## Ethernet Duplex issues

Many people love to slam hardware vendors for compatibility issues, me included. However I've generally found that Ethernet duplex configuration is often misunderstood, and results in performance issues. I have written this paper to help others navigate their way to a better networking experience. Before I dive into the details I challenge you to answer these questions based on the diagram below.

1) Which of the following will operate end to end in full duplex mode?
2) Which of the following will drop packets due to one or more duplex mismatches?
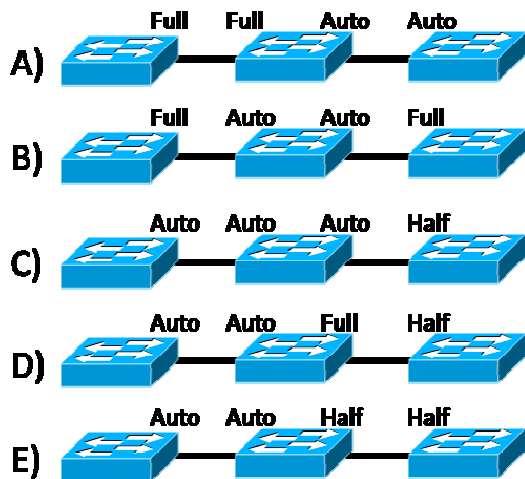
The answers come later, let's go over some of the facts and fiction.

Auto negotiation is described by 802.3u and can negotiate things like speed, duplex and flow control. We're going to try to stay focused on duplex.

What is duplex?

It describes a two way conversation. Setting the duplex mode determines if both parties are allowed to talk at the same time. If you and I are talking to each other, and either of us must always remain quiet when the other talks, that's called "**half-duplex**". In the Ethernet world, that was necessary back when coax cables were used and all parties used the same wire, and the same frequency, to talk. Now that Category 5 cable (and Cat5+, and Cat6...) has separate pairs for transmit and receive, it's possible to send and receive at the same time without confusing the signal. When traffic is able to be sent in both directions at the same time it is called "**full duplex**". Clearly running in full duplex mode allows for more data to be transferred than half duplex, so it's more desirable.

How is duplex negotiated?

10M, 100M and 1G Ethernet can negotiate the duplex mode. The key thing to understand is how it's negotiated. The protocol requires that if I'm trying to negotiate duplex with you, and you don't respond I must fall back to half-duplex mode. A device which is set to full-duplex doesn't negotiate. So, if I'm set to auto-duplex, and you are set to full-duplex, the end result will be that I will operate in half-duplex mode (because you didn't answer my negotiation attempts) and you will operate in full duplex mode. No big deal right? Wrong.

Duplex behaviour

In half-duplex mode, I have to wait for silence before speaking. Due to the speed the data travels on the wire and the fact we could both send at the same instant, and there is a period of time where we both could start sending without knowing the other end has started sending. In half duplex mode this is a "collision" and it will happen occasionally and is normal.

However if one of us is set to full-duplex (you) and the other is in hal-duplex mode (me), and we both start sending, very soon after starting to send, I will see data start to come in from you (you're in full-duplex mode) while I'm trying to finish sending. From my "half-duplex" perspective, this is expected to happen at times even though it's not desired. As soon as I see data coming in while I'm sending, I'm required as a half-duplex device to consider that a "collision" and I must abort my transmission and start a random timer before I try to resend. Half-duplex Ethernet is quite resilient, it will deal with trying to send a packet many times, each time waiting longer (on average, it's a random timer) before resending. If you are running in full duplex mode, your perspective on what's happening is a little different.

You don't care if you receive something while you're sending, so you keep sending. However you do notice that I stopped sending to you in the middle of a packet and will log errors on your receive side like "runts", "incomplete packets", and "aborts". "Late collisions" can be a sign of a duplex mismatch too (logged at my half-duplex end). The errors logged by each device are clues to duplex mismatches. Note though, that when both ends are running half-duplex, collisions are normal and expected.

When both ends are half duplex, throughput is reduced because one direction is always silent, but delivery of packets is pretty reliable. When running half duplex, the maximum throughput tends to be about 40% of the line rate. As you approach 50% utilization, you have a 50/50 chance of colliding with traffic coming the other way. This causes both ends to retransmit, thus increasing the link utilization and making it all worse. A good rule of thumb is to plan for a maximum of 40% throughput on half duplex.

When both ends are full duplex, there are never collisions, and traffic can flow in both directions at the same time, so throughput is better and packets are not typically dropped.

When one end is half-duplex (me) and the other is full-duplex (you), and the amount of traffic increases, packets will be lost. I will see lots of collisions. You will see lots of aborted packets.

Over 90% of traffic is TCP based, and that protocol deliberately slows down (windowing) when packets are lost, compounding the impact of packet loss due to duplex mismatches. In my experience, around 2% packet loss starts to become painful for the end users and the impact accelerates as the packet loss increases. Under heavy load, duplex mismatches cause much more than 2% packet loss.

"It can't be a duplex problem"

Many times I've heard people say "it can't be a duplex mismatch because it worked fine when we first installed it.". You must remember, when there is one or more collisions when a device trying to send a single packet, the protocol will retransmit the packet, and it will usually get through. From an end user perspective, the packet made it through, and the delay wouldn't be noticed in human terms. It is only after 16 collisions for that specific packet that it will actually be dropped. So there may be errors happening, but you never lose a Ping packet when you do a quick test. Most people don't check the port counters if a ping test works.

When you first turn on an Ethernet connection, it doesn't typically have a lot of traffic on it. Collisions are far less likely when neither side is saying much. Also, if you test with a file transfer, that tends to be a lot of data in one direction, but only acknowledgement packets in the other. This also doesn't lend itself to a lot of collisions. It's only when there is a lot of traffic in both directions that collisions become frequent. This isn't a huge problem if both ends are running half-duplex, because they both recognise collisions and are smart enough to back off and give the other end a chance to send. Clearly very specific testing must be done to spot a duplex mismatch when you first turn on a link.

"But my router and his router are both set to auto-duplex"

Every Ethernet device matters, not just the ones with IP addresses. Even though you ping between two IP devices, there may be several layer 2 devices in between, for example Ethernet switches, line drivers, and Sonet switches. Here is an example below.

Some people assume that the duplex settings on router A must match the settings on router F, which is the next hop in the layer 3 (IP) network. Ethernet is layer 2, and each end on each of the 5 hops shown above must match the other end of that hop.

The next layer 2 hop can be different. On hop 2, "B" must match "C", the other hops don't matter to hop 2.

Media converter, Line Driver, Repeater… ?

Now just to muddy the waters, some devices which convert between copper and fiber participate in the Ethernet, and therefore they break the link into multiple Ethernet segments as described above. This is the behaviour I've seen most often. However, I have also seen devices which simply convert the electrical signal to an optical signal and remain transparent to the devices on either side (personally I'd call this a "media converter", but see my note on names below). These devices do not break the Ethernet into two segments, and therefore can not mediate a duplex mismatch (reminder: copper GigE can only do auto-duplex, fiber GigE can do auto or fixed). The bottom line is that even though it's a relatively simple device you stick in to convert from copper to fiber, you must understand how it functions and if

- it has options for duplex ,or
- if it is transparent.

This knowledge allows you to set the neighbouring devices correctly. Don't trust the names of these devices to tell you what they do, people seem to use the terms interchangeably (media convertor, line driver, repeater, FOIRL…).

So back to the questions at the beginning.

1) Only A
2) B and D

How do I recognise duplex mismatches?

Clue number 1… performance is poor. Look further.

There are some applications on the market that allow you to identify that there may be a duplex mismatch, but they typically can only narrow it down to the granularity of an IP hop (which may contain several Ethernet hops). In a multiple hop IP network, these tools tend to get less accurate the further you get away from the test source. Problems close to the test source overshadow the results shown for hops farther away, so be careful with interpreting test results. Appcritical by Apparent Networks is one application which usually identifies duplex mismatches. www.apparentnetworks.com

One of the simplest methods is to do a multipacket ping. I don't mean ping 10 times. Most ping applications pause 1 sec between pings. This means the response to the ping is never returning at the same time as the ping request is going out. The ping request is called an "echo request", and the response is called an "echo reply". If you change your ping size to 8,000 bytes, it means the ping application must break that up into multiple packets (Ethernet typically is limited to 1500 byte packets). It is multiple packets, but counts as one ping. All packets of a ping must still be received before a response is sent, but it that still creates an environment which is more likely to cause enough collisions to lose packets. Try pinging with 8,000 byte pings.

It should be noted that some devices have a limit on how big a ping packet can be, so getting no response at all may mean you've hit the limit. Also, pings can be used in denial of service attacks, so some firewall configurations limit the ping size they permit. That will hamstring your ability to test for mismatches.

Of course looking at the interface errors on the devices is an ideal method, but you may not always have the ability to log onto a device (like a FORIL) to check statistics.

Some additional notes

Before you change anything!!!!...

As you will see below, when you correct your duplex settings, there is a very real possibility you will take the link down completely. This isn't a big deal if you are at the site where the equipment is, but if you are remotely connected to the device you are changing, you could knock your link to the site down and be unable to reach it to change the settings back, or check and change the other settings mentioned below. So here are a couple options:

- Make these changes when you are on site, or
- Before making the changes, try it in a lab, or
- Before making the changes, save the pre-change config of the device, and then schedule a reload in 10min. If the change you make causes a failure, the reload will back out the change. If the change works, cancel the reload.

Speed and duplex linked?

On some devices the configuration of duplex is linked to the configuration of speed (i.e. setting auto-speed may also set auto-duplex), this can have a nasty effect if you were not aware of it. For example on a Cisco Catalyst switches running **CatOS**

"use **set port speed 1/1 auto** command to configure auto-negotiation for both speed **and duplex** mode... There is no **set port duplex {***mod_num/port_num***} auto** command"

1G 802.3z does support half duplex, but Cisco doesn't. Cisco supports auto negotiation on GigE only to inform the other end to use full duplex. Cisco GigE based ports that use an SFP or GBIC do not negotiate (source: Cisco tech note, 2009, *Configuring and Troubleshooting Ethernet 10/100/1000Mb Half/Full Duplex Auto-Negotiation*.)

On Cisco Catalyst switches running **IOS**, there are separate commands for speed and duplex, but they are not always completely independent, for example...

```
NativeIOS(config-if)#duplex full
Duplex will not be set until speed is set to non-auto value
!--- Error: On this platform, you must set the speed before
the duplex.
!--- Not all switch platforms have this command ordering
requirement.
NativeIOS(config-if)#speed 100
NativeIOS(config-if)#duplex full
```

If the link goes down when you change to a fixed duplex setting, it could be because the speed is no longer negotiated. I've seen this on Cisco 2800 series routers. The carrier circuit was set to 10M Full Duplex, and the 2800 was set to auto speed, auto duplex. They had performance issues. When the duplex was changed to full duplex, the link failed. The 2800 (without being asked to) also switched the speed to it's default (100M). Changing the port to 10M brought the link up and resolved the initial packet loss problem by allowing non-negotiated full duplex.

Auto crossover (MDI/MDIX)

Many Ethernet switches (typically not routers) have the ability to sense whether a port is connected to a workstation or to another switch. They can then change which cable pair they use for transmit and receive. This prevents the need to install crossover cables. However, in some Ethernet switches, the automatic features are either "all on", or "all off". So if you switch to manually setting duplex, and the link fails between two devices (most commonly one end is a switch), you may need to replace the straight through cable with a crossover cable. (Another reason to schedule a reload before making a change)

Is GigE different?

Copper based GigE (1000BASE-T) is different than fiber based GigE (1000BASE-X). Copper GigE specifies that it <u>must</u> operate in Auto mode. "Duplex configuration during 1000BASE-X (fiber based GigE) operation can be handled either through Auto-Negotiation <u>or through manual selection</u>" *source Interpretation Number: 2-07/05 IEEE802.3af-2003interp-6[1]*

802.3z does not specifically define a way to turn Autonegotiation off, for both 1GigabitEthernet and 10GigabitEthernet. This is likely because autonegotiation of a timing source is mandatory. While negotiation support is clearly a requirement, it's not clear to me whether autonegotiation of the speed and duplex options are mandatory for GigE and 10GigE. The Cisco tech note quoted above (*Cisco tech note, 2009, Configuring and Troubleshooting Ethernet 10/100/1000Mb Half/Full Duplex Auto-Negotiation*) seems to indicate they don't negotiate. At this point GigE negotiation behaviour and vendor conformance to standards are things I don't have enough real life data on. (input gratefully accepted)

What other devices should I be concerned about?

People typically recognise right away that Ethernet switches and routers are the devices which they must configure for duplex. The following are other devices that must be considered.

Fiber Optic Inline Repeater Link (**FOIRL**) devices are also key to duplex issues. A FOIRL is often called a "media convertor" because it typically converts between copper and fiber. Many of these devices are simple and either have only one way they work for duplex, or have DIP switches to configure them. You can not ignore them. They must match duplex settings of the device on the other end of the wire or fiber.

SONET mulitplexors (muxes) with Ethernet cards must also be considered when looking at duplex issues. I've seen one where it was not programmable, it only worked one way.

PC Duplex is not typically an issue. Vendors almost exclusively default the Ethernet cards to auto negotiate. However I did have one issue when using a laptop to troubleshoot where I don't believe it switched to full duplex mode as requested. Unfortunately, I wasn't the one running the PC and wasn't on site so I was never able to validate the report.

What if the device options are limited and they can't match?

On occasion, when two pieces of equipment can never be made to match their duplex settings (i.e. neither end is configurable, and one end is full-duplex and the other end is auto-duplex). I've had to install an intermediate switch between the two devices so that the intermediate switch can be configured to match with both existing devices. It has always made a large improvement to performance.

Misc.

802.3X describes both duplex and flow control, and DIX framing (aka Ethernet II)

Summary

The moral of the story, is that both ends of the Ethernet cable need to be set the same (Auto and Auto, or, Full and Full). Auto-Duplex at one end and Full-Duplex at the other will cause the Auto end to fall back to half-duplex, and you REALLY don't want that. When "Auto" end falls back to half-duplex (and there is a mis-match) it will see and count collisions, and abort sending whenever it starts receiving. The Full-Duplex end will not see collisions, but will see and count aborted frames from the other end.

Making the changes to correct duplex mismatches could cause changes to other settings like speed and crossover (MDI/MDIX), so care must be taken when implementing the changes.

I welcome additional information, clarifications or corrections.

My Business email: glen.govier@bell.ca

My Residential email: ggovier@shaw.ca