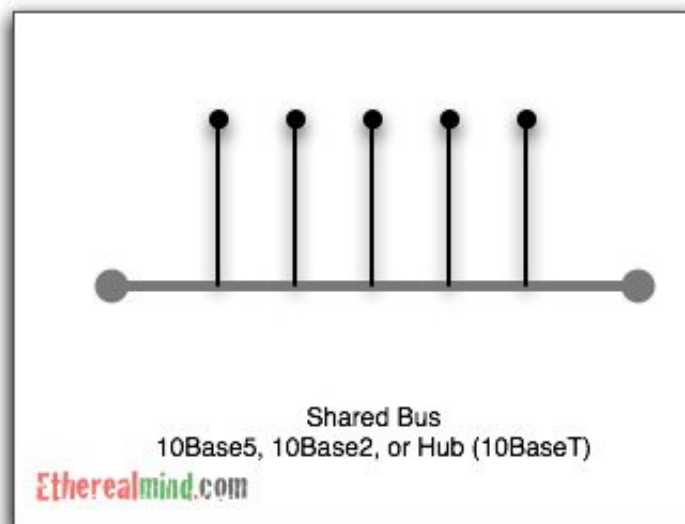


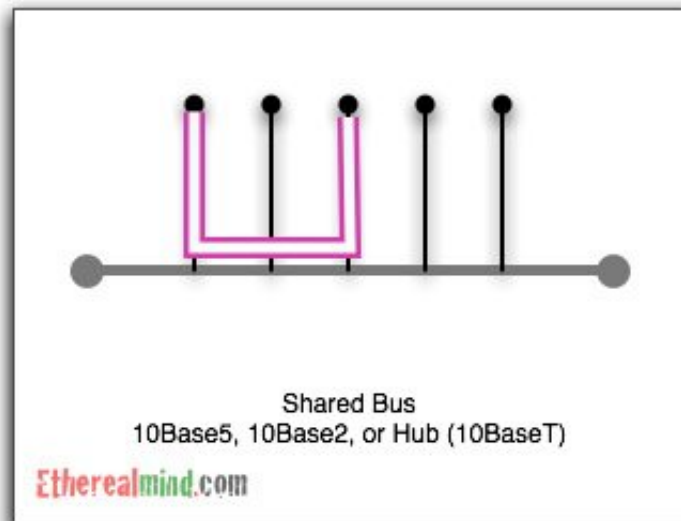
The marketing people in IT tend to be overwhelmed by complexity and deep technology. For many liberal arts graduates, they take the drowning option and latch onto certain terms and then grossly abuse it. The most egregious abuse today is “cloud” but “fabric” comes a close second. In this series of posts I want to look at what **is** a *FABRIC* and provide a canonical look at what it does and how it works for us.

Some History

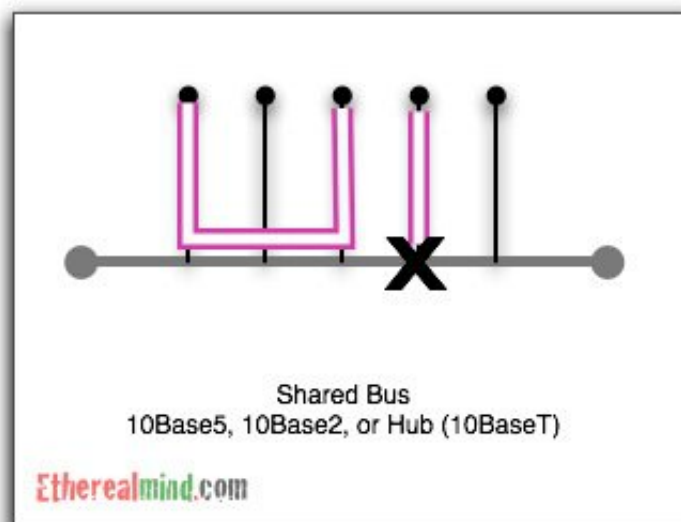
While I sometimes indulge in nerd history for fun, we need to look carefully at the way Ethernet developed before moving into more complex purposes. The first versions of Ethernet used a shared bus technology



where all connected systems would simply transmit onto the wire:



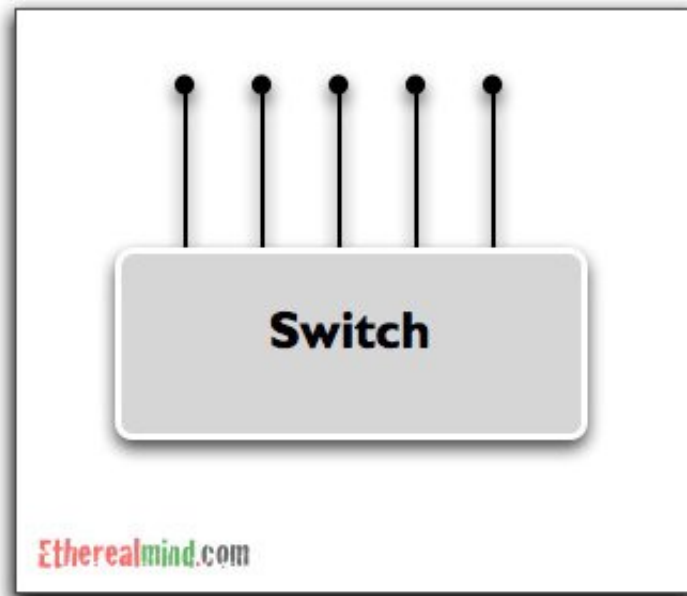
Of course, if two systems attempted to transmit at the same time, then frame loss occurred



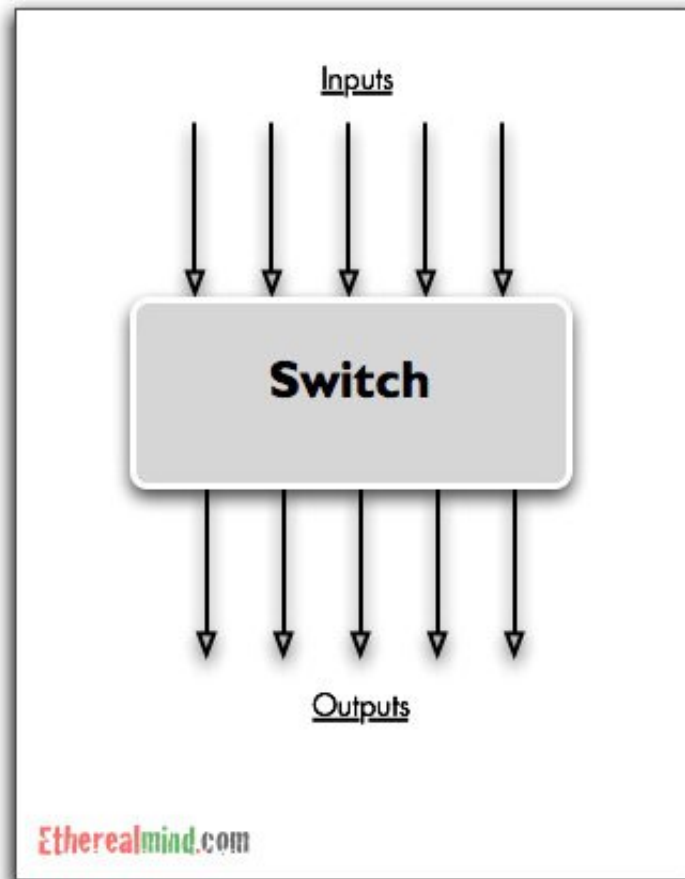
This was solved by defining CSMA/CD (Collision Sense Multiple Access / Carrier Detect) and the Ethernet card would check that the bus was clear before transmission and transmit if there was no carrier. In the event that another system transmitted while the Ethernet frame was spooling onto the wire, the collision sense would detect this and it would back off for a pseudo-random interval and retry. In practice, for low levels of utilisation (less than 60%) this worked OK, but as peak utilisation increases this solution becomes a deadlock and few frames are successful.

Enter Switching

In order to reduce the size of collision domains and thus reduce the number of collisions, Ethernet bridges reduced the number of systems per segment. This later developed into switches (also more correctly known as multi-port bridges) and each connection is a single “bus”, like so:

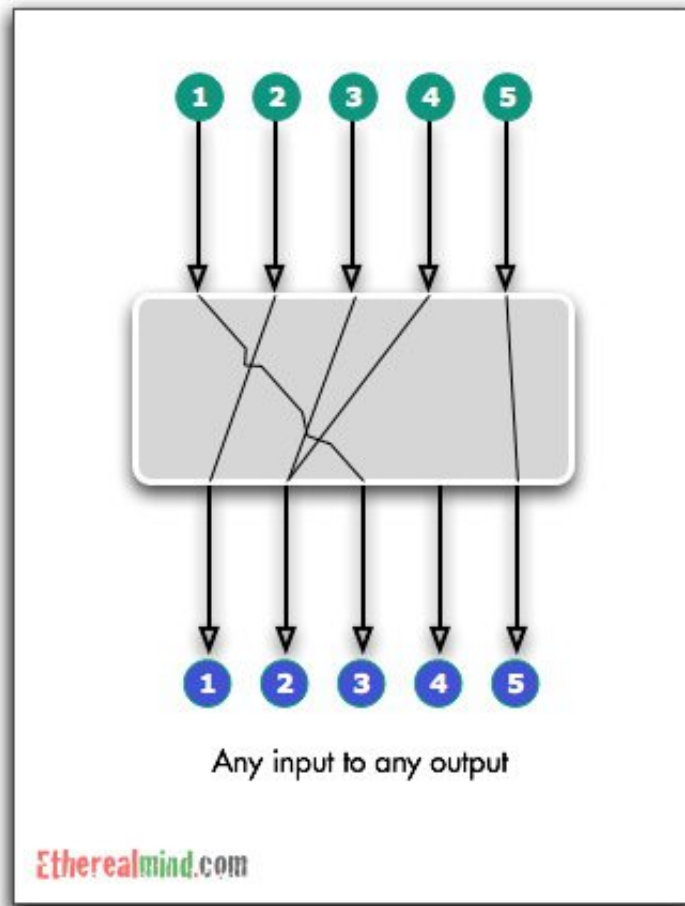


This led to the development of full-duplex Ethernet – because only a host and switch are connected, there is no change of sharing the “bus”. So full-duplex Ethernet meant that the switch becomes an input *and* output device.

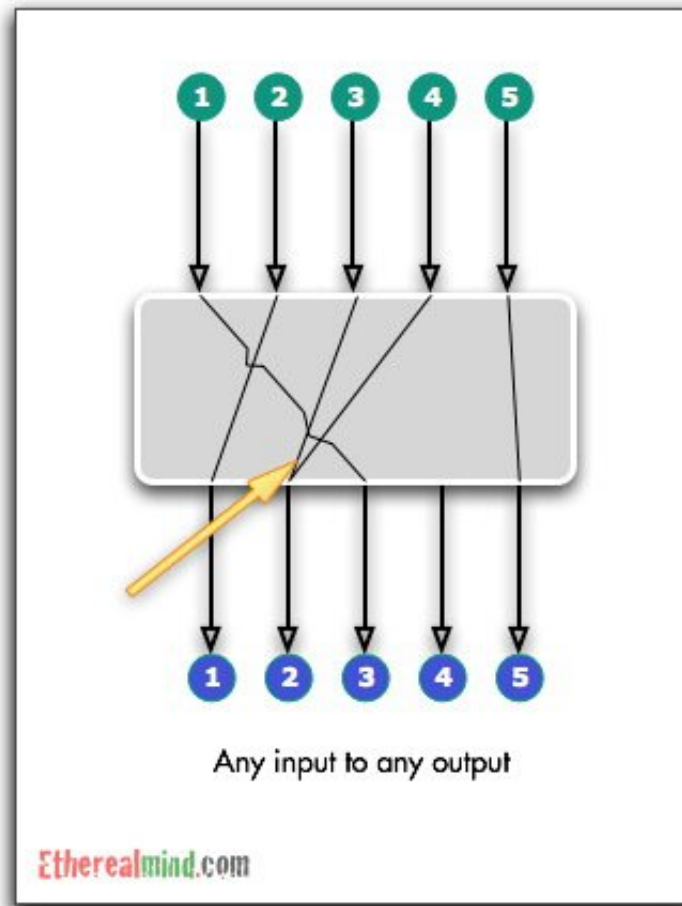


Any to Any Switching

The challenge now becomes that any input needs to forward frames to any output and there is no collision detection. This “any to any” function is a complex function to achieve:



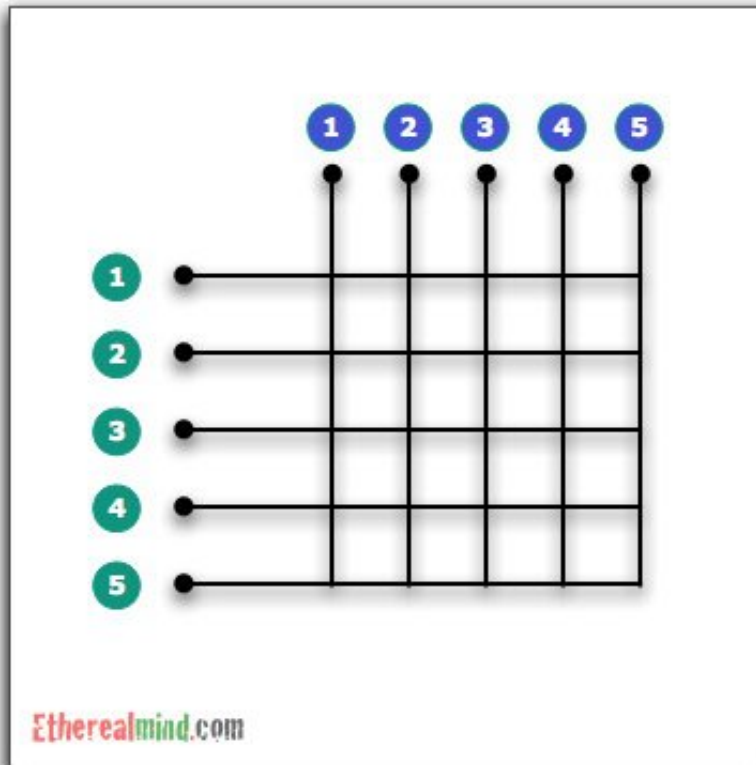
For example, if port 1 and 4 attempted to transmit to Port 2 at the same time (because a server is connected there) a collision would occur.



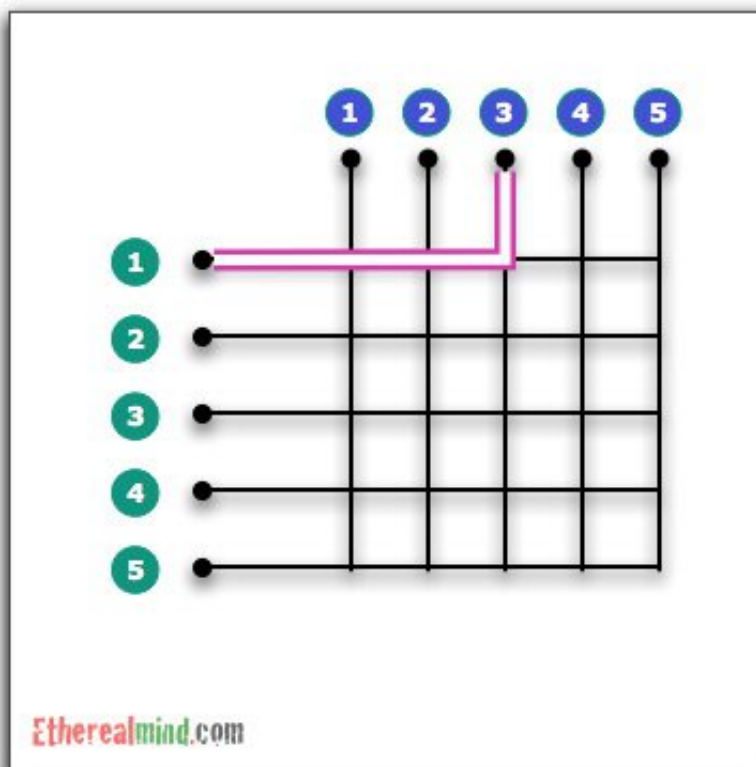
If you realise that this looks like Time Division Multiplex circuit switching – you would be correct. The crossbar theory is derived directly from telephony, and reflects the nature of the flow. That is, a frame has a source and a destination

Enter the CrossBar Switch

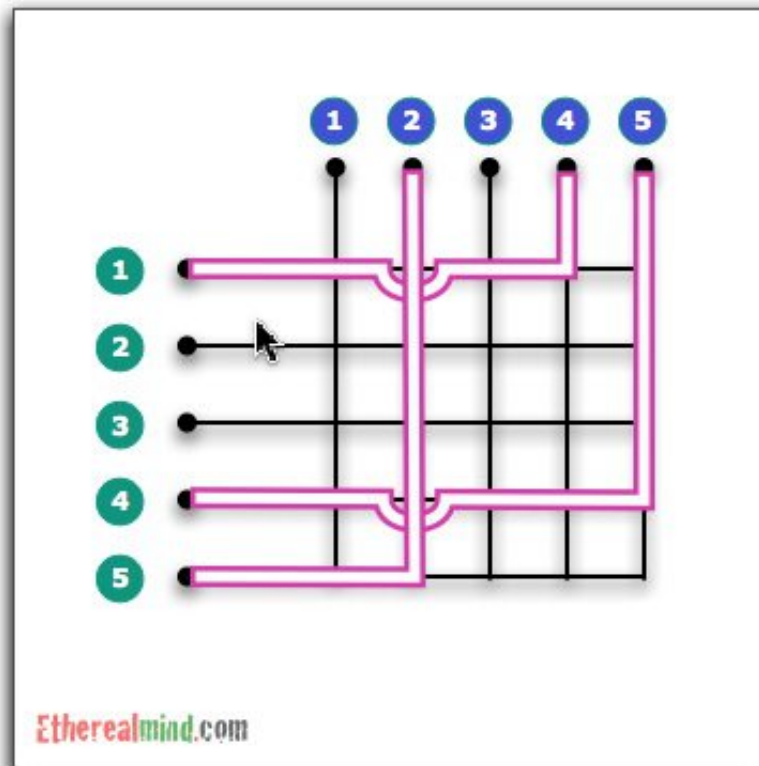
The Networking industry then looked to telephone exchanges of the era which had exactly the same challenge of switching TDM circuits and copied the [Crossbar switch](#) theory but developed it in silicon form. The basic purpose is to switch any input to any output.



Lets consider Input 1 sending a frame to Output 3:



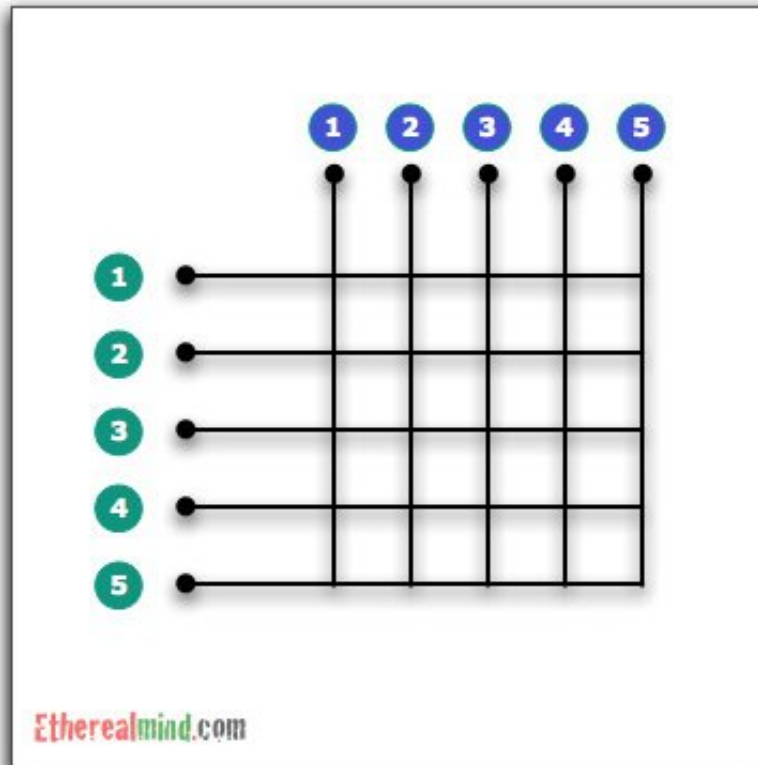
Lets consider multiple inputs forwarding to multiple outputs:



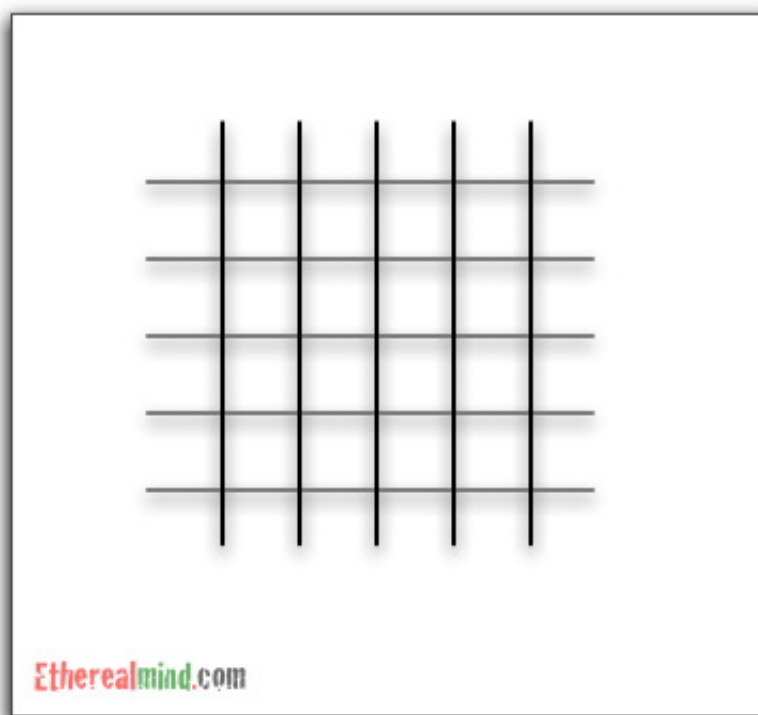
It's worth remembering that these forwarding events **occur at the same time**. At a given time interval, this diagram shows that three frames are concurrently being forwarded in the same time slice.

Where does the term Fabric come from ?

The diagram for a crossbar switch is correctly shown as this:



But most commonly, the marketing people find the lack of symmetry displeasing and aesthetically insensitive. As a result most PowerPoint waterboarding of the 1999/2000 era showed the crossbar as this:



And this looks like a woven piece of fabric.



Hence the “crossbar silicon switch chip” became known as the “silicon fabric” and later was further shortened to just “fabric” out of laziness and incomprehension.

The EtherealMind View

Now that we have established what a Silicon Fabric is, I’ll extend into a series of posts that look into some of the more advanced features of switch fabrics, and other more technical aspects that affect your day to day working.

Other Posts in A Series On The Same Topic

1. [© What's Happening Inside an Ethernet Switch ? \(Or Network Switches for Virtualization People \)](#) (11th January 2013)
2. [Tech Notes: Juniper QFabric - A Perspective on Scaling Up](#) (14th February 2012)
3. [Switch Fabrics: Input and Output Queues and Buffers for a Switch Fabric](#) (6th September 2011)
4. [Switch Fabrics: Fabric Arbitration and Buffers](#) (22nd August 2011)
5. [What is an Ethernet Fabric ?](#) (21st July 2011)
6. [What is the Definition of a Switch Fabric ?](#) (30th June 2011)
7. [Juniper QFabric - My Speculations](#) (1st June 2011)

Related Posts & Articles

- [Switch Fabrics: Fabric Arbitration and Buffers](#)
- [Switch Fabrics: Input and Output Queues and Buffers for a...](#)
- [What Is an Ethernet Fabric ?](#)
- [© What's Happening Inside an Ethernet Switch ? \(or...](#)
- [Defining Flow Forwarding Instead of Switch or Routing](#)