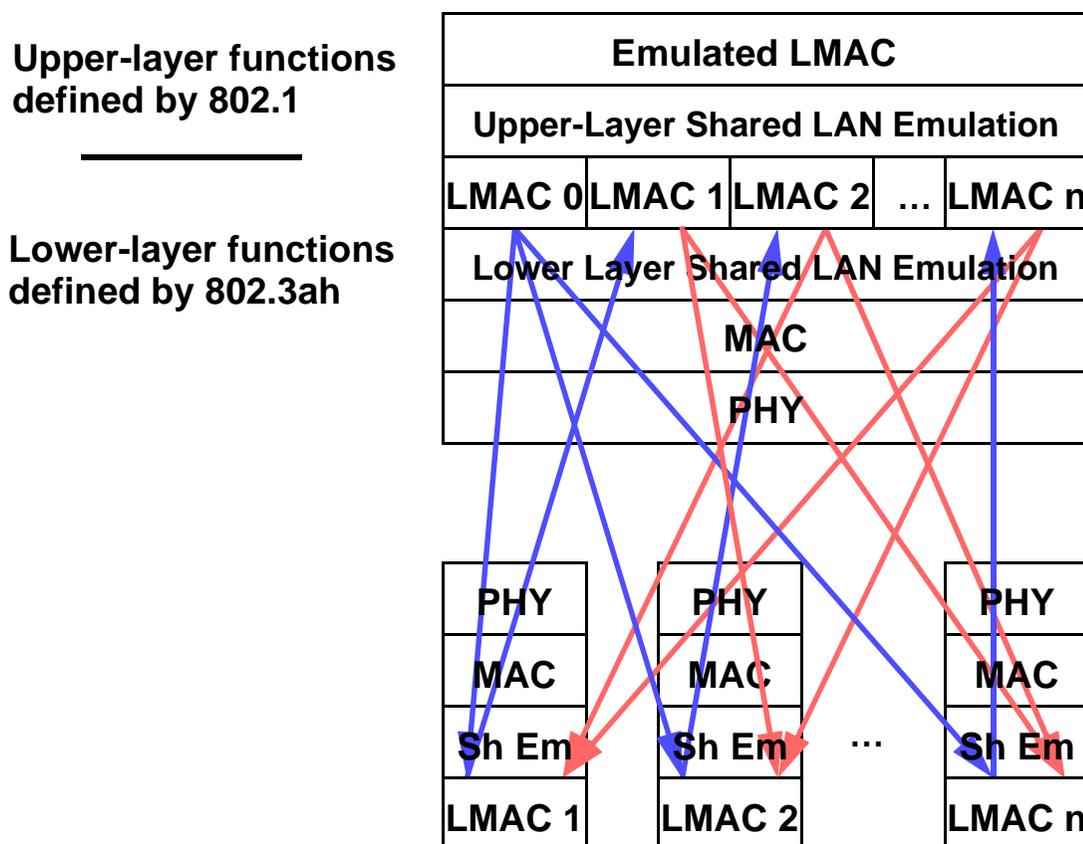


FIGURE 3. MAC model for Shared LAN Emulation

2.5 Advanced Upper-Layer Shared LAN Emulation

We can imagine also IEEE 802.1 defining an upper-layer shared LAN emulation module which functions very much like a bridge. This Advanced Upper-Layer Shared LAN Emulation module uses $2n+1$ lower-layer LMACs to support n ONU LMACs. As shown in Figure 5, on page 7, the OLT requires n p-p LMACs and n “anti-LMACs”, one “all ONUs” LMAC, and the Emulated LMAC, which to the ULSLE, faces in the opposite direction. The operation of this layer can be summarized in a manner that is very reminiscent of the operation of a normal IEEE 802.1D bridge. (The following discussion is expressed in terms of learning only MAC addresses, as does a standard IEEE 802.1D bridge. Of course, an 802.1Q VLAN compatible bridge learns {MAC address, Database ID} pairs, rather than simply learning MAC addresses.)

1. The source MAC address of every frame received on every lower-layer LMAC is learned by ULSLE in its own equivalent of a bridge’s Filtering Database (FDB), along with the LMAC ID of the lower-layer LMAC on which it arrived. (No frames ever arrive on LMAC 3.)
2. The source MAC address of every frame passed down from upper layers through the Emulated LMAC is also learned in the ULSLE’s FDB as coming “down from above”.

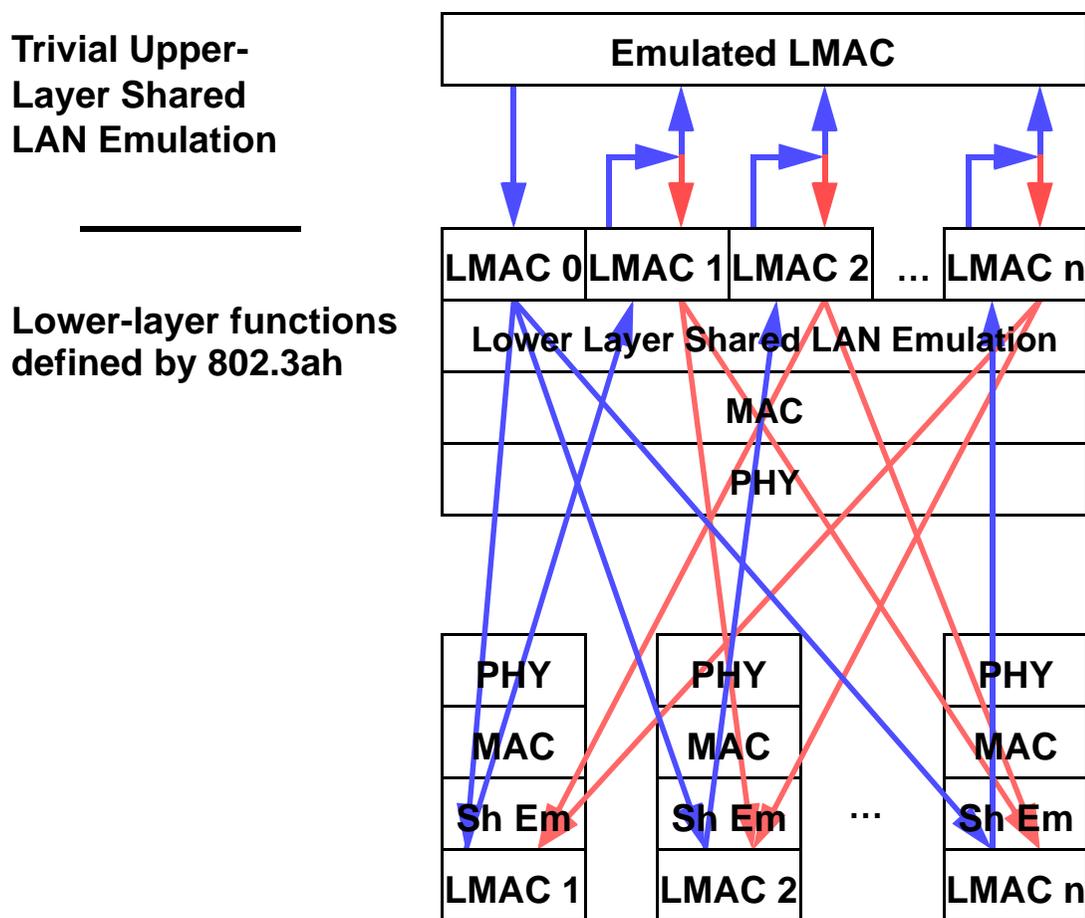


FIGURE 4. Trivial Upper-Layer Shared LAN Emulation

3. Every frame passed down from upper layers through the Emulated LMAC is directed to the appropriate p-p LMAC, if its destination MAC address has been learned on that p-p LMAC, or to the “all ONUs” LMAC, if not.
4. For every frame received on lower-layer p-p LMAC n , the advanced ULSLE looks up its destination MAC address in the ULSLE’s FDB.
5. If the destination is a unicast address learned either from the Emulated LMAC, or from the lower-layer LMAC on which it was received, the frame is passed up the stack and not reflected back down to LMAC n . (The device above the Emulated LMAC is unlikely to pass the frame on, but may need to learn from the frame’s source MAC address.)
6. If the destination is a multicast address which is, by GMRP or some other means, known to not be wanted by any ONU LMAC except, perhaps, the one corresponding to the LMAC on which it was received, then the frame is passed up the stack and not reflected back down to anti-LMAC n .
7. Any other frame is both passed up through the Emulated LMAC and reflected back down to anti-LMAC n . Such frames include those whose destination MAC addresses are:
 - a. unicast addresses unknown to the ULSLE;
 - b. unicast addresses known to reside on some lower-layer LMAC other than the one on which it was received;

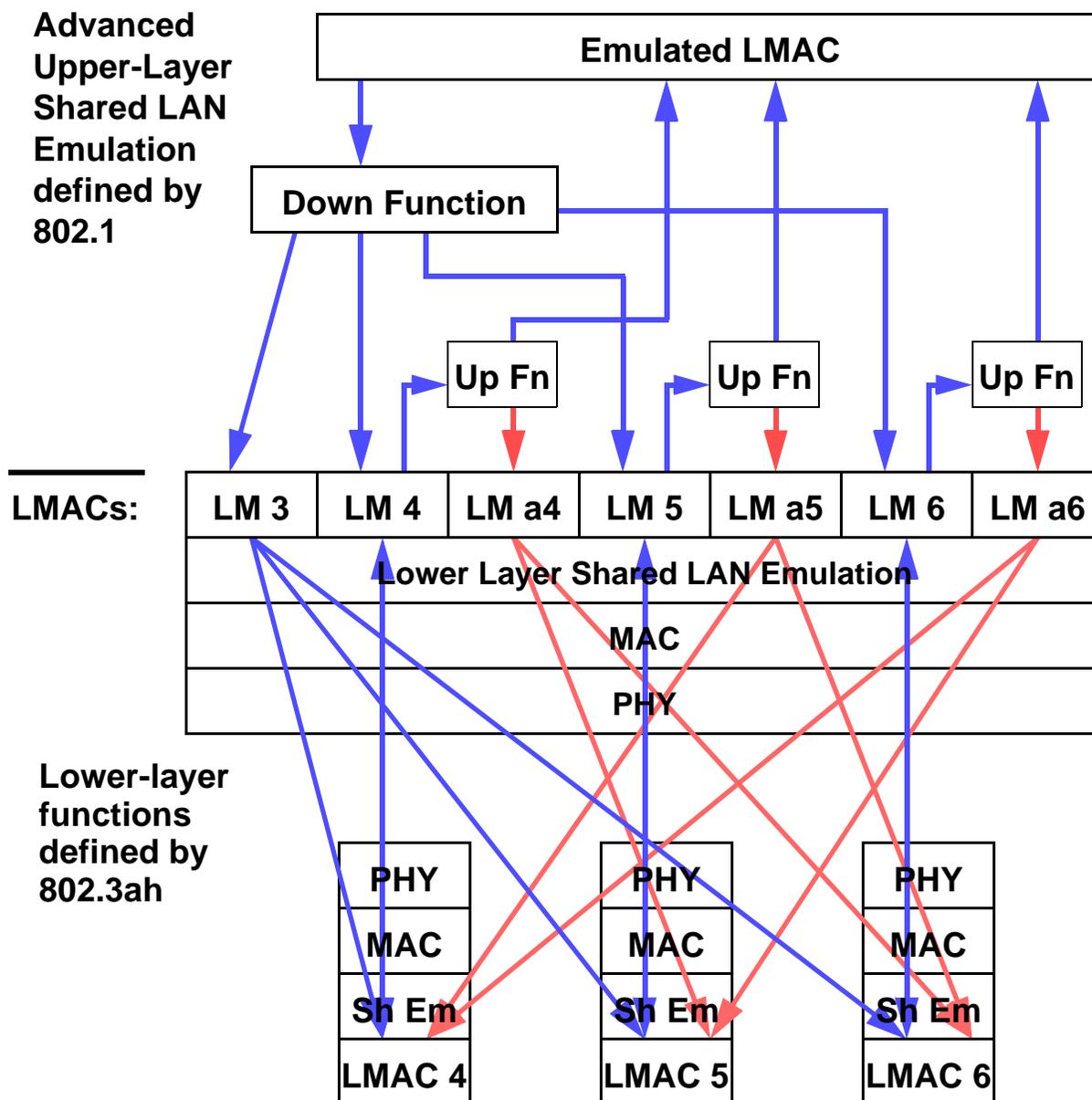


FIGURE 5. Advanced Upper-Layer Shared LAN Emulation

- c. multicast addresses which are not known (e.g. by GMRP) to be *not* wanted by all of the other lower-layer LMACs below the Emulated LMAC; and
 - d. all broadcast frames.
8. Any IEEE 802.1D spanning tree BPDUs which pass through the ULSLE must be inspected to see whether they contain Topology Change Notices. Transmission of a TCN in either direction must be handled appropriately by the ULSLE, which means that the ULSLE must forget, or accelerate the timing out, of all, or certain classes of, MAC addresses. This behavior is required whether or not the device “above” the Emulated LMAC is a bridge.

9. If VLANs are used, then the ULSLE must be configured to make the same associations of MAC addresses, VLANs, and Filtering Databases as do any bridges attached to the EPON. Again, this behavior is required whether or not the device “above” the Emulated LMAC is a bridge.

Thus, the Upper-Layer Shared LAN Emulation behaves like an n -port bridge while deciding whether to forward a frame. Although the ULSLE does not need to generate IEEE 802.1D spanning tree BPDUs, it must be aware of their passage through the ULSLE, so that it can forget MAC addresses appropriately.

If implemented as specified above, a ULSLE can be attached to any device, whether a bridge, a router, or a host computer. Any device attached to the other side of the Emulated LMAC believes itself attached to a standard Ethernet shared LAN. If the ULSLE is attached to a router or host computer, one would assume that the “owner” device’s own MAC addresses would be permanently “learned” by the ULSLE to reside on the Emulated LMAC, and that these addresses would never be “forgotten” by the ULSLE. Similarly, standard MACs provide the means to specify to the Emulated LMAC which multicast MAC addresses the “owner” device wishes to see.

The reader familiar with bridge implementations may recognize that the ULSLE’s requirements are very similar to the extra information learned by a bridge which connects to any of a number of common Ethernet emulation technologies, including ATM LAN emulation and MPLS-VPN Layer 3 tunnels. The information is slightly different than those technologies, and applied in a somewhat different way, because we are not always trying to send frames to exactly the right ONUs.

As described so far, the Advanced Upper-Layer Shared LAN Emulation is not too different from the operations of a normal IEEE 802.1D bridge. The “Down Function” is a bit odd, because a normal bridge would direct the downward frame to some number of p-p LMACs, rather than having to choose between one p-p LMAC and the “all ONUs” LMAC. Another optimization which could be performed, but is not shown in Figure 5, would be to perform the same choice in the “Up Function”. Here, of course, the choice would be between transmitting the frame on one p-p LMAC other than the one the frame was received on, or to the anti-LMAC it was received on.

This implementation of an Advanced Upper-Layer Shared LAN Emulation module removes the single greatest disadvantage of the trivial shared LAN emulation of Section 2.4 on page 4; most upstream traffic is not reflected back down the stack.

Of course, “one skilled in the art” can think of other things that will improve the intelligence of the Advanced Upper-Layer Shared LAN Emulation of Section 2.5 even further. For example, propagating any static VLAN permit/deny configuration to the lower-layer LMACs would assist greatly in keeping down unnecessary reflections, especially of broadcasts, multicasts, and unicast floods. The GVRP mechanism could be used to help contain flooding for the enterprise; the service provider would probably want to use static configuration.

2.6 Special Control Functions

If the device above the Emulated LMAC is a router or an end-station, the ULSLE as so-far described is sufficient. If the Emulated LMAC is serving as a Bridge Port of an IEEE 802.1D bridge, however, certain additional control functions are required. IEEE 802.1D must be modified in order to make use of these functions at the appropriate times.

1. **Disable/Enable Learning.** Whenever the Bridge Port enters the Blocked state, the bridge needs to signal the ULSLE to disable the learning of received MAC addresses, and to forget all those it has learned. Similarly, the bridge needs to enable MAC address learning at or before the time the Bridge Port enters the Forwarding state.
2. **Forget Learned MAC Addresses.** In certain situations, for example when an STP Topology Change Notification is received on another Bridge Port, the bridge may need to signal through the Emulated LMAC that the ULSLE is to forget its learned MAC address information. Depending on the Spanning Tree algorithm used, this may involve accelerating the timeout period used by the ULSLE for all MAC addresses, or may involve immediately deleting learned information.
3. **Parameter Value Controls.** Parameters such as the background timeout period for forgetting learned MAC addresses may vary dynamically with the particular Spanning Tree algorithm used by the bridge. It would be most convenient for the bridge to be able to control parameters in the ULSLE which correspond to similar parameters in an IEEE 802.1D bridge.

Additional control functions may be required, as well.

3.0 Shared LAN Usage by Service Providers

We have not, so far, addressed the other concern of the service provider community: keeping traffic for different customers separate. In one sense, this request is asking for antagonistic goals: we want to keep traffic for different customers separate, but we want to send exactly the same packet to many customers.

Fortunately, we have, already, a number of tools that can help us. The two most obvious are to use 802.1Q VLANs, and to run both point-to-point and shared LAN emulations over the same EPON.

3.1 Separating Customers Using VLANs

IEEE 802.1Q provides for tagging frames with VLANs. By configuring the ONUs appropriately, it is fairly easy to use VLANs to separate the customers, while using the Shared LAN Emulation mode to control the bandwidth utilization. A number of techniques are in common use, today, to accomplish just this feat using standard IEEE 802.1Q bridges. Since VLANs can be configured to have separate MAC address data bases, it can be made impossible for users of one VLAN to interfere with users of another VLAN. As mentioned, above, the VLAN techniques can also improve bandwidth utilization in the shared LAN emulation, e.g. by not reflecting frames arriving on the only lower-layer LMAC that supports the frame's VLAN-ID.

3.2 Running Both Point-to-Point and Shared Emulations

If service providers are required by law to share an EPON, then using VLANs for customer separation is not satisfactory. There are no standard VLAN-based techniques that would allow providers to separate their traffic, and their customers' traffic, without trusting each other, and perhaps their customers, to handle the untagged Layer 2 control frames fairly and correctly. Fortunately, there is another possibility that would provide complete provider and customer isolation, efficient

bandwidth allocation, and some degree of downstream broadcast utilization, while still remaining entirely within the standards.

Suppose we take two of the three point-to-point emulated LANs in Figure 2, add two examples of the shared emulated LANs in Figure 3, one three-node and one two-node, and combine one of the point-to-point and one of the shared ONUs into a single ONU with two Logical MACs. Then one gets something like the system shown in Figure 6.

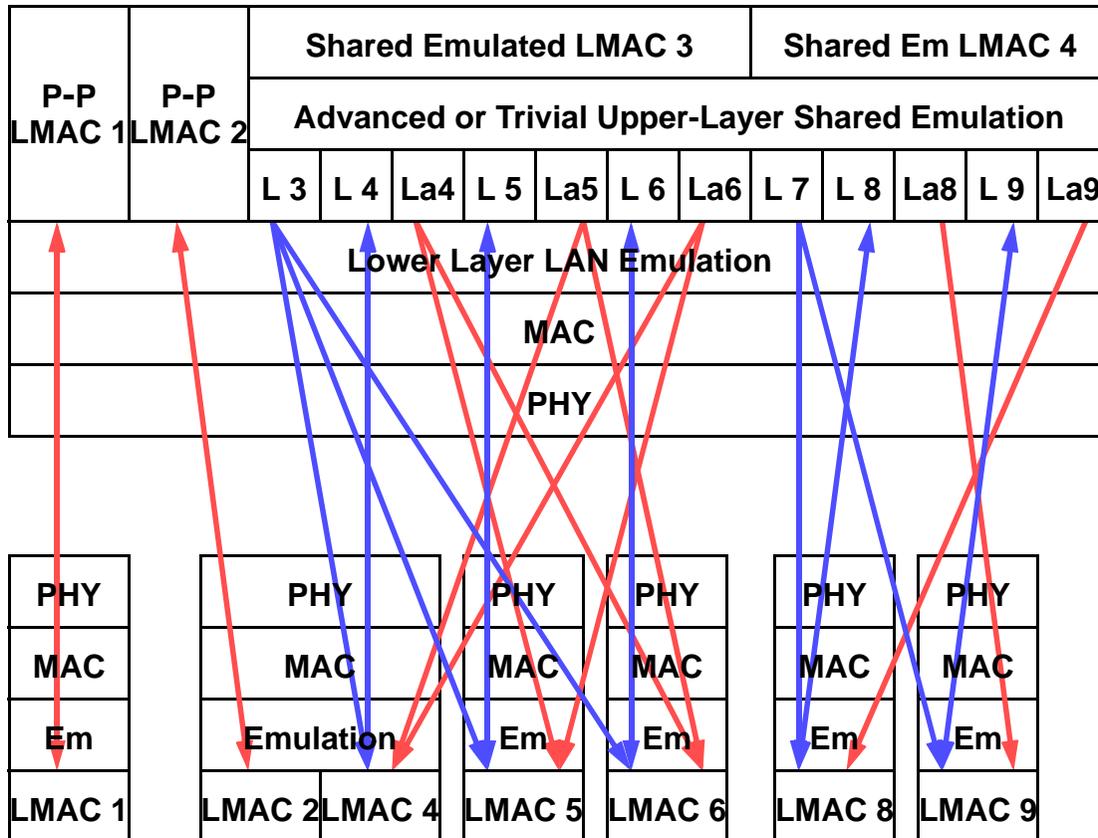


FIGURE 6. Multiple Combined emulations

Please notice that this combined scheme does *not* provide for an over-arching broadcast facility. It is assumed that the need to separate providers makes this impossible. This combination is for strict customer separation. To achieve customer separation plus broadcast reachability, one uses the VLAN-based techniques of Section 3.1 on page 9, perhaps on top of an ONU-based provider separation plan.

To make this combined scheme a possibility, some additional means of marking and recognizing data frames is required, as outlined in Section 4.2 and Section 4.3.

In addition, it is important to understand that no one ONU LMAC can participate in both a point-to-point and a shared LAN; it must be connected to one or the other. The reason is best described in terms of ONU-OLT frames. If they are sent on the point-to-point LAN, then that ONU's shared LAN connection is unidirectional. If they are sent on the shared LAN, then the point-to-point LAN is unidirectional. Either case is clearly incompatible with existing standards.

4.0 Frame Tagging Techniques

There are several means by which OLT-ONU and ONU-OLT frames can be labeled, including methods that support both point-to-point and shared LAN emulations can be performed.

4.1 LMAC ID Plus Broadcast Bit

Table 1 illustrates a scheme in which each frame transmitted by either an ONU or the OLT is labeled with one LMAC ID and a Broadcast Bit. The ONU never sets the Broadcast Bit; the OLT anti-LMAC sets the Broadcast Bit before reflecting a frame. (An ONU could set the Broadcast Bit if it is attached to an emulated shared LAN, but it is perhaps better to make the ONU a constant and have the OLT set the Broadcast Bit.) The OLT's "all ONUs" LMAC and the anti-LMACs set the Broadcast Bit; the OLT p-p LMACs clear it. This scheme allows the EPON to attach as many ONUs as can be encoded in the LMAC ID field in point-to-point LAN emulations, and the same number in one single shared LAN emulation. It does not allow multiple shared LAN emulations to take place on the same EPON.

TABLE 1. LMAC ID Plus Broadcast Bit Frame Reception

Frame Tag		Receiver Action	
Broadcast Bit	LMAC ID	OLT to ONU	ONU to OLT
0	match	Accept	Accept and maybe reflect
1	match	Discard	Accept and maybe reflect
0	no match	Discard	Discard
1	no match	Accept	Discard

4.2 LMAC ID Plus Group ID

A straightforward extension of the plan in Section 4.1 would accommodate multiple providers, strictly separated from each other. That is, multiple shared LANs can be emulated. In this plan, the "Broadcast Bit" is expanded to a "Group ID", as shown in Table 2 and Table 3. The Group ID is used to strictly separate various emulated LANs, whether point-to-point or shared. One encoding of the Group ID is used for the "Unicast" value.

TABLE 2. LMAC ID Plus Group ID Transmission

LMAC Type	Frame Tag Sent	
	Group ID	LMAC ID
OLT p-p LMAC	"Unicast"	LMAC ID
OLT anti-LMAC	Group ID	LMAC ID
ONU LMAC	Group ID	LMAC ID

These diagrams show that the ONU behavior and the OLT LMAC behavior do not change, whether they are participating in a point-to-point or in a shared LAN emulation. Only the assignment of LMAC IDs to OLT LMACs and the presence and the behavior of the Upper-Layer Shared

TABLE 3. LMAC ID Plus Group ID Reception

Frame Tag		Receiver Action	
Group ID	LMAC ID	OLT to ONU	ONU to OLT p-p LMAC
“Unicast”	match	Accept	Accept and maybe reflect ^a
“Unicast”	no match	Discard	Discard ^a
no match not “Unicast”	any	Discard	Discard
match	match	Discard	Accept and maybe reflect
match	no match	Accept	Discard

a. This should never happen. We may wish to make this a Discard.

LAN Emulation determines whether a given OLT LMAC is part of a point-to-point LAN or a shared LAN.

4.3 LMAC ID Only

It is also possible, as shown in Table 4, to make a trade, reducing the number of ID bits carried in each frame for additional configuration in each ONU. As for the “Group ID” scheme of Section 4.2, every ONU works the same way, whether participating in a point-to-point or a shared LAN, as does every OLT LMAC. In this scheme, each ONU or OLT LMAC possesses a configured LMAC ID. Each ONU LMAC possesses a configured recognition bit table. This table must be large enough to span all possible LMAC IDs, so that a 10-bit LMAC ID space in the frames requires a 1024-bit recognition bit table in each ONU LMAC. Then:

1. Both OLT-ONU frames and ONU-OLT frames are tagged with the sender’s LMAC ID.
2. The ONU has the same LMAC ID as the corresponding OLT p-p LMAC.
3. Any frame received by an ONU LMAC whose LMAC ID is programmed into the recognition bit table is accepted; others are not.
4. Any frame received by an OLT p-p LMAC whose LMAC ID matches the OLT LMAC’s is accepted; others are not.
5. OLT “all ONUs” LMACs and anti-LMACs do not receive frames.

The emulation layers for the ONU LMACs are programmed to recognize LMAC IDs as illustrated in Table 4, which gives the programming for Figure 6, on page 10.

The number of LMAC IDs required to support p ONU LMACs spread over q shared LANs, plus m ONU LMACs in point-to-point LANs, is equal to $(p+m$ OLT point-to-point LMACs) + $(p$ OLT anti-LMACs) + $(q$ “all ONUs” LMACs) = $2p+m+q$. In the worst case, which is to divide up all of the n ONU LMACs into “shared” LANs with one ONU LMAC each, it takes $3n$ LMAC IDs. Thus, an 8-bit LMAC ID tag space will support 256 point-to-point LANs, one 128-node (including the Emulated LMAC) shared LAN, 102 shared LANs each consisting of two ONU LMACs and one OLT Emulated LMAC, or 85 “shared” LANs each with one ONU LMAC and one OLT Emulated LMAC.

TABLE 4. Programming ONUs' LMAC ID Recognition Tables in Figure 6

ONU LMAC OLT LMAC ID	1	2	4	5	6	8	9
1	Y	-	-	-	-	-	-
2	-	Y	-	-	-	-	-
3	-	-	Y	Y	Y	-	-
4	-	-	Y	-	-	-	-
a4	-	-	-	Y	Y	-	-
5	-	-	-	Y	-	-	-
a5	-	-	Y	-	Y	-	-
6	-	-	-	-	Y	-	-
a6	-	-	Y	Y	-	-	-
7	-	-	-	-	-	Y	Y
8	-	-	-	-	-	Y	-
a8	-	-	-	-	-	-	Y
9	-	-	-	-	-	-	Y
a9	-	-	-	-	-	Y	-

Please note that this bit-vector scheme has nothing to do with the scheme described in Section 5.0. The “recognition bit table” is a bit vector that is present in the ONU, and configured once to set up a shared LAN comprising some number of ONUs. The bit vector discussed in Section 5.0 is carried in every frame.

5.0 A Completely Separate Alternative

Consider Figure 2, on page 4, with a very different Point-to-Point Emulation module. Let us assume that there is *no* facility except an array of point-to-point LMACs available to the upper layers above any OLT or ONU. That is, all connections are point-to-point. With this type of emulation, all standard higher-layer protocols would be happy.

But, how do we get efficient downstream broadcast? Here is one way:

1. Every frame transmitted in the OLT-ONU direction carries a bit vector, with one bit representing each ONU LMAC.
2. Each frame passed down through OLT LMAC x gets the bit corresponding to ONU LMAC x set to 1, and all others set to 0.
3. The Point-to-Point Emulation layer in the OLT examines all frames passed down through the various OLT LMACs; and compares them to each other. If any set of frames are identical, then instead of transmitting multiple copies, the emulation layer transmits one copy, with each LMAC's bit set, all the way to a frame passed through all LMACs with all bit vectors set.

Of course, it is doubtful that many actual implementations would work the way described in point 3. A bridge, for example, would typically find it relatively easy to generate a vector similar to the actual frame bit vector when transmitting a frame to multiple point-to-point LMACs.

This type of emulation has a number of positive and negative characteristics, compared to a system based on the Upper-Layer Shared LAN Emulation module of Section 2.0 on page 2, and perhaps should be explored, further. Some of these characteristics are favorable to the ULSLE:

1. A bit vector in every frame is difficult to scale up to a large number of ONU LMACs; the ULSLE can clearly be scaled as far as bridging.
2. Certain functions and/or implementations above the OLT LMAC(s), for example, routed IP multicast, find a single shared LAN for a large number of ONU LMACs much easier to deal with than an array of the same number of point-to-point LANs.

On the other hand, some are favorable to frame bit-vector tagging:

3. The latest versions of IEEE 802.1D bridging converge much more quickly over point-to-point LANs than over shared LANs.
4. Although a clearer statement of the requirements and behavior of the point-to-point emulation layer using frame bit vectors is needed, this technique would appear to be simpler than the ULSLE. In particular, the frame bit vector method does not require learning MAC addresses, nor does it interact in any way with the Spanning Tree functions of a bridge.

6.0 Summary

Do these scenarios answer all questions that a service provider might have? Of course not! We in IEEE 802.3ah are not here to define residential broadband service at all higher layers! We are here (or at least the author is here) to define a medium that will 1) give *both* enterprise and residential broadband providers the medium they need, and 2) still be an Ethernet. The first is important because there are so many potential users of the new medium. The second is important, because to define a fundamentally new MAC service would be a tremendous disservice to the public. Furthermore, meeting the second goal is the best guarantee of meeting the first.

The biggest reason for the demand for Ethernet in the First Mile is that, historically, any upper-layer protocol based on Ethernet has been able to utilize, without change, every new development from IEEE 802.3. Few other media can approach this track record, which is now longer than the careers of most IEEE 802 attendees. Such compatibility must be maintained if IEEE 802 is to maintain its enormous credibility.

In the long term, the user community in general, and other standards organizations such as IETF in particular, may well wish to take advantage of the “native” mode of an EPON, utilizing the point-to-point LMACs and the “all ONUs” LMAC of Section 2.5 on page 5 directly, without an intervening emulation function. This does not, of course, relieve IEEE 802 of the need to provide a compatibility with other 802.3 media.

I do not claim to have described the only reasonable plan(s) for IEEE 802.1 Higher-Layer LAN Protocols and IEEE 802.3ah EFM Point-to-Multipoint. I do claim to have shown there exist at least two means of meeting both the goal of efficiency and the goal of complete standards compliance, and claim that anything less should be unacceptable to IEEE 802 as a whole.