Scholastic Networking 101

ATLANTIC

Computing Technology Corporation

ATLANTIC.com

Scholastic Networking 101

Version 1.0

Scholastic Network Primer

This is an introduction to the kinds of network technologies and challenges that are at work within a typical school network. If you're the network administrator, then many of these topics will be familiar to you. If you serve a different role, we hope that the language will be transparent enough that it helps you understand the school network better and will provide an appreciation for the work that goes into maintaining a school network. I've always mused that school networks are the most demanding of all enterprises: the user density is greater than that of any business; the whole population gets up and moves every 45 minutes; the network is being hacked from within. Accordingly, there's more that can go wrong and more people to be affected by it.

An enterprise network, like yours, is a mix of devices that *describe*, *move*, *filter* and *authorize* network traffic.

- Routers and switches move traffic from place to place.
- DHCP, DNS, OSPF and ARP are examples of services that describe the network to its participants.
- Directories, RADIUS and TACACs are examples of elements that are used to authorize traffic.

- Firewalls and web appliances filter network traffic.

These four concepts underlie all wired, wireless, local area network (LAN) and wide are network (WAN) connectivity.

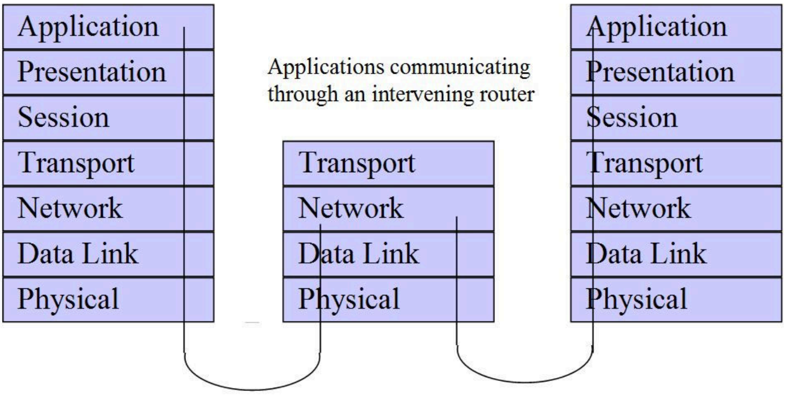
Moving Network Traffic

At their most basic level, data networks are computers communicating across an intermediate medium, like a wire. In fact, the very first networks were just that—two computers on each side of a wire. To make data networking scale, there have to be standards and methods for sending signals, for addressing other computers, for gaining access to the network (like raising your hand to take your turn), for finding other computers by name, and for sending data near and far.

ISO model

It's dry, but convenient, to use a model to describe how computer networks are built. Terminology from the model makes it possible for two people to talk about specific kinds of network functions without having to describe what kind of network they are using.

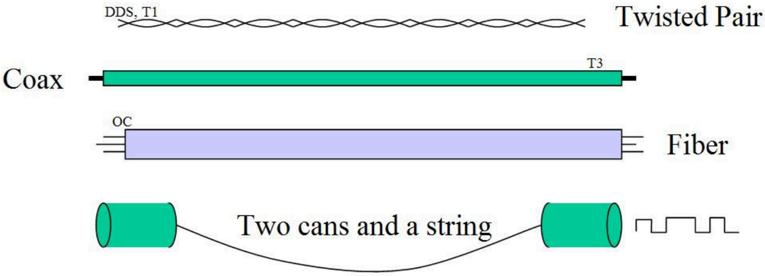
The standard for network description is the OSI reference model. It describes seven peer-protocol layers. The idea is that all primitive networking functions can be ascribed to one of these layers, and that a carefully crafted set of protocols might be able to stack neatly, such that you could swap out functions at one level without modifying any of the others.



Seems abstract? You benefit from this kind of stacking on your own network, every day. For instance, you might talk about accessing a server at the Board of Ed from a laptop at the high school. Between the laptop and the BoE, there may be several kinds of networks, including WiFi, copper and fiber. Your experience isn't affected by the technology differences along the way because the networking you are interested in occurs at a higher level. The bottom layers are different, but the network layer is the same. You don't feel the potholes.

Physical layer

The bottom layer, *physical layer* (sometimes shortened to "phy"), describes the medium that network traffic runs over. It could be twisted pair, coax, fiber or two cans and a string.



Network interfaces modulate data across the physical layer using electrical signals, pulses of light or radio, depending on the transmission ability of the medium. For two cans and a string, it would be vibrations.[1]

Data Link Layer

The next layer up, the *data link layer* (often called "layer 2"), is concerned with accessing the physical media, encoding data and communicating. It is concerned with how to send a 0 or a 1 (data bits). It's also concerned with how to take turns sending 0s and 1s, and how to direct them to another computer.

If we want a given device on the medium to be able to communicate with another device, we need to be able to identify the destination for the message (called the media access control or *MAC* address) and have a method of entering the medium and sending our message (logical link

[1] I once worked for a company that sent signals up from an oil drilling operation by creating pulses in drill bit lubricating mud. The data rate was 1.5 seconds per bit.

control or *LLC*). It might be by light pulses, electricity or over the airwaves. Each type of medium has its own notion of *logical link control* (LLC) and *media access control* (MAC).

In the picture above, the tin can on the left could have MAC address ‘1’; the can on the right could be ‘2’. When ‘1’ wants to send a message to ‘2’, it will address the message using can ‘2’s address. Can ‘1’ will gain access to the medium (the string) and send its message down the string according to agreed-upon standards for cans and strings. On a network with more than two participants, the MAC is what identifies which message is for whom.

```
<start><from can #1><to can #2>hello!<end>
<start><from can #2><to can #1>hello back!<end>
```

In your network, every port, every transceiver (GBIC/SFP) and every wireless network device uses agreed-upon LLC access methods that describe how to send data. Every device has a unique MAC address. The MAC address is usually made part of the device at the time of manufacture; it’s not something that we get to choose.

Fortunately (and it didn’t have to be this way; wasn’t always), the same *Ethernet* MAC address format is used on all of the media (wire, fiber, WiFi) that you have in your schools. The format is 48 bits, represented as xx:xx:xx:xx:xx:xx, in hexadecimal. An example might be an address like:

0C:C4:7A:6B:23:E7

Because the format is the same across your media, it is possible to extend one logical network across multiple media types. This is how we can make one virtual LAN (VLAN) extend across wired, wireless and optical networks, and across schools.

Note that up to this point, we haven't said anything about Internet Protocol (IP) addresses or IP networks or the Internet. The concepts of layer 2 LLC and MAC underlie many layer-3 protocols, including NetBIOS, DECNet, IPX *and* IP. The goal of the ISO model was to separate the network layers so that there could be independence between them, and to even make it possible to run multiple network protocols or multiple IP networks on the same layer-2 network.

We explain IP, next.

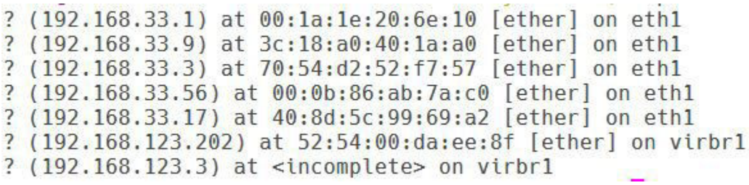
IP Network Layer

Whereas layer 2 addresses are generally hardwired into network interfaces, and not ours to choose, IP addresses are arbitrary—we can assign any IP address we like to any interface. Internet Protocol (IP) addressing allows us to build a network of networks, where participating computers can distinguish between local neighbors and far-off destinations.

Say that we give one computer an IP address of 10.10.10.1, and give another an address of 10.10.10.2. Each of the two computers now has an assigned IP address, and each came from the factory with a unique MAC address. As we saw above, network communication takes place using MAC addresses—not IP addresses. In order for 10.10.10.1 to send a message to 10.10.10.2, we need some method to map IP addresses to Ethernet MAC addresses.

```
ARP, Request who-has 192.168.33.56 tell 192.168.33.8
ARP, Reply 192.168.33.56 is-at 00:0b:86:ab:7a:c0
```

Network devices make the MAC/IP address association by a mechanism called *Address Resolution Protocol* (ARP). Each host keeps a cache of the MAC addresses of devices it finds on directly connected networks. If one IP host wishes to communicate with another for which it has no MAC entry, it broadcasts an *ARP request* containing the IP address it is trying to resolve. The host owning the IP address (or a proxy) responds to the broadcast with the matching MAC address. With each others' MAC addresses in hand, the two hosts can now communicate at layer 2.

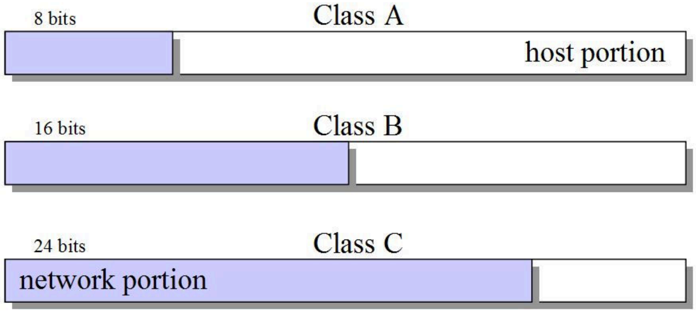


```
? (192.168.33.1) at 00:1a:1e:20:6e:10 [ether] on eth1
? (192.168.33.9) at 3c:18:a0:40:1a:a0 [ether] on eth1
? (192.168.33.3) at 70:54:d2:52:f7:57 [ether] on eth1
? (192.168.33.56) at 00:0b:86:ab:7a:c0 [ether] on eth1
? (192.168.33.17) at 40:8d:5c:99:69:a2 [ether] on eth1
? (192.168.123.202) at 52:54:00:da:ee:8f [ether] on virbr1
? (192.168.123.3) at <incomplete> on virbr1
```

Above is an ARP table correlation of IP addresses to MAC addresses on a computer. The command to produce this listing is 'arp –a'.

Routing

An IP address has two parts—a host portion and a network portion. When we give a device a network address, we also specify a network mask. The mask tells the device which part of an IP address describes 'host' and which part describes the 'network.'

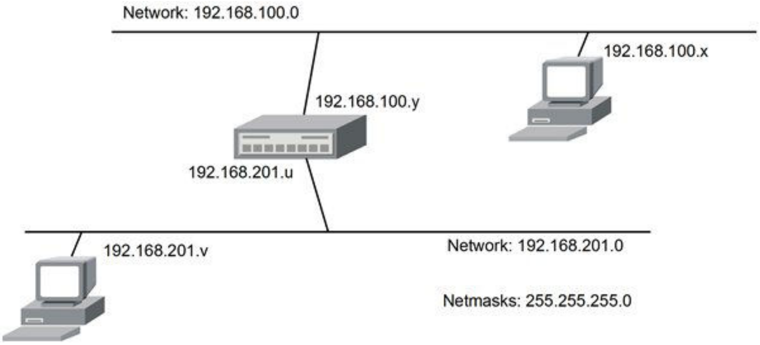


For example, if we were to give a computer an address of 192.168.100.1 with a network mask of 255.255.255.0 (24 bits), we would be telling the computer how to identify the network portion of the address versus the host portion. The computer would derive the network portion by applying the network mask. The result would be 192.168.100.0. Eliminating the network portion of the address with the inverse of the mask gives the host part—0.0.0.1. From the process, we learn that we're talking about host '1' in network '192.168.0.'

When a computer wants to talk to another, it makes a simple calculation: the sender takes its own address and masks out the host portion. It also takes the receiver's address and masks out the host portion. Then it compares the two results. If the answer is the same, then the sending computer knows that the receiving computer is on the same layer-2 network[2], and that ARP resolution (described above) will indicate which MAC address will receive the traffic.

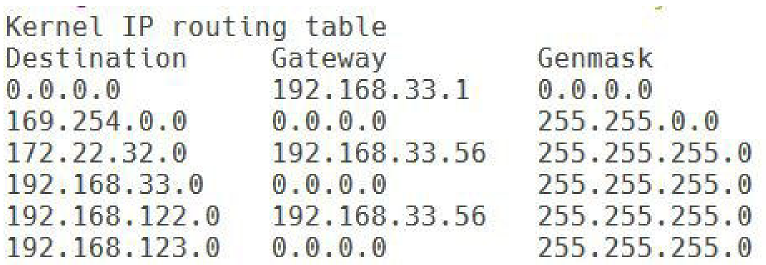
If the answer is *different*, then the sending computer knows that the destination is for another layer-2 network, somewhere else, and that the message has to be routed through a *gateway* or *router*.

[2] 192.168.100.1 (logical AND) 255.255.255.0 = 192.168.100.0
192.168.100.2 (logical AND) 255.255.255.0 = 192.168.100.0
The result is the same. Therefore, these two IP addresses are on the same network.



In this picture, assume that the computer at 192.168.100.x wants to send something to a computer at 192.168.201.v. The mask calculation creates a comparison of 192.168.100.0 versus 192.168.201.0, which shows that the network portions don't agree, and that the traffic has to be routed.

Many times, the only gateway a computer knows about is a *default gateway* or *default route*. The default gateway is the forwarding destination for all traffic a device cannot deliver locally using layer-2 networking or by some other route.



```
Kernel IP routing table
Destination     Gateway         Genmask
0.0.0.0         192.168.33.1    0.0.0.0
169.254.0.0     0.0.0.0         255.255.0.0
172.22.32.0     192.168.33.56   255.255.255.0
192.168.33.0    0.0.0.0         255.255.255.0
192.168.122.0   192.168.33.56   255.255.255.0
192.168.123.0   0.0.0.0         255.255.255.0
```

Each device keeps its own routing table. The table does not describe the paths to reach other networks; it only lists a next hop in the journies. The routing table can be populated manually (static routing), by DHCP or via a routing protocol, of which OSPF, RIP and BGP are examples.

Potential Failure Modes:

1. Bad route – if DHCP or a routing protocol serve up a bad route, traffic may not get out.

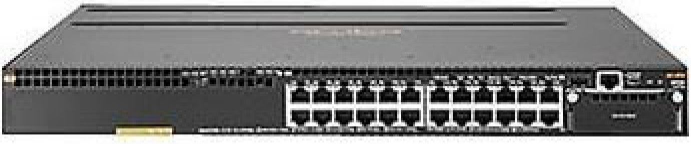
2. ICMP redirects – if a device has an incorrect route that points to another device that *has* the correct route, then the second device may provide an ICMP redirect. This will temporarily correct the routing error in the first device.

3. Split-horizon routing: in environments with stateful firewalls or NAT, the return route for traffic needs to match the outbound route. Split horizon routing problems look like bad routes, but they can also express themselves as state table bloating on a firewall.

Switching, VLANs and trunking

Network switches are layer 2 devices that simulate a physical network inside a box (like Ethernet or two cans and a string). The broadcasts and data exchanges that might occur over a fiber, copper or two cans and a string take place inside hardware—only more efficiently. A network switch watches MAC addresses on its interfaces and, over time, builds a topology map that makes it possible to forward traffic only to those interfaces where it is needed.

The term *Virtual LAN* (VLAN) comes from partitioning the interfaces of switch into different logical layer-2 networks. So, for instance, the first 12 network ports on a switch might be associated with one network; the second 12 might belong to another.



The first switches manufactured were strictly layer-2 devices; they did not have IP addresses. However, all switches you would buy today are layer 2/layer 3. That means that in addition to having the ability to act as multiple logical networks, they also can route traffic between the networks. In order to be able to do this, one must assign IP addresses to the VLANs on a switch. With IP addresses, a layer 2/layer 3 switch will be playing the part of the router in the diagram above.

In order to tell them apart at layer 2, VLANs are numbered. VLAN tagging (also known as 802.1q *trunking*) provides a way to extend multiple VLANs from one switch to another across a single interface, using their VLAN numbers. The network traffic is identical to that of a non-trunk port, except that trunked packets are preceded with a VLAN tag that contains the VLAN number. An individual switch port can be combination of 802.1q trunked traffic and *access* traffic (for one VLAN).

Layer-2 and Layer-3 Networks in Your District

Fortunately, because the MAC formats are the same across all of the different network types in your environment, we can build a mixed layer-2 and layer-3 network without caring too deeply about what is underneath. Concepts like *VLANs* and *IP Networks* are common to all the physical networks at your schools, and we can talk about a VLAN being on a wired network and wireless network, and appearing at multiple schools at the same time.

In a school district, IP network traffic is often routed between schools; each school will have its own IP network. At the same time, some emergency services and phone services may be *bridged*, meaning that one layer-2 network is extended between schools. VLAN trunking makes mixing traffic in this fashion possible.

Still, other traffic can be tunneled, which means that a network can be encapsulated within another network. The encapsulation is removed at the endpoints. The typical reason for tunneling is to allow a layer-2 network to hide inside a routed, layer-3 network, and thus extend its reach. *Wireless traffic is often tunneled.* You might associate your computer with a wireless network and be placed on a VLAN reserved for wireless devices (a *wireless VLAN).* This VLAN might exist only in the air, and back at the wireless controller and on a core switch. It travels over the rest of the network encapsulated in tunnels—invisible and often encrypted.

You might ask why not create a single, large layer-2 network? Why bother with routing and tunneling? The reason is that network protocols are chatty with maintenance traffic. Broadcasts, ARP requests, DHCP, routing announcements, multicast—these all reach every node on a layer-2 network. If we deployed a single, huge layer-2 network, then all of the maintenance traffic from hundreds or thousands of devices would burden the network.[3] In order to make a large network manageable, it has to be partitioned into smaller, layer-2, chunks, and then routed.

[3] The Internet could not run as a layer-2 network; it would collapse.

Network Services for Describing the Network

This section talks about a few of the critical network services in your network.

Address Resolution Protocol (ARP)

ARP is the mechanism by which an IP address (which is something that we assign) becomes associated with a media access control (MAC) address (which is something assigned to hardware at the factory). Network traffic is never delivered to IP addresses; it is delivered to MAC addresses. ARP makes it possible to learn which IP address is associated with which MAC address.

We discussed ARP in more detail, above.

Dynamic Host Configuration Protocol (DHCP)

DHCP is primarily for configuring IP network parameters onto computers that join the network. When a computer first brings up its network interface, it may make a request to the network for an address and other parameters. A typical set of parameters is:

- Host address and network mask
- Default gateway
- DNS nameserver(s)

This is a subset of what's possible.

DHCP is a network service that is provided by a DHCP server—a computer on the network. In its simplest configuration, a DHCP server listens on a layer 2 network for DHCP requests. A newly-joining host will broadcast a DHCP request to the network and the DHCP server will respond with an *offer*. The requesting host will assign itself the offered address and other parameters.

DHCP responses come with a *lease time*[4] that describes how long the address is available for use by the client. At half the lease time, the host will re-apply to the DHCP server for renewal, and periodically thereafter if no response is given. Once a lease is 'up', or if the DHCP server revokes or replaces it, the client must abandon the IP address.

There are a few failure modes that may occur with DHCP:

1. If the DHCP server fails it can take some time before the network users become aware of it. This is because existing IP address leases continue while new requests are ignored. It may be a day before it becomes apparent that there's a failure.

2. The DHCP server is configured with a *scope* that describes a range of IP addresses to be handed out. If the scope overlaps with statically-coded addresses,

[4] A typical lease time is between 8 and 72 hours

the DHCP server may hand out an IP address that's already in use. This can cause ARP churn as the two identically-addressed machines respond to ARP requests with two different MAC addresses.

3. If there are two DHCP servers on the network, then a requesting client will get two responses. If the scopes of the two DHCP servers overlap then duplicate addresses may be handed out. If the scopes are on different networks, then clients on the same VLAN may not be able to talk to one another.

4. If the scope on a DHCP server is exhausted then the server won't hand out additional addresses. This can happen if there are too many clients on a layer 2 network for the DHCP server to support.

5. DHCP can hand out bad info if not configured correctly. For instance, a DHCP server could hand out a bad default gateway address. The client will be stuck with it until half the lease time expires or until the client reboots.

6. After replacing a DHCP server, there will be a period when the new server may hand out addresses that clients are already using. This is because the new server will not be aware of its predecessor's leases.

DHCP Helper

DHCP can also provide service to *other* layer two networks via *DHCP Helper* service.

Say, for example that the DHCP server is at 192.168.200.10, and has a scope configured for another IP network at 192.168.99.0/24. A switch with a VLAN address of 192.168.99.1 might hear a DHCP request on this VLAN. It will forward the request to the DHCP server using its 192.168.99.1 address in the request. By this, the DHCP server knows where the request came from, knows what scope is appropriate, and can respond to the helper. When the helper switch receives the response from the DHCP server, it can forward it to the requesting client.

Failure modes:

1. The DHCP server has to be able to route its response back to the requesting switch or the client will not get an address.

2. At half lease time, the client will directly request lease renewal. In order to receive the response, the DHCP server has to be able to route to the requesting network. If there is a routing issue, a client may be unable to renew its lease. Once the lease expires, the client will start over. This will manifest itself as a periodic interruption in service.

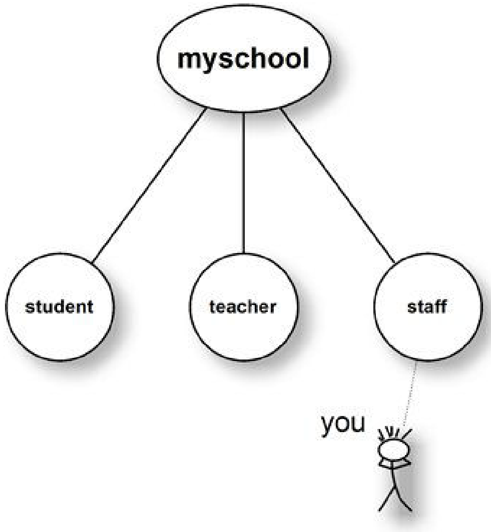
Domain Name Service (DNS)

DNS translates addresses that are convenient for humans into addresses that are convenient for computers—e.g. www.atlantic.com translates to 69.60.120.80. Depending on the organization, a local nameserver may be used with a local namespace (e.g. '.myorg') or external caching nameservers may be used. DNS can also be directed to filtering services to attempt to limit access based on a client's ability to look up a name.

DNS settings are typically given out via DHCP. When DNS is slow or intermittent, users often fault the underlying network. Since a given commercial web page may pull content from dozens of web servers, slow DNS can be particularly painful for browsers.

Directories and RADIUS Authentication

Organizations usually have a central directory that lists all of the members of the organization, associated identity information plus authentication information, like passwords. The directory will be organized hierarchically. One's place in the directory will often be chosen in correspondence with one's function. So, for example, a school called *myschool* could have three groups: students, teachers and staff. If you're part of the staff, you would be included in the staff group.



The two most common directories are Microsoft's Active Directory Service and Apple's Open Directory, though social media and Google credentials can be leveraged, too. Third-party products connect to the directory servers for the purpose of querying directory membership or performing authentication. School computers may be listed in certain directories too, which makes it possible to verify that a computer is known to the school.

Your wireless (and wired) network can use the directory services to make decisions about wireless access on a per-person basis, typically as a function of a person's group membership.

The WiFi Network

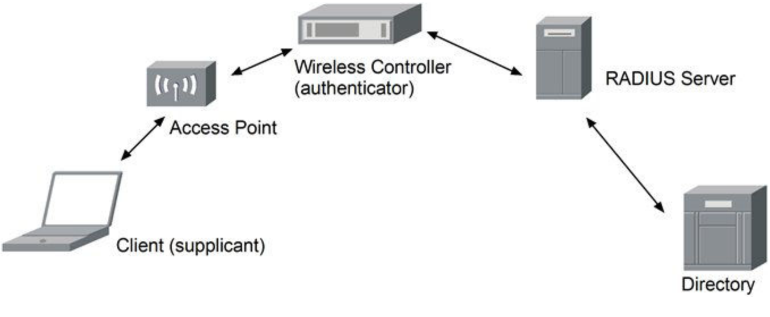
Your wireless network probably includes an SSID for guests and another for authenticated users. Schools often have a third, pre-shared key[5] (PSK) SSID for devices that cannot participate in WPA Enterprise (802.1X) authentication and encryption. This may include Apple TVs, printers and perhaps Chromebooks[6]. Typically, each SSID is connected to its own dedicated wireless VLAN. It is also possible to assign users to different VLANs as a function of their ability to authenticate.

Before your WiFi device even gets an IP address, it participates in a layer 2-based conversation with an authentication back-end. Each computer runs a software component called an 802.1X *supplicant.* The supplicant seeks network access from an *authenticator,* which for WiFi would be a wireless network controller.

[5] mPSK or multiple pre-shared keys are now possible—one per device or user

[6] Chromebooks *can* participate in 802.1X with a shared domain account or through the Chrome Console via ClearPass

The authenticator cannot see the client's credentials; they're encrypted; the authenticator passes them to a RADIUS (Remote Access Dial-In User Service) server. The RADIUS server talks to a directory server (like AD or OD, discussed above). Once the user is authenticated by the directory server, the RADIUS server forwards information that permits the authenticator to determine what privileges to give the device (or to deny access). When the wireless device is finally assigned a VLAN, all of the DHCP, ARP, DNS services we discussed in previous sections become relevant and function as for a computer on a wired network.



Because the access level granted can be based on the response from the directory, we can have a single authentication process and a single SSID shared by staff and students. Each person or group can be assigned a separate role and VLAN as a function of their AD group memberships or ability to authenticate.

Within the 802.1X protocol is a collection of Extensible Authentication Protocols (EAP) that describe how authentication is affected. A simple password exchange protected by a server-side certificate (EAP-PEAP) assures the client that they are authenticating to the valid network and encrypts the password verification. EAP-TLS uses client-side and server-side certificates to prove that the network and client are both bona fide school devices. Many other EAP methods are available.

Individual computers can be subject to access restriction too. One can vet machines by their ability to authenticate, the use of a certificate, their existence in an MDM database or their MAC address. The ability to distinguish users and machines forms the basis for secure access to the network. It also provides a basic form of bring your own device (BYOD) differentiation.

Directory services and authentication can be complex. The most common failure modes for RADIUS authentication that we see are:

1. RADIUS certificate expiry: clients trust the RADIUS server based on an X.509 certificate that is either issued by a trusted local PKI infrastructure or signed by one of the big Certificate Authorities. If the RADIUS server certificate expires, users will suddenly find that they cannot get onto the network. PSK or open networks won't have a problem. This can be fixed by renewing the RADIUS server certificate.

2. Slow authentication/failed authentication: the RADIUS server performs a lot of crypto

math. If the RADIUS server becomes overloaded, the user experience may be long network association times or occasional network drops. Likewise, the back-end directory, if overloaded, can be slow in responding to the RADIUS server. These problems can be exacerbated during class changes, when the school population roams.

Firewalls and Filters

Whereas all of the facilities we have discussed to this point are concerned with providing access, firewalls and URL filters are designed to limit traffic. Common to all security products is the notion of a list or *sieve*. A packet or session is tested against a list of rules, starting from the top and working toward the bottom, one rule at a time. The first rule that matches the network traffic wins, and some corresponding action is taken.

Firewall

Rule	**Test**	**Action**
1	If source is 172.0.0.0/8	Allow
2	If destination tcp port = 80	NAT…
3	Protocol is ICMP	Permit
4	If destination is 172.0.0.0/8	Drop
5	Any	Drop

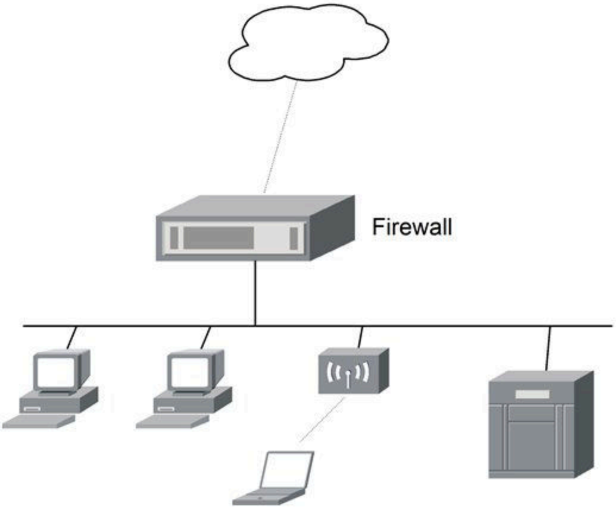
This is a sample firewall rule sieve. We start at the top and work our way down. It says that we allow any traffic coming from the 172.x network to pass through the firewall. If this rule matches, we do not test any further rules. The next rule says that traffic from anywhere else, bound for port 80 (HTTP) will have a Network Address Translation applied, and then will be passed. If this rule matches, we do not test any further rules. Then, any ICMP traffic (ping, redirects, etc), coming from anywhere, will pass.

If this rule matches, we do not test any further rules. The fourth rule says that anything bound for the 172.x network from anywhere will be dropped. If this rule matches, we do not test any further rules. The last rule, *Any/Drop,* is typically implicit, but is often included for completeness.

Firewalls may be layer-2 or layer-3 devices, or both. They usually sit inline with network traffic at the network edge, between the network provider's router and the customer's network.

URL Filter

Filters are programmed in the same way as firewalls, with a rule sieve. The difference is that the 'test' may be a traffic *category* such as 'sexually-explicit content' or 'known virus source' or 'web proxy.' URL filters may either sit inline with network traffic or alongside traffic on a layer-2 network. In the latter case, filtering is affected by issuing a reset to the client as soon as an unsanctioned web request is made.



Inspecting Web Traffic

Inspecting the content *inside* a web request is much more complicated than simply denying a request. Most web traffic is SSL-encrypted. In order to ‘see’ web traffic that is encrypted, one has to intercept the web traffic, decrypt it, inspect it, re-encrypt it, and send it on its way. However, this ‘man-in-the-middle’ intercept of a private transmission is precisely what encryption is meant to prevent. In order to intercept a student’s web session, some preparation is necessary.

When your web browser wishes to engage in an encrypted session with a web site, encryption keys are negotiated. The process depends upon trust, which is established in advance. Say, for instance, you are visiting your bank’s web site. The bank will have a *certificate* that has been digitally signed by a public key authority that your web browser (Chrome, Safari, Mozilla, Opera, etc.) already trusts. Your browser trusts the bank's certificate because it comes with a pre-installed copy of a certificate from the same public key authority. In a nutshell, if a website offers your browser a certificate that you already trust, then your browser will trust the site. If your browser does not trust a site, it will warn you, vociferously. This is to avoid having you mistake an imposter for a legitimate site.

Now imagine how a school would intercept, decrypt, inspect and re-encrypt traffic: suppose a student tries to connect to Google. The connection is intercepted; the contents are unloaded, inspected, repackaged and forwarded to Google. The response from Google is intercepted and returned to the student. The technical problem is that Google is not cryptographically responding to the student's browser's request. Rather, it is the web traffic filter that is negotiating a session key, and this will look to the browser like a man-in-the-middle attack (which it is).

For the purpose of intercepting https web traffic, it is possible for a school to create a limited capability *wild-card certificate* (matches all web sites). Trust for the wild-card certificate will have to be loaded into every student's browser as an additional root certificate.

An alternative to intercepting web traffic is explicitly assigning the URL filter as a web proxy. In this case, the browser contacts the URL filter directly, and asks that it make web requests on its behalf. Proxy settings can be loaded into browsers by MDM or updates, or set in DHCP responses.

UTM

The term *Unified Threat Management* applies to basic firewalls that also offer virus scanning, URL filtering and traffic inspection, and intrusion detection/prevention. The man-in-the-middle attack we discussed above makes it possible for a firewall or web filter to inspect the contents of an encrypted web session. By this, it is possible that a school's perimeter security will detect a virus, malware or ransom attack payload, even when hidden inside an encrypted session.

Signatures and heuristics

There are two basic methods for identifying that network content poses a threat; based on a signature pattern—a bit string—discovered within the traffic or based upon the way it is packaged, where it comes from and how it is delivered. The two methods may be used in combination.

Signatures are delivered as part of a subscription service. They represent the collective experience of others across the Internet and of the firewall vendor's security teams. If a malevolent piece of software is discovered in one part of the network, its characteristics may be shared with the rest of the (subscribed) world so that others avoid contact.   In order for a signature to be included in the subscription, someone else has to have been bitten by it.

Heuristics try to capture the way bad software is delivered so that it can be discovered even if it hasn't hurt anyone yet, or if tweaks in the payload obscure signatures. Heuristics can fail, and they can also cause false-positive detection. Heuristic algorithms will also be delivered by subscription.

DNS/address-based security

Security via reputation is becoming more commonplace. With DNS/address filters, traffic from known-bad sites and connections to known-bad servers is a fairly reliable method to keep school computers from being exposed to malware.   Reputation-based filtering does not protect against an attachment being forwarded by a friend, however.

Summary

We have discussed some of the digital plumbing that makes up a school network: network traffic forwarding, networks services, authentication and security. A school IT specialist has to be adept with all of these, *plus* wireless networking, domain services, device management, network management and support. It's a lot to know.

Atlantic Computing specializes in HPE/Aruba products in scholastic and higher education environments.

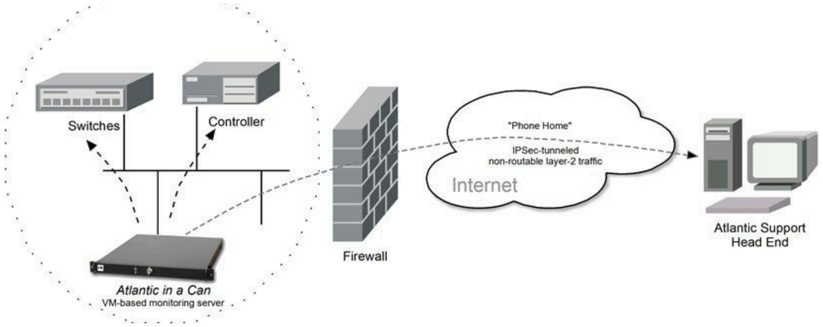
Atlantic can make the complicated simple and remove some of the barriers to running your network. Technical consultation is always free. Our managed services infrastructure offering is very affordable, secure, and will provide more of the expertise you see here.

-Kevin Dowd, Atlantic Computing

Atlantic in a Can – FREE site support server

Atlantic in a Can is available at no charge to Atlantic customers. It provides a window to the network for us when you have a question. It monitors some network functions so that we may detect a problem in advance. For those customers that wish to contract for a higher level of service, the same platform provides a base for delivery.

Security is our first concern. We have assembled a server package that "phones home" from within the customer network. It cannot be reached from the Internet. Most of its ports are shut down. All traffic is encrypted within an IPSec tunnel carrying a non-routable segment.



The *Atlantic in a Can* platform can be used for Category 2 Managed Internal Broadband Services (MIBS), which is E-Rate eligible. Under managed services, Atlantic will make the first diagnosis and interact with technical support on your behalf. Typical time-to-response is less than four hours. All levels of monitoring include bi-annual operating system updates, as necessary.* Services do not include network re-engineering, administration of authentication databases, new equipment provisioning or on-site services. Ask for details.

*Support and updates require current vendor support contract, provided separately

Atlantic Computing
22 Pennsylvania Ave, #4
Niantic, CT 06357
860 313 0727
dowd@atlantic.com

ATLANTIC.com