



Steve Friedl's Unixwiz.net Tech Tips

Deploying Mitel IP Phones in a VLAN/DHCP Environment

- [Home](#)
- [Contact](#)
- [About](#)
- [TechTips](#)
- [Tools&Source](#)
- [Evo Payroll](#)
- [Research](#)
- [AT&T 3B2](#)
- [Advisories](#)
- [News/Pubs](#)
- [Literacy](#)
- [Calif.Voting](#)
- [Personal](#)
- [Tech Blog](#)
- [SmokeBlog](#)

Table of Contents

- [Background](#)
- [PoE Tips/Cautions](#)
- [Networking Configuration](#)
- [DHCP Configuration](#)
- [Phone boot sequence](#)
- [Other resources](#)

A customer recently purchased a Mitel phone system with IP phones, and though there was no VoIP external to our network, we still had to do a fair amount of work to get the phones to play well internally. Mitel had no resources at all for how to do this, so I'm memorializing this to help the next guy. It's likely that this will be somewhat applicable to other phone vendors.

This is the paper that Mitel should have been giving their customers.

General background

Most IP phones are designed to use [Power over Ethernet](#) (PoE), which means that power is supplied by the network switch in the computer room rather than by a power brick plugged into the wall near the phone itself. This helps keep the user's desk a lot neater by eliminating one of the power sources.

In addition, rather than require a separate network cable for the phone alongside the cable for the user's computer, most IP phones have a tiny built-in network switch that rides a single network cable to the computer room and supplies a network connection to the PC plugged in behind it. This saves the enterprise from an intrusive and potentially costly rewiring project.

Surprisingly (to me, at least), the phones even support the notion of putting the phone on one VLAN and the PC behind it on another, allowing for a segregation of the network. Apparently this is a common feature.

Selection Tip: the phones themselves don't need a speedy network connection, so Fast Ethernet (100 megabit) is plenty fast, but the device *behind* the phone might need more. But considering that GigE phones and the GigE switches they plug into are both substantially more expensive than FastE versions, it may well be that your office users can live with FastE to their desktop. Ours could.

Our intention was to keep the office PCs on VLAN1 while putting the phones (and phone system) on VLAN4. This took some doing.

Power over Ethernet Tips and Cautions

As noted, Power over Ethernet goes a long way to making the user's

desktop area neater than it otherwise would be, as it largely avoids a power brick (aka "wall wart") among the other clutter. But there are lots of options here, including quite a few ways to make a mistake.

- PoE is **not** implemented by naively applying power to some unused pins on the RJ45 cable (as some have assumed). Applying power to a device that wasn't prepared to accept it would be dangerous, so there's a fairly intricate protocol that the switch uses in order to make sure that it and the device on the other end can play ball. Don't ever be tempted to hack this together yourself.
- It's possible to use **PoE injectors**, which are small powered devices that take regular Ethernet in one side and provide 802.3af PoE on the other side but you still need one for each phone, this clutters a server room very quickly. It's **so** much neater to use a real PoE switch, though it does create a single point of failure.
- **Caution:** Cisco made a number of switches with a "[pre standard version of PoE](#)" that won't work with modern devices; those shopping for used gear should pay attention here and look specifically for **802.3af standard** support.
- **Caution:** Some models of switches have *some* PoE ports, but the rest non-powered, and some online vendors (CDW, for instance) make this very easy to miss. I initially spec'd a Cisco Catalyst 2960-24LT-L 24-port FastE switch but found that only the first 8 ports are PoE with the other 16 being regular, non-powered ports. Though this can be useful for applications requiring only a few powered ports, such as driving a handful of security cameras or wireless access points, it was not at all helpful for our officewide IP phone deployment and came as an unhappy surprise. Ordering the **Cisco 2960-24PC-L** got all 24 ports of FastE PoE.

Networking Configuration

As one would expect, there are many ways to integrate these phones into a larger network, with the "easiest" being to configure them with a static IP on the same subnet as the rest of the network, as this avoids having to deal with DHCP or VLANs, but we really wanted to do it properly. Unfortunately, we found that Mitel had very few resources for us so we had to navigate this almost entirely on our own.

Our network had two subnets of interest:

- 192.168.1.0/24; VLAN1; general office network
- 192.168.4.0/24; VLAN4; IP phone network (including the phone system)

We used our internet firewall to route between the two subnets, though others may use a layer3 switch for this purpose. Though the phone traffic itself doesn't need to travel between the two networks,

the phone-control application on the user's PC will need to talk to the phone system.

After creating the VLAN, insure that each switchport used for the phone is set to a **tagged** (aka trunked) interface, because untagged traffic will default to the switch's native VLAN (which for us is VLAN1). Instructions for Cisco devices look like:

```
interface FastEthernet0/7
description Mike's phone
switchport mode trunk
```

DHCP configuration

We're mostly a Windows shop so we used the Microsoft DHCP server on the domain controller for this purpose. This is not only convenient, but it provides tight integration with DNS (which we care about).

These steps detail how to create a second DHCP scope as well as the DHCP options that support the phones:

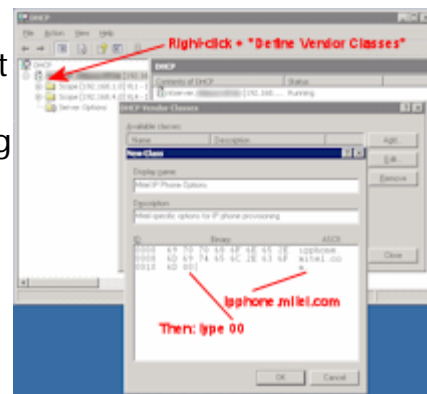
■ Define a second DHCP scope for the phones

This is done in more or less the usual way; we chose 192.168.4.0/24 with the networking options that matter for this scope (default gateway, DNS servers, etc).

■ Define a Vendor Class for the Mitel phones

A vendor class matches a string included in the device's DHCP broadcast and allows the DHCP server to recognize that it's a Mitel phone making this request and not something else.

- Right-click on the server name + "Define Vendor Classes"
- Click [Add]
- Enter "Mitel IP Phone Options" as the display name
- Enter "Mitel-specific options for IP phone provisioning"
- Enter "ipphone.mitel.com<NUL>" for the match string. Important: this has to be done **exactly** right! Click the mouse pointer into the ASCII area and type `ipphone.mitel.com`, then click at the end of the binary area (after the 6D) and type **00**. Missing this NUL byte means that the server won't recognize this class, and we burned a lot of time before we figured this out.
- Click [OK] to commit



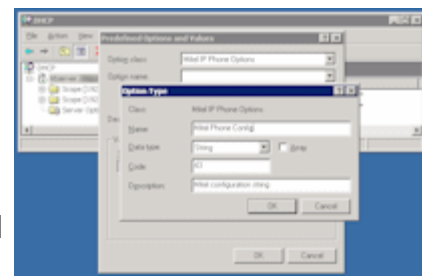
(Click to enlarge)

■ Define the Mitel DHCP option

DHCP option #43 is used for Mitel configuration, and this provides parameters to the phones by telling them where the phone system is, which VLAN they should use, and the like. What exactly goes into this string is the next item, this just creates the option that will contain it.

To enter this string into the DHCP server:

- Right-click on the server name + "Set Predefined Options"
- In "Option class", drop down "Mitel IP Phone Options"
- Click [Add]
- For name, enter "Mitel Phone Config"
- For data type, choose **String**
- For Code, enter **43**
- For description, enter "Mitel configuration string"
- Click [OK] to commit



(click to enlarge)

■ Configure the actual Mitel option string

The configuration string will vary for every instance, but there are common patterns and they are fairly well described in the Mitel documentation. Note: though the Mitel docs suggest that the parameters can be provided independently by multiple separate DHCP options, this is apparently deprecated by the DHCP standard, so the proper solution is to put them in a single vendor option #43 that we defined in the previous step.

Our magic Mitel string looks like:

```
id: iphone.mitel.com;sw_tftp=192.168.4.41;call_srv=192.168.4.41;vlan=4;12p
```

This string includes IP address of the phone system (once for the TFTP server that provides firmware and configuration, and again for the call control server), the VLAN the phone should be on (4), and a few other parameters that should be entered literally. Your IPs may vary.

Curiously, this option must be visible to both the main DHCP scope **and** the phone scope, we chose to enter this string into the global scope and not in each one. This allows us to insure that the option always presents itself the same way and that it's not a maintenance problem.

- Right-click "Server Options" and select "Configure Options..."

- Click the Advanced tab at the top
- Drop down the vendor class to select "Mitel IP Phone Options"
- Leave the User class alone
- Double-click the "Mitel Phone Config" option and enter your long magic string into the area provided
- Click [OK] to commit

At this point, the DHCP server should be prepared to offer the special Mitel options to any Mitel phone that asks for an address, no matter which subnet it asks on.

■ DHCP special cases

There are a number of ways to configure DHCP, and we chose to configure our DHCP server (which is also our domain controller) with a separate network connection directly in VLAN#4 (with an IP address of 192.168.4.7) — this allowed it to hear DHCP broadcasts directly on VLAN4 as well as allowing it to serve DNS and other requests directly without having to rely on the firewall to route to the main office subnet.

But it's possible to configure a DHCP server with a single NIC to serve up requests for multiple subnets, including those it can't talk to directly, by the use of **DHCP relay** (sometimes known as a DHCP Helper or a UDP helper).

This is configured in the firewall/router by telling it: if you hear a DHCP request on this separate subnet, forward it to the main DHCP server at this IP, and add your remote subnet IP to the packet in the Relay Address slot.

When the DHCP server gets the packet, it knows that it wasn't broadcast - it's coming from the inside address of the firewall - but it uses the DHCP relay address inside the packet to select a scope and send the reply.

~~It's a clever system that I nevertheless couldn't get to work, and since I had a NIC to spare, probably won't revisit it. But other environments probably should.~~

DHCP relay started to work when I insured that **all** phone switchports to be in trunk mode; it's not clear why this needs to be, but it made the difference for DHCP relay mode.

■ IP Phone Boot Sequence

These are the steps that the phones take to boot, and the fact that it requests a DHCP address twice was the final piece of the puzzle to allow us to understand this. It was very, very frustrating trying to figure this out.

1. The phone wakes up with no knowledge of anything, including which VLAN it's on (or *should be* on), so it puts out a DHCP request via broadcast.
2. The switch hears the request as untagged (without a VLAN tag), so it sends the request to the switch's *native VLAN*, which is VLAN#1 (the office network).
3. DHCP server hears broadcast request on the VLAN1 interface, allocates an address out of the 192.168.1.X pool. The response includes DHCP option 43 with the magic Mitel string.
4. The phone gets the IP address on the office subnet (and is unaware of which VLAN it's on), but notices via the magic Mitel string that it's suppose to be on VLAN4. Therefore, it releases the IP address it just obtained, changes its VLAN membership to #4, and goes through the process again by putting out a second DHCP request *on VLAN#4*.
5. DHCP server hears the broadcast request on the VLAN4 interface and issues an address out of the 192.168.4.X pool. The response includes the same DHCP option 43 with the magic Mitel string.
6. The phone gets an IP address on the VLAN4 subnet and inspects the magic Mitel options. Since the options are the same as last time, it figures that it's done setting up on the network so plants itself on VLAN4 with an 192.168.4.X address. **Note:** I'm just guessing on the phone's rationale for behaving this way, there are certainly finer points I'm not aware of.
7. Using information in the magic Mitel string in DHCP option 43, it (ahem) phones home to the phone switch for continued configuration.
8. The phone starts passing traffic from the PC device behind it: if it's untagged traffic, the switch sees the lack of a VLAN tag and puts it on the native VLAN 1.

As far as we know, the only way to avoid the phone making two DHCP requests per boot is to configure it with a static IP directly on VLAN4, but we preferred the features of automatic provisioning from DHCP.

Other Resources

This Tech Tips was created strictly to document on particular experience on one particular customer's network, but it seems likely that Mitel is not the only phone system that works this way. As we learn of other configurations that work in a similar way, we'll document them here.

Blogged [here](#).

First published: 2012-05-15

[More Tech Tips](#)



[Home](#) ■ [Stephen J. Friedl](#) ■ [Software Consultant](#) ■ [Orange County, CA USA](#) ■
steve@unixwiz.net ■ 