Proceedings of 28^{th} *Annual IEEE International Conference on Local Computer Networks, Bonn/Königswinter, Germany, October 2003*

Thinking Outside the Box: Extending 802.1x Authentication to Remote "Splitter" Ports by Combining Physical and Data Link Layer Techniques

Arun Saha and Mart Molle
Department of Computer Science and Engineering
University of California, Riverside CA 92521
{saha, mart}@cs.ucr.edu

Abstract

We present a novel switched full-duplex LAN architecture which can greatly simplify the cabling requirements in areas that must support high port densities and/or are subject to frequent changes. Instead of providing a separate cable to connect each host to a dedicated port on a monolithic switch behind the wall, we emulate the shared bus topology from the early days of Ethernet by daisy-chaining a series of small network-powered "slave" bridge modules called Ethernet Splitters from a single port on the "master" switch. Our partitioned switch architecture enforces network privacy throughout the entire splitter chain, so no host can view any traffic belonging to another host. The splitters also authenticate the point of origin for every frame, independent of the value contained in its source address field thus providing the same level of security as a monolithic switch under the 802.1x Port Based Access Control protocol.

1 Why Switched LANs?

In recent years, Ethernet-based Local Area Networks have been transformed. The old *shared half-duplex network* paradigm — in which multiple hosts must take turns transmitting frames over a common medium known as a "collision domain", according to the well-known CSMA/CD medium access control protocol — has been replaced by a new *full-duplex switched network* paradigm — in which each host is connected to a separate port on an IEEE 802.1d Transparent Bridge (commonly referred to as a LAN "switch") via a dedicated, collision free, full-duplex link segment (see Fig 1).

Switching was originally conceived as a means for substantially increasing the overall capacity of a network, using *filtering* to avoid transmitting frames to those network segments known not to contain the destination address. We will not consider this performance advantage any further in this paper. Instead, we will focus our attention on privacy and authentications issues, and how this migration to full-duplex switched networks has enabled dramatic improvements in these areas compared to earlier half-duplex shared Ethernet systems.

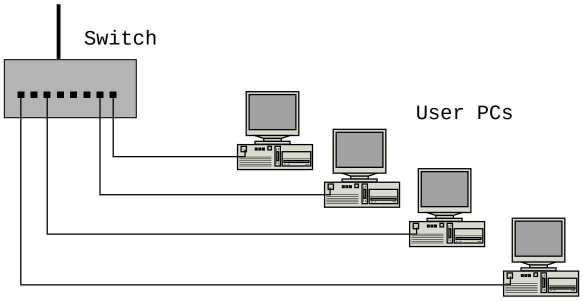


Figure 1: **Four hosts having dedicated access to four ports of a switch**

1.1 Shared LANs lack privacy

In shared half-duplex networks, each *receiver* is free to examine all frames transmitted over the shared network — independent of the frame's source and destination address, and without any of the other hosts being able to detect this breach of privacy — simply by setting its network interface to promiscuous receive mode.

Conversely, switched full-duplex networks provide *disjoint paths* from each host to a dedicated switch port, and force all host-to-host communications to pass through the switch. Thus, as soon as the switch learns the addresses and port assignments for all active hosts, its standard traffic filtering algorithm will render promiscuous receive mode completely ineffective.[1]

1.1.1 IEEE 802.1q Virtual Bridged LANs

Switched full-duplex networks can also be configured to provide a much stronger level of isolation between different groups of hosts through the use of Virtual Bridged LANs

[1] A comparable privacy feature was defined for half-duplex repeaters in [1]. In this case, the repeater learned a single destination address per port. For each incoming frame, the port logic looks for an exact match between the destination address of this incoming frame and the learned address for this port. If the two addresses match, the repeater sends the frame through this port. Otherwise, it substitutes an equal number of "garbage" bits to ensure that the attached host will sense carrier at the proper times.

(VLANs) [2]. VLANs provide a mechanism for partitioning the physical network resources into multiple, disjoint logical broadcast domains. Traffic cannot cross from one VLAN to another except through a router, which can enforce an arbitrary set of policies covering access rights, security and performance issues.

VLANs are created by specifying the criteria for membership, such as: (i) a set of switch ports together with the associated links and hosts that are directly connected to those ports; or (ii) all frames that carry a specific VLAN tag value within the optional VLAN ID field. Note that a *VLAN trunk* is a single link that carries frames belonging to multiple VLANs — all of which must carry the appropriate VLAN tag value.

1.2 Shared LANs hide the sender's identity

There is no way for a host to determine the origin of any incoming frame except by reading its source address field. Since the source address is inserted into each outgoing frame by software executing on the sending host, a malicious source could easily hide its identity by placing a (sequence of) different value(s) into the source address field of its outgoing frames.

Unfortunately, this authentication issue does not go away when we migrate to a switched full-duplex network. Once a malicious sender transmits a bogus frame to the switch, it is accepted unconditionally and relayed to the destination based on its destination address, without any regard for the accuracy of its source address — leaving the receiver with no way to determine its point of origin.

Indeed, since switches use the source address field from every frame to update their filtering database (which holds the list of known MAC addresses and their current port assignments) — and being able to update that filtering database at wire speed without any impact on performance is viewed as a competitive feature among switch vendors — a malicious attacker can use this feature to hijack traffic that is addressed to another host. Each time the attacker transmits a frame that includes the victim's MAC address in the source address field, the switch will update its filtering database to send all of the victim's traffic *to the attacker only* until it sees another transmission by the victim. Thus, an attacker who uses this technique sparingly can obtain a sampling of the victim's traffic with little risk of detection; a more aggressive use would generate a *denial of service* attack against the victim.[2] As a result, the IEEE 802.1d transparent bridging standard was recently extended to include a new port-based authentication method.

1.2.1 IEEE 802.1x Port Based Access Control

The IEEE 802.1x Port Based Network Access Control Standard [3] defines a framework by which a "client device", known as the *supplicant*, is authenticated by its "first point of attachment", i.e., the switch at the edge of the network known as the *authenticator*. This framework contributes to layer-2 security, supplementary to the security of upper layers. 802.1x uses the Extensible Authentication Protocol (EAP) [4]. EAP can support multiple authentication methods and can work over Ethernet or wireless links.

Initially the supplicant's port is blocked except for relaying the EAP frames (i.e., authentication messages) between the supplicant and authentication server. Once authenticated, the controlled port is opened and all kind of frames are allowed.[3] This authentication process must be repeated each time the Ethernet physical layer transceiver reestablishes the link after a loss of carrier, even if the link is reserved for a single host (e.g., a staff person's desktop PC, or a networked printer). This re-authentication requirement is intended to prevent a network security breach if the host operating system is compromised, or if the network cable is moved from the usual host to an intruder's laptop computer.

The need for re-authentication of the client is even greater if the port serves a shared-use facility, such as an instructional laboratory on a university, a public Internet access point in a library, etc. In this case, users with different access privileges may use the same host at different points in time, and the authentication mechanism must be general enough to allow the administrator to assign a different set of access rights to each user upon successful authentication.

It is important to note that 802.1x authentication represents a "Maginot Line" view of network security. Every client node is assumed to have its own dedicated full-duplex connection to a separate port on a trusted edge switch (authenticator). Thus, since each client must first satisfy an authenticator before it gains access to the network, only frames sent by authenticated sources can enter the network. Unfortunately, this authentication can easily be defeated by inserting a man-in-the-middle attacker (as described below) or an alien switch into the link that connects the supplicant to the "trusted" switch port.

2 Why Switch Partitioning?

2.1 Motivation: emulating "shared" cabling

Consider a large open-plan office, a computer lab in a university, or a call/data center staffed by operators sitting in front of computer screens. In all of these cases, we must accommodate large numbers of networked computers within a single room, and only a small fraction of them can be placed next to a wall where it would be most convenient to provide a data jack.

In the old days of shared half-duplex networks (e.g., 10BASE-2 "thinnet"), the network access in such a room would have been handled by daisy chaining multiple computers along the same shared coaxial cable segment, to minimize wiring clutter. However, the cabling requirements for switched full-duplex networking in such an environment seem quite clumsy in comparison. To prevent physical tampering, we expect the switch to be housed inside a locked

[2] To prevent such attacks, some switches have an option for "locking down" the MAC address assigned to a port. However, these countermeasures can easily be defeated if the network contains multiple switches.

[3] In some implementations, the client may be granted a user-specific set of network services, or perhaps assigned to a different VLAN.

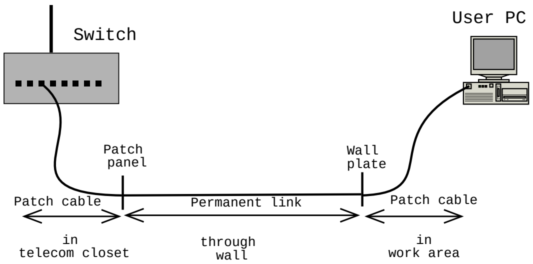


Figure 2: **Detailed view of a single link segment.**

telecom closet. In order to use the 802.1x port based authentication protocol described above, we must provide a dedicated full-duplex connection from each host to a separate switch port. Current horizontal cabling standards for commercial buildings [5] permit only *two intermediate connection points* in each host-to-switch link, one at the patch panel located in the telecom closet and the other at the data jack located in work area (see Fig 2). Thus, each host-to-switch link consists of: (i) one patch cable from the switch to the patch panel in the telecom closet; (ii) one permanent link connecting the patch panel to a data jack in the work area; and (iii) one patch cable from the data jack to the host in the work area. If we later decide to reconfigure furniture in the room, we must rearrange and/or replace all the patch cables in the work area, and possibly install some new data jacks and permanent wiring if additional network connectivity is required in some parts of the room. Clearly, we must pay a very high price in terms of higher cabling costs and reduced flexibility to enjoy the increased performance and security of switched full-duplex networking in this type of high density environment!

We are thus motivated to find a means to combine the *convenience* of shared cabling with the superior *performance* and *security* of switched full-duplex operation.

2.2 Feasibility by combining facts

Some recent technological advancements, stated below, motivates us to propose partitioned switch architecture.

2.2.1 Highly scalable data rates

Ethernet supports 10 Mbps, 100 Mbps and 1000 Mbps operation over the same horizontal twisted pair cabling. Thus, we can assign a higher data rate to the shared “backbone” links (1000 Mbps, say) than to the individual “access ports” for each host (which are limited to 100 Mbps, say) to prevent the backbone from becoming a performance bottleneck. This speed disparity also reduces the buffering requirements at each access port.

2.2.2 VLAN tags enable simple Ethernet multiplexors

Assume that every access port is assigned a unique VLAN ID, and that all frames traveling up or down the chain’s backbone links must carry the appropriate VLAN ID within their tag field. In that case, every access port is connected to the master switch through a dedicated virtual link. Hence all incoming frames from a given host are tagged with the VLAN ID of its access port and sent directly to the master switch. Similarly, all outgoing frames addressed to the given host are tagged with the appropriate VLAN ID by the master switch and sent directly to the corresponding access port. This strategy (see [6]) allows us to centralize the implementations of complex policy decisions within the master switch module while at the same time reducing the forwarding decisions at each access port to a simple VLAN tag lookup.

2.2.3 Switch-on-a-Chip Design Possible

To reduce costs and prevent tampering, our goal is to keep the design of the remote access module simple enough to permit a single-chip implementation. Several vendors already offer a single-chip implementation of a complete 10/100 switch including all physical layer transceivers. Although current 1000BASE-T transceivers occupy an entire chip, some quad transceiver chips for 1000BASE-X (Gigabit Ethernet over fiber) are already available, so it should not be long before we see single-chip switches that include a few gigabit ports.

2.2.4 Powered Ethernet

The IEEE 802.3af standard [7] defines a method for distributing DC power from the telecom closet to remote equipment through the horizontal twisted pair cabling system. If the single-chip access port module runs on DTE power, it would be as easy to install as a passive telephone line coupler.

2.2.5 Configuration via Auto Negotiation

Clause 28 of IEEE Std. 802.3-2002 [8] defines an Auto Negotiation protocol for establishing the operating parameters for Ethernet transceivers operating over twisted pair segments. Upon the initial establishment of a physical link between the two Ethernet transceivers, and thereafter each time one of those transceivers is powered up, reset or a renegotiation request is made, the transceivers exchange a series of fast link pulses which encode the set of options supported by each transceiver. These information are exchanged in predefined Link Code Word format. Thus a single device can communicate with different devices at different link speeds.

2.2.6 DSP-based Transceivers

Currently, many transceiver designs for 100 Mbps and 1000 Mbps operation over twisted pair cabling are based on sophisticated digital signal processing algorithms. Such DSP transceivers collects a wealth of data about the electrical properties of the physical link. Some of this information can be used to improve the security of our authentication procedure by allowing us to estimate the round-trip propagation delay over the link.

Network Jack [9] is an unmanaged ”in-the-wall” switch which enables four network devices to be connected to a single Ethernet connector. It is not VLAN capable. Probably, more than one of them cannot be combined to facilitate a group of users.

3 Partitioned Switch Architecture

In the remainder of this paper, we introduce a “partitioned switch” architecture. More specifically, we replace the monolithic switch by combination of a “master” switch module, which remains safely locked inside the telecom closet, and a collection of small “slave” modules called *Ethernet Splitters*. Each splitter consists of a single-chip implementation of a complete VLAN-capable bridge module powered by the Ethernet cable, together with three or more external ports.

Two of the splitter ports are called *backbone ports*, labelled N and S, which are used to link a *string* of splitters to a single port on the master switch module in a linear daisy-chain topology that emulates the old-style shared cabling topology. We assume that port N points towards, and port S points away from, the master switch module, but obviously these roles need not be “cast in silicon” and may be established through the auto-negotiation process during link startup. The remaining splitter ports are *access ports*, which are used to connect individual hosts to the network. A splitter need not have any hosts attached to it. There can be an open ended cable attached to the last splitter in the chain. Figure 3 illustrates a partitioned switch configuration in which a string of four splitters serve four hosts.

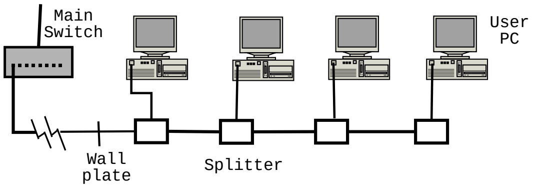


Figure 3: **Main switch, four splitters and four hosts**

From the user perspective, splitters act like the familiar passive couplers we use to share a single AC power outlet between two appliances or a single telephone jack between a FAX and an answering machine. However, each slave module is actually an active electronic device, i.e., a complete Ethernet bridge incorporating a few special features, which allows the master switch module to maintain the same degree of control over all switch ports residing in the remote splitters as if they were part of a monolithic switch.

3.1 Reduced Cabling Costs

Using this partitioned-switch approach, we can greatly decrease the wiring clutter in a building full of computers by laying out strings of splitters throughout the rooms, thus ensuring that each host is adjacent to its network access port. In addition, we also reduce the equipment costs considerably. For example, consider the effect of grouping m hosts located in the same work area to form a single splitter string instead of the standard cabling approach shown in Fig. 2. In this case: (i) the component count for items *inside the telecom closet* (i.e., switch ports, patch cables, and connectors in the patch panel) are each reduced from m to 1; (ii) the component count for (permanent) items *inside the walls* (i.e., permanent links and wall plate connectors in the work area) are each reduced from m to 1; and (iii) the component count for (moveable) items *inside the work area* are increased from m to $2m$ (for patch cables) and from 0 to m (for splitters), assuming a worst-case topology where each splitter supports only one host. We can offset the extra items of (iii) against the saved items from (i), since the two types of patch cables are equivalent and we expect that the combined cost of a switch port and a patch panel connector will be at least as high as the cost of one splitter. This leaves us the $(m-1)$ permanent items at (ii) as net savings for the partitioned switch approach. Thus, given the high cost of labor, together with the fact that the maximum length for the permanent link is 90 meters (compared to 5 meters each for the two patch cables), we expect the partitioned-switch approach to yield a considerable cost savings in high density work areas.

3.2 Linear Topology

We assume that multiple splitters will only be linked together into a linear *string* topology through the two backbone ports. This restriction greatly simplifies the topology-related issues that must be handled locally by the splitter logic. All inbound frames should be sent to the master switch, and all outbound frames are either addressed to one of the splitter’s own access ports or simply relayed to the next splitter in the string. In addition, we don’t need to run the Spanning Tree protocol to detect and eliminate cycles, since we can define the master switch as the “root” bridge for each splitter string, and any confusion over which of two adjacent splitters is closer to the root is trivial to solve based on the direction of DC power distribution. The only possible topology “mistake” is to have both ends of the same splitter string connected to master switch ports. However, this problem is easily detected during link startup by the two adjacent splitters when the last patch cable is connected. Moreover, such redundant connections do not cause any harm during normal operation (since they would be made inactive during link startup), while at the same time allowing the string to survive a single link or splitter failure.

3.3 Single Management Interface

The network administrator must be able to control all features provided by the entire partitioned-switch system from the management interface on the master switch. Thus, we centralize the implementation of complex features (access control policies, user authentication, etc.) in the master switch module and try to limit the splitters to act as dumb multiplexors that collect and distribute traffic between the master switch and its collection of remote access ports. Any managed object within a splitter can be remotely read or written from the master switch by exchanging control frames with target splitter's control interface.

3.4 Equivalent Security to a Monolithic Switch

There is *no difference* between the level of security provided through the 802.1x authentication process if a supplicant node accesses the network through a dedicated link segment terminating at a free port on the master switch, or through dedicated link segment terminating at a free access port on one of the splitters. Either way, the supplicant's network connection remains blocked (except for exchanging EAP frames with the master switch) at the "first point of attachment" until it successfully authenticates itself to the master switch using 802.1x. Thereafter, all traffic to or from this host is subject to the same policy controls as it passes through the master switch port, before it can reach any other host.

3.5 Splitters are Trustworthy

The integrity of all privacy and security policies applied to the partitioned switch system depend critically on the assumption that we can trust the splitters to (i) maintain the separation between traffic tagged with different VLAN IDs, and (ii) prevent an intruder from gaining undetected access to the backbone link connecting two adjacent splitters — which would allow it to read and/or tamper with traffic belonging to other hosts.

3.6 Backbone Traffic not Encrypted

After successful authentication, we assume that all host traffic is sent between the splitter access ports and the master switch in plaintext. An alternative would be to encrypt all traffic being sent over the backbone, using a unique key for each access port. We rejected this approach because encrypting all data is computationally expensive, which increases power consumption and cost of the splitter. It also increases the response time, which is undesirable due to reasons mentioned below. Moreover, since a malicious intruder anywhere along the path to the master switch could masquerade as trusted third-party authenticator server, it is not clear whether encryption over the links would really help.

4 Splitter Authentication

The solution to splitter authentication problem is our main result, which is the focus of the remainder of this paper. More specifically we now present a sequential splitter-authentication procedure that grows a string of "trusted" splitters, starting from the master switch port, by adding one new splitter at a time to the end of the string. The novelty in our approach is to incorporate specific information about the *physical layer properties of the link*, which are obtained from the DSP transceiver, into our packet-level challenge-response authentication protocol. Our approach allows each peer node at the boundary of the "trusted" string to determine that its partner in the challenge-response dialog is indeed another "trusted" splitter, rather than some intruder masquerading as a trusted node. More importantly, the peer nodes are also able to guard against a "man-in-the-middle" attack by verifying that timing of the responses matches the measured delay properties of the link.

4.1 Alien versus Bonafide Splitters

For cost and interoperability reasons, we assume that all splitters are built from standard commercially available components. This means that anyone, including an adversary trying to break into the network, can purchase a splitter. Hence, we need a mechanism by which the system can identify that a particular splitter is a *bonafide* member of the network, as opposed to an *alien* device brought in by someone seeking unauthorized access.

In our problem, when we say that a splitter is authenticating itself, we do not mean that splitter is attempting to establish its singular identity by serial number or something of that kind. Instead, the splitter must simply demonstrate that it has been properly scrutinized by the network administrator and pronounced fit to be connected to the network. During this inspection process, the local administrator writes a small amount of site-specific secret data into the splitter memory, which must be protected against disclosure using "smart card" techniques. Since all splitters are functionally equivalent, it is the ability to respond to challenges that depend on knowing the secret data for this site which earns a splitter the right to join the network.

Suppose splitter V has just been powered on and wishes to authenticate itself to its neighbor U, which is already part of the authenticated chain. After U and V exchange some information, U must classify V among the following choices:

- V is a splitter that successfully responded to challenges from U, which requires V to know the site-specific secret data. In that case, V must be a bonafide splitter which has passed the network administrator's inspection and can be trusted.
- V failed to respond to the challenges from U correctly. Hence V is an alien splitter, or perhaps a completely different device masquerading as a splitter, and cannot be trusted. At this point U can either treat V as a user who connected to the end of the string, or simply disable the link.

Interestingly, there is also a possibility that the splitter V does not know the secrets, yet it was able to respond correctly to U's challenges. How is that possible? It can happen that V is connected to U on one side and T on the other. V relays the challenges from U to T and the responses from T back to U. This is well known Man-In-The-Middle Attack or Bucket Brigade Attack [10].

Suppose, a novice player is playing chess with two top players simultaneously, in different rooms. The novice player is playing white on one board and black on the other. He looks at the white move at board-1, plays it at board-2, waits for the response there and plays it back at board-1. The point is, unless the top players come to know otherwise, they will never realize that they are actually playing with someone else, not the person who is making the move in front of them.

Lets look at the timing analysis diagram (Fig 4) as the signal travels back and forth where splitter U sends a query and receives a response. We consider two situations here. First, when the immediately neighboring splitter (X) is responding on its own. Second, when X is posing as the Man-In-The-Middle and taking help from the next splitter Y.

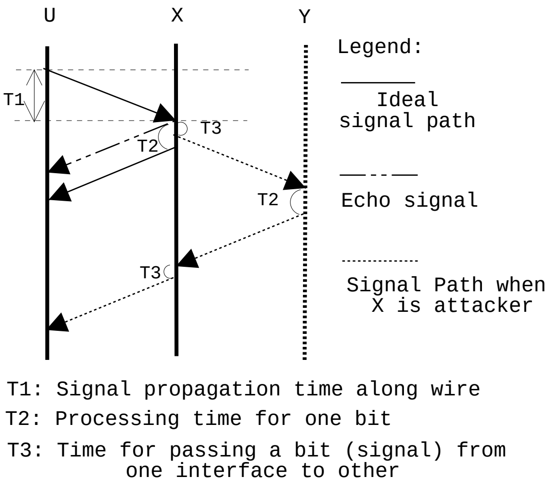

U
X
Y
Legend:
Ideal signal path
Echo signal
Signal Path when X is attacker
T1
T2
T3
T2
T3
T1: Signal propagation time along wire
T2: Processing time for one bit
T3: Time for passing a bit (signal) from one interface to other

Figure 4: **Timing Diagram with and without attacker**

The question is, how does U ensure that the response it receives was generated by its immediate neighbor and not just relayed from another splitter further down the chain?

When U gets response directly from X, i.e., the normal case, the response time is

$$T_1 + T_2 + T_1 = 2T_1 + T_2$$

When X is an attacker, the response time is

$$\begin{aligned} & T_1 + T_3 + T_1 + T_2 + T_1 + T_3 + T_1 \\ = \;& (2T_1 + T_2) + 2(T_1 + T_3) \end{aligned}$$

We see that there is a difference in response time, $2T_1 + 2T_3$, at U between the two cases.

Propagation time T_1 and time required for copying between interfaces T_3 are constant in a particular setup. But, T_2 depends on the nature of challenge i.e., the amount of computation required to formulate the response. As the required computation time grows, T_2 starts increasing. At some point, the difference in response times between these two scenarios, $(2T_1 + 2T_3)$, will become indistinguishable relative to the honest response time $(2T_1 + T_2)$.

This leads us to the notion that the authentication protocol should be designed in such a way that the overall authentication process may depend on an arbitrary amount of pre-computation, but the specific responses to individual challenge messages should require a minimum amount of "on-line" computation.

4.2 Obscurity Can't Save Us

The threat is that the plaintext communication is exposed to the man in the middle. If somehow it was possible to communicate authentication messages between bonafide splitters through some covert channel that the attacker could not decipher, then our problem would immediately be solved. For example, we might consider trying to encode hidden information by intentionally introducing distortion into the analog waveform that represents a given symbol being sent over the physical channel. One likely candidate is the Fast Link Pulses (FLP) sequence that is used during Ethernet auto-negotiation to select the correct data rate and duplex settings before the first bit of valid data has been sent over the link [8]. The minimum, typical and maximum values of clock pulse to clock pulse interval are specified as 111, 125 and 139 microseconds respectively. Thus, a covert channel could be created by using an interval of less than 120 microseconds between successive clock pulses to represent a hidden 0-bit and an interval of greater than 130 to represent a hidden 1-bit. In this case, a man-in-the-middle attacker would fail because his ordinary Ethernet transceivers would be unaware of (and hence unable to relay) the data being sent over the covert channel. Unfortunately, in order for this scheme to work among the bonafide splitters, they must agree upon some standard encoding (proprietary or open) for representing the hidden data. Once that standard is known, it becomes useless.

4.3 Exposing the Man-In-The-Middle

Returning to Fig 4, suppose there was some method by which the known bonafide splitter (here U) could measure the physical properties of the link segment to determine the round-trip propagation delay, $2T_1$, across the attached link segment to the unknown splitter (here X). How can U take advantage of this physical layer echo time for detecting a man-in-the-middle attack? In this case, U can subtract the round-trip echo time from the elapsed time for receiving the response from X, which we call the *excess delay* (beyond the physical layer round-trip echo time) for receiving the response to each of its challenges. If X is a bonafide splitter, then each excess delay should be approximately $T_{bonafide} = T_2$. Conversely, if X is a man-in-the-middle, then each excess delay should be approximately $T_{alien} = (2T_1 + 2T_3 + T_2)$. Thus, the key to catching the man-in-the-middle is designing the authentication protocol in such a way to make $2T_1 + 2T_3$ as large as possible relative to T_2. In other words, we want

$$\frac{T_{alien}}{T_{bonafide}} \approx 1 + \frac{2(T_1 + T_3)}{T_2}$$

to be significantly greater than 1.

Now suppose that the U's challenge is constructed in such a way that X can use a deterministic algorithm to generate the response, and that the "online" portion of the response computation uses a constant number of bit operations. Since the challenge and response messages will be sent as normal Ethernet frames after the link has been established, it makes sense to measure the excess delay in units of baud rate for the link. In the case of Gigabit Ethernet, each symbol is a PAM-5 codeword, transmitted in parallel across all four pairs in the cable, that delivers a block of 8 bits of user data in parallel once every $8ns$.

Clearly, $T_3 \geq 8ns$ because an alien splitter cannot relay the codeword to Y before it has been received from U, and in practice T_3 may be much larger than this because the data must be passed from one physical port to another.[4] In addition, if we assume a segment length of $1m$ and a signal velocity of $2 * 10^8$ m/s through copper cable, we find $T_1 = 5ns$. Finally, we will show below in section 4.6 how T_2 can be reduced to approximately 2 symbol periods or $16ns$. Therefore, since

$$\frac{T_{alien}}{T_{bonafide}} \approx 1 + \frac{2 * (5ns + 8ns)}{16ns} > 2.5,$$

we can expose the man-in-the-middle if: (i) the authenticator can measure the round-trip echo time, T_1; and (ii) the supplicant can respond to each challenge sufficiently quickly, i.e., within approximately two channel symbol periods.

In full-duplex baseband communication over the 1000BASE-T channel, the echo of transmitted signal is mixed with received signal and further distorted by far and near end cross talks with neighboring transceivers. Still, the value of T_1 can be estimated from channel equalization data that is generated by DSP-based physical layer transceivers (for e.g., [11]) for their own use, independent of any packet level data sent by the other device. Here are three possible methods. First, the link is equipped with *a digital echo canceler circuit* that removes a weighted moving average of its previous transmissions from the received signal before the transceiver attempts to decode it.[5] Echo is created as the signal encounters small variations in impedance as a function of distance. The cable discontinuity at each connector is usually (although not always) recognizable as a discrete echo source. The round-trip delay estimate is based on the index of the last non-zero weight. Second, the link is equipped with *an automatic gain control circuit* to adjust the transmitted power level to compensate for the signal attenuation over the link. Since the rate of signal attenuation per unit distance through the cable is specified to fall within a narrow tolerance, we can estimate the cable length with an error of less than $10m$. Third, if this link supplies DC power to the next node, then we can instrument the *inline power detection algorithm* to obtain an estimate of the round-trip delay before the next node has actually received the power needed to turn itself on. The basic idea is the following. To prevent damage to legacy devices that neither require DC power delivery through the Ethernet cable nor are aware that such power might exist, the IEEE 802.3af standard [7] requires each power source to execute a discovery algorithm before turning on the power. An Ethernet-powered device in its passive *unpowered* states "advertises" its need for power by connecting a $25K\Omega$ resistor across two pins in the CAT-5 cable, which the power source detects by sending a series of pulses across the link and looking for return of the echo after passing through the resistor.

By adopting one or more of these methods, U can obtain a reliable estimate for the round-trip echo delay for the link. Although a truly determined attacker could compromise this estimate, the cost would be too high compared with other methods for compromising the link (e.g., reading the data stream by monitoring the EMI generated by the cable). In particular, attacker would need to create a new DSP transceiver design which can inject false echos at larger round-trip delays to defeat the first method, tolerate excessively high signal levels to defeat the second method, and fake the response of a distant coupling resistor to a power discovery pulse to defeat the third method. Furthermore, the goal of our approach is simply to protect the integrity of (and to create a single management interface for) the partitioned switch system *as far as the splitter access ports*. Stopping the attacker from compromising the naïve security of 802.1x, by inserting another device into the middle of the patch cable connecting the host to its access port, is beyond the scope of this work.

4.4 Splitter Authentication Procedure

From now on, we assume that the authenticator splitter is able to identify if someone is playing the Man-In-The Middle attack.

Earlier, the need for simplicity in computation at responder was identified. Inspired by the philosophy of one-time pad (or Vernam cipher) [15], we use bitwise exclusive-or operations in our approach. Lets try some simple authentication mechanisms.

P1: Some secret key K is known only to bonafide splitters. U sends a nonce[6] to X. X XORs the nonce with K and sends back to U.

However, **P1** is flawed as exploited by the following attack. X is connected to U but keeps the link down. Sometime later, Y come to join the network. Since X is physically at the end, Y connects to X. Y is unaware of the fact that $U - X$ link is down. Now, X sends some a nonce to Y.

[4] Indeed, the maximum one-way circuit delay permitted by the 1000BASE-T specification to pass data between the physical connector (MDI) and the MAC layer transmit/receive finite-state machine is $132ns$, which includes 84 bit times to pass through the transceiver logic according to Table 40-14[8], MDI to GMII delay constraints (full duplex mode), plus an additional 48 bit times to pass through the reconciliation sublayer according to Table 35-5[8], MAC delay constraints (with GMII).

[5] Weights assigned to different feedback times are selected during link startup, by exchanging a fixed pattern of non-data code groups and adaptively adjusting those weights to minimize the mean square error. Thereafter weights adapt slowly to changes in the characteristics of the link. General information about DSP echo canceler may be found in [12] [13], while their application to 1000BASE-T Ethernet is described in [14].

[6] A nonce is some information that is *fresh*, i.e., has never appeared before.

From the response, X easily finds out K. Then, X makes the $U - X$ link up and responds to U's challenge correctly.

This is an example of Chosen Plaintext Attack. This attack also points out an important dimension. The integrity of the authenticator must be validated (explicit or implicit) by the new splitter (supplicant) before revealing *all* its secrets.

P2: Instead of a *single* key fixed in advance, this time keys will be agreed upon as the need arises.

All the bonafide splitters and the main switch in a particular administrative area will have knowledge to a prime number p, a number α relatively prime and smaller than p.

When a splitter i is powered up, it chooses a random number (private key) i_{pri}. Another corresponding number (public key) is computed as

$$i_{pub} \equiv \alpha^{i_{pri}} \bmod p$$

Adjacent splitters exchange their public keys, i.e. U gets X_{pub} and X gets U_{pub}. Then they use Diffie-Hellman [15] mechanism to agree on a common number. U calculates

$$\begin{aligned} b_1 &\equiv X_{pub}^{U_{pri}} \bmod p \equiv (\alpha^{X_{pri}} \bmod p)^{U_{pri}} \bmod p \\ &\equiv \alpha^{X_{pri} U_{pri}} \bmod p \end{aligned}$$

X also calculates in identical fashion and finds

$$b_2 \equiv \alpha^{U_{pri} X_{pri}} \bmod p$$

U and X computes b_1 and b_2 separately, but they are the same number, say B. As a matter of fact, lets assume all the computation is carried out in a sufficiently wide w-bit datapath. B will be used as mutual secret key for rest of this authentication session.

Some reasons for choosing the above procedure are:

- U and V agrees on a number (B) without transmitting that on the wire.
- Both U and V share equal importance in choosing B.
- If the splitters choose different private keys on power up, then, even the same pair of splitters will agree on different B in different sessions which contributes to improved security.

Thus we see that splitters exchange their public keys and compute a mutual secret key known to no-one else. Now, the splitters can check each others authenticity by the process of sending a w-bit nonce and expecting a response which is XORed with the mutual secret key. Who will send the challenge first? The answer is, whoever sends first, U can be fooled. If X is allowed to precede, then it can learn the key simply XORing the sent nonce with the received response. If however, U precedes then X can play the attack as in **P1**. Lets try to avoid this by making them to send their nonces simultaneously. This is possible in case of full-duplex channel. Consider the following attack. After sending all the bits of nonce, X waits until the first response bit arrives from U. From that it can infer the first bit of key, compute the first bit of own response and send to U. Same for subsequent bits. Thus, we realize that in addition to the mutual shared key agreed upon at runtime, there needs to be some additional shared secret that can be referenced during authentication process.

In the mechanism described below, a particular bit string, A, is shared among all bonafide splitters in a domain. This is the site-specific secret data that the network administrator stores in splitter during the inspection process.

4.5 Challenge-Response Messages

We assume that all bonafide splitters contain an array of bits A whose length is 2^l. The splitters exchange k ($k > l$) bit authenticating messages which contains two parts:

1. *Position*: a l bit string which is the starting index in the array A from where $(k - l)$-bits are used to answer the current challenge.
2. *Body*: a $(k - l)$ bit string which is response of the last challenge from the peer or the current challenge.

Instead of being concatenated, the Body and Position strings are interleaved. The bit positions that contain the Position string are determined by the mechanism described below.

The splitters use an r-bit ($r > l$) linear feedback shift register (LFSR) [15]. It is initialized with rightmost 'r' bits of B as computed above. All the splitters use the same polynomial for this LFSR. Lets exemplify with the following numbers:

l = 5
A [0-31] = 0000 1101 0101 0010 1011 0110 1000 0111
B = 0010 0001 1001 1100 1010 0110
k = 16
r = 16
LFSR polynomial = $x^{16} + x^{15} + x^{14} + 1$

The LFSR corresponding to this polynomial is shown in Fig 5. On every iteration, k_{14}, k_{13}, $\ldots k_0$ would each be shifted one stage to the left. The new value of k_0 will be $k_{15} \oplus k_{14}$.

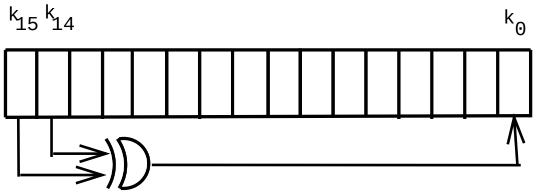


Figure 5: **LFSR corresponding to polynomial $\mathbf{x^{16} + x^{15} + x^{14} + 1}$**

The number of bits required to index a k-bit long authentication message is $p = [log_2 k]$. So, after each iteration, the least significant p bits of the shift register contents are noted. The LFSR contents are left shifted until l (leaving the initial one) unique least significant p bit contents are obtained. These numbers will be the indices to the Position string bits in the k-bit authenticating message.

The LFSR contents after each stage of shifting are shown in the following table.

Initial Contents	1001 1100 1010 0110
After 1 shift	0011 1001 0100 1101
After 2 shifts	0111 0010 1001 1010
After 3 shifts	1110 0101 0011 0101
After 4 shifts	1100 1010 0110 1010
After 5 shifts	1001 0100 1101 0100
After 6 shifts	0010 1001 1010 1001

Thus 5 unique rightmost 4 bits from LFSR after minimum shifts are 1101 (=13), 1010 (=10), 0101 (=5), 0100 (=4) and 1001 (=9). These will be the bit locations of Position string in authentication message. The mask for selecting the hidden Position string bits in an authentication message is 0010 0110 0011 0000.

U sends the first message U_1. X responds as $f(U_1)$ immediately and sends its own challenge X_1. In this way, challenge-response pairs $(U_1, f(U_1))$, $(X_1, f(X_1))$, $(U_2, f(U_2))$, $(X_2, f(X_2))\dots$ takes place in sequence (see Fig 6). No delay is required between the response of one pair and the challenge for the next pair. And except for a little phase shift as explained in section 4.6, a challenge and its response, though appear sequential in diagram, actually traverse in parallel.

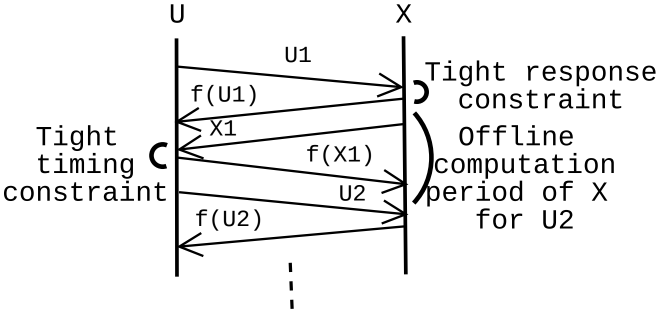
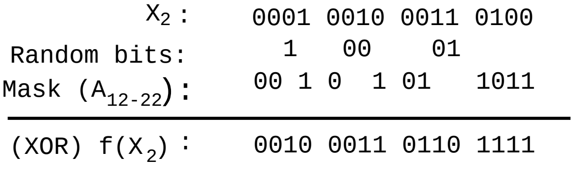

U
X
U1
Tight response constraint
f(U1)
X1
Tight timing constraint
f(X1)
Offline computation period of X for U2
U2
f(U2)

Figure 6: **Challenge Response Messages**

For $i = 1, 2, \dots$, Position bits in U_i are used to create mask string to answer challenge U_{i+1}. Also, Position bits in X_i are used to create mask string to answer challenge X_{i+1}. For example, assume the Position bits in X_1 are 01100. So, $(k - l)$ bits of A starting from 12, i.e. A_{12-22} will be used as mask to reply the challenge X_2. Assume $X_2 = 0001\,0010\,0011\,0100$ and l pseudo-random bits chosen by U are 10001. The computation of $f(X_2)$ is shown in Fig 7. The challenge bits are XORed with either random bits or mask bits depending on the bit position.



```
            X2 :   0001 0010 0011 0100
   Random bits:      1   00    01
Mask (A12-22):     00 1 0  1 01   1011
--------------------------------------
 (XOR) f(X2) :    0010 0011 0110 1111
```

Figure 7: **Bitwise response computation**

Techniques described in section 4.3 are exercised by the challenger to detect any possible man-in-the-middle attack. If X responds correctly to all challenges, it is accepted. Otherwise U rejects X's admission as a bonafide splitter.

4.6 Exchanging Authentication Messages

The authentication messages could be exchanged as Link Code Words (LCW) during the auto-negotiation phase or as the payloads in ordinary Ethernet frames after the link has been established. During auto-negotiation, the same LCW is sent multiple times in both directions to ensure the link partner receives it correctly. Sometimes same LCW is sent multiple times with only changing the 'Ack' bit. Clearly this approach greatly increases the time available for the supplicant to respond to a challenge message beyond the physical layer round-trip delay, and hence would be ineffective exposing man-in-the-middle attacks. Thus, we assume that authentication messages are sent as ordinary Ethernet frames and that the link is operating in full-duplex mode.

In this case, the finite-state machine representation of the Ethernet MAC layer needs to be modified to prevent normal Ethernet operation until the authentication phase has been successfully completed. Therefore, the destination and source addresses for an authentication message are irrelevant and be arbitrarily set to the broadcast address and null (all-zero) address. In addition, since we know in advance that only authentication frames will be sent at this time, the transceiver at responder, say X, can do some preprocessing to minimize T_2 as follows. As soon as the start of the preamble for an incoming frame is detected by the receive logic within X's transceiver, its transmit logic immediately starts sending its own preamble. After the remainder of the preamble and the fixed-format frame headers have been exchanged in this manner, X receives the first octet of the challenge from U. Thereafter for the duration of this frame, each incoming octet received from U is decoded, XORed with (offline computed) waiting mask and random bit strings and immediately re-encoded as the next outgoing symbol that will be transmitted by X. Thus, our approach requires a phase shift of only one octet at X between the reception of each octet from the challenge sent by U and the transmission of the corresponding octet of the response generated by X.

5 System Issues

5.1 VLANID assignment

If X responds to U's challenges correctly, then U sends a special AUTHENTICATED message to X, which also includes the next available VLANID. Then X replies to this message by sending its port count, n say, to U, which relays all the information about the newly-authenticated splitter, including X's public key, to the main switch.

5.2 Frame Scheduling

The backbone links connecting the chain of splitters to the master switch operate at a much higher data rate than the access links for connecting individual hosts to a splitter. The use of different speeds is intended to reduce congestion along the backbone. Obviously, this approach handles outbound traffic very well, since it is being distributed from a single source (i.e., the master switch port) to multiple destinations (i.e., the appropriate access port). Hence a splitter will never need to buffer any outgoing frames waiting for transmission via backbone port S, independent of the total number of hosts connected to the string. However, the problem is significantly more challenging in the case of inbound traffic, which is collected from multiple sources (i.e., the set of all access ports) for delivery to a single destination (i.e., the master switch port). Since the traffic volume *increases* as we move closer to the destination because of the addition of traffic originating at local access ports, a splitter may need to buffer outgoing frames waiting for transmission via backbone port N. If the total number of active access ports in the entire chain is limited to the ratio of speeds between the backbone links and access links, then we can establish a finite upper bound to the worst-case queue size. However, if the total number of active access ports is greater than this speed ratio, then the worst-case queue size is unbounded and we face a serious fairness problem in allocating the inbound bandwidth among the different VLAN flows.

Fortunately, the bandwidth allocation problem in shared, unidirectional bus networks has been widely studied in past. In particular, Manjunath *et. al.* [16] have proposed an *optimal work conserving preemptive scheduling algorithm* for a network model that exactly matches our partitioned switch architecture. Thus, we will assume that the splitters use the optimal work conserving preemptive scheduling policy as described in [16]. As a result, a splitter (near to master switch) will sometimes swap an incoming frame arriving from backbone port S with a frame waiting in its local transmit buffer.

6 Conclusion

In this paper we extended the switching functionality beyond the traditional switch box using splitters. Splitter-based networks provide flexibility and easy reconfigurability to high densely environments. The key problem of splitter-to-splitter authentication is studied in detail. The main challenge is solving the man-in-the-middle attack. Our solution introduces a novel cross-layer application of the physical channel parameters, obtained from a DSP transceiver, to estimate the round-trip delay over the link.

Using this timing information, we define a splitter-to-splitter authentication method that can resist man-in-the-middle attacks. The proposed mechanism operates without any sort of key server, and none of the entities must disclose all its secrets to the other. In addition, we believe that the general concept of authentication through one's *location* rather than *identity* may be of interest in other applications.

Acknowledgement: The authors would like to thank Sailesh Rao for his helpful comments about 1000BASE-T transceiver design.

References

[1] Geoffrey O. Thompson. *Hub Privacy Filter for Active Star CSMA/CD Network (United States Patent 5,251,203).* Xerox Corporation, October 1993.

[2] IEEE Computer Society. *IEEE Std 802.1s-2002 (Amendment to IEEE Std 802.1Q, 1998 Edition).*

[3] IEEE Computer Society. *IEEE Std 802.1X-2001.*

[4] L. Blunk and J. Vollbrecht. PPP Extensible Authentication Protocol (EAP). *RFC 2284,* March 1998.

[5] Technical Committee TR-42. Commercial Building Telecommunications Cabling Standard - Part 1: General Requirements. *Telecommunications Industry Association,* ANSI/TIA/EIA-568-B.1-2001, April 2001.

[6] Paul James Frantz and Geoffrey O. Thompson. *VLAN Frame Format (United States Patent 6,111,876).* Nortel Networks Limited, August 2000.

[7] IEEE Computer Society. *IEEE Std 802.3af-2003.*

[8] IEEE Computer Society. *IEEE Std 802.3-2002.*

[9] 3Com Corporation. 3Com Network Jacks Family Overview. http://www.3com.com/ -to- Network Jacks, 2003.

[10] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach featuring the Internet.* Addison Wesley Longman, 1999.

[11] Marvell Semiconductor Inc. Marvell Virtual Cable Tester Software Solution. http://www.marvell.com/products/vct_soft.jsp, 2003.

[12] David G. Messerschmitt. Echo Cancellation in Speech and Data Transmission. *IEEE Journal on Selected Areas in Communications,* SAC-2(2):283–297, March 1984.

[13] David G. Messerschmitt. Asynchronous and Timing Jitter Insensitive Data Echo Cancellation. *IEEE Transactions on Communications,* COMM-34(12):1209–1217, December 1986.

[14] Gigabit Ethernet Alliance. Gigabit Ethernet 1000BASE-T Whitepaper. http://www.10gea.org/GEA1000BASET1197_rev-wp.pdf, 1997.

[15] Douglas R. Stinson. *Cryptography: Theory and Practice.* Chapman & Hall, 1st edition, March 1995.

[16] D. Manjunath and Mart L. Molle. The Effect of Bandwidth Allocation Policies on Delay in Unidirectional Bus Networks. *IEEE Journal on Selected Areas in Communications,* 13(7):1309–1323, September 1995.