Audinate

Kieran Walsh – Director of
Application Engineering

Beyond Certification

Dante AVNW Amsterdam 2018

What have networks ever done for us?

IP Networks

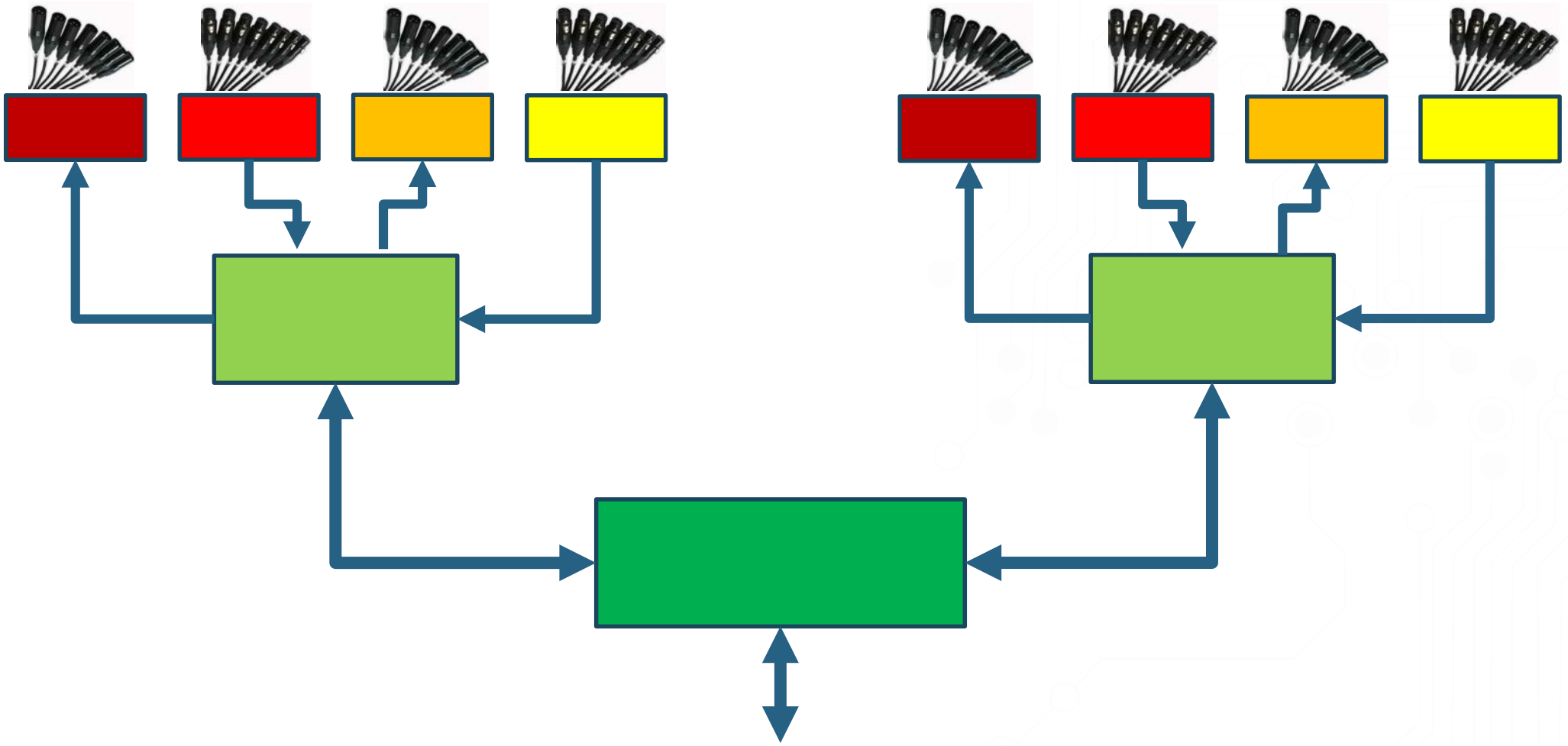
- Networks exist to join devices together
  - Email
  - Web Browsing
  - Voice and Video calling/conferencing
  - Audio
  - Video

- Why IP networks?
  - Global Scale
  - Most ubiquitous method for connection
  - Most supported method

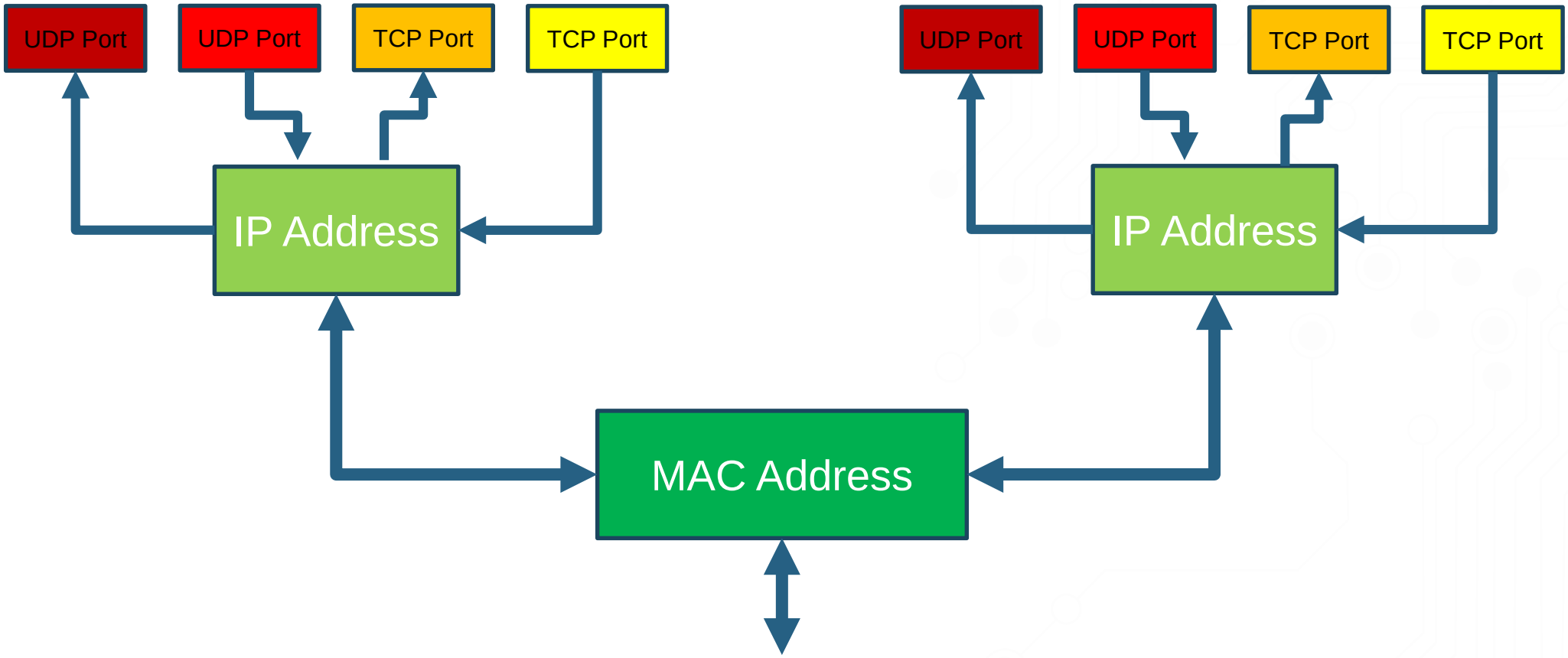
IP Networks

- What makes a network useable?
  - Names with meaning
  - Automated configuration
  - User shouldn’t worry about different services
  - It needs to “just work”
- What makes networking seem “difficult”
  - “rules” that attempt to “oversimplify”
  - Fear of “complexity”
  - Very wide subject
  - Seemingly fast-evolving

Does this scare you?

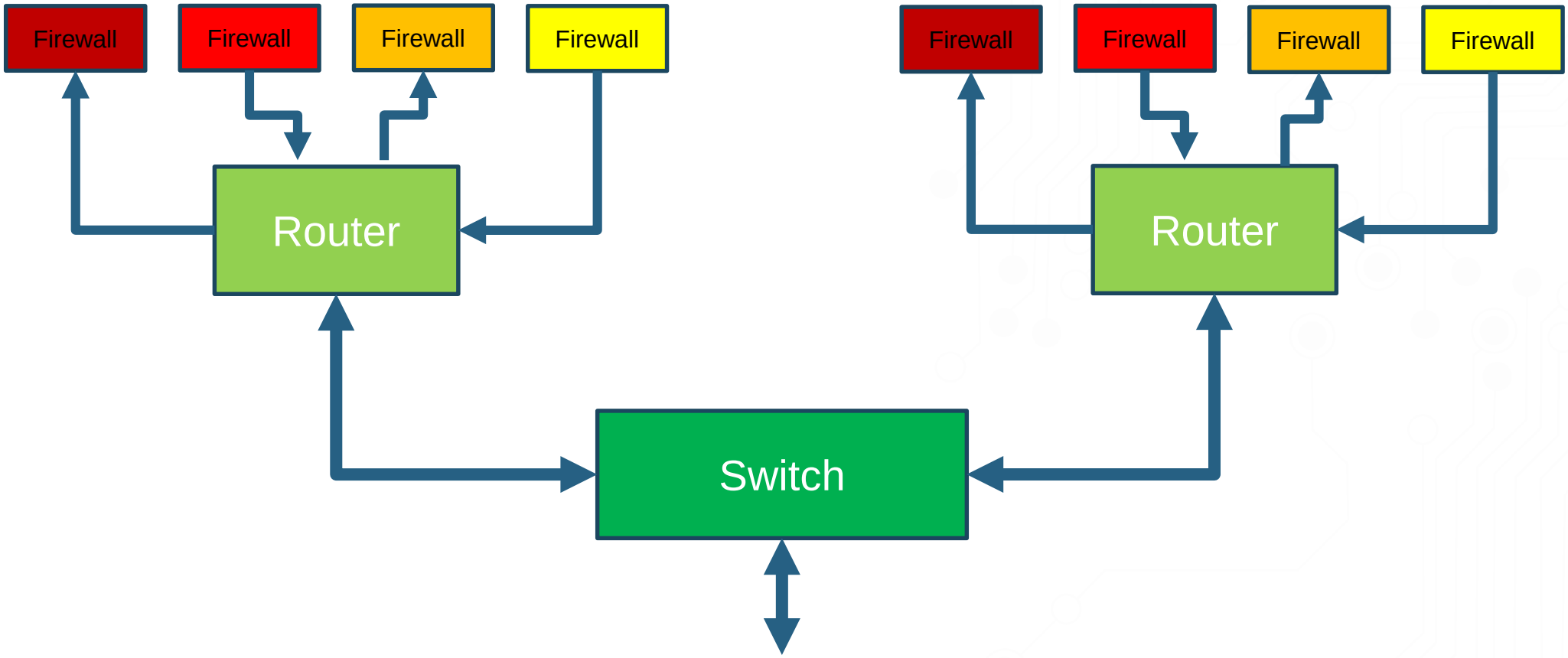


So why does this?



AV on an IP network

OR this?



Maybe its folks talking about QoS?

Or Multicast/Broadcast Management?

<< very nasty noise audio goes here >>

Maybe it was “easier” with Analogue?

Take the weight off your feet…

How do we make Networking Difficult?

- The way we acquire initial knowledge is key to comfort
  - Some console manufacturers have mute buttons
  - Some console manufacturers have an “on” button
  - Some consoles (helpfully) label a pre-fade aux as “monitor”
  - Other consoles have a pot with the legend “monitor” for a local loudspeaker output
- Interpret terminology based upon manufacturers context
- No Difference when it comes to networking

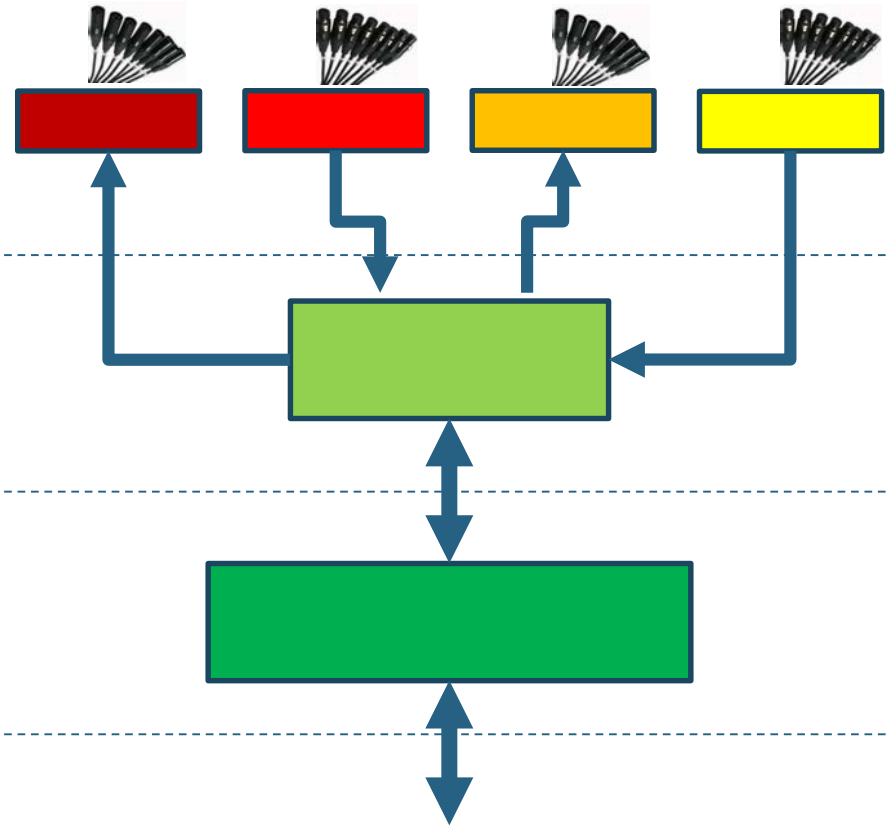
Jargon

Audio	Networks
Aux – Is it Pre or Post Fade?	VLAN – Port-based or dot1q?
XLR – Pin 2 or Pin 3 hot?	Trunk – Cisco or HP?
Three Phase – Star or Delta?	Switching – Layer 2 or Layer 3
Snake / Multicore	Socket / Port
Stagebox – remote head amps or Splits?	Brigde or Switch?
Isolation – Transformer or Buffer	Interface – Switchport or VLAN

IP Networking Fundamentals

First steps

- Layered models underpin Networks



7 – Application

6 – Presentation

5 – Session

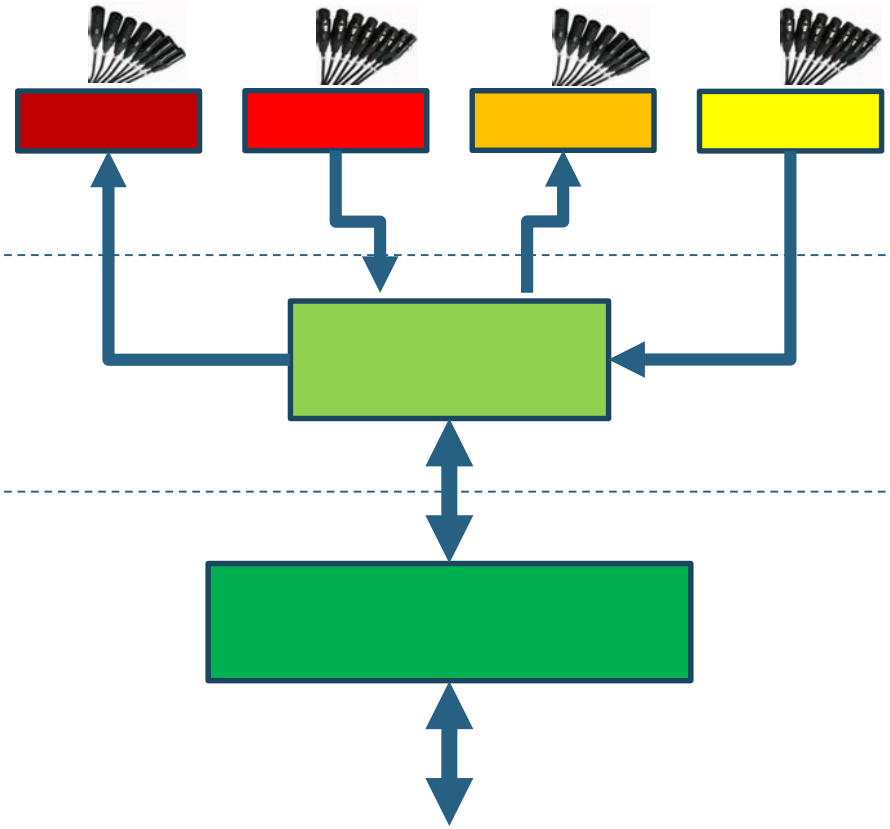
4 – Transport

3 – Network

2 – Datalink

1 – Physical

- Layered models underpin Networks



Application

Transport

Internet Layer

Network Access

First steps

- Layered models underpin Networks
  - Two Models (because one would be too easy)
    - OSI – 7 Layer model
    - TCP/IP or DoD or Internet (its got 3 names!) – 4 layer model
  - Why have 2?
    - Because neither 100% match what is really going on
  - So why bother ?
    - Useful for design
    - A good starting point for troubleshooting
    - Helps build a real implementation

So what is really going on?

Network Connections

- Lets start with something familiar
- Following our network model
  - What is the application?
  - At the “stage end”
    - Kick Drum
    - Snare Top
    - HH
    - Tom 1
    - ……
    - CD Right
  - At FOH – it’s the channel number

WEST COAST BAND

Patch list

Revised 19/06/2007
Tech. David Debris : 06.15.89.87.76

Ch	Instrument	Micro	Stand	48 V	Inserts
1	Kick Drum	Beta52 ou SM91	Small	X	Gate
2	Snare Top	E 604	Pince		Compressor
3	HH	SM 81	Small	X	
4	Tom 1	E 604	Pince		Gate
5	Tom 2	E 604	Pince		Gate
6	Tom 3	E 604	Pince		Gate
7	Floor Tom	E 604	Pince		Gate
8	Over Head left	SM 81	Large	X	
9	Over Head right	SM 81	Large	X	
10	Bass	DI		X	Compressor
11	Bass	E 421	Small		Compressor
12	Electric guitar	E 609			
13	Electric guitar Pat	E 609			
14	Accoustic Guitar 1	DI		X	
15	Accoustic Guitar Pat	DI		X	
16	Accoustic 12 strings	DI		X	
17	Accoustic Guitar 3	DI		X	
18	Keyboard 1 left	DI		X	
19	Keyboard 1 right	DI		X	
20	Keyboard 2 left	DI		X	
21	Keyboard 2 right	DI		X	
22	Vocal Guitar	Beta 58A	Large		Compressor
23	Vocal Dolly	Beta 58A	Large		Compressor
24	Vocal Pat	Beta 87A	Large	X	Compressor
25	Vocal Accoustic Gtr	Beta 58A	Large		Compressor
26	Reverb 1 Left	Type SPX 990			
27	Reverb 1 Right	Type SPX 990			
28	Reverb 2 Left	Type SPX 990			
29	Reverb 2 Right	Type SPX 990			
30	Delay mono	Type SPX 990			TAP TEMPO
31	CD Left				
32	CD right				

Network Connections

- Lets start with something familiar
- Following our network model
- Key to understanding
- Does the stage tech care which lines Are used on house multi?
- Does the FOH engineer care either?
- NO!

WEST COAST BAND

Patch list

Revised 19/06/2007
Tech. David Debris : 06.15.89.87.76

Ch	Instrument	Micro	Stand	48 V	Inserts
1	Kick Drum	Beta52 ou SM91	Small	X	Gate
2	Snare Top	E 604	Pince		Compressor
3	HH	SM 81	Small	X	
4	Tom 1	E 604	Pince		Gate
5	Tom 2	E 604	Pince		Gate
6	Tom 3	E 604	Pince		Gate
7	Floor Tom	E 604	Pince		Gate
8	Over Head left	SM 81	Large	X	
9	Over Head right	SM 81	Large	X	
10	Bass	DI		X	Compressor
11	Bass	E 421	Small		Compressor
12	Electric guitar	E 609			
13	Electric guitar Pat	E 609			
14	Accoustic Guitar 1	DI		X	
15	Accoustic Guitar Pat	DI		X	
16	Accoustic 12 strings	DI		X	
17	Accoustic Guitar 3	DI		X	
18	Keyboard 1 left	DI		X	
19	Keyboard 1 right	DI		X	
20	Keyboard 2 left	DI		X	
21	Keyboard 2 right	DI		X	
22	Vocal Guitar	Beta 58A	Large		Compressor
23	Vocal Dolly	Beta 58A	Large		Compressor
24	Vocal Pat	Beta 87A	Large	X	Compressor
25	Vocal Accoustic Gtr	Beta 58A	Large		Compressor
26	Reverb 1 Left	Type SPX 990			
27	Reverb 1 Right	Type SPX 990			
28	Reverb 2 Left	Type SPX 990			
29	Reverb 2 Right	Type SPX 990			
30	Delay mono	Type SPX 990			TAP TEMPO
31	CD Left				
32	CD right				

Network Connections

- Lets start with something familiar
- Following our network model
  - What is our transport protocol ?
    - Analogue “purists” would divide line and Mic level signals into different multis ;)
- In networking we care about data size
- This is how we determine our transport

WEST COAST BAND

Patch list

Revised 19/06/2007
Tech. David Debris : 06.15.89.87.76

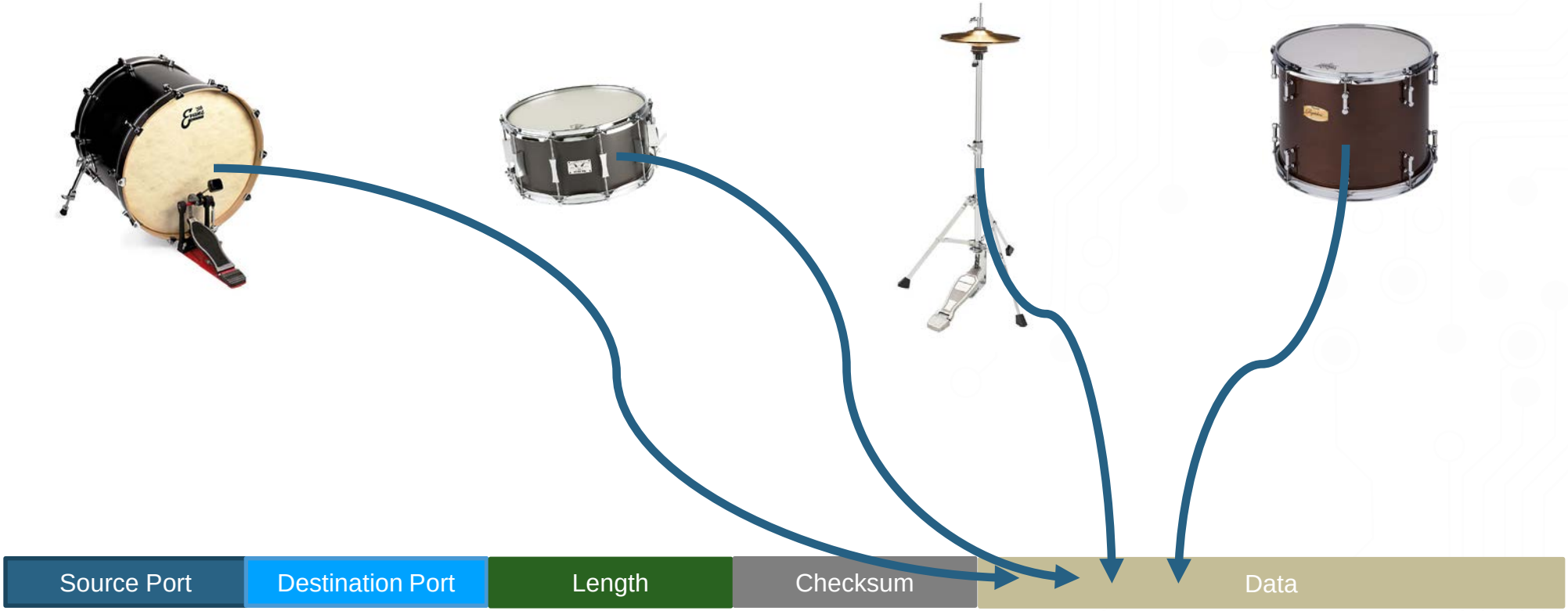
Ch	Instrument	Micro	Stand	48 V	Inserts
1	Kick Drum	Beta52 ou SM91	Small	X	Gate
2	Snare Top	E 604	Pince		Compressor
3	HH	SM 81	Small	X	
4	Tom 1	E 604	Pince		Gate
5	Tom 2	E 604	Pince		Gate
6	Tom 3	E 604	Pince		Gate
7	Floor Tom	E 604	Pince		Gate
8	Over Head left	SM 81	Large	X	
9	Over Head right	SM 81	Large	X	
10	Bass	DI		X	Compressor
11	Bass	E 421	Small		Compressor
12	Electric guitar	E 609			
13	Electric guitar Pat	E 609			
14	Accoustic Guitar 1	DI		X	
15	Accoustic Guitar Pat	DI		X	
16	Accoustic 12 strings	DI		X	
17	Accoustic Guitar 3	DI		X	
18	Keyboard 1 left	DI		X	
19	Keyboard 1 right	DI		X	
20	Keyboard 2 left	DI		X	
21	Keyboard 2 right	DI		X	
22	Vocal Guitar	Beta 58A	Large		Compressor
23	Vocal Dolly	Beta 58A	Large		Compressor
24	Vocal Pat	Beta 87A	Large	X	Compressor
25	Vocal Accoustic Gtr	Beta 58A	Large		Compressor
26	Reverb 1 Left	Type SPX 990			
27	Reverb 1 Right	Type SPX 990			
28	Reverb 2 Left	Type SPX 990			
29	Reverb 2 Right	Type SPX 990			
30	Delay mono	Type SPX 990			TAP TEMPO
31	CD Left				
32	CD right				

Network Communications

- Networks “encapsulate” data
- This is a familiar idea (just with a different name!)

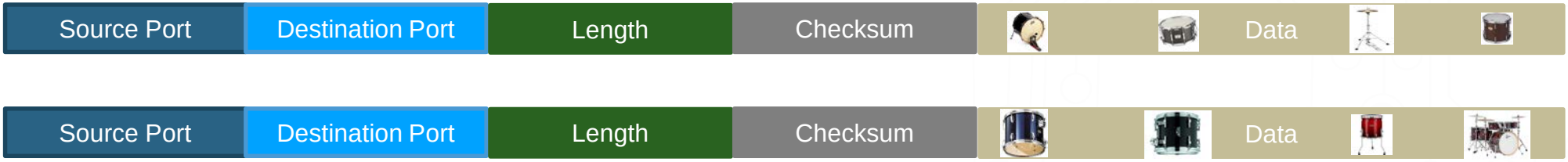
Network Communications

- Networks “encapsulate” data
- This is a familiar idea (just with a different name!)



Network Communications

- Dante Encapsulates up to 4 channels in a unicast flow
- But my box has more than 4 channels!
- Easy – create another flow!



TIP!

For efficiency – Dante Multiplexes channels into a single flow
Think – 4 channels only use one core on the “multi”

The number of cores available on a device is the number of Flows the device supports

Network Communications

- Dante uses UDP at the transport layer
- We care about data size to select transport protocol
- Why?

It’s the Ethernet Cops!

Network Communications

You are in breach of IETF RFC 894, and consequently IEEE 802.3 – your data payload exceeds the Maximum Transmission Unit (1518 bytes)

Unlike human cops – the Ethernet Cops are really mean

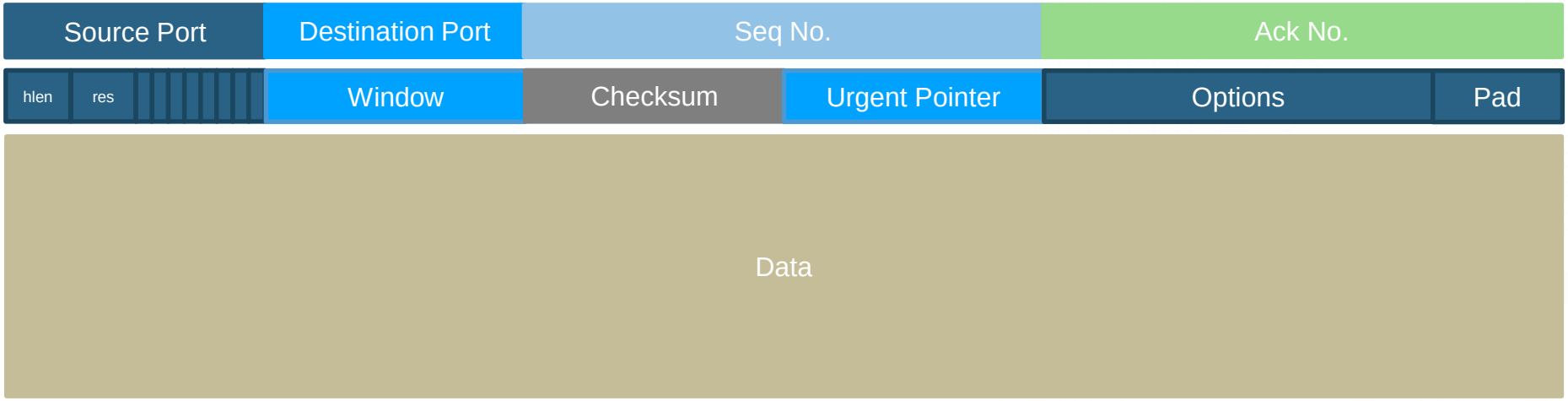
No Fines
No Warnings
Just…

Network Communications

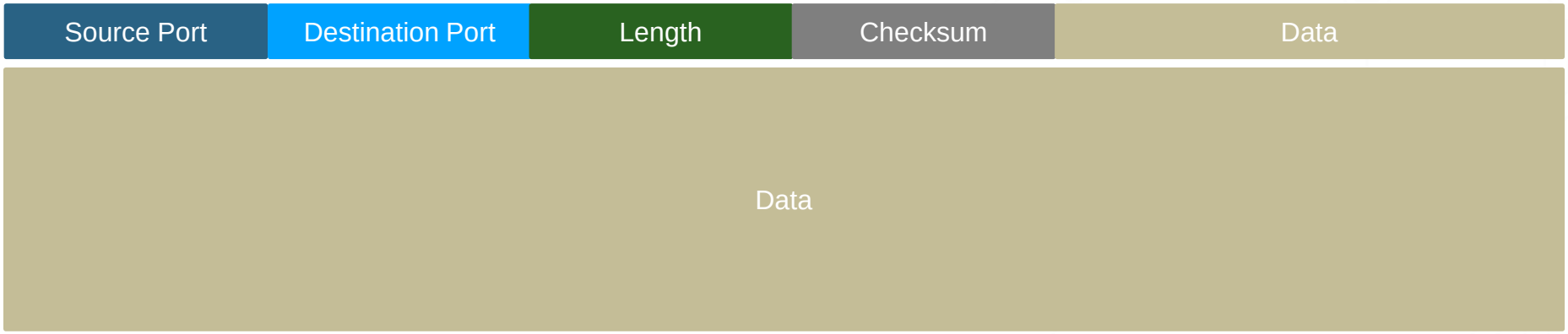
Network Communications

- Getting around the Ethernet cops
  - Same way as with anything larger than “maximum” size
  - Use more “trucks”
- Simple UDP doesn’t work so well for this
- TCP can deal with it
  - Higher “overhead”
  - Possibly slower
- TCP uses session layer for segmentation

TCP header



UDP Header



Network Communication

- Myths concerning TCP and UDP
  - TCP was called a “reliable” protocol
  - UDP was called an “unreliable” protocol
  - This was caused by early worries about UDP
- The Truth
  - TCP is a “connection oriented protocol”
  - UDP is a “connectionless protocol”
- Both Protocols are useful
  - Sports cars are good
  - Tractors are good
  - Lamborghini make both ;)

Network Communication

- Why is “reliability” a “myth”
  - TCP uses sequence numbers to ensure all information is delivered as part of the transport header
  - Nothing is stopping a system using UDP from doing this “inside” the packet
  - Flagging “errors” “inside” a system using UDP means that a “stateful paradigm” could be used (only errors are reported)
  - TCP reports EVERY single received packet back to the sender (normally a success) – arguably from a “reliability” perspective “excessive overhead”

Network Communication

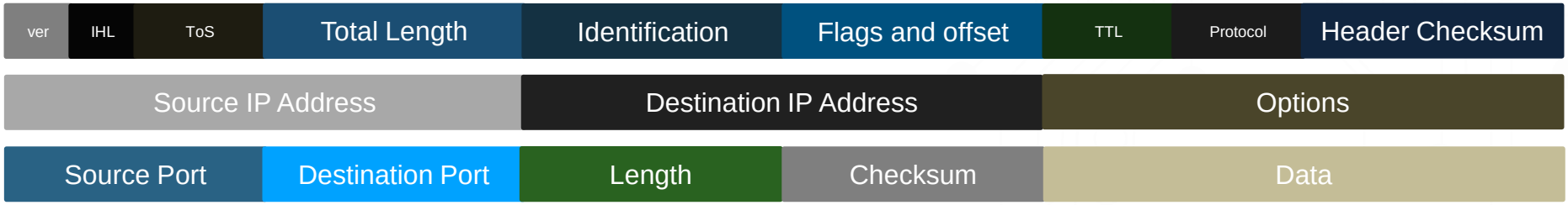
- Where to use TCP
  - When the data is a known size, and is bigger than MTU
    - EG video file, audio file
  - When the data is not “time critical”
  - When the data is likely to pass along different routes through the network on a per packet basis (sequence numbers are good for reassembly)
- Where TCP breaks
  - When the data is an unknown size
    - EG an audio or video stream
  - Where low packet jitter is required (ack increases this)

Network Communication

- Where to use UDP
  - Where the data to be carried is smaller than MTU
  - Where the amount of data to be sent is unknown
    - Eg a stream that has to stay “up” 24/7/365
  - Where packet jitter is to be minimized
- Where UDP breaks
  - (potentially) more work for application software engineers
  - Extremely hostile multi administrator internets (eg the Public Internet… but by no-means very often)

Network Communication

- The next Layer of encapsulation is to create an IP packet
  - We take our TCP “segment” or our UDP “datagram”
  - and put it inside an IP packet



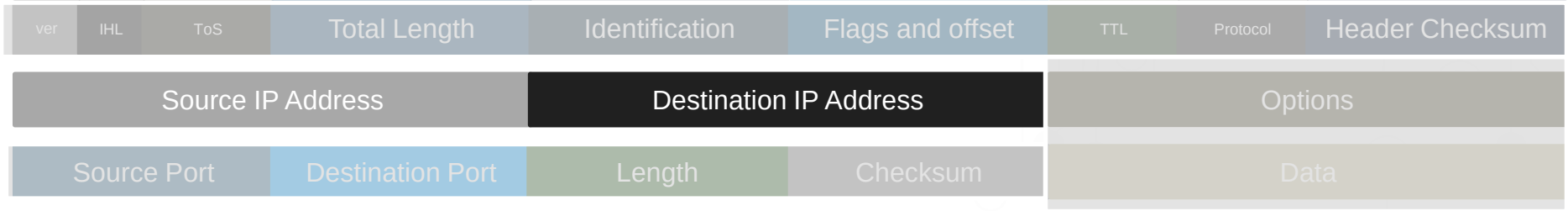
*IPv4 Packet (IPv6 header is bigger)

- This is where IP addresses, and IP subnets become involved
- In our analogue analogy we are still inside the sub-stage box

IP Addresses

IP addresses

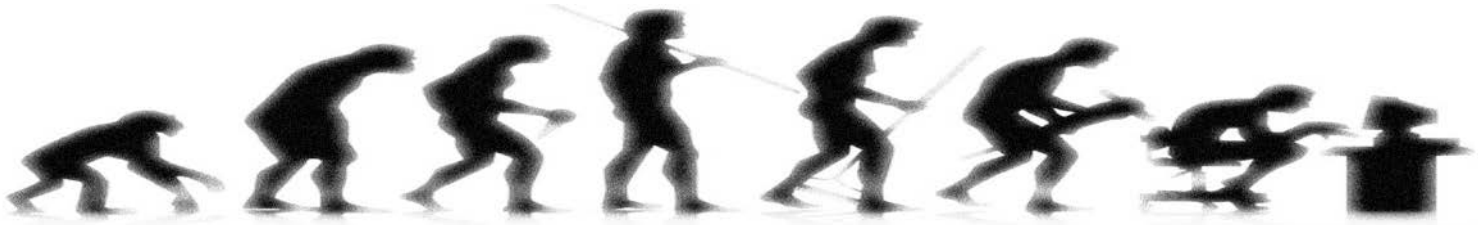
- Static IP addresses / DHCP Served IP addresses / Link-Local ?
  - Remember the “On” button vs the “Mute” button?
    - Lets look at what is really changed from a network traffic Perspective



    - Changing the IP address ONLY changes this bit!
    - So why is this a popular conversation?

IP addresses

- Similar to so much in IT/Networking and Audio its all about
  - History
  - 2 main Operating system families today
  - Unix
    - Linux OS family
    - Apple Mac OS (above version 10)
  - Microsoft Windows family



IP addresses

- Unix based Operating Systems
  - Started in mid 1960's as experimental OS
    - MIT, AT&T, Bell and GE built Multics
    - Designed for Mainframe computing
  - Linux arrived in 1991
  - Mac Darwin Kernel 1997
- Windows based Operating Systems
  - Windows version 1.0 released 20th November 1985
  - Oriented to desktop computing
  - Separate development track (Windows NT) for Servers

IP addresses

- So what have Operating Systems got to do with this?
  - Most common way to access the Application Layer
  - Network connected applications run on an OS
- Key point
  - Mainframe computing requires tight network integration
  - Desktop computing doesn't… (well… didn't)
- Method for communication between processes in Unix is the same as method to communicate between devices (crucial for mainframe)
- Network integration in Windows was an extension (Windows 3.11)

IP addresses

- Systems admins job is to “make it work”
  - Used to be easy to spot the difference between Unix and Windows admins (working methods, and attitudes)
- OS integration of networking today
  - Very mature on all platforms
  - More difficult to spot the difference more recently
- Result
  - IP addressing is just not such a big thing anymore
  - Knowledge takes time to filter down

IP addresses

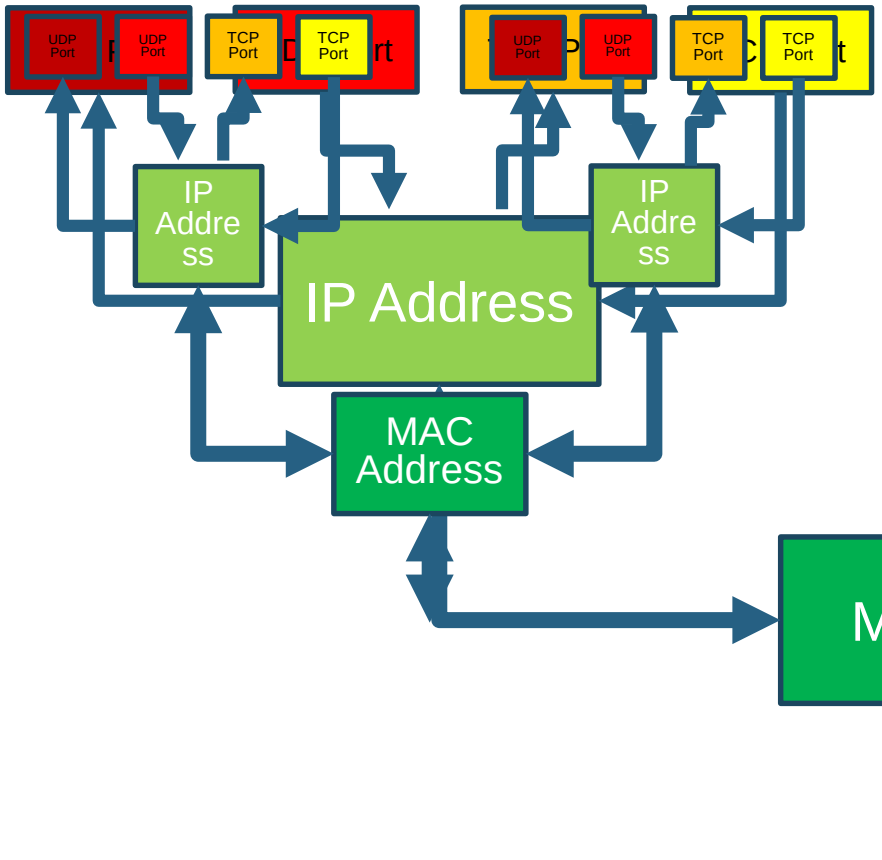
- It wasn’t just Operating Systems that evolved
- To add complexity to the working system
  - Network technologies evolve
  - Network standards follow (and evolve)
  - Network applications evolve
- Not too different to audio
  - Mixing consoles evolved
  - Amplifiers evolved
  - Loudspeakers evolved

What is an IP address

- An IP address (should be) an unique address for a network stack on a host
  - It allows the possibility to communicate with any other host connected to the same infrastructure
  - It can also refer to a group of hosts a “network” address
- Why “should be”?
  - IPv4 lacks a large enough address space to cater for the Public Internet
  - It still works because some clever workarounds were created
  - IPv6 addresses that problem (spectacularly)

What is an IP address

Lets Revisit this…



Note carefully:

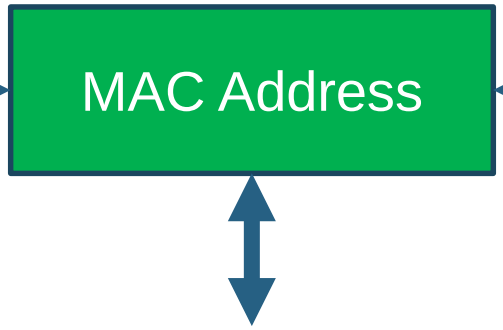
We have 2 IP addresses “sharing” the same MAC address (this is very common)

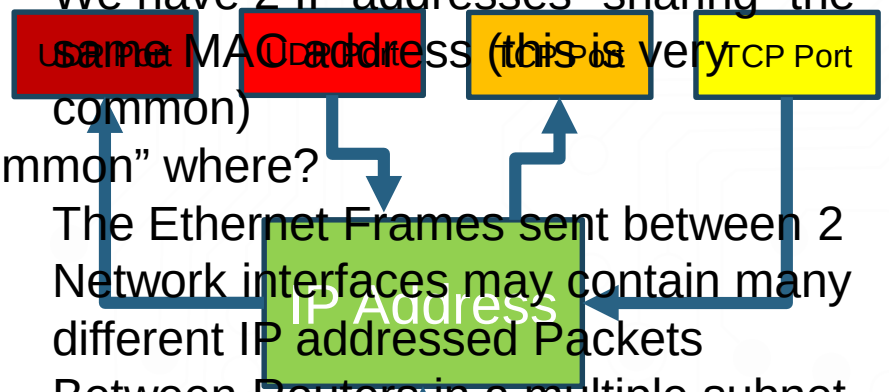
If its “common” where?

The Ethernet Frames sent between 2 Network interfaces may contain many different IP addressed Packets

Between Routers in a multiple subnet internet

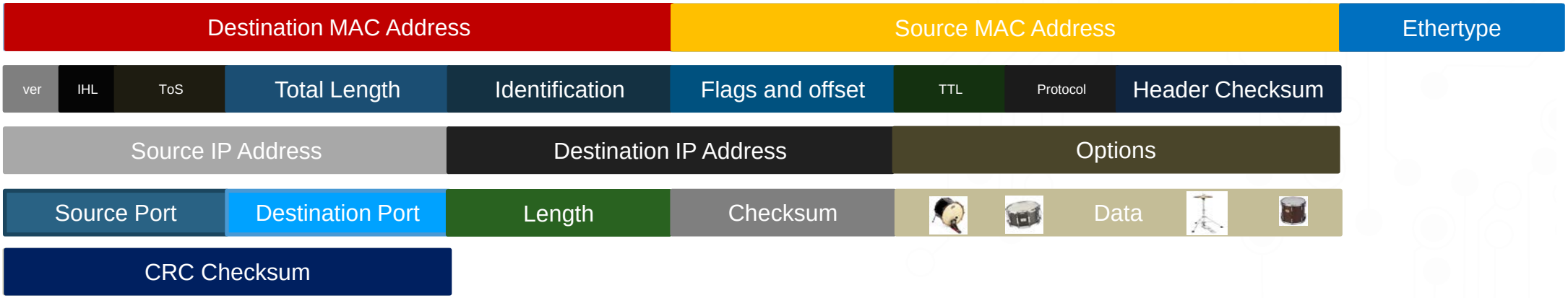
In a Virtualized server environment





What is an IP address

- We have our UDP datagram
- We have encapsulated this in an IP Packet
- The next stage is to encapsulate the IP Packet into an Ethernet Frame
- Whenever a Packet passes through a Router, the frame encapsulation is changed

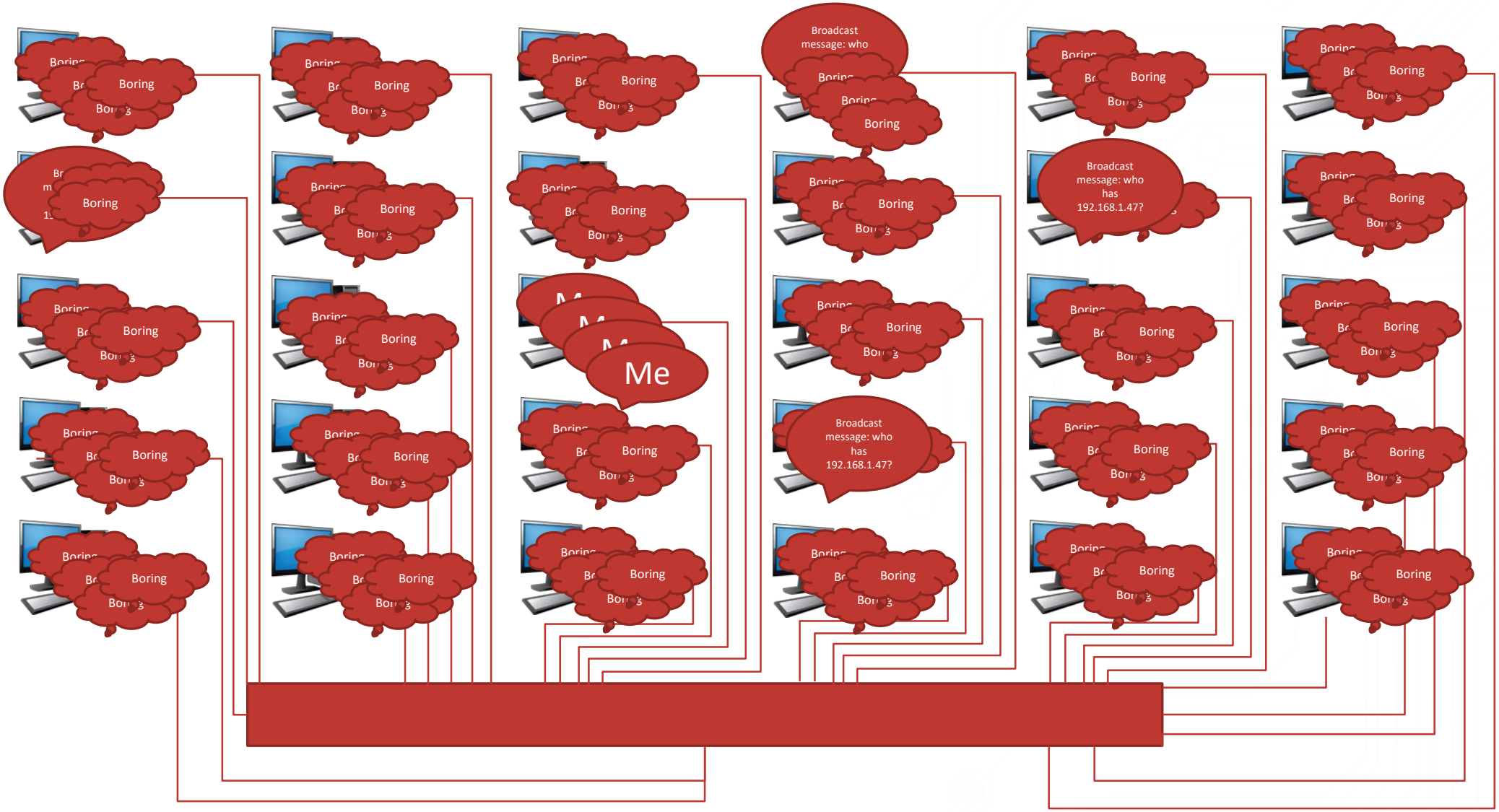


- Why change the Frame encapsulation?

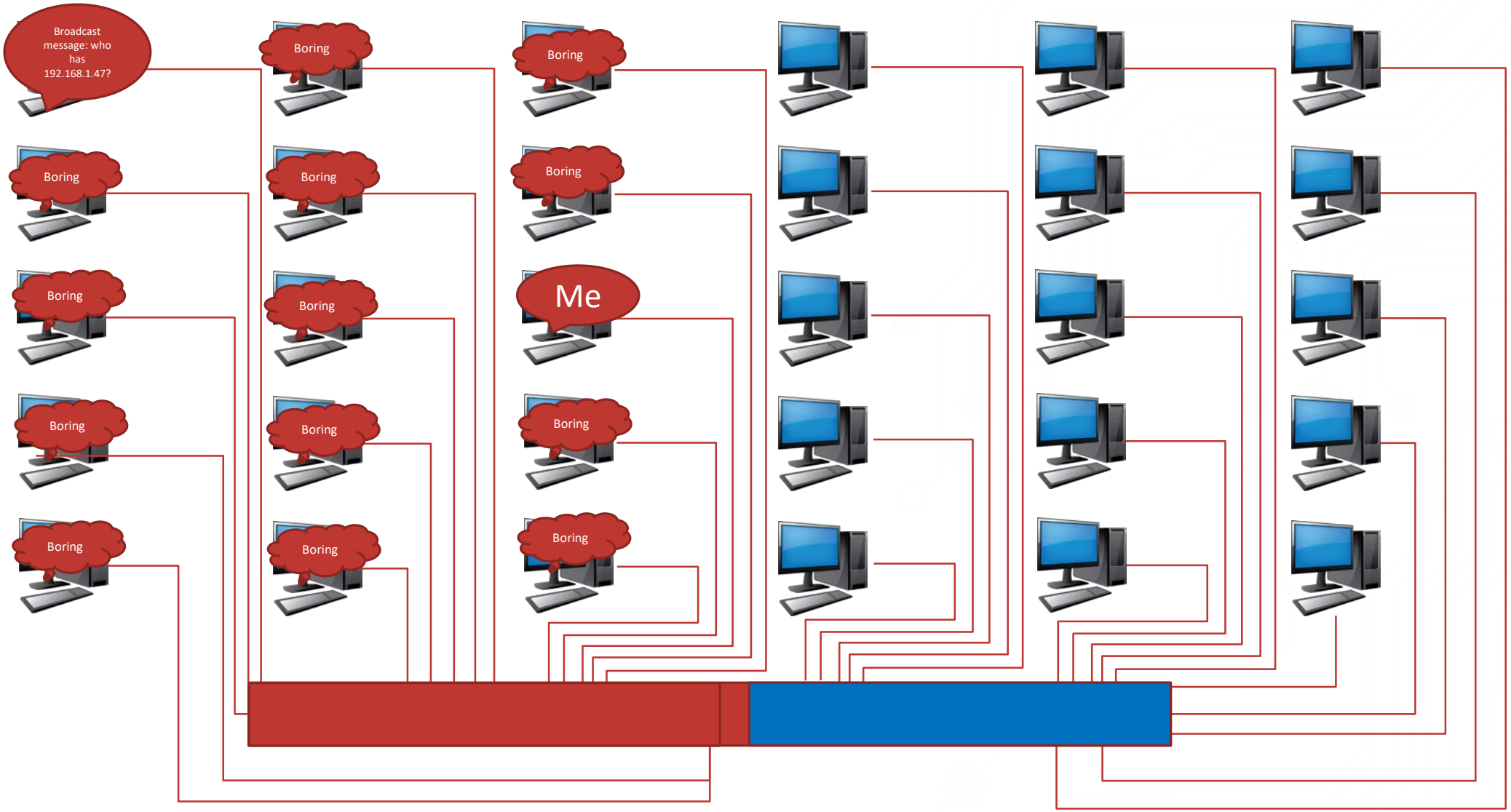
LAN MAN WAN

- “Layer 2” networks – sometimes called LANs or Local Area Networks inhabit one IP subnet
- They are joined together by switches
- Switches only care about MAC addresses
- A LAN is considered to be a single “Broadcast Domain”
- A “Broadcast domain” is the segment of a network where a “broadcast message” can be received
- Routers segment (will not forward traffic) “Broadcast Domains”
- A WAN – Wide Area Network consists of multiple IP subnets
- Therefore a WAN consists of multiple “Broadcast Domains”

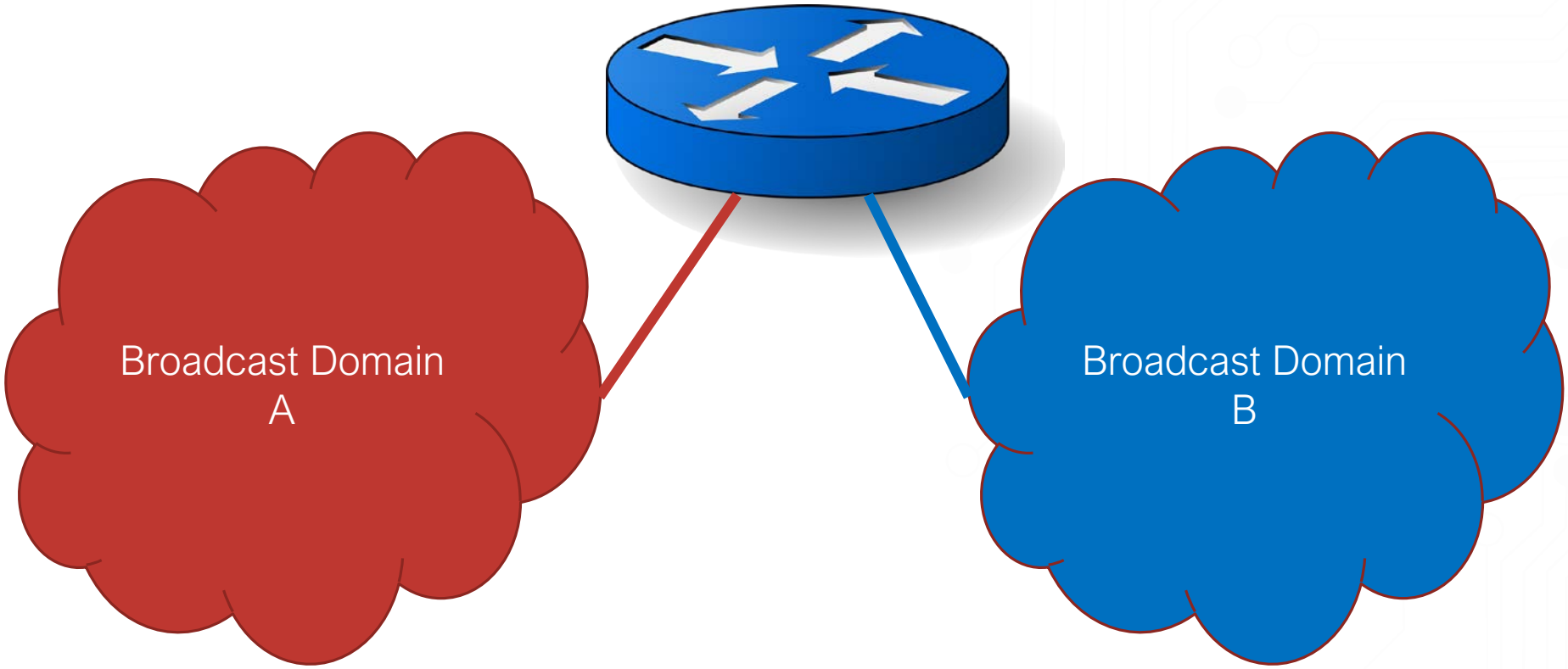
Broadcast Domain



Segmenting Broadcast Domains



Segmenting Broadcast Domains



Why we have IP subnets

- For a datagram to “leave” a Local Area Network (and by extension a Virtual Local Area Network) it NEEDs to be IP encapsulated
- Because the Frame Encapsulating the IP packet is destroyed, and re-built when traversing a Router
- Routers only care about IP addresses
- Routers drop all broadcast messages
- IP subnets, and therefore Routes allow us to communicate between LANs
- This is why IP Subnets exist

IP Subnetting

IP subnetting

- The WORST part of IPv4 addressing is the confusing mixture of binary and decimal
- This makes more difficult to learn than it should be
- An IPv4 address is 32 bits long
  - 4,294,967,296 possible unique addresses
  - It is written using a dotted decimal notation
  - aaa.bbb.ccc.ddd giving “decimal” values for 4 “octets”
  - Convert Binary to Decimal? – Windows calculator is really good!
- Using decimal notation is counter-intuitive
  - The device after number 255 should be 256
  - NOT 1.0

IP Addressing

- In Binary
  - The address after 11111111 is obviously 00000001.00000000
- Computers (and therefore networks) use Binary (fundamentally)
- IPv4 subnetting makes more sense in Binary
- IPv6 addresses are notated in Hexadecimal (much more sensible)
- A quick refresher on Binary
- With 1 bit I can have 2 values (a 1 or a 0)
- With 2 bits I can have 4 values (00 01 10 11)
- With 3 bits I can have 8 values (000 001 010 011 100 101 110 111)
- This pattern is “powers of 2” $2^1 = 2$ $2^2 = 4$ $2^3 = 8$ etc
- Binary “place value” is different to decimal

IP addressing

- Decimal and Binary Place Value

decimal	100s	10s	1s	128	64	32	16	8	4	2	1
123	1	2	3	0	1	1	1	1	0	1	1
55	0	5	5	0	0	1	1	0	1	1	1
200	2	0	0	1	1	0	0	1	0	0	0

- We take it for granted (in a decimal system) that 123 is arrived at by
  - (1x100) + (2x10) + (3x1)
  - Which simplifies to 100 + 20 + 3
- In Binary (to convert to decimal) we do
  - (0x128) + (1x64) + (1x32) + (1x16) + (1x8) + (0x4) + (1x2) + (1x1)
  - Which simplifies to 64+32+16+8+2+1

IP Addressing

- An essential element of using IP is “sub-netting”
- We can describe a single node using an IP address
- We can describe a group of nodes using an IP address
- To understand whether an IP address refers to a group or “network” or a single device we use an IP subnet mask
- The IP subnet mask is crucial for a device to understand whether it is connecting to a device in its own IP subnet (local)
- Or whether it is connecting to a “remote” network
- On a local network – the device can communicate “directly”
- To connect to a remote network the device must connect via a gateway

IP addressing

- The IP subnet mask is held in all devices in a network
- It is used to define the “size” of the local network
- A device applies a logical truth table to determine a local or remote destination 1=true 0=false
- Because the mask refers to the IP address – it is the same “length”
- In IPv4 this is 32 bits

IP addressing

- If I use an IP address that we are all familiar with
  - 192.168.1.1
  - Translated to Binary
  - 11000000.10101000.00000001.00000001
  - And apply a familiar subnet mask
  - 11111111.11111111.11111111.00000000 (255.255.255.0)
  - In the subnet mask 1 means “must match” 0 means may match (for destination to be local)
  - If I send to 11000000.10101000.00000001.00000010 (192.168.1.2)
  - Compare with subnet mask 1= match 0 =no match
  - 11111111.11111111.11111111 (I can stop now… its local!)

IP addressing

- I can “extend” the subnet mask (more but smaller subnets)
  - 192.168.1.1
  - Translated to Binary
  - 11000000.10101000.00000001.00000001
  - And apply a VLSM – Variable Length Subnet Mask
  - 11111111.11111111.11111111.10000000 (255.255.255.128)
  - In the subnet mask 1 means “must match” 0 means may match (for destination to be local)
  - If I send to 11000000.10101000.00000001.00000010 (192.168.1.2)
  - Compare with subnet mask 1= match 0 =no match
  - 11111111.11111111.11111111.1 (I can stop now… its local!)

IP addressing

- I can “shrink” the subnet mask (fewer but larger subnets)
  - 192.168.1.1
  - Translated to Binary
  - 11000000.10101000.00000001.00000001
  - And apply a VLSM – Variable Length Subnet Mask
  - 11111111.11111111.11111110.00000000 (255.255.254.0)
  - In the subnet mask 1 means “must match” 0 means may match (for destination to be local)
  - If I send to 11000000.10101000.00000001.00000010 (192.168.1.2)
  - Compare with subnet mask 1= match 0 =no match
  - 11111111.11111111.1111111 (I can stop now… its local!)

IP addressing

- IP subnet masks can only be 1s read left to right (as notated this way)
- Subnet Masks – the “magic” numbers

decimal	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0
128	1	0	0	0	0	0	0	0
192	1	1	0	0	0	0	0	0
224	1	1	1	0	0	0	0	0
240	1	1	1	1	0	0	0	0
248	1	1	1	1	1	0	0	0
252	1	1	1	1	1	1	0	0
254	1	1	1	1	1	1	1	0
255	1	1	1	1	1	1	1	1

IP addressing

- We look at one “octet” at a time to create decimal notation
- CIDR notation /n is the number of bits in the WHOLE (32 bit) subnet mask

decimal	128	64	32	16	8	4	2	1	CIDR 1st octet	CIDR 2nd octet	CIDR 3rd octet	CIDR 4th octet
0	0	0	0	0	0	0	0	0	0	8	16	24
128	1	0	0	0	0	0	0	0	1	9	17	25
192	1	1	0	0	0	0	0	0	2	10	18	26
224	1	1	1	0	0	0	0	0	3	11	19	27
240	1	1	1	1	0	0	0	0	4	12	20	28
248	1	1	1	1	1	0	0	0	5	13	21	29
252	1	1	1	1	1	1	0	0	6	14	22	30
254	1	1	1	1	1	1	1	0	7	15	23	31
255	1	1	1	1	1	1	1	1	8	16	24	32

IP addressing

- Number patterns
- An IP subnet has its own address
  - This ALWAYS ends Binary 0
- An IP subnet has a broadcast address
  - (by convention) This ALWAYS ends Binary 1 for all host bits
- We haven't mentioned classful IP addresses
  - Its true – Modern IP has no class
  - BUT (to continue the theme of patterns)
  - A Class A IP address ALWAYS starts with a 0
  - A Class B IP address ALWAYS starts with a 10
  - A Class C IP address ALWAYS starts with a 110
  - A Class D IP address (multicast) ALWAYS starts with 1110

IP addressing

- Why are number patterns interesting?
- When network equipment was slower – these patterns helped with processing reduction – increased speed
- Lets compare some examples
  - Subnet Mask
    - 255.255.255.0 11111111.11111111.11111111.00000000
  - Network address
    - 192.168.1.0 [redacted].00000000
  - first host
    - 192.168.1.1 11000000.10101000.00000001.00000001
  - Last host
    - 192.168.1.254 11000000.10101000.00000001.11111110
  - Broadcast
    - 192.168.1.255 [redacted]11111111

IP addressing

- Why are number patterns interesting?
- When network equipment was slower – these patterns helped with processing reduction – increased speed
- Lets compare some examples
  - Subnet Mask
    - 255.192.0.0 11111111.11000000.00000000.00000000
  - Network address
    - 10.0.0.0 [redacted]000000.00000000.00000000
  - first host
    - 10.0.0.1 00001010.00000000.00000000.00000001
  - Last host
    - 10.63.255.254 00001010.00111111.11111111.11111110
  - Broadcast
    - 10.63.255.255 [redacted]111111.11111111.11111111

Redundancy Techniques

Redundancy

- An Audio engineer's second worst nightmare!
  - The Cable got broken, so I lost the audio
- Second? So what's the first?
  - The Radio Mic ran out of battery, because I forgot to change it
- Ideally a redundant system should be
  - Unheard (it is there to catch you without missing a beat)
  - Simple (because complicated makes it more likely to fail)
  - An asset to the system (not a liability… this has been true)
- Truly redundant – not "high availability" (there is a big difference)
- Minimize (or eliminate) points of failure

Misunderstanding Redundancy

- VERY FEW technologies exist specifically for redundancy
- Some non-specific technologies can be used to create a resilient system by exploiting technical aspects
- Examples:
  - Spanning Tree Protocol
    - Primarily exists to prevent loops in networks
    - Can give high availability by deliberately creating a loop
  - LACP (Link Aggregation Control Protocol)
    - Primarily exists to create load-balanced, aggregated links where capacity is limited
    - Can provide the appearance of redundancy if certain criteria are met
  - Dynamic Routing Protocols (Layer 3)
    - Give high availability by monitoring link states between routers

Misunderstanding Redundancy

- Dante Redundancy
  - Proprietary technology SPECIFICALLY DESIGNED to achieve redundant connections
- If a technology relies on “switching” connections it is an High Availability method, and is not strictly redundant
  - Spanning Tree Protocol
  - LACP
  - Layer 3 Routing Protocols
- Dante Redundancy DOES NOT SWITCH FROM PRIMARY to SECONDARY!
  - Duplicate signals are sent ALL OF THE TIME on BOTH Networks

Creating Resilient Networks

- Infrastructure technologies can however compliment Dante Redundancy
- REMEMBER – if you are implementing a resilience standard
  - Know how it works
  - Know its limitations
  - Understand how to predict these

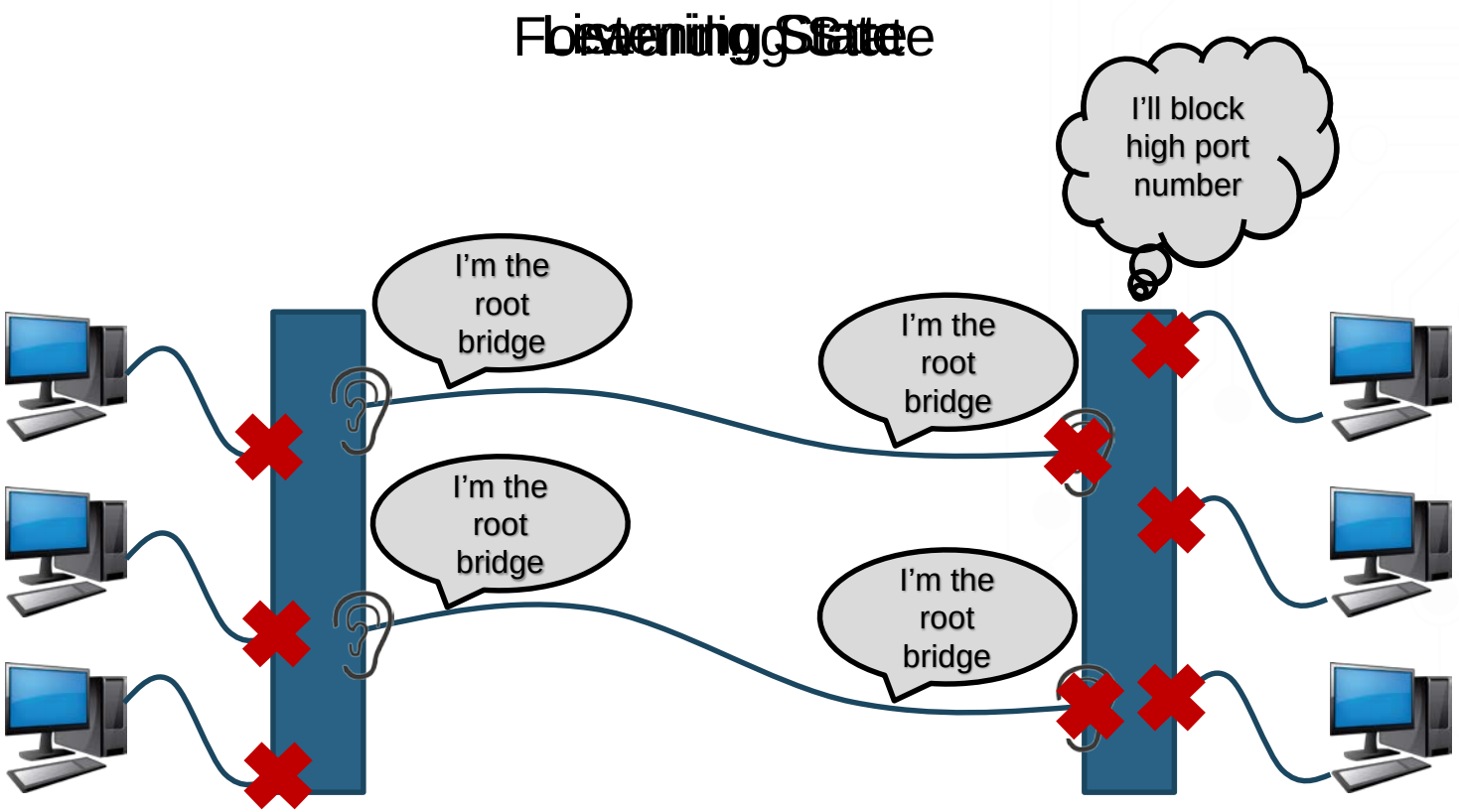
STP – Spanning Tree Protocol

- The Algorithm guarantees one and ONLY one Link between hosts
- How?
  - BPDUs are sent (Bridge Protocol Data Unit)
    - Multicast Frames (STP is a Layer 2 technology)
    - Configuration BPDU (sends information to all STP switches)
    - TCN (Topology Change Notification) – something has happened
    - TCA (Topology Change Acknowledgement) – Thanks message received
  - Hello Time
    - How often BPDUs are sent
    - Default every 2 seconds
  - Forward Time
    - How long a switch stays in the “listening” state before going to “learning” before going to “forwarding” state
    - Can be adjusted from 4 to 30 seconds
    - Be Careful with this value on large networks!
  - Max Age
    - How long a BPDU is stored in a switch, before it listens for new BPDU

STP Spanning Tree Protocol

- “Classic” Spanning Tree Protocol Works as follows:
- Like many network technologies there is an election process
- On startup all switches send BPDUs assuming that they are the Root Bridge (the principal point of reference)
- All switches are in a “listening” state – they do not forward any other traffic
- If a switch receives a BPDU from another switch with a superior “priority” number, it forwards that BPDU, and stops sending its own BPDU
- This “listening” state lasts for 15 seconds (by default) – forward time
- The switch then goes into a “learning” state for 15 seconds (default)
- After 30 seconds, “normal” traffic is forwarded

Spanning Tree Protocol



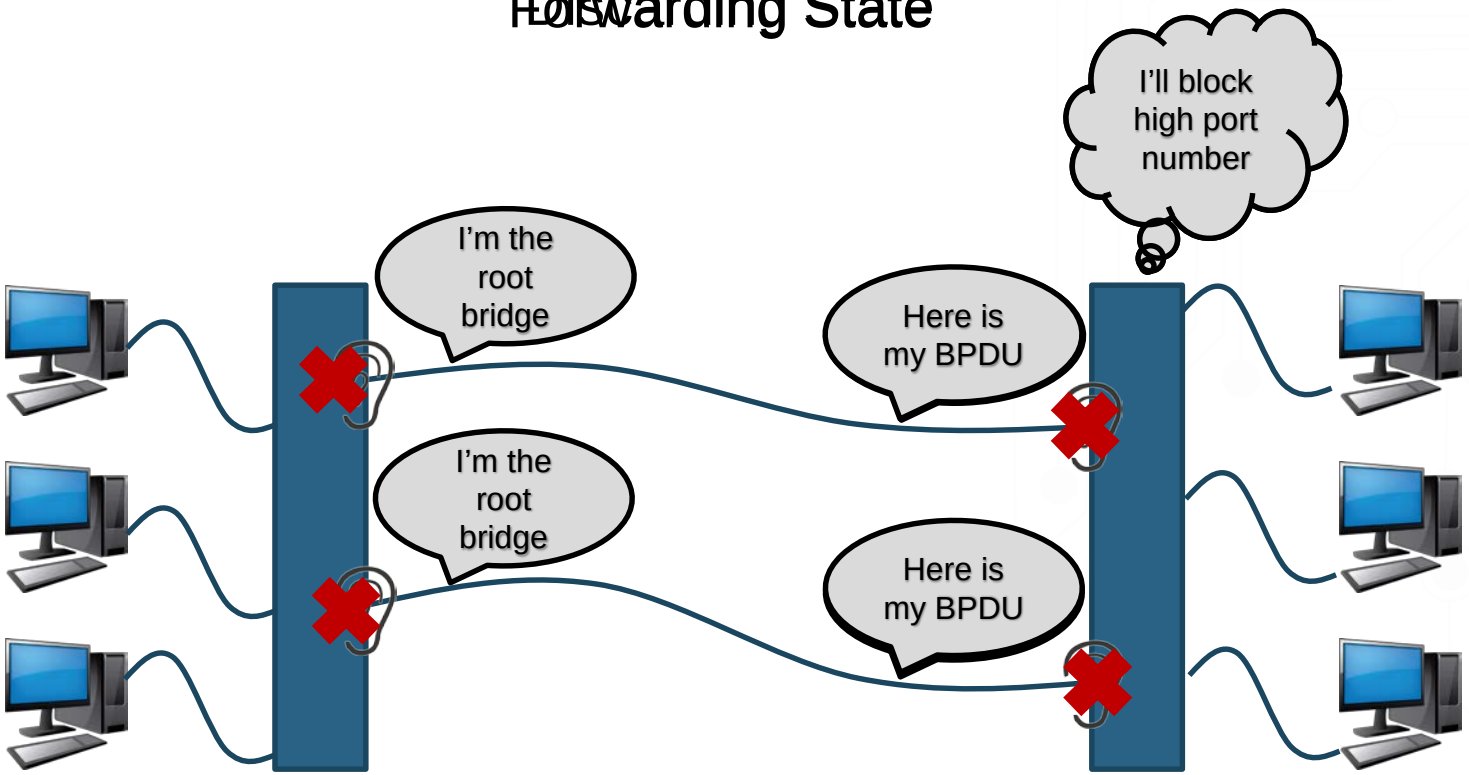
Spanning Tree Protocol

- 30 seconds without data (and longer without audio!)
- Is that redundant?
  - It's probably quicker than
    - Finding the broken cable
    - Unplugging the broken cable
    - Replacing the broken cable
    - On a big network
- It is still a very long time!
- So…

Rapid Spanning Tree Protocol

- RSTP combines the listening and learning states
- Every switch always sends its own BPDUs all the time
- This means that all switches have up to date information
- “discarding” state replaces the distinct Listening and Learning states
- 3 Hello times are all that is needed (default 3x2 = 6 seconds)
- 6 Seconds… its still very noticeable

Forwarding State



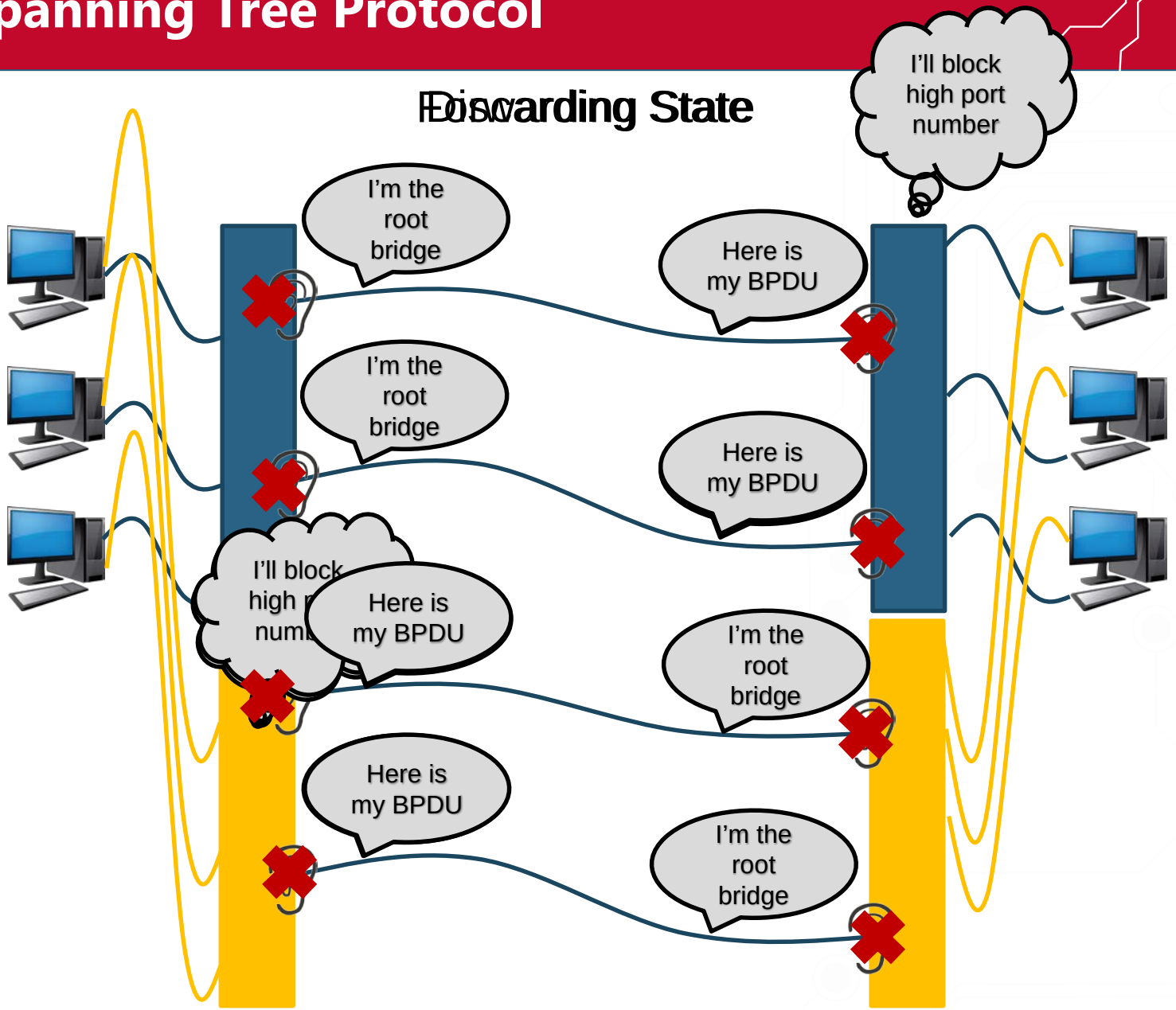
Spanning Tree Diameter

- This is the most Dangerous part of spanning tree
- Especially in large temporary installations
- Normally the “diameter” is 7
- This means that any switch more than 7 switch hops from the Root Bridge is unaware of the Root Bridge ID
- How did I find this out?
- Setting up a gig… it was “educational”
- Symptoms:
  - Random weirdness
  - Whole Tree becomes unstable
  - Links come up, packet storms start, and abate, randomly
  - It gets worse and worse
- Spanning Tree Diameter can be increased… BUT
  - You must use a longer Hello time
  - Using a longer forward time is VERY important
  - Result – much longer convergence times

PVSTP and MSTP

- Modern switches have higher processing capacities
- We can create multiple “instances” of spanning tree
  - Several instances in a VLAN – breaks up network into zones
  - Do a Spanning tree instance “per VLAN”
- Combining Per VLAN spanning tree with Dante Redundancy
  - Extreme redundancy!

MSTP Spanning Tree Protocol



LACP Link Aggregation Control Protocol

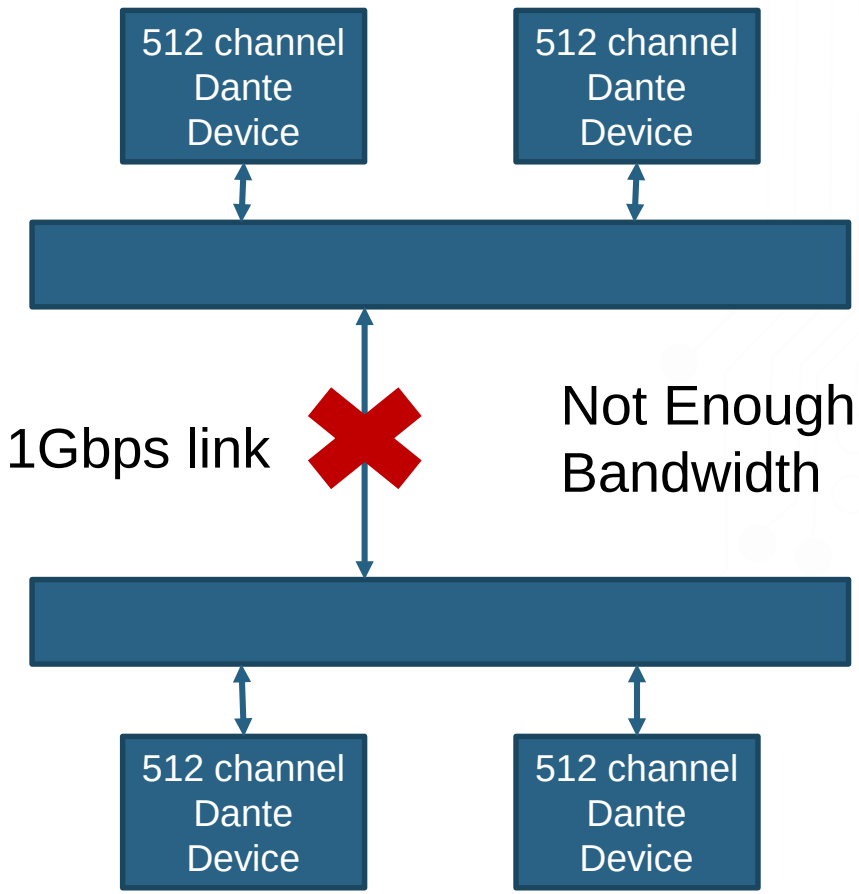
- LACP is a method for ostensibly creating higher bandwidth links
- BEWARE – it is NOT a true method for redundancy
- Why?
  - LACP “load balances” across member physical links
  - The “standard version” uses source MAC address for the load balance differentiator
  - Some Vendor Proprietary implementations may use
    - IP address
    - Port Address

LACP Link Aggregation Control Protocol

- Why does it “feel” redundant?
- The biggest Dante Endpoint sends under 1gbps per MAC address
- If using 1gbps links, then, in practice this isn’t an issue
- If aggregating multiple 100mbps links…
  - You WILL NOT get more than 100mbps per MAC address!
  - No Matter how many links you aggregate!

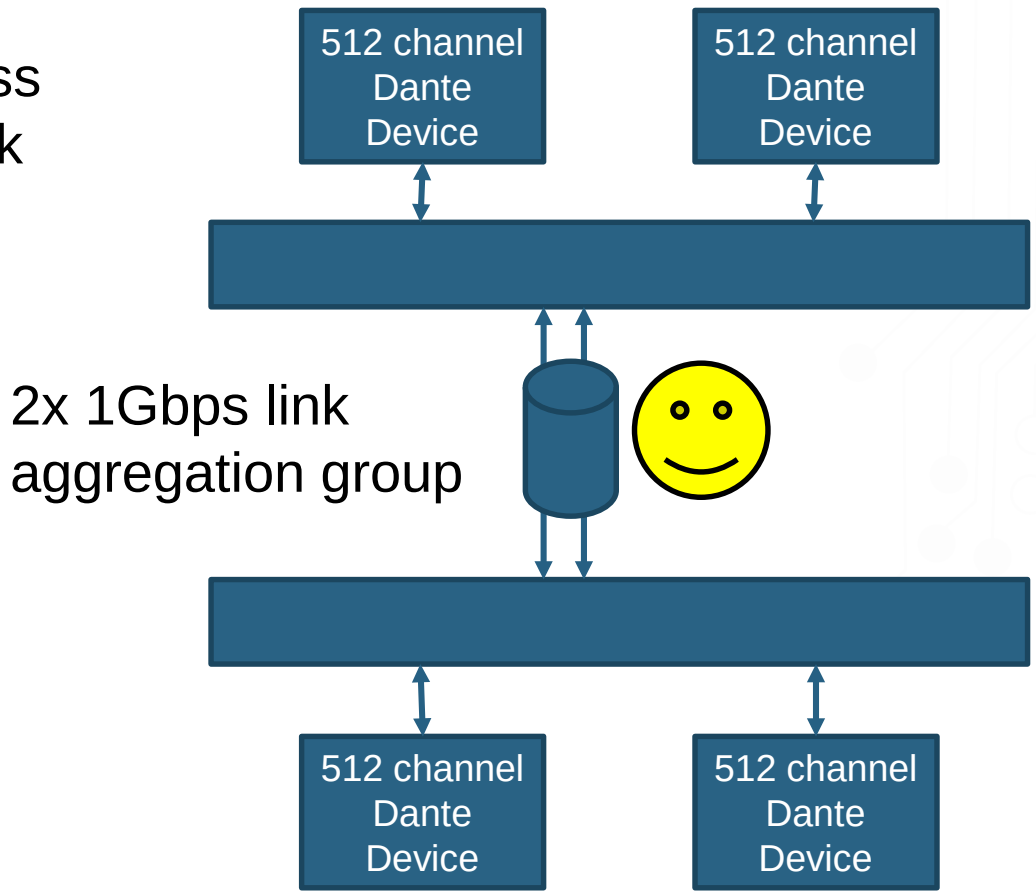
LACP

Sending 1024 channels across Inter-switch link



LACP

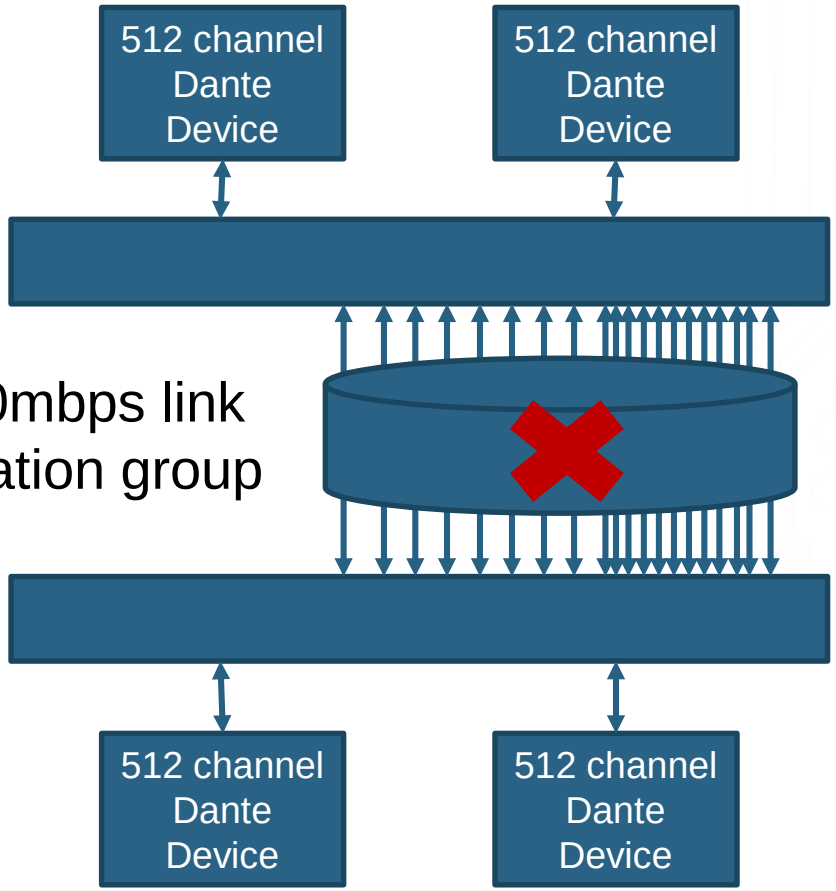
Sending 1024 channels across Inter-switch link



Load balancer will use one link per source MAC address

LACP

Sending 1024 channels across Inter-switch link



20x 100mbps link aggregation group

Load balancer will use one link per source MAC address

Links 1 & 2 will fill to 100mbps, links 3-20 will do nothing

LACP

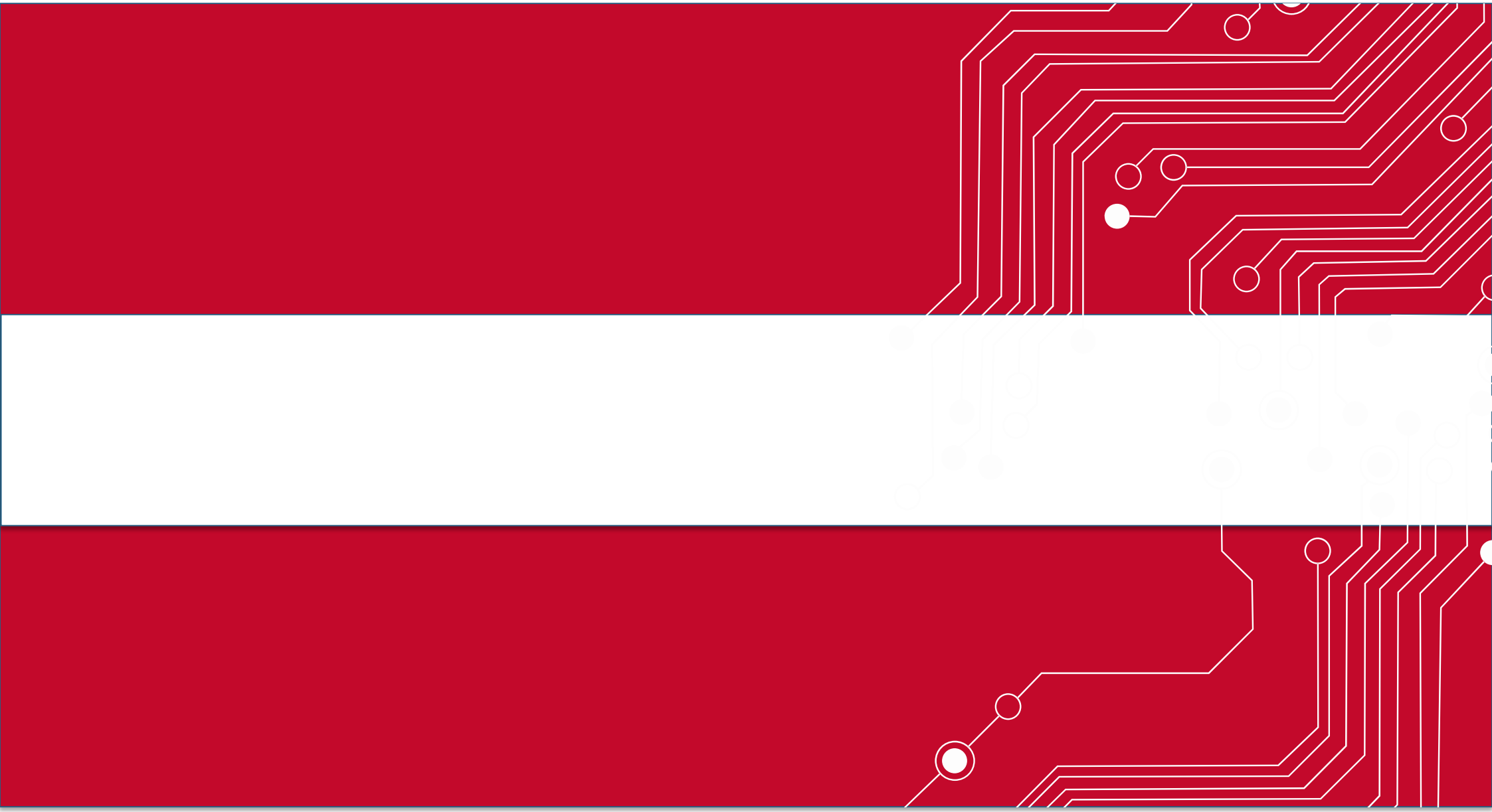
- Each link aggregation group is a “point to point” link
- Members of a LAG normally have to have the same speed
- On a 2 member LAG
  - If the used capacity is <50% of the total
  - AND no single MAC address can send >50% of the total
  - Then there is a rapidly converging (1 frame) high availability link
- This is easy to calculate in an Audio only network
- “dropout” would be maximum of a few samples (if any)
- More difficult to predict in a multi traffic network

Dante Redundancy

- Dante Redundancy is true redundancy
- The two network stacks on a redundant Dante Device separately encapsulate the audio
  - all the way from acquisition (application Layer)
  - Through the UDP Datagram
  - To separate IP addresses
  - Through Separate MAC addresses
  - Through separate Physical connectors
  - And “back up again”
- As long as the correct sample arrives “on time” at either or both destination IP stacks, audio will be played out

Dante Redundancy - Confusion

- The HUGE risk of a true redundant audio system
  - How do you know there is a “fault” if audio is still working?
- Dante is not just a media transport layer
- Dante has a rich API that allows for reporting and measurement of thousands of parameters
- The comprehensive Dante Control and Monitoring protocol reports any errors to a host system (the Audio part carries on working)
- “Error” reports are not very “human friendly” in text (they are very precise)
- Sometimes these messages are given “simplified” “friendly” translations in some software applications
- This causes misunderstandings for users reading the translated message
- Dante Redundancy NEVER “switches to secondary”
- This usually means “an error was detected on the primary”



Security

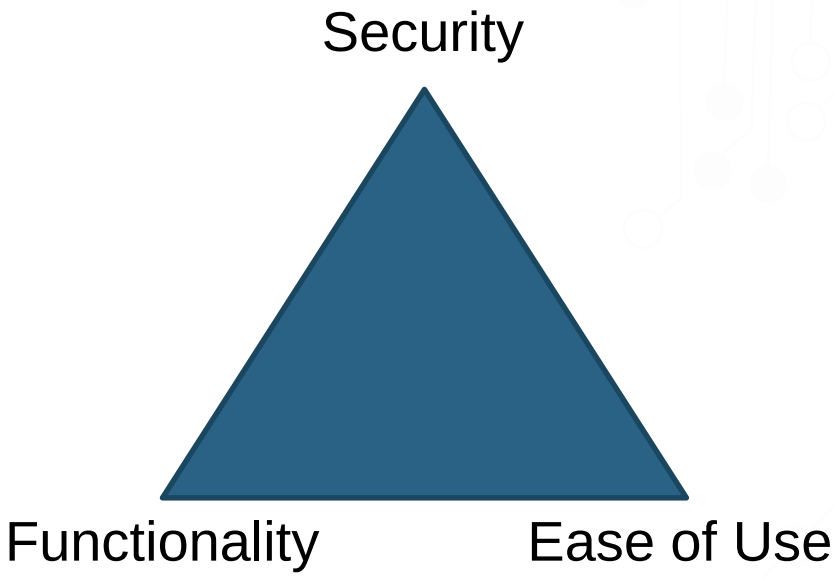
Securing a Network

- There are many ways to secure a network
  - Many are good
  - Many are mature
- Security is simple
- Making a secure communications system is not
- Effective security implementations also follow a layered model approach
- Security systems must be designed before being implemented
- “ad-hoc” security is nearly a paradox

© Scott Adams, Inc./Dist. by UFS, Inc.

Layer 1 Security

- Layer 1 security is the most effective security method
- Layer 1 security is also the least useful method to allow communication
- Security can be thought of as a triangle – choose your position inside



Layer 1 Security

- Layer 1 is the physical layer
- To secure Layer 1 - don't allow the possibility of making a physical connection

- The above is effective at denying access
- The best security is binary (at every layer) Permit/Deny – no maybe

Analogue Security

This piece of tape describes the consequences of tampering with the settings (quite graphically)

Layer 2 Security

- Assuming that we have found a way to permit physical access at layer 1 – we move to Layer 2
- Layer 2 is the Datalink Layer, and devices use MAC addresses
- A MAC address is a UUID – Universally Unique Identifier
- This is where a little intelligence needs to be applied
- Did I get to this point before someone nearly wet themselves about MAC address spoofing?
  - Yawn
- Using A MAC address whitelist as a permit/deny criteria is perfectly valid
- Is it totally secure in and of itself? No
- Is it more than most networks do today? absolutely

Layer 2 Security

- Assuming that we have permitted a MAC address
- What can we do at Layer 2?
- We can connect to another layer of security at Layer 3
- Because we don't completely trust MAC addresses
  - We temporarily assign permitted MAC addresses to a single VLAN
  - We then move on up to Layer 3
- We "could" theoretically "spoof"

a whitelisted MAC address – maybe a Dante Device to gain access

(more on this later)

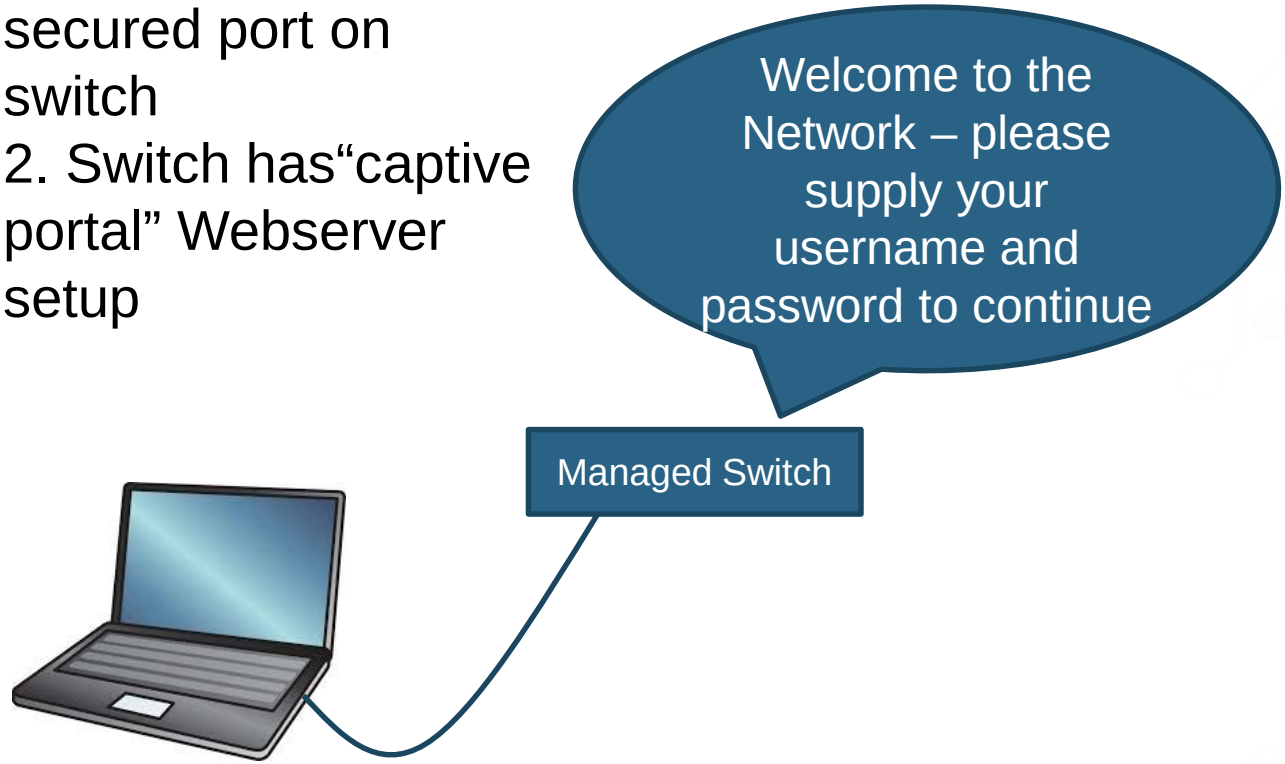
Layer 3 Security

- At Layer 3 we have a device that has passed MAC permit/deny
- We can assign the device an IP address using DHCP
- Keeping the device in a quarantined IP subnet allows us to connect to the application layer, and do further authentication
- This gets around “static IP address” attacks
  - If a different IP subnet is entered statically – no communication can happen with the authentication application
  - If a static IP address is used in our quarantined IP subnet… well the device will still have to connect to the application that decides permit/deny… and will (even if permitted) fail to access the secure side of the network (because it will not be able to change IP address automatically)

A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 1

1. Laptop connects to secured port on switch
2. Switch has“captive portal” Webserver setup



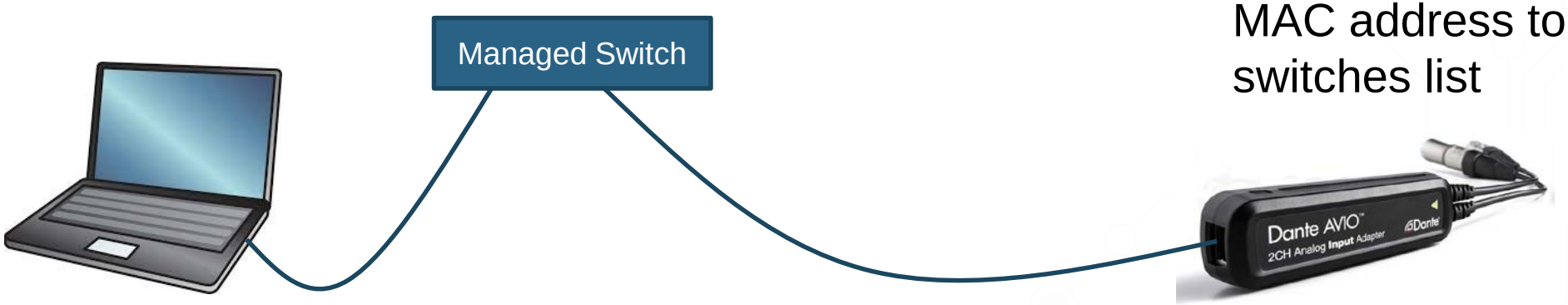
3. Switch has a list of usernames and passwords locally
4. Switch is the DHCP server
5. Switch blocks everything apart from authentication webpage

A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 2

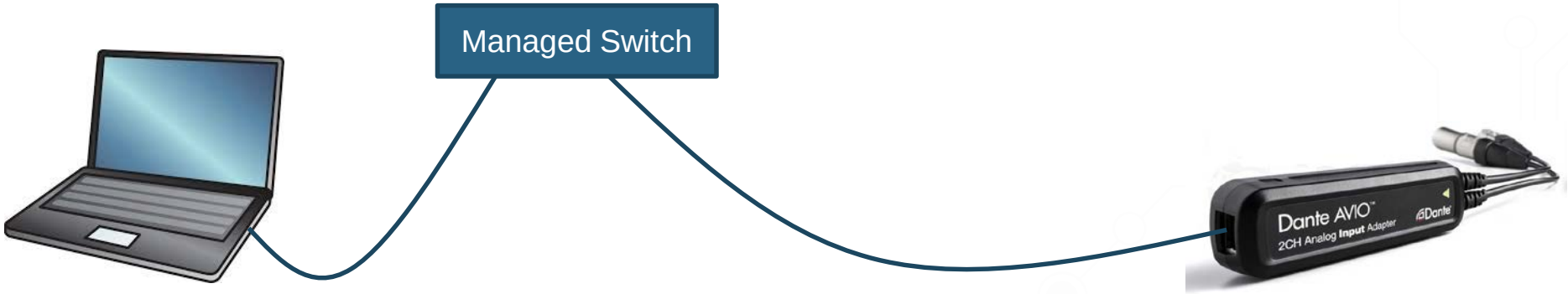
1. The computer authenticated successfully
2. Now we want to connect a Dante Device

3. There is no way to send a username and password from Dante device
So we use a different port on the switch, and add the Device MAC address to the switches list



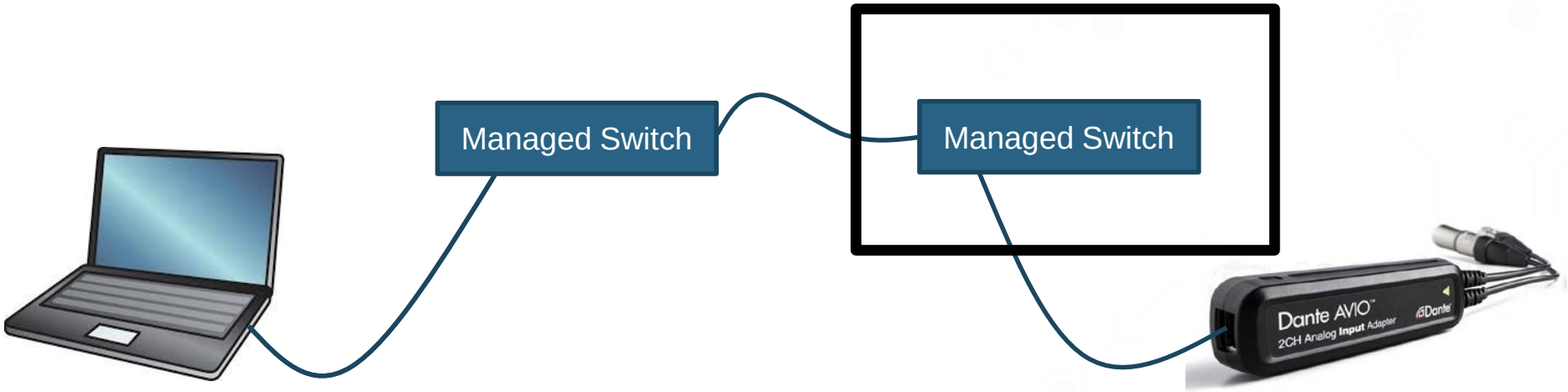
A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 2
- What prevents us from “spoofing” the Dante Device MAC address and plugging into that port?
- In this setup – absolutely NOTHING!
- BUT – apply layer 1 security (and 2 and 3)



A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 3
- Use a switch located for easy access with captive portal authentication
- Use a physically secured switch running a MAC address whitelist
- Easy – but not perfect (but better than 99% of networks today)

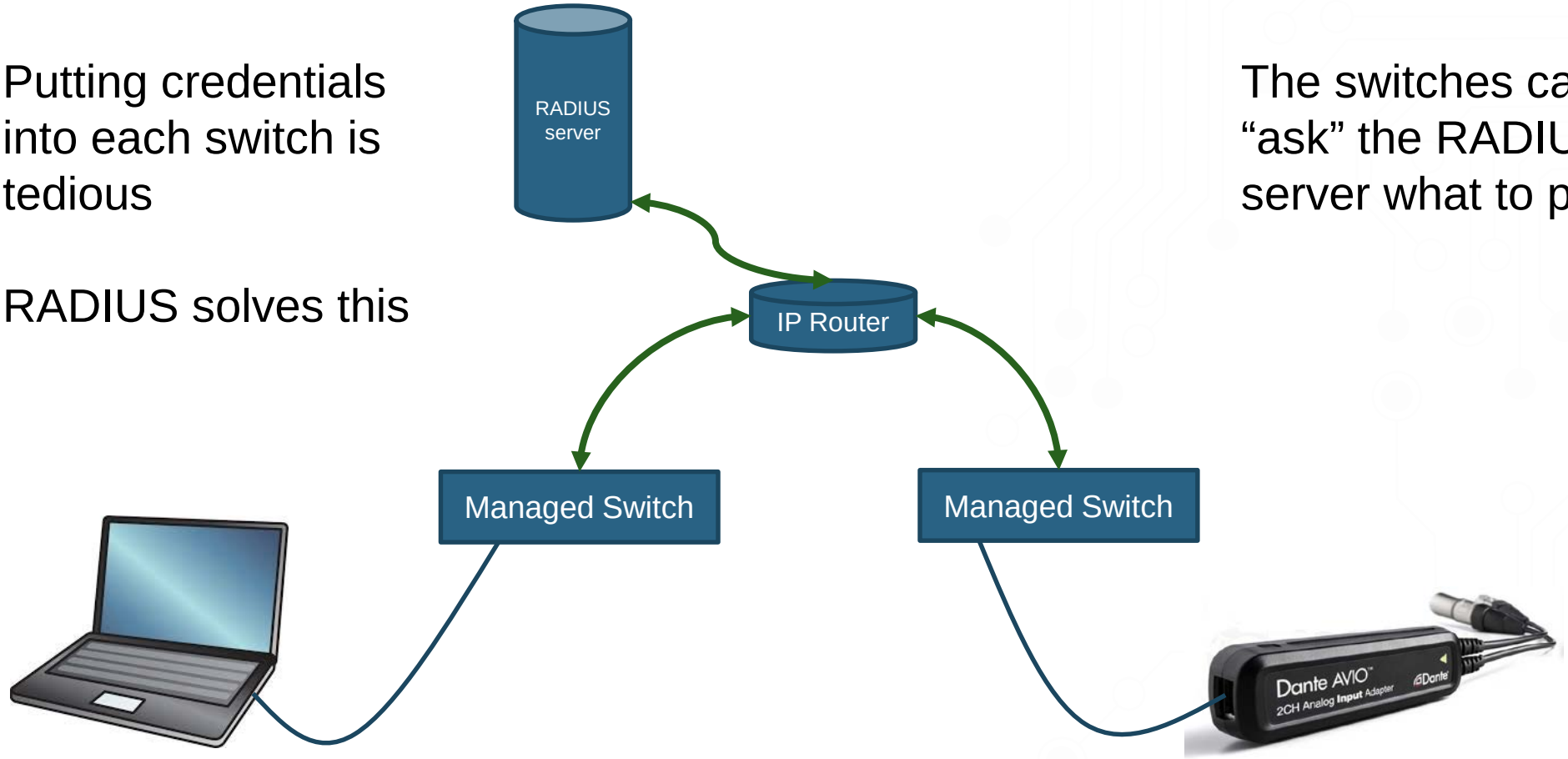


A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 4

Putting credentials into each switch is tedious

RADIUS solves this



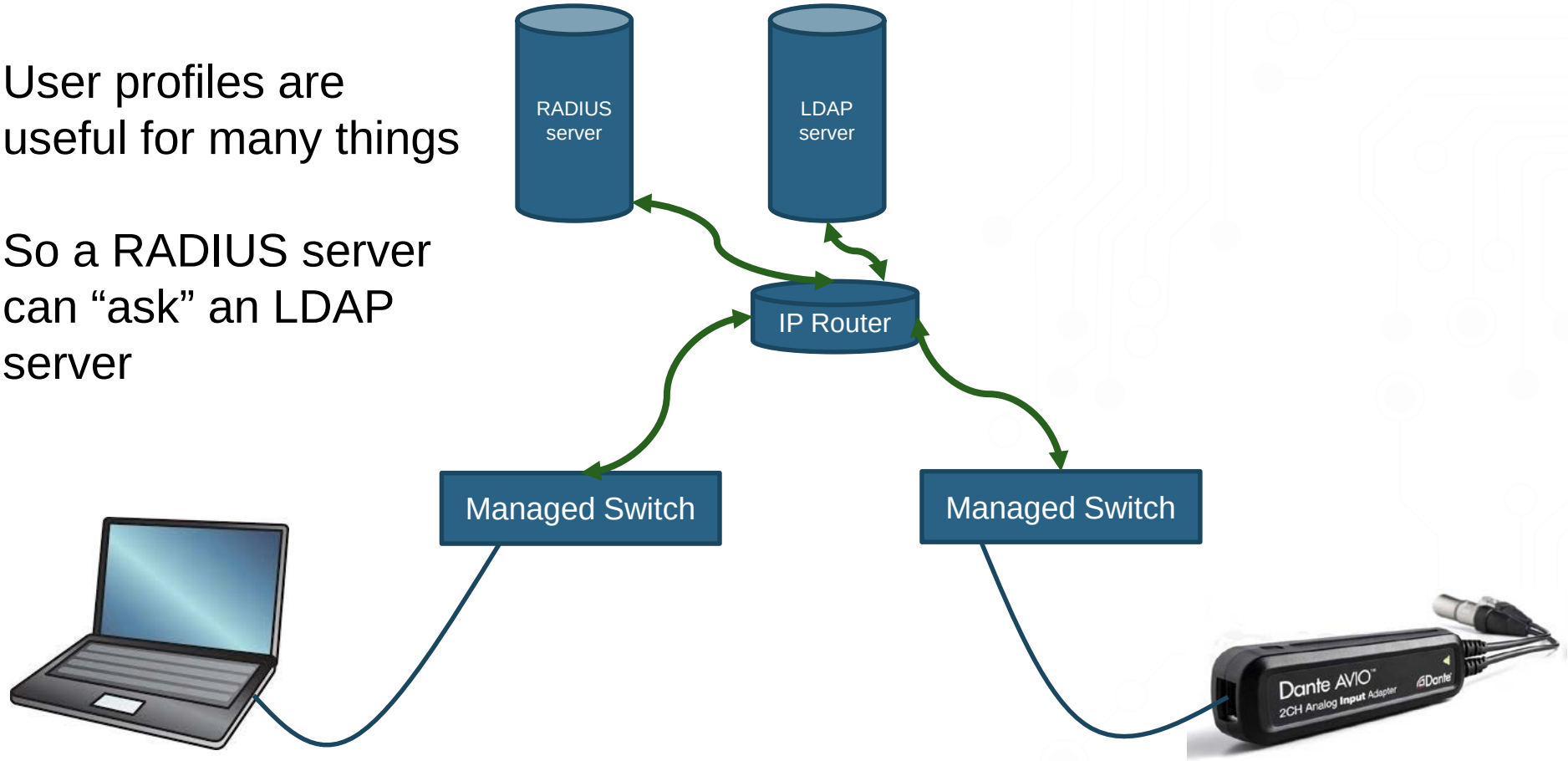
The switches can "ask" the RADIUS server what to permit

A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 5

User profiles are useful for many things

So a RADIUS server can “ask” an LDAP server

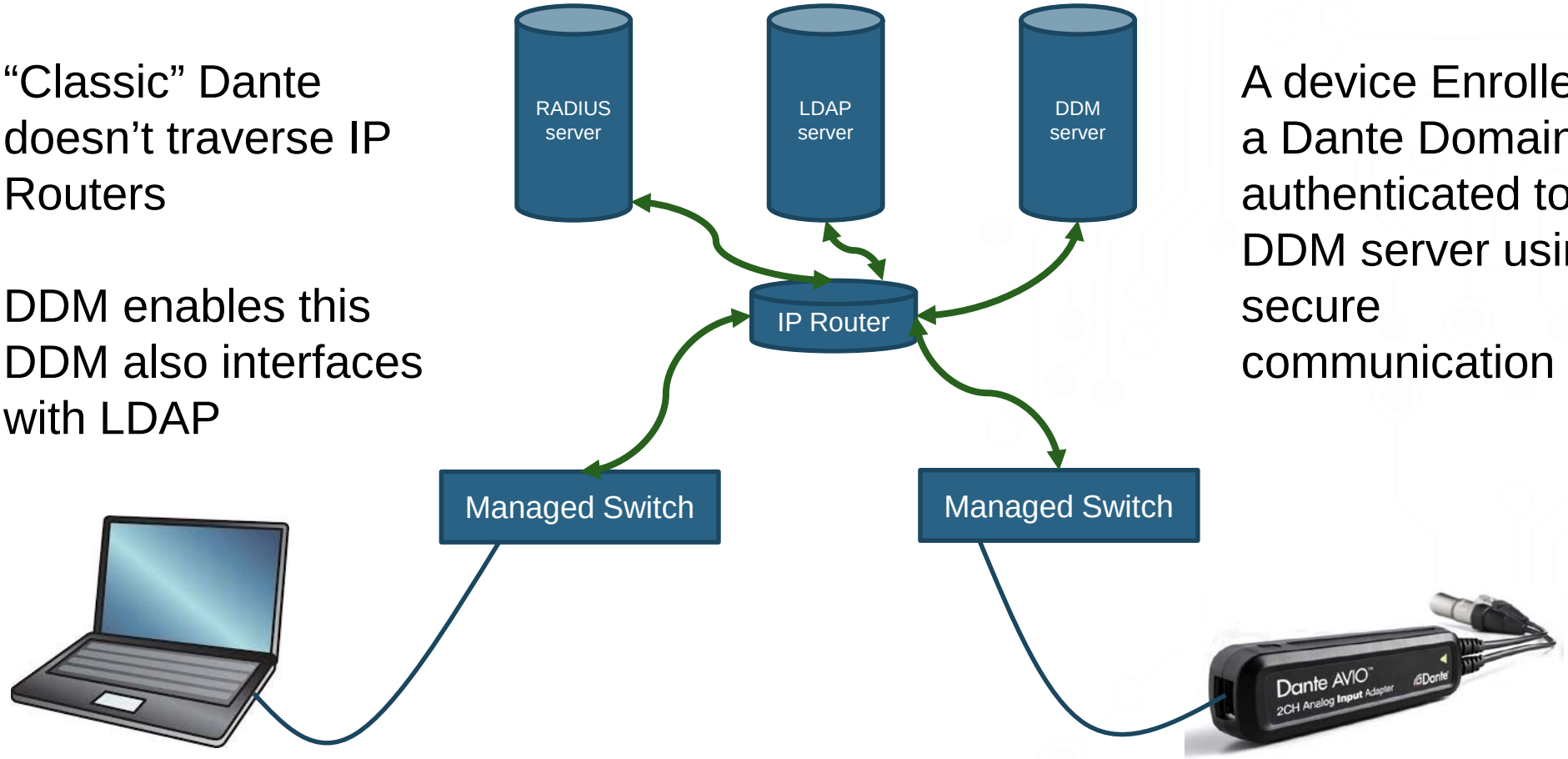


A Secure Network – A possible implementation

- Theory is all well and good – but reality is better
- Scenario 6

"Classic" Dante doesn't traverse IP Routers

DDM enables this DDM also interfaces with LDAP



A device Enrolled in a Dante Domain is authenticated to a DDM server using secure communication

DDM Security Workflows

Windows/OSX device

Connects to Switch

Switch webserver asks for username and password

Switch requests match from RADIUS server

RADIUS server asks LDAP server

LDAP server replies to RADIUS server

RADIUS server replies to switch

Switch permits/denies based on response

Unenrolled Dante Device

Connects to switch

Switch requests MAC address match from RADIUS server

RADIUS server replies to switch (and may instruct switch which VLAN to place device in)

Switch permits/denies based on response

Enrolled Dante Device

Connects to switch

Switch requests MAC address match from RADIUS server

RADIUS server replies to switch (and may instruct switch which VLAN to place device in)

Switch permits/denies based on response

Device connects to DDM server

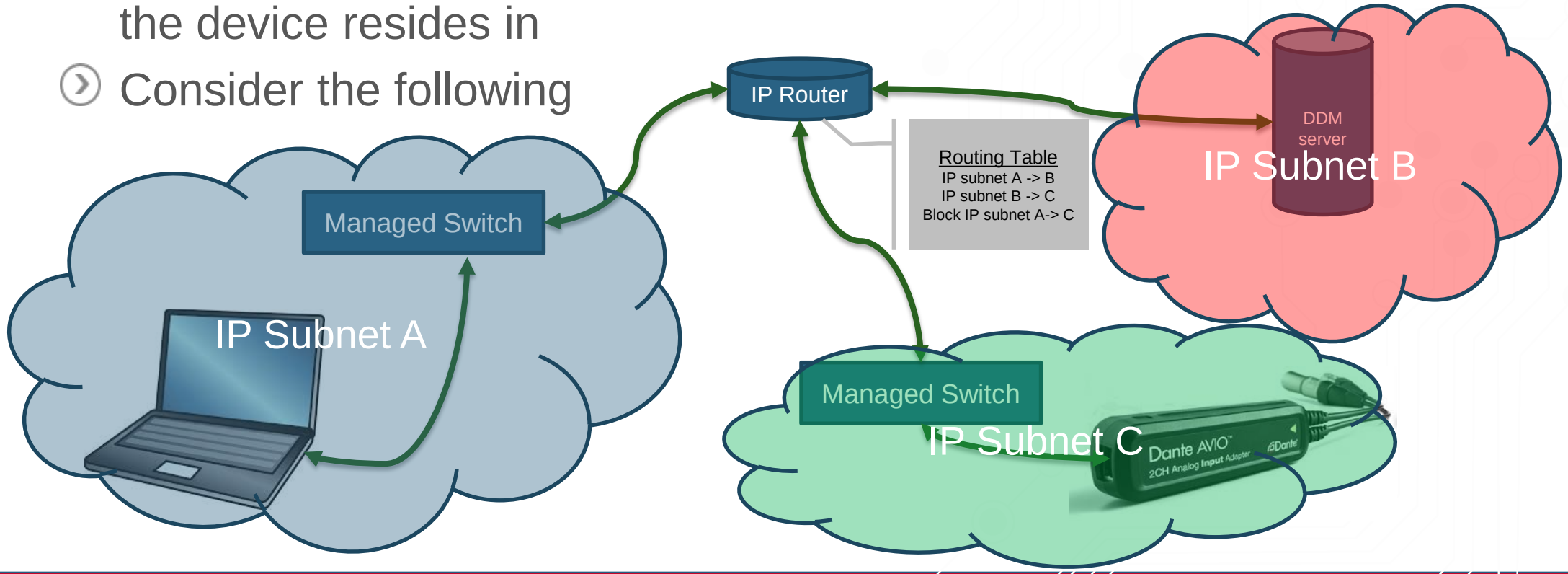
Device will only exchange control messages with DDM server

How this aligns with common methods

- 802.1x is a common method to secure networks
- It relies upon a “supplicant” an “authenticator” and an “authentication server”
- Desktop Operating Systems integrate 802.1x easily (just enable in Windows network setup)
- This is replicated in this example by the “captive portal” webserver as a manual method for supplication
- It can be automatic in Windows/Mac OS
- Difficult for Dante devices, printers security cameras etc
- Hence MAB (MAC Authentication Bypass) extension (common)
- The Authentication Server is the RADIUS server (in this example)
- Using MAB and DDM in concert replaces the supplicant in 802.1x in at least as a secure way

DDM Layer 3 Security tips

- Unlike “classic” Dante controllers can’t send control messages direct to domain-enrolled devices
- Routing and discovery is done through the DDM server
- There is no need for a controller to have a route to the subnet that the device resides in
- Consider the following



Thinking point

- AV is about providing a reliable service
- AV professional's thoughts are orientated to "making stuff work"
- AV professionals have pride in their work
- Audio and Video content is "valuable" to AV professionals
- The "service" is more valuable than the content to everybody else
- Network security is multidimensional
- Acquiring valuable content is actually very hard (AV especially)
- Disrupting or denying the service is a lot easier
- Social engineering is the most effective way to gain content
- DOS – Denial of Service is the most effective way to cripple a target
- Financial victory can be won by preventing a competitor delivering