Understanding Standards Track IETF MIB Modules

(SNMP Counters Tutorial)

Chris Elliott
Cisco Systems, Inc.

Michael R. MacFaden
Riverstone Networks, Inc.

draft#18

Goals

- Be able to differentiate between CLI counters and SNMP counters
- Learn how to locate counters in IETF Standards track documents
- Understand various SNMP counter types
- Review SNMP counters defined/Layer 1-3

Why Counters?

- Why counters matter…
  - Subsystem performance monitoring
    - errors
    - utilization/measure of activity
  - Most debugging activities require counters
    - fault isolation
  - Resource usage evaluation/planning
    - trending & thresholds
  - Basis for most billing applications

CLI Counters

- Command Line Interfaces
  - No standards body currently defines one
    - Yet most CLIs have common traits
  - Each counter is named
    - packets input, packets output
  - CLI Counters start at zero and increase in value
    - base starting point undefined, usually system start
  - CLI Counters may also decrease in value
    - Telco style event performance counters

CLI Counters

- The definition of what a given counter counts is dependent on vendor documentation
  - and on independent observation
- Are formatted for direct human consumption
  - 0 packets input, 0 packets output
- Many implementations provide command to clear/reset counter
  - clear interface ethernet 3

CLI Counters

- **Show** commands and expect scripting remain basic way of life in element management.

```
c4500#sh int e1
Ethernet1 is up, line protocol is down
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 input packets with dribble condition detected
    187352 packets output, 11347294 bytes, 0 underruns
    187352 output errors, 0 collisions, 3 interface resets
```

SNMP Counters

- Allow you to compare apples to apples
  - Counters have standard definitions
    - as defined by IETF, IEEE, some vendors…
    - regardless of network element type or vendor
  - and globally unique, hard to pronounce names
    - 1.3.6.1.2.1.17.2.4 dot1dStpTopChanges
- Have a well specified size
  - 32 or 64 bits wide
    - 64 bit data-type available in SNMP v2c or v3
    - Hacks for SNMPv1 include split counters

SNMP Counters

- Counters do not necessarily start at zero
  - Vendor implementation friendly
- Are not for direct human consumption
  - require a DELTA function to compute rate
- Can tell if the counter value polled is valid
  - Each counter has a well defined indicator that represents the validity of the sample taken known as a “discontinuity”

SNMP Counters

- Have well defined semantics

ifHCInOctets OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."
::= { ifXEntry 6 }

SNMP Counters

- Good counters are generally derived from underlying protocol specification

dot1dTpPortInFrames OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The **number of frames** that have been received by this port from its segment. Note that a frame received on the interface corresponding to this port is **only counted by this object if and only if it is for a protocol being processed by the local bridging function, including bridge management frames**."
REFERENCE
"IEEE 802.1D-1990: Section 6.6.1.1.3"

Units specified

Clearly specifies what to count

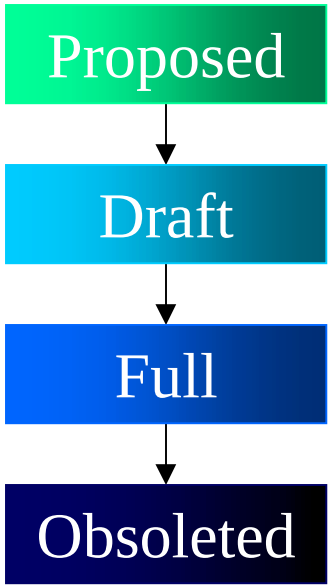
Internet Standards

RFC 1156 May 1990

The IAB has designated the SNMP, SMI, and the initial Internet MIB to be full "Standard Protocols" with "Recommended" status. By this action, the IAB recommends that all IP and TCP implementations be network manageable and that the implementations that are network manageable are expected to adopt and implement the SMI, MIB, and SNMP...

IETF Standards

- Beware, an RFC is not necessarily a standard.
  - Internet Drafts (I-D) (118 MIB modules in 101 drafts currently)
  - Standards Track Process defined in RFC 2026
    - Proposed (111 MIB modules in 105 RFCs)
    - Draft (25 MIB modules in 21 RFCs)
    - Full (11 MIB modules in 9 RFCs)
    - Obsoleted (83 MIB modules in 72 RFCs)
  - Non-standards-track MIB modules
    - Experimental (9 MIB modules in 9 RFCs)
    - Informational (9 MIB modules in 8 RFCs)
    - Historic (6 MIB modules in 5 RFCs)
  - IANA maintained documents: IANA-IF-TYPES



MIB module information

- How to stay informed on MIB Modules
- Operations & Management Area of
  - http://www.ietf.org http://www.rfc-editor.org
  - Specific web site for O&M
  - http://www.ops.ietf.org
- A mailing list:
  - mibs@ietf.org
- Bill Fenner's site:
  - http://www.aciri.org/fenner/mibs/

- Understanding various SNMP counter types

SNMP Counter Types

- Structure of Management Information
  - Version 1 RFC 1155
  - Version 2 RFC 2578-2580
  - Counter32 / Counter64
  - ZeroBasedCounter32
  - Integer32, Gauge32, are **not** counters
    - but can be the basis for new counter Textual-Conventions

RFC 2493
PerfCurrentCount
PerfIntervalCount
PerfTotalCount

RFC 2856
CounterBasedGauge64
ZeroBasedCounter64

SNMP Counter Types

- RFC 2578 Section 7.1.6. Counter32

  The Counter32 type represents a non-negative integer which monotonically increases until it reaches a maximum value of 2^32-1 (4294967295 decimal), when it wraps around and starts increasing again from zero.

  Counters have no defined "initial" value, and thus, a single value of a Counter has (in general) no information content.

NOTE: Counters may increase by more than one

SNMP Counter Types

RFC 2493 Jan 1999

perfCurrentCount ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A counter associated with a performance measurement in a current 15 minute measurement interval. The value of this counter starts from zero and is increased when associated events occur,

until the end of the 15 minute interval.

[snip]

SYNTAX Gauge32

Counter Identification

- Need a scheme that allows two vendors or products within a vendor to compare like items.
  - Object Identifiers (OID) were chosen as the identification scheme.
  - An OID is an ordered sequence of non-negative integers written left to right, containing at least two elements (0.0)
  - Bound to simple names in MIB Modules:
    - “ifInOctets” is 1.3.6.1.2.1.2.2.1.10

Counter Identification

- OIDs are not limited to SNMP protocol
  - Are useful, globally unique values that can be used for identifying anything.
- Once a MIB module is published, OIDs are bound for all time to the objects defined.
  - Objects can not be deleted! See RFC 2665
    - Can only be made obsolete
    - Even minor changes to an object are discouraged

Counter Identification

- Most common prefixes are:
  - 1.3.6.1.2.1 - contains MIB-II/std. objects
  - 1.3.6.1.3 - experimental MIB modules
  - 1.3.6.1.4.1 - contains vendor's objects
- IEEE 802.3ad Link Aggregation is:
  - 1.2.840.10006.300.43
- Enterprise OIDs are delegated by IANA

Tools for Managing OIDs

- Useful tools for managing OID/names
  - libsmi (open source)
    - http://www.ibr.cs.tu-bs.de/projects/libsmi/
      - smidump -f identifiers
  - SMICng (commercial)
    - http://www.snmpinfo.com
      - smicng -L

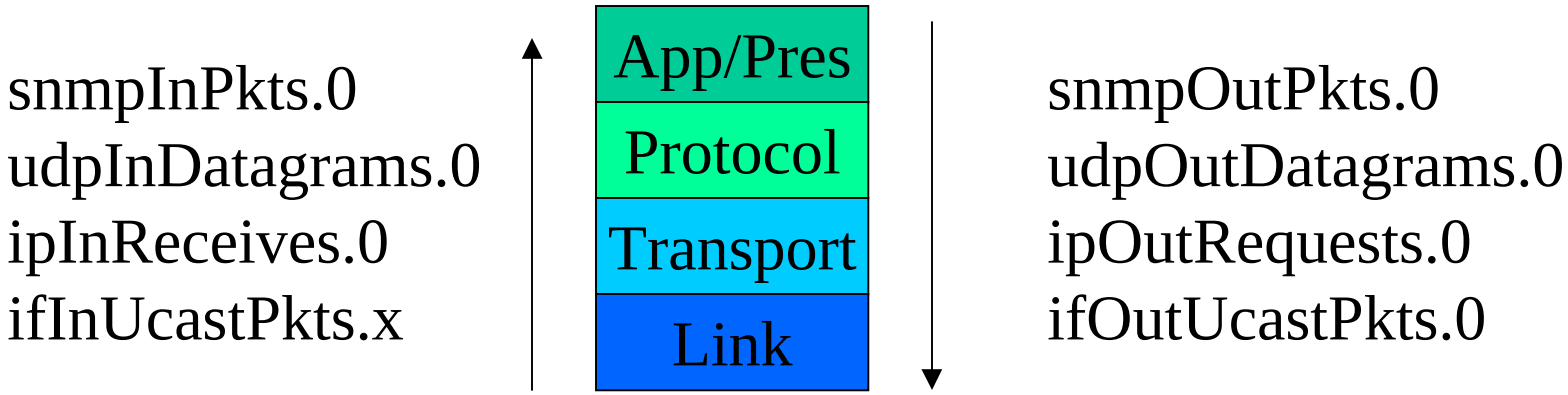
Example OID Report

```
-- List format from SMICng version 2.2.0.7
1.3.6.1.2.1.31.1.1 TOT: ifXTable[IF-MIB]
1.3.6.1.2.1.31.1.1.1 ROT: ifXEntry[IF-MIB] aug: ifEntry[IF-MIB]
1.3.6.1.2.1.31.1.1.1.1 COT: ifName[IF-MIB] syn: DisplayString[SNMPv2-TC]acc:ro
1.3.6.1.2.1.31.1.1.1.2 COT: ifInMulticastPkts[IF-MIB] syn: Counter32 acc: ro
1.3.6.1.2.1.31.1.1.1.3 COT: ifInBroadcastPkts[IF-MIB] syn: Counter32 acc: ro
1.3.6.1.2.1.31.1.1.1.6 COT: ifHCInOctets[IF-MIB] syn: Counter64 acc: ro
1.3.6.1.2.1.31.1.1.1.7 COT: ifHCInUcastPkts[IF-MIB] syn: Counter64 acc: ro
1.3.6.1.2.1.31.1.1.1.8 COT: ifHCInMulticastPkts[IF-MIB] syn: Counter64 acc: ro
1.3.6.1.2.1.31.1.1.1.9 COT: ifHCInBroadcastPkts[IF-MIB] syn: Counter64 acc: ro
1.3.6.1.2.1.31.1.1.1.14 COT: ifLinkUpDownTrapEnable[IF-MIB]
         syn: ENUM{ enabled(1) disabled(2) } acc: rw
1.3.6.1.2.1.31.1.1.1.15 COT: ifHighSpeed[IF-MIB] syn: Gauge32 acc: ro
1.3.6.1.2.1.31.1.1.1.17 COT: ifConnectorPresent[IF-MIB] syn: TruthValue
1.3.6.1.2.1.31.1.1.1.18 COT: ifAlias[IF-MIB] syn: DisplayString
1.3.6.1.2.1.31.1.1.1.19 COT: ifCounterDiscontinuityTime[IF-MIB] syn: TimeStamp
```

TOT - table object type
ROT - row object type
COT - columnar object type
SOT - scalar object type

Getting Counters

- Beware: retrieving counters can affect the values one is retrieving inband/out-of-band.
  - A given SNMP GET/GETNEXT to a network element will increment at least these counters:



Getting Counters - PDU Size

- Understand how large your PDU's are.
  - Standard specifies agent must support 484
  - MTU of most networks is 1500 bytes
  - The max SNMP/UDP/IP PDU can be 65518 with ip fragmentation, but is very, very costly and may not be supported by many agents and managers.
  - Agents have a max PDU size they accept and create
    - else snmpInTooBig, snmpOutTooBig will increment

Getting Counters - PDU Size

- one ifTable counter, community = 5 bytes
- net-snmp 4.0 (open source)/snmpget
  - Can fit 80 32-bit integer varbinds per 1500 byte MTU

```
# snoop -S between mgr agent
  mgr -> agent     length: 1498  UDP D=161 S=37913 LEN=1464
  agent -> mgr     length: 1447  UDP D=37913 S=161 LEN=1413
```

- SNMP Research 15.1.0.8(commercial)/getone
  - Can fit 83 32-bit integer varbinds per 1500 byte MTU

```
  mgr -> agent     length: 1402  UDP D=161 S=53411 LEN=1368
  agent -> mgr     length: 1513  UDP D=53411 S=161 LEN=1479
```

Data Collection Best Practices

- How you poll counters/form requests can impact the quality of the data for analysis
- How you poll for counters can skew your information/graphs.
- Skew defined per Webster's dictionary:
  - To give a bias to; distort.
- Time is the major factor causing skew

Data Collection Best Practices

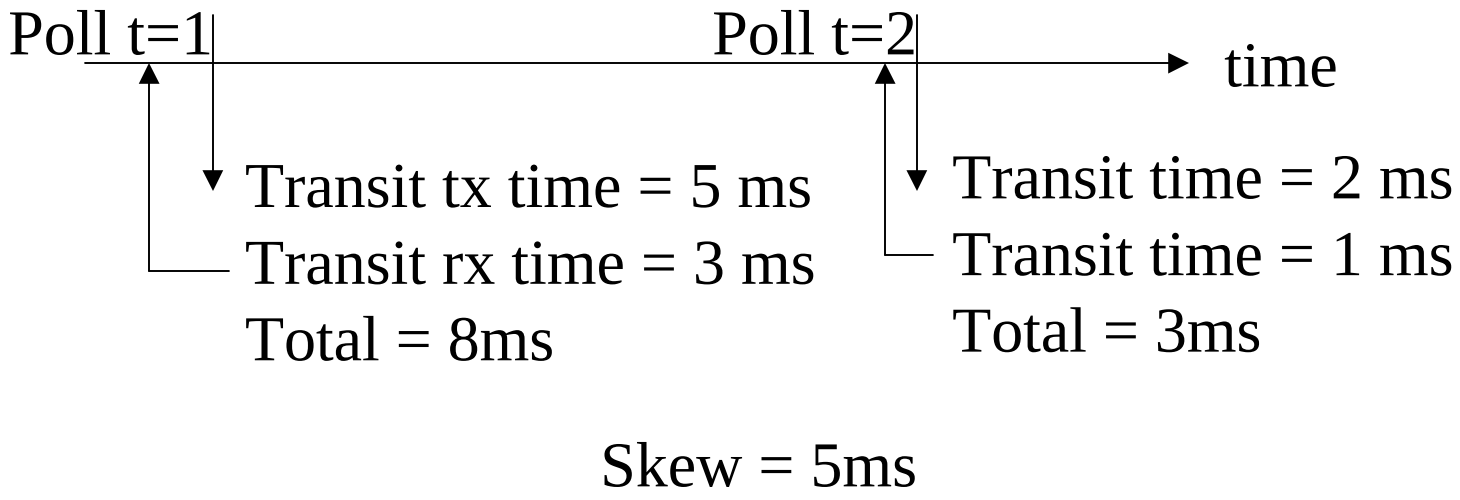
Time Skew

- Group multiple objects in a given Get or GetNext request to minimize time differences in sampling like objects.
- $GET_{t=1,2,3,..}$ { (x = ifIndex, y = time)
  - ifInUCastPkts.x, ifOutUCastPkts.x,
  - ifInDiscards.x, ifOutDiscards.x
  - ifInErrors.x, ifOutErrors.x
  - sysUpTime.0, ifCounterDiscontinuityTime.x }

Data Collection Best Practices

Time Skew

- When calculating the delta time between two polling requests, use sysUpTime from the device itself and not the management station to avoid transit time skew.



Data Collection Best Practices

Using Perf Counters

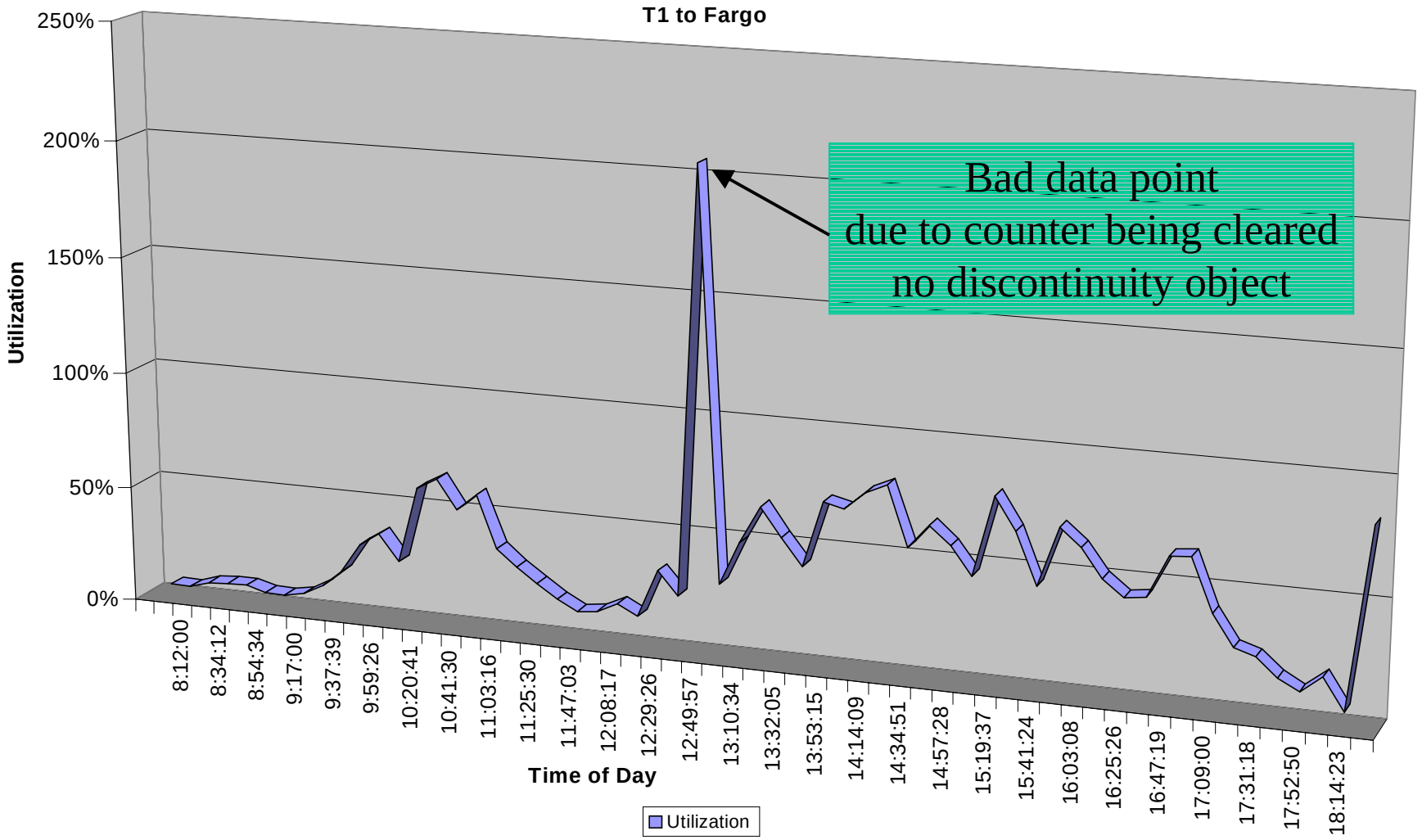
- All digital circuit interfaces (DS0, DS1, E1, DS3, E3, SONET, SDH) use time based counters
  - **PerfCurrentCount** (RFC 2493)
    - Current Interval counters can decrease in value
    - Must align polling with device on 15 minute boundaries
  - **PerfIntervalCount**
    - Provides history up to 24 hours in 96 15-minute intervals
- All devices and management stations need to be in time sync—use NTP

Data Collection Best Practices
Counter Discontinuity

- Counters should not be reset without a way to determine the reset
  - Leads to inaccurate delta calculations
- Two ways to determine a counter reset:
  - Polling sysUpTime for reset
    - Reset every time SNMP agent is reset
    - Note: sysUpTime wraps every 1.36 years
  - Poll the discontinuity timer if it exists
    - Look in the description of the counter in the MIB module

Data Collection Best Practices

Counter Discontinuity



Data Collection Best Practices

Counter Discontinuity

ifCounterDiscontinuityTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of sysUpTime on the most recent occasion at which any one or more of this interface's counters suffered a discontinuity. The relevant counters are the specific instances associated with this interface of any Counter32 or Counter64 object contained in the ifTable or ifXTable. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object contains a zero value."
::= { ifXEntry 19 }

Data Collection Best Practices

Counter Discontinuity

- For each counter polled:

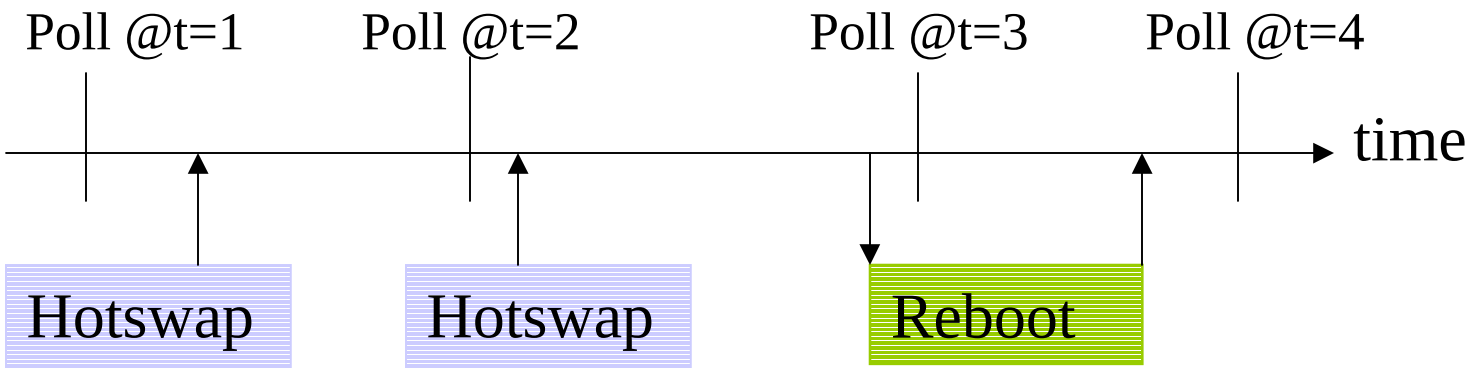
  collect the discontinuity managed object

- $GET_{y=1,2,3,..}$ **{** (x == ifIndex, y=time)
  - **ifCounterDiscontinuityTime.x,**
  - sysUpTime.0
  - ifInUCastPkts.x, ifOutUCastPkts.x,
  - ifInDiscards.x, ifOutDiscards.x
  - ifInErrors.x, ifOutErrors.x **}**

Throw out deltas where discontinuity does not match previously polled value

Data Collection Best Practices

ifIndex Changing



If same type of line-card is reinserted into same slot, ifIndex must be reused.

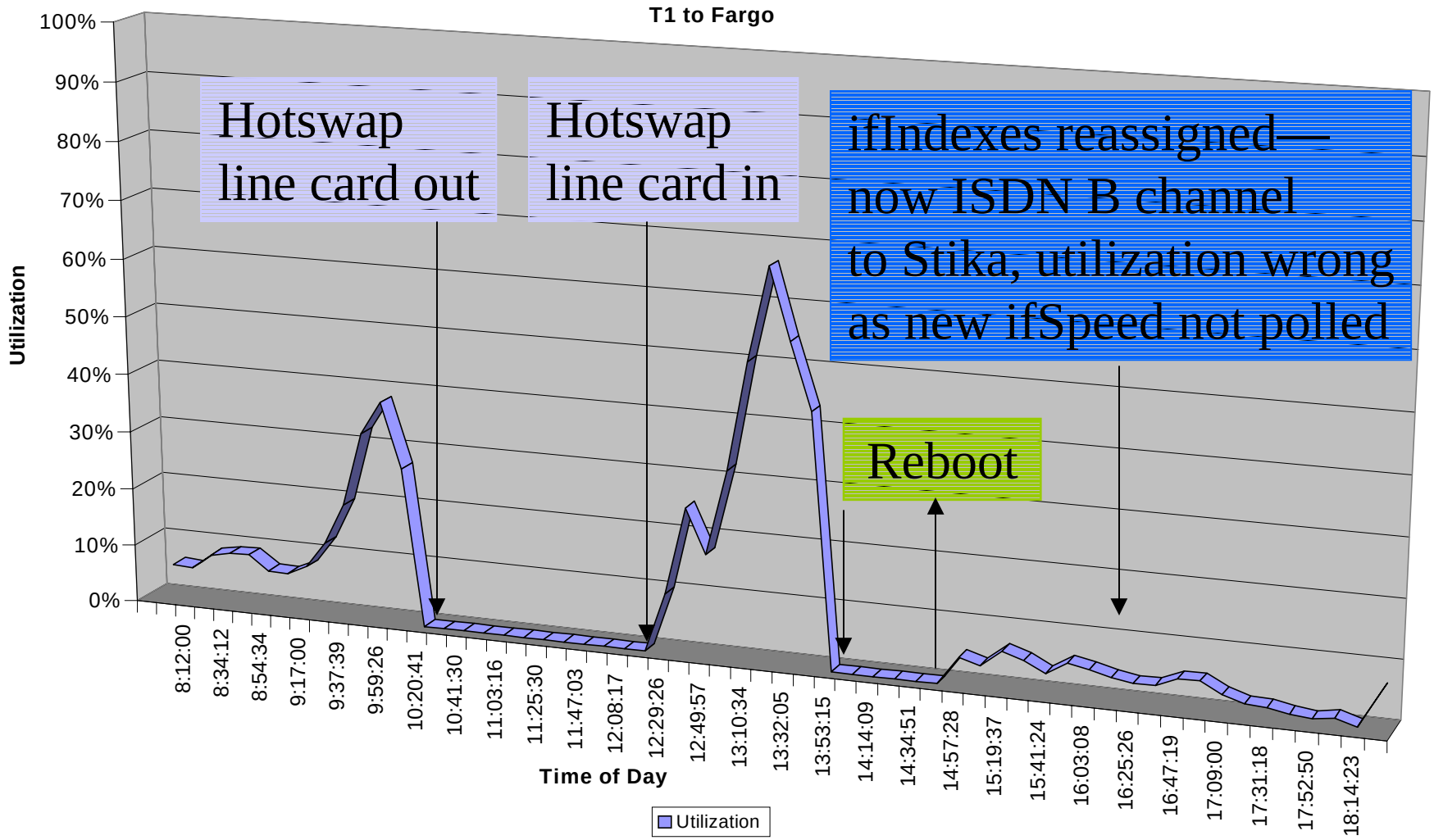
RFC 2863

ifIndexes can be reassigned across reboots.

Use ifAlias to track reassignment

Data Collection Best Practices

ifIndex Changing



Data Collection Best Practices

Setting Minimum Poll Interval

- What is the minimum polling interval?
  - Different implementations of a given counter can differ reasonably differ in refresh rate:
  - When testing vendors products, determine if the counter refresh rate is acceptable for your deployment needs
    - < 1 second, 1 minute, 5 minutes, 15 minutes, …
  - Remember to calculate data storage requirements
  - Careful! Check vendor hardware/software counter size underlying the SMI counter !!

Data Collection Best Practices

Setting Minimum Poll Interval

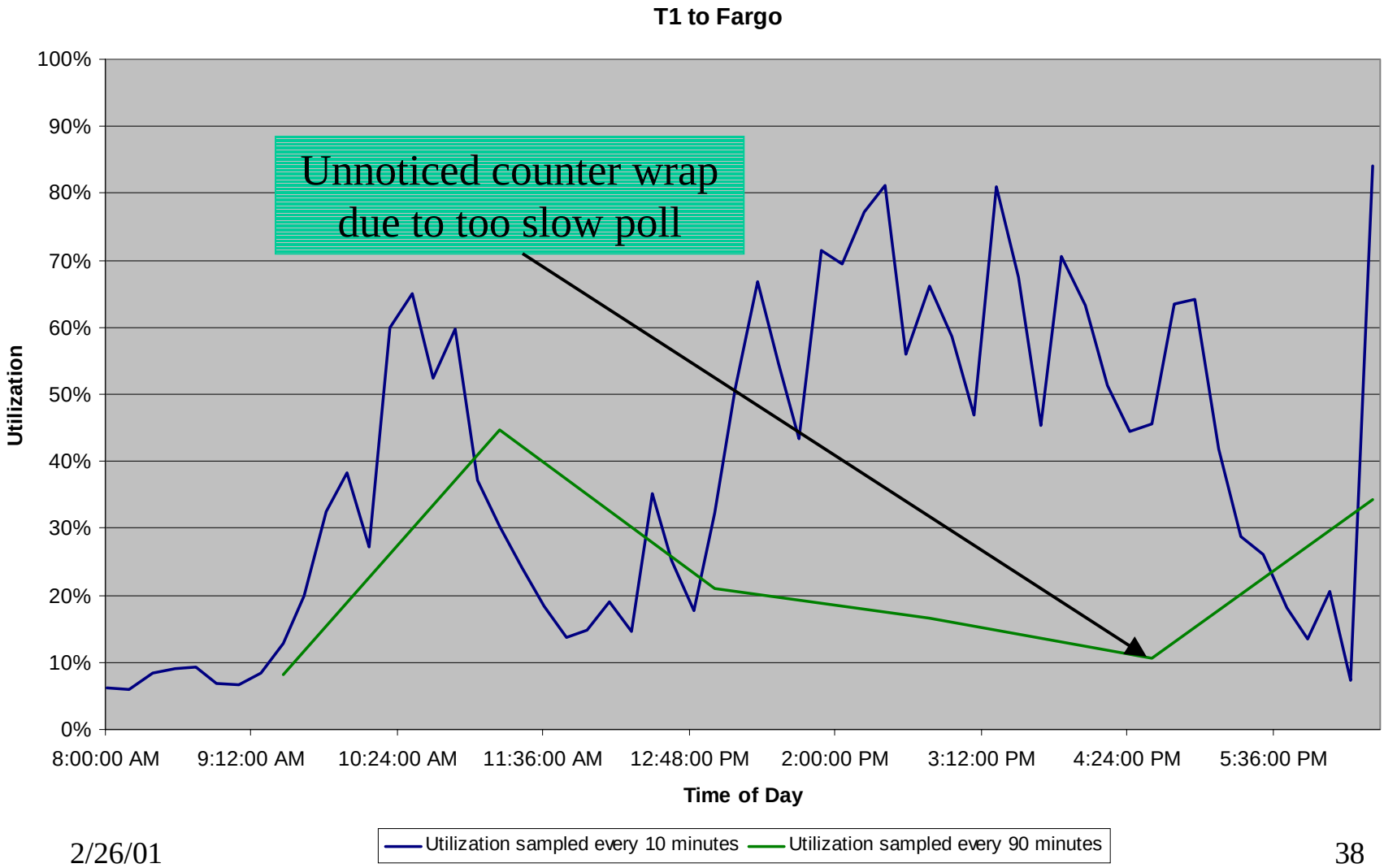
- Additional checks in determining minimum poll interval
  - Verify CPU Load on device is acceptable
  - Verify management traffic load created is acceptable
  - Wrap time for a given counter:

32-bit counters by link speed/sec:	
10M	57.26 minutes
100M	5.73 minutes
155M	3.69 minutes
1Gig	34 seconds

64-bit counters by link speed/sec:	
1 Terabit	~5 years
81,000,000 Terabits	30 minutes

Data Collection Best Practices

Setting Minimum Poll Interval



Data Collection Best Practices
Counter Width

- Determine which counter size to poll for a given managed object.
  - 64 bit counters are often named “High Capacity” or HC as in ifHCInOctets
  - Another strategy is high/low 32 bit objects
  - Section 3.1.6 of RFC 2863 provides to vendors as follows for IETF Standards abiding Agent implementations for byte/packet counters

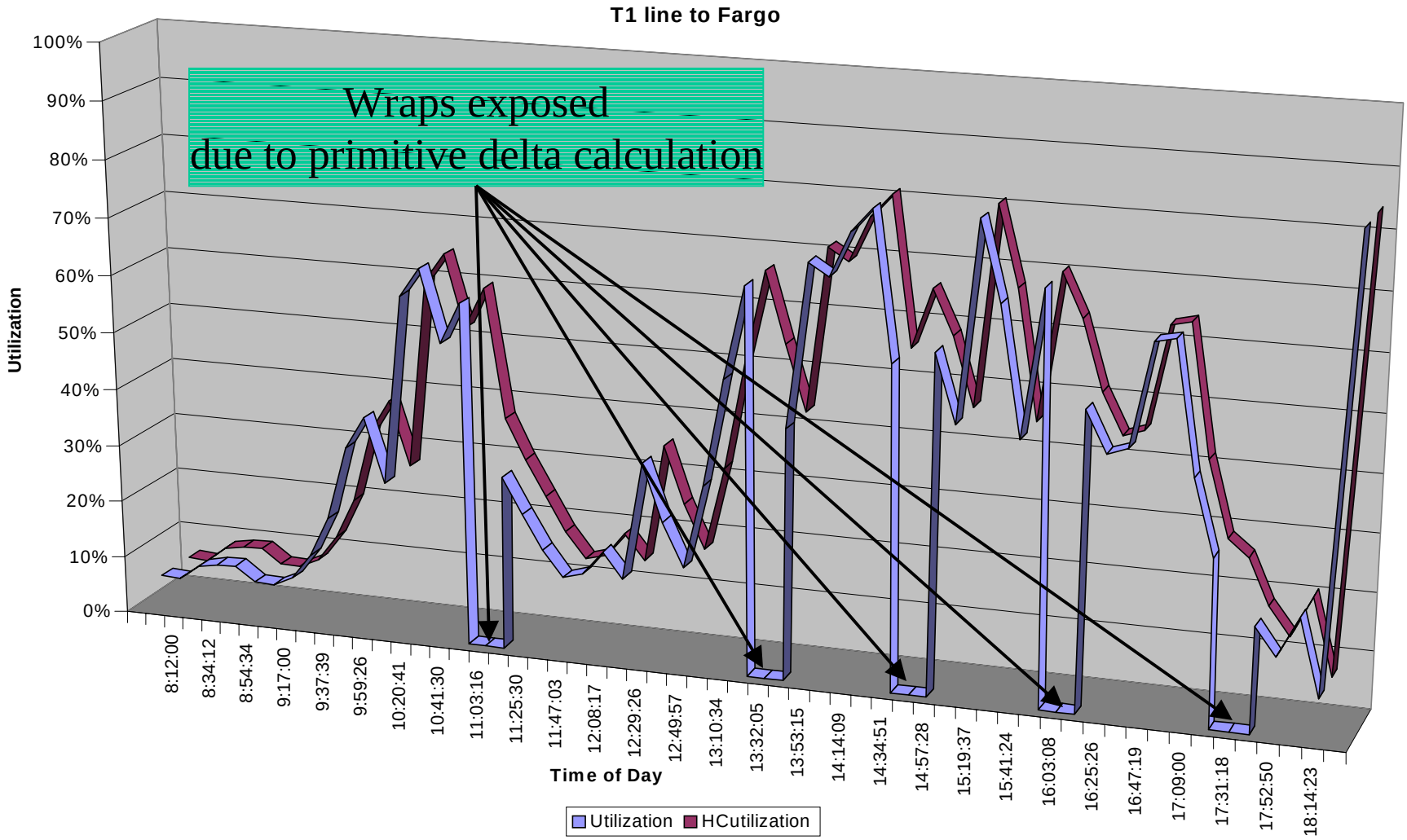
Data Collection Best Practices

Counter Width

- Abiding IETF Implementations will provide byte/packet counters at widths of:
  - ifSpeed <= 200 Mbps
    - 32-bit byte and packet counters
  - ifSpeed > 200 Mbps && < 650 Mbps
    - 32-bit packet counters and 64-bit byte counters
  - ifSpeed >= 650 Mbps
    - 64-bit byte and packet counters
  - Implementations may provide additional counters, i.e. 64-bit byte counters for 100M interfaces

Data Collection Best Practices

Visualization of Wraps on T1 line



Where to find SNMP Counters

- MIB I RFC 1066 defined in 1988
  - 79 Counters defined
    - interface: 11
    - ip: 16
    - icmp: 26
    - udp: 4
    - tcp: 7
    - egp: 4
- 21 Defined interface types
  - ethernet, token-ring, fddi, isdn, t1, etc.

Where to find SNMP Counters

- MIB-2 (RFC 1213) March 1991
  - RFC 1573 updates Jan 1994
    - Deprecated some counters
    - ifInNUCast deprecated and replaced by two counters: ifInMCast, ifInBCast…
    - Modified how ifIndex is used
    - Allowed sparse tables—i.e. data that doesn't make sense shouldn't exist (if[In|Out]Errors for subinterfaces, for example)
  - new device management and applications SHOULD use new MIB modules/counters and fall back to older or deprecated counters only as necessary. Now 10 years of compatibility !

Base Counter Set

- November 2000, MIB2 exists in 7 RFCs

	Counters		
	MIB-II	Subsequent	
– 1907 system	0	0	
– 2863/4 interfaces	22	11	
– 2096 ipCidrRouteTable	0	0	
– 2011 ip counters	17	1	
icmp counters	26	0	
– 2012 tcp counters	9	2	
– 2013 udp counter	4	0	
Totals	**78**	**14**	**= 92**

Counter Organization

- Two major indexing methods exists in Standards based SNMP Agents to identify physical and logical ports on IP routers and bridges:
  - ifIndex (1..2147483647)
    - RFC 2683 IF-MIB
  - dot1dBasePortNum (1..65535)
    - RFC 1493 BRIDGE-MIB, RFC 2674 Q/P-BRIDGE
- Most Enterprise MIB modules will provide a mapping to these standard indexes.

Network Interfaces

- Is an ifIndex logical or physical?
  - In old days, one used ifType, but these values are not generally well defined. Some RFCs define the value to use EtherLike-MIB, Frame Relay, ATM... others do not, such as 802.1Q VLANS in RFC 2674.
  - Use ifConnectorPresent.x to determine if you are looking at a logical or physical interface.

Accessing Counters - Indexing

- The BRIDGE-MIB provides a mapping from dot1dBasePort to ifIndex in the dot1dBasePortTable:

dot1dBasePortIfIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The value of the instance of the ifIndex object, defined in MIB-II, for the interface corresponding to this port."
::= { dot1dBasePortEntry 2 }

Accessing Counters - Indexing

- IfIndex's are not by required to be fixed for all time to a given slot/port.
  - Only rule is that ifIndex can't change from one media to another w/o disrupting sysUpTime
- Routed vs Bridged Ports
  - All ports that bridge have an ifIndex and dot1dBasePortNum assigned
  - If a port does not perform bridging, no dot1dBasePortNum is assigned

Identifying Indexes

Or how do I find counters for POS3/5?

- What ifIndex corresponds to my CLI description of “interface ethernet 0”?
- Use ifName managed object in ifXTable!
  - Its sole purpose is to cross reference the Command Line Interface representation of a given port, ex: ifIndex 23 ~ pos 3/5
- ifDescr or ifAlias may also have it, but that is at the discretion of the vendor.

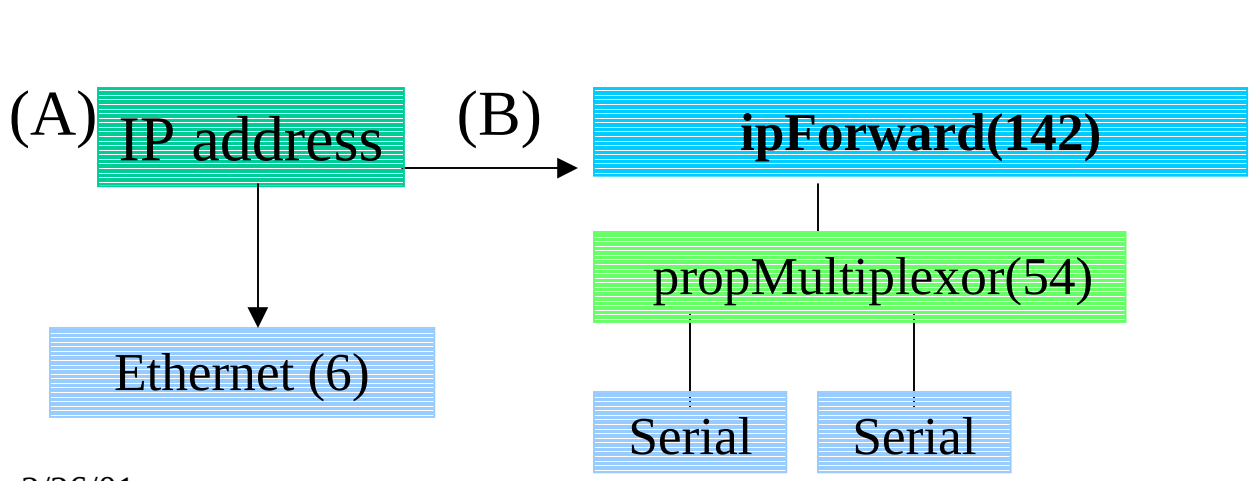
Helpful CLI mappings

- Sometimes CLI will also show translation back to SNMP...

```
Console> (enable) show vlan 998
VLAN Name                        Status    IfIndex Mod/Ports, Vlans
---- ------------------------------ --------- ------- ----------------------
998  VLAN0998                     active    357
```

Interface Layering

- What ifIndex corresponds to my IP address?
- ipAddrTable - lists all IP Addresses
- Provides an ifIndex pointer to interface
  - (A) For some devices this is the physical port
  - (B) For others, it is a pointer to a logical layer



ifStackTable provides ifIndex mappings

Layer 2 Addressing

- What ifIndex corresponds to a given MAC?
  - Two standard ways to find this:
  - Use the ARP Table
    - ipNetToMedia
      - Indexed by ifIndex, and MAC address
  - Use the transparent Bridge Table
    - dot1dTpFdbTable
      - Indexed by source MAC address

- Review SNMP counters defined/Layer 1-3

Layers 1-2

- Ethernet
- 803.3ad Link Aggregation
- Serial
- Cable Modem
- PPP
- Frame Relay
- ATM
- MPLS
- 802.1D/Q Bridging
- RMON

Ethernet

- EthernetLike-MIB (dot3) RFC 2665
  - obsoletes RFC 2358 which obsoletes RFC 1650
- Now supports all flavors of Ethernet
  - 10, 100, and 1000 megabit/sec

New Gig Ethernet counters
dot3InPauseFrames,
dot3OutPauseFrames
dot3ControlInUnknownOpcodes

dot3PauseTable provides: dot3StatsDuplexStatus

Ethernet RFC 2665

- dot3StatsTable
  - By IfIndex (dot3StatsIndex)
- dot3StatsAlignmentErrors
- dot3StatsFCSErrors
- dot3StatsSingleCollisionFrames
- dot3StatsMultipleCollisionFrames
- dot3StatsDeferredTransmissions
- dot3StatsLateCollisions
- dot3StatsExcessiveCollisions
- dot3StatsInternalMacTransmitErrors
- dot3StatsCarrierSenseErrors
- dot3StatsFrameTooLongs
- dot3StatsInternalMacReceiveErrors

Ethernet

- MAU-MIB RFC 2668
- New Gig Ethernet Support:
  - rpMauFalseCarriers
    - A count of the number of false carrier events during IDLE in 100BASE-X links
  - rpMauMediaAvailableStateExits
    - A count of the number of times that rpMauMediaAvailable for this MAU instance leaves the state available(3).

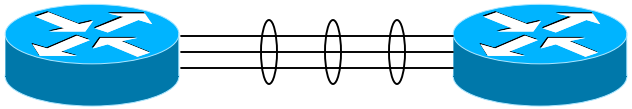
Ethernet

rpJackTable - Provides connector type

other(1),
rj45(2),
rj45S(3), -- rj45 shielded
db9(4),
bnc(5),
fAUI(6), -- female aui
mAUI(7), -- male aui
fiberSC(8),
fiberMIC(9),
fiberST(10),
telco(11),
mtrj(12), -- fiber MT-RJ
hssdc(13) -- fiber channel style

Link Aggregation

- An IEEE 803.2ad/D3.0 Specification
  - A group of links between two nodes that appear to a MAC client as if they were a single link.
  - Defines LAG-MIB—dot3adAggPortStatsTable



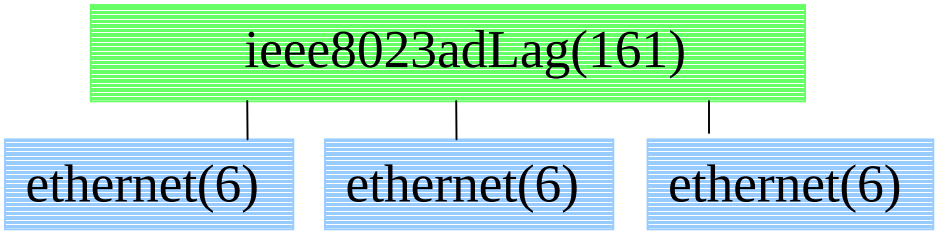
LA Control Protocol

Counters Indexed by: ifIndex

dot3AggPortStatsLACPDUsRx
dot3adAggPortStatsMarkerPDUsRx
dot3adAggPortStatsMarkerResponsePDUsRx
dot3adAggPortStatsUnknownRx
dot3adAggPortStatsIllegalRx
dot3adAggPortStatsLACPDUsTx
dot3adAggPortStatsMarkerPDUsTx
dot3adAggPortStatsMarkerResponsePDUsTx

Link Aggregation

- ifTable row uses ifType
  - ieee8023adLag(161)
    - Vendor pre-standard implementations may use propMultiplexor(54)



    - To find port membership, use ifStackTable or use the dot3adAggPortListPorts object
      - Latter provides a more compact representation of port membership w/o details of what these ports are.

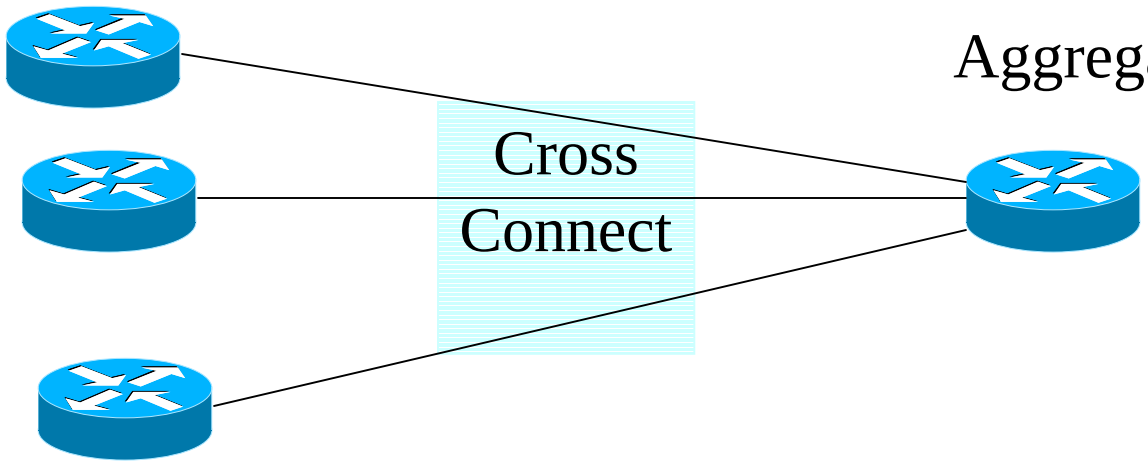
Serial Interfaces RFC 2494-6

- Serial Interfaces (DS0/DS1/E1/DS3/E3/SONET/SDH)
  - Use of PerfHist counters which are GAUGE
    - current values can decrease
  - Some IP devices keep only current 15 minutes others can keep complete 24 hours worth of 15 minute statistics.
  - No requirement for 15 minute intervals to match current wall clock time.

Serial Interfaces

- Since counters are kept in 15 minute buckets, one can correlate performance across a TDM circuit.
- Performance is calculated using error counters instead of packet discard counters, i.e.: Bit Error Rate
  - Errored Seconds
  - Severely Errored Seconds
  - Unavailable Seconds

Serial Interfaces



Chan T1/E1 ifTable

ds0bundle(82)
ds0(81)$_{1..n}$
ds1(18)*
v34(45)

*DS1/E1 should use the Same ifType, DS1(18) as of RFC 2495. Look at the DS1-MIB to distinguish. Same for DS3/E3—use the DS3-MIB (RFC 2496).

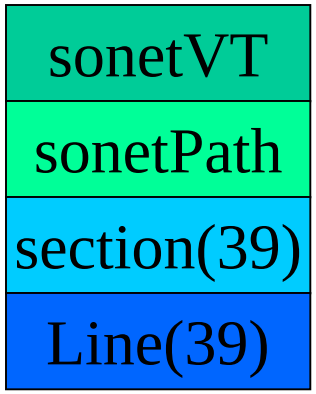
Chan T3/E3 ifTable

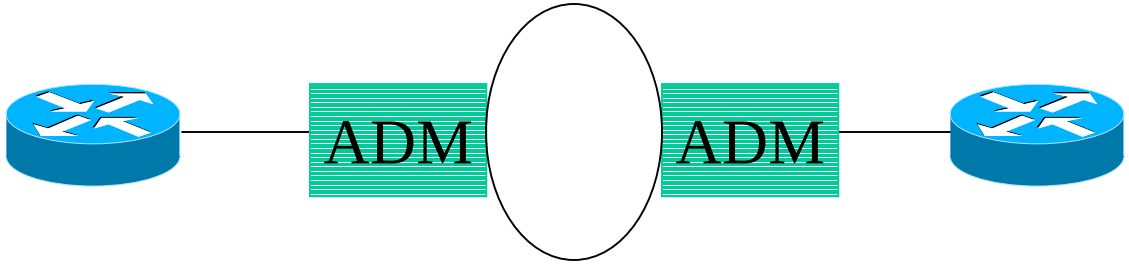
ds0bundle(82)
ds0(81) $_{1..n}$
ds3(30)*
hssi(46)

optional

Serial Interfaces

ifTable/ifXTable





End to End Circuit Monitoring of

- Coding Violations (ES)
- Defects (LOS, SEF, AIS)

Current 15 minutes	
UAS	0
ES	1
SES	2
SEF	1
CV	15

IntervalTable (ifIndex.(1-96)		
UAS	0	
ES	1	
SES	2	
SEF	1	
CV	15	
ValidData	True(1)	

ANSI T1.231 UAS begins at onset of 10 SES.
All stats other than UAS must be frozen during this time.
As such, all counters must pass through a 10 second delay.

SONET/SDH

- Linear Automatic Protection Switching
- MIB Module currently under development
  - `draft-ietf-atommib-sonetaps-mib-02.txt`
- 1:n and 1+1 support

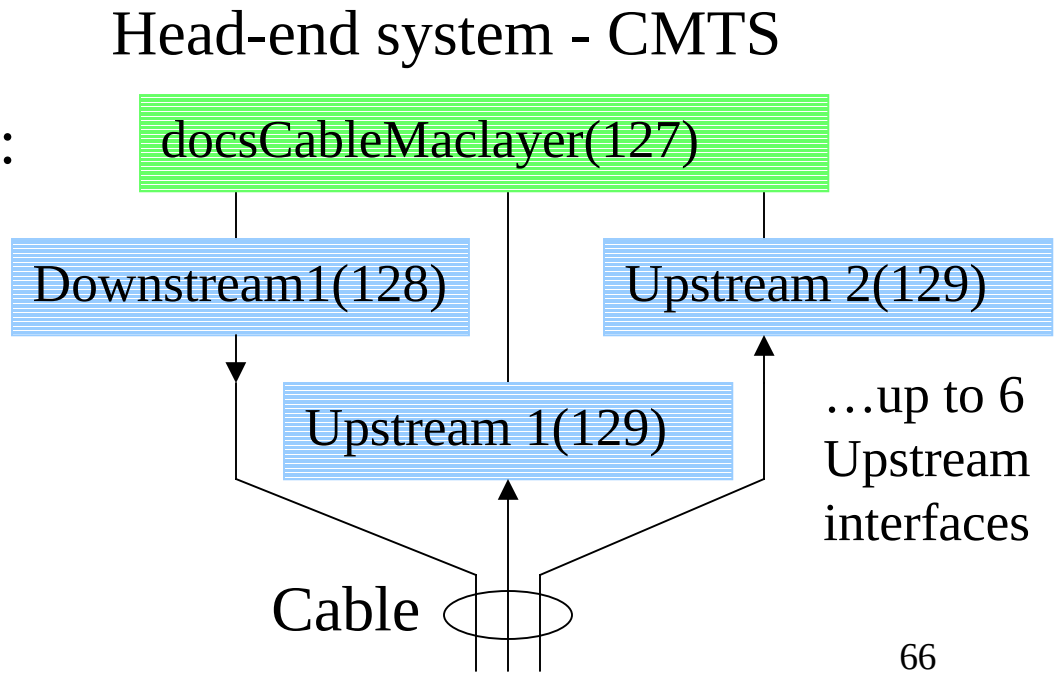
```
apsStatusTable
indexed by:
   apsConfigName
apsStatusModeMismatches
apsStatusChannelMismatches
apsStatusPSBFs
apsStatusFEPLFs
```

```
apsChanStatusTable
indexed by:
   apsChanGroupName,
   apsChanNumber
apsChanSignalDegrades
apsChanSignalFailures
apsChanSwitchovers
apsChanLastSwitchover
```

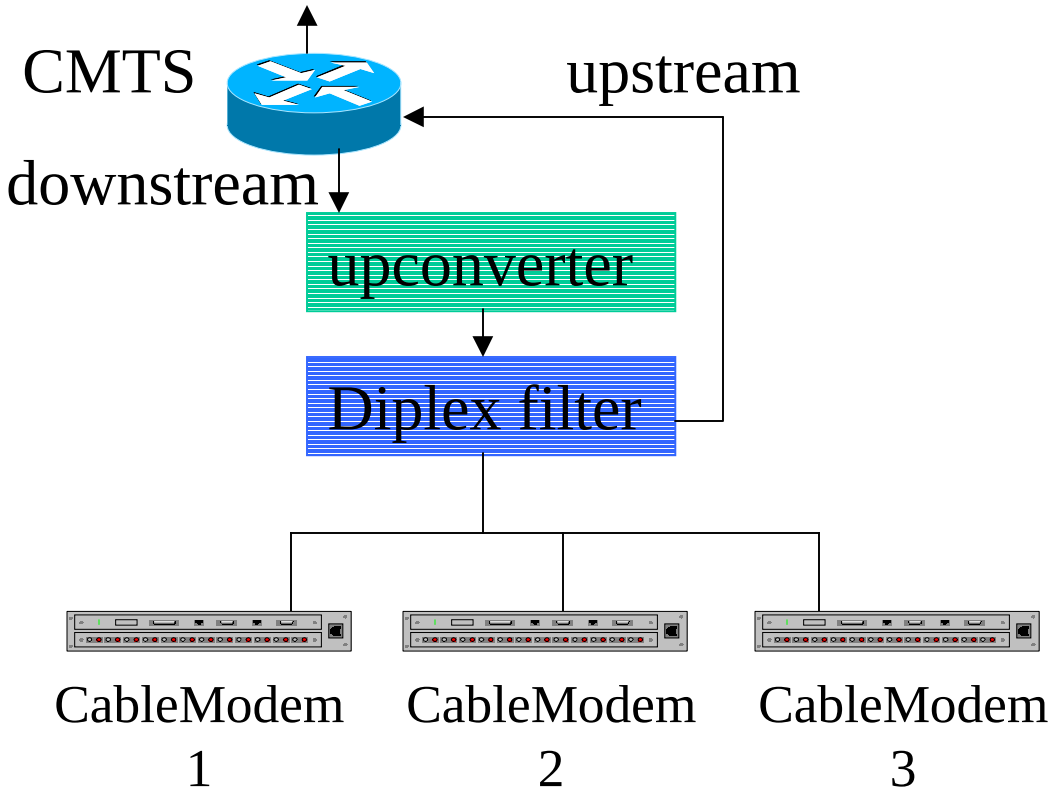
Cable Network RFC 2669-70

- A single logical Ethernet-like interface is made up of physical unidirectional sub-interfaces:

For CMTS, trick is getting Cable Modem to:
1. Range
2. Register



Sample Cable Network



Cable Networks

- RFC 2670 DOCS-IF-MIB provides 3 tables
  - All indexed by ifIndex
- docsIfCmtsStatusTable (CMTS only)
  - docsIfCmtsStatusInvalidRangeReqs
  - docsIfCmtsStatusRangingAborteds
  - docsIfCmtsStatusInvalidRegReqs
  - docsIfCmtsStatusFailedRegReqs
  - docsIfCmtsStatusInvalidDataReqs
  - docsIfCmtsStatusT5Timeouts
- docsIfSignalQualityTable (both)
  - docsIfSigQUnerroreds
  - docsIfSigQCorrecteds
  - docsIfSigQUncorrectables

•docsIfCmStatusTable (CM only)

•docsIfCmStatusResets
•docsIfCmStatusLostSyncs
•docsIfCmStatusInvalidMaps
•docsIfCmStatusInvalidUcds
•docsIfCmStatusInvalidRangingResp
•docsIfCmStatusInvalidRangingResponses
•docsIfCmStatusInvalidRegistrationResp
•docsIfCmStatusInvalidRegistrationResponses
•docsIfCmStatusT1Timeouts
•docsIfCmStatusT2Timeouts
•docsIfCmStatusT3Timeouts
•docsIfCmStatusT4Timeouts
•docsIfCmStatusRangingAborteds

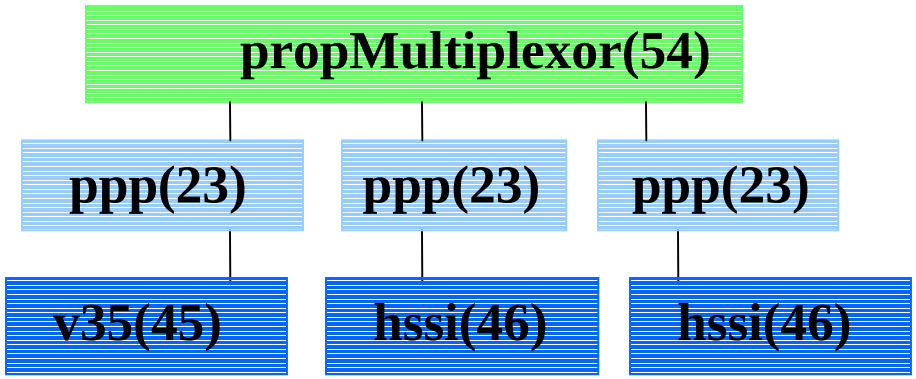
PPP - RFC 1471-4

- 1471 PPP-LCP-MB
  - Per ifIndex

    - pppLinkStatusBadAddresses
    - pppLinkStatusBadControls
    - pppLinkStatusPacketTooLongs
    - pppLinkStatusBadFCSs
    - pppLqrInGoodOctets
    - pppLqrInLQRs,
    - pppLqrOutLQRs

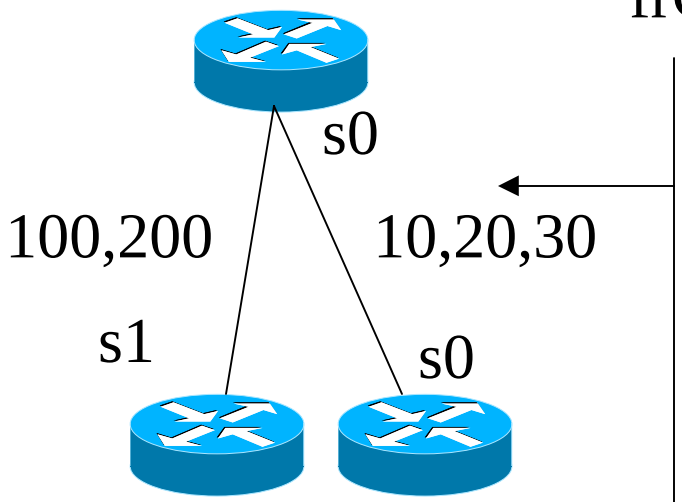
MultiLink PPP

- No RFC defines ML-PPP
  - However, can be represented in a standard way using RFC 2863 in ifTable/ifXTable/ifStackTable
  - Higher layer Counters are sums of underlying counters



pppMultilinkBundle(108) may also be used

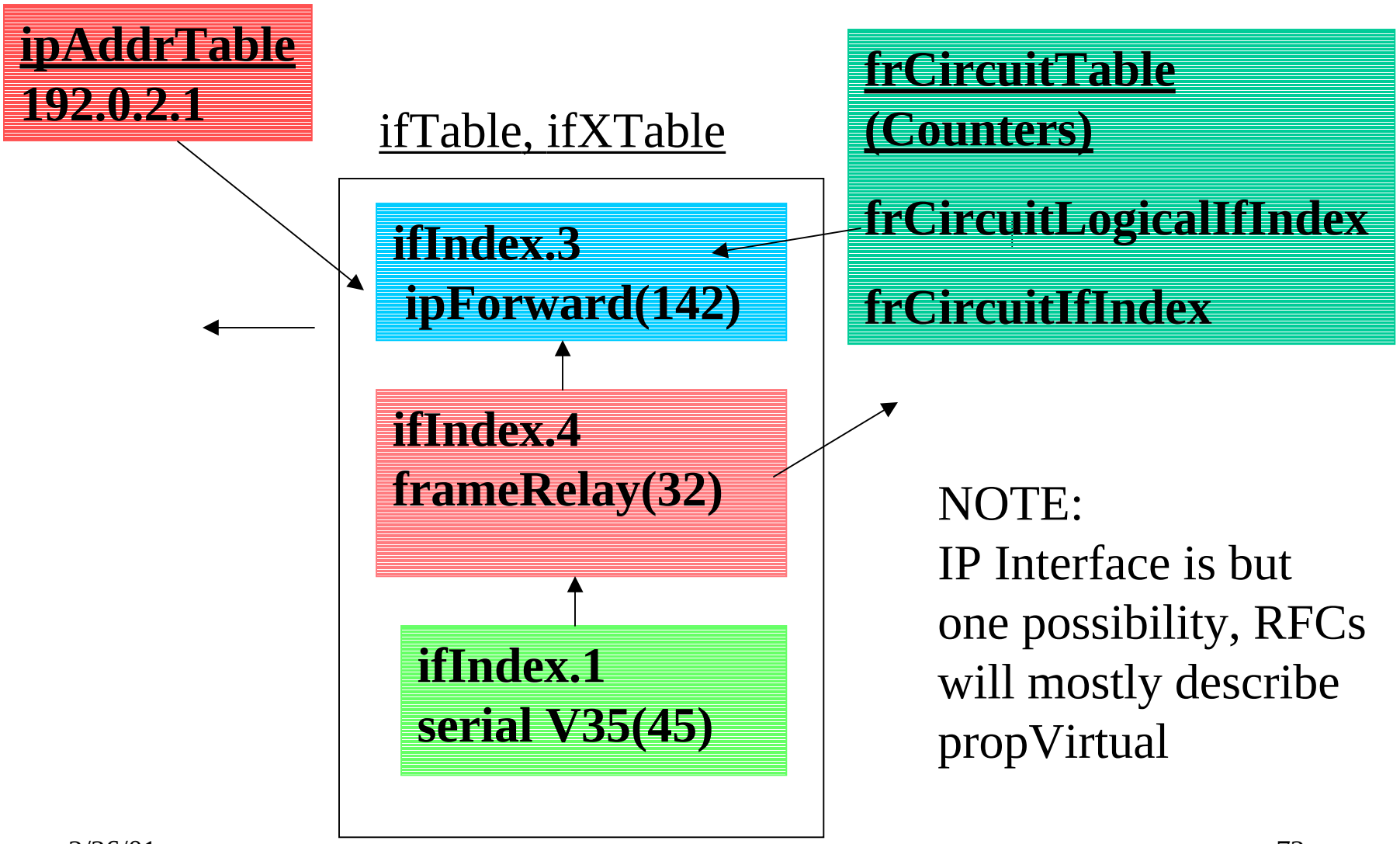
Frame Relay RFC 2115



frCircuitTable (Per DLCI)

- frCircuitReceivedFECNs
- frCircuitReceivedBECNs
- frCircuitReceivedDEs
- frCircuitSentDEs
- frCircuitReceivedFrames
- frCircuitReceivedOctets
- frCircuitSentFrames
- frCircuitSentOctets
- frCircuitDiscards

Frame Relay



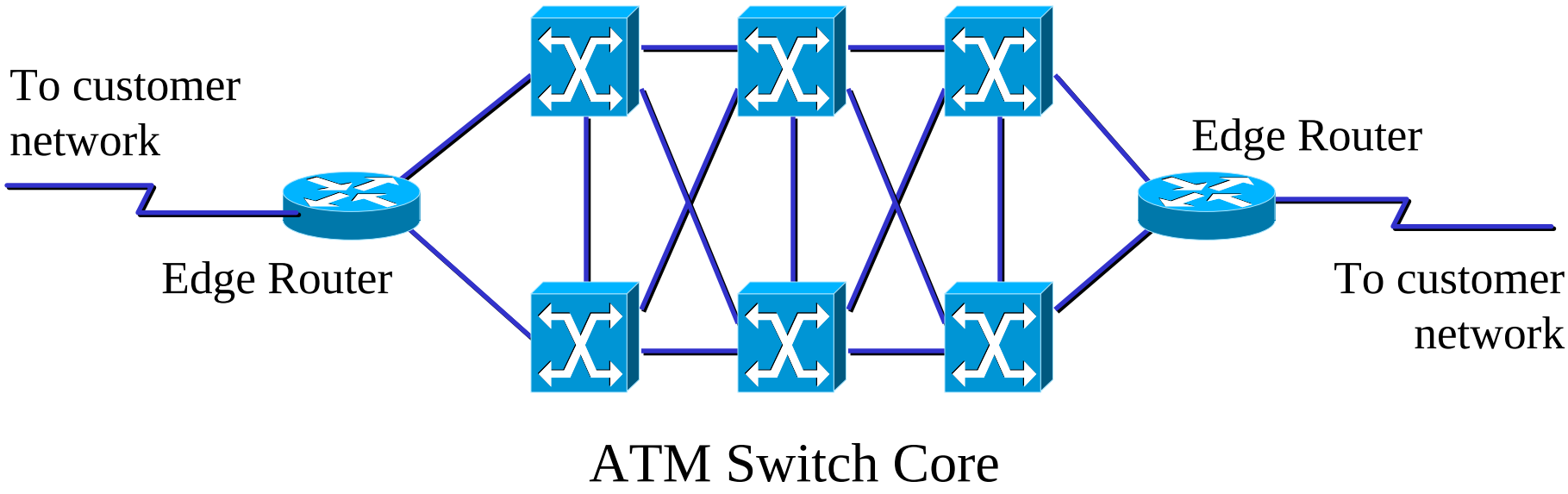
NOTE:
IP Interface is but one possibility, RFCs will mostly describe propVirtual

ATM

- RFC 2515 ATM-MIB (M2)
- ATM over DS3/SONET by ifIndex
  - atmInterfaceDs3PlcpSEFSs
  - atmInterfaceDs3PlcpUASs
  - atmInterfaceOCDEvents
- AAL5 by ifIndex, aal5VccVpi, aal5VccVci
  - aal5VccCrcErrors
  - aal5VccSarTimeOuts
  - aal5VccOverSizedSDUs

Byte, Packet, discard counters in ifTable/ifXTable by ifType aal5(49)

ATM Sample Network



ATM

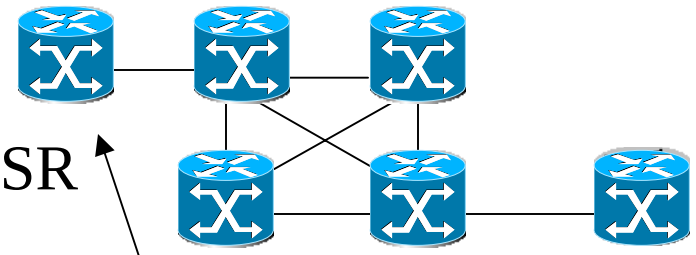
- atmVclTable manages per VC
  - Indexed by {ifIndex, atmVclVpi, atmVclVci}
  - Has ifTable like capabilities:
    - atmVclAdminStatus / atmVclOperStatus
    - atmVclLastChange
  - Unfortunately: no ifIndex pointer to identify upper layer components, use Enterprise MIB to map to IP address or MAC addresses associated with this.
  - Per VC stats found in B-ICI Specification,
    - ATM-RMON MIB (Cisco) plus CISCO-AAL5-MIB

MPLS

- Three drafts current in development:
  - draft-ietf-mpls-ftn-mib-00.txt
    - MPLS-FTN-MIB
    - Associate FEC with LSP
  - draft-ietf-mpls-lsr-mib-07.txt
    - MPLS-LSR-MIB
    - setup mpls per interface, LSPs, cross-connect, etc
  - draft-ietf-mpls-te-mib-05.txt
    - MPLS-TE-MIB
    - configures tunnels, tunnel resources

MPLS Sample Network

FTN-MIB



Edge LSR

LSR

Edge LSR

mplsFTNPerfTable
index:IfIndex, MapCurrIndex
mplsFTNMatchedPackets
mplsFTNMatchedOctets

ifTable Counters

mplsTunnel(150)

mpls(166)

ethernet(6)

LSR-MIB

mplsInterfacePerfTable
Index: ifIndex

mplsInterfaceOutFragments
mplsInterfaceFailedLabelLookup

MPLS

- Currently draft-ietf-mpls-lsr-mib-07.txt
  - mplsInSegmentPerfTable
    - Index: mplsOutSegmentIndex
  - mplsInSegmentOctets
  - mplsInSegmentPackets
  - mplsInSegmentErrors
  - mplsInSegmentDiscards

MPLS

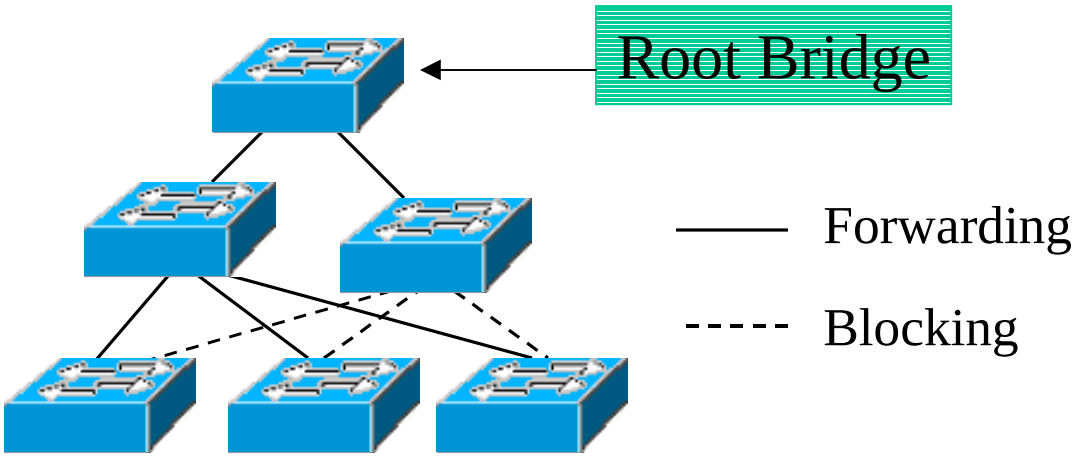
- MPLS-TE-MIB
  - mplsTunnelPerfTable
  - indexed by
    - mplsTunnelIndex, mplsTunnelInstance,
    - mplsTunnelIngressLSRId, mplsTunnelEgressLSRId
  - Counters provided
    - mplsTunnelPerfPackets
    - mplsTunnelPerfPackets
    - mplsTunnelPerfErrors

mplsTunnelPerfHCBytes
mplsTunnelPerfHCPackets

IEEE 802.1D Bridging

- Most important counters deal with knowing stability of topology. RFC 1493 defines the following for Spanning Tree Protocol (STP):
  - dot1dStpTopChanges
    - Number of times Spanning Tree recalculated topology
  - dot1dStpTimeSinceTopologyChange
    - Indicates how long bridged network has been stable
  - dot1dStpDesignatedRoot
    - mac addr of root bridge - should track this
  - dot1dStpRootPort
    - The port by which the root bridge is reached

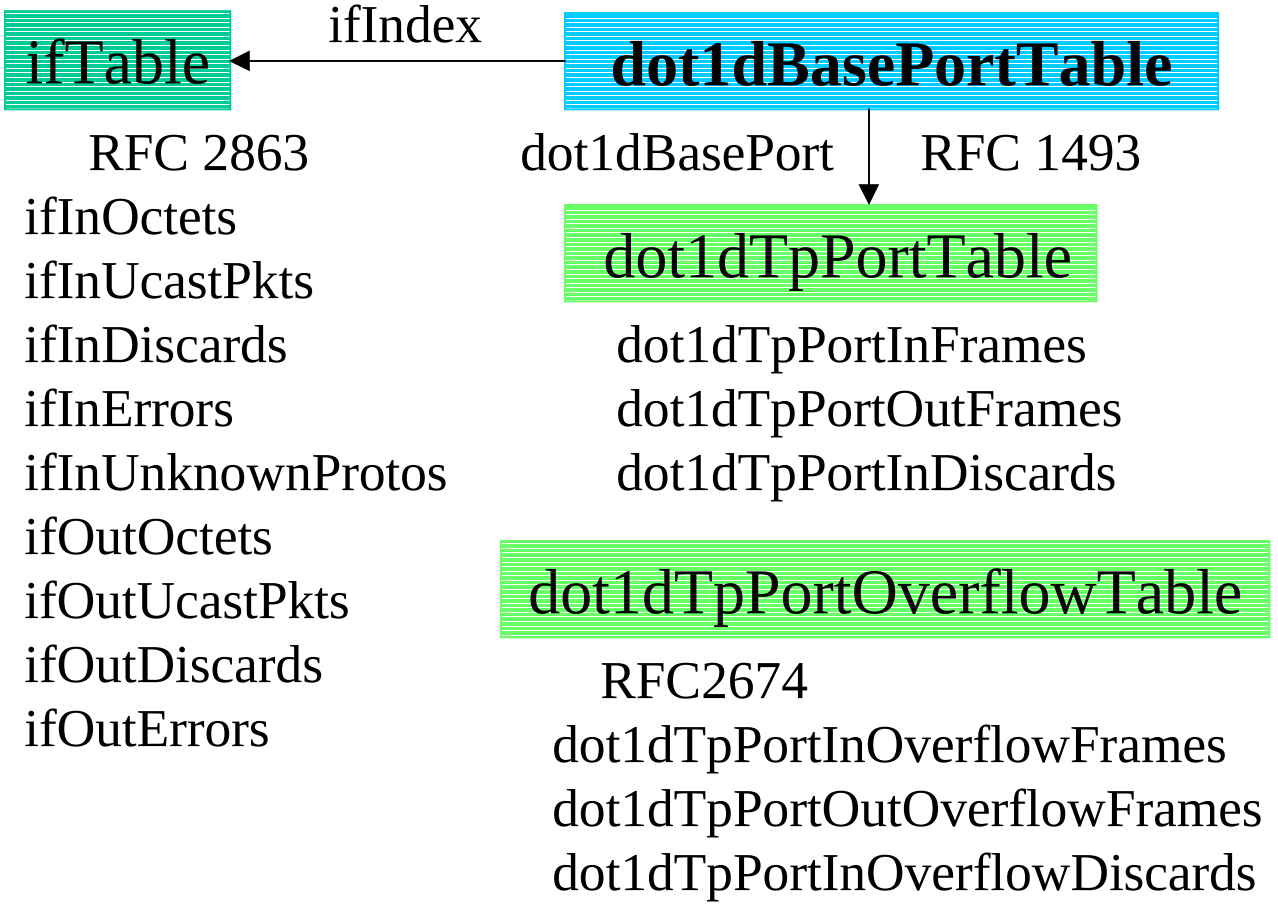
802.1D Bridging



No Standard MIB module for RSTP/802.1w exits to date
Per VLAN spanning tree protocols still proprietary
till 802.1s ratified.

Bridged vs Non-Bridged

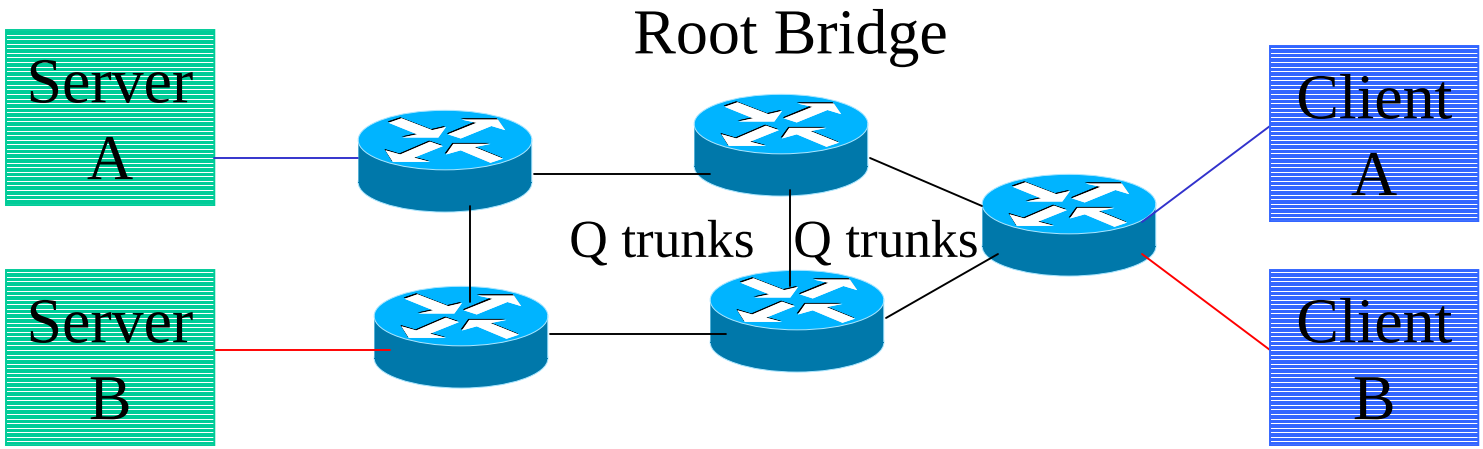
- Dot1d counters should be a less than ifTable counters if a port both bridges and routes pkts.



IEEE 802.1P/Q

- RFC 2674 defines Virtual LANS
  - Q-BRIDGE-MIB
  - P-BRIDGE-MIB
    - dot1dTpPortOverflowTable extends RFC 1493 counters
- Per VLAN (vid: 1-4094) Stats
  - dot1qTpVlanPortInFrames
  - dot1qTpVlanPortOutFrames
  - dot1qTpVlanPortInDiscards

VLAN 802.1P/Q



GARP/GVRP

dot1qPortGvrpFailedRegistrations
dot1qVlanNumDeletes

RMON

- RFC 2819 defines RMON-MIB
  - Passive Monitor for Ethernet
    - etherStats, etherHistory Tables provide
      - etherStatsCRCAlignErrors
      - etherStatsUndersizePkts
      - etherStatsOversizePkts
      - etherStatsFragments
      - etherStatsJabbers
      - etherStatsCollisions
    - Uses OID of ifIndex as dataSource pointer
  - RFC 2665 provides more detail

RMON

- Traffic patterns
  - Host Table provides bytes, pkts per MAC
    - Like dot1dTpFdbTable w/o counters
  - Matrix - Bytes Packets
  - Ethernet Packet distributions

    etherStatsPkts64Octets
    etherStatsPkts65to127Octets
    etherStatsPkts128to255Octets
    etherStatsPkts256to511Octets
    etherStatsPkts512to1023Octets
    etherStatsPkts1024to1518Octets

Layers 3-7

- IPv4 Stack/CIDR Route Table
- VRRP
- BGP
- OSPF
- IS-IS
- RMON 2

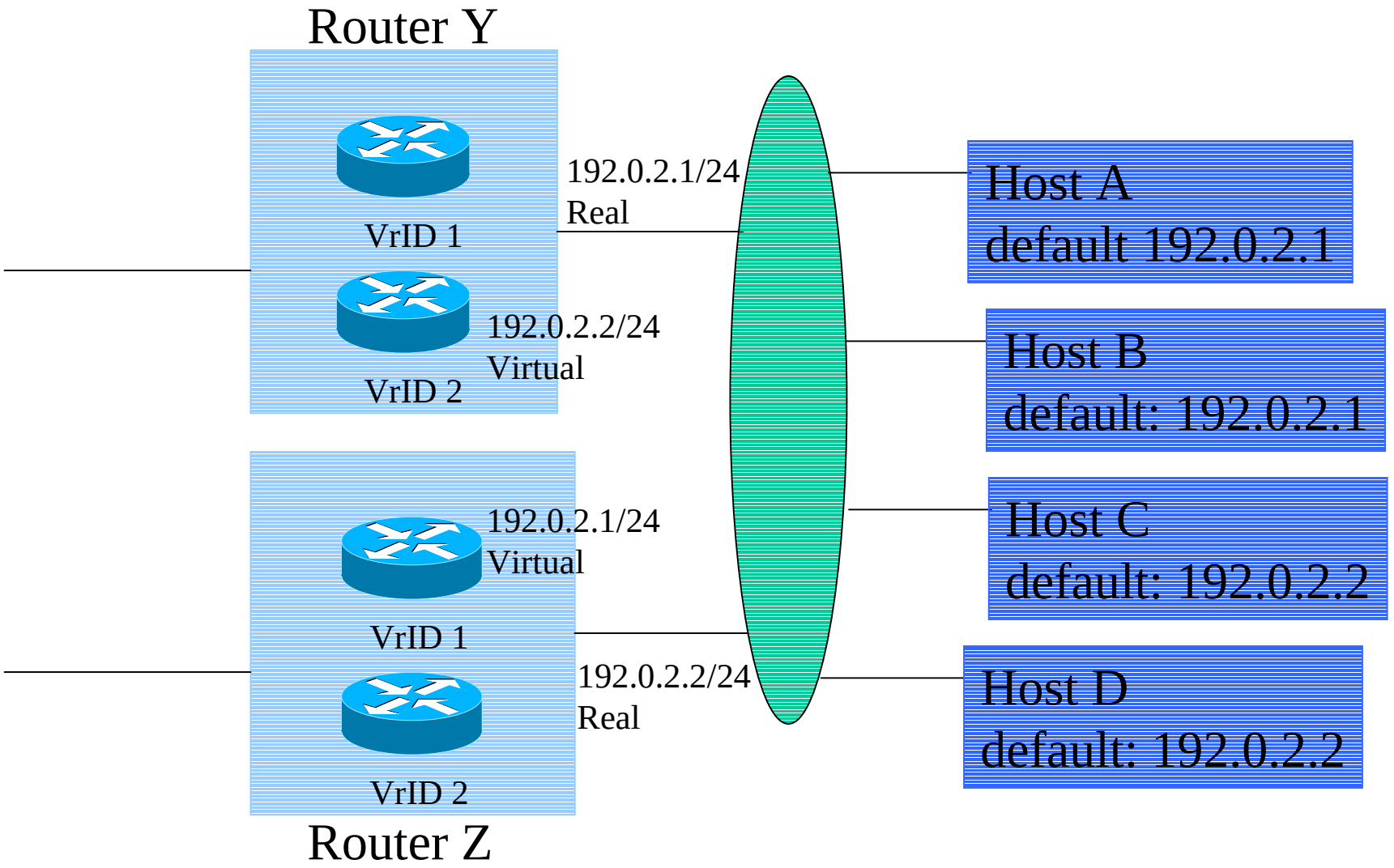
IPv4

- RFC 2011 defines IP Layer
  - defines seventeen counters
  - If a device supports multiple IP Stacks, these counters will apply to one of the stacks per vendor's discretion
  - If the device supports the ENTITY-MIB (2737)
    - then multiple MIBS can be represented in a given agent MIB (BRIDGE, OSPF, BGP)

IPv4

- RFC 2096 ipCidrRouteTable
  - Indexed by:
    - dest ip, netmask, tos, next hop
    - Provides:
      - Age of route
      - Next hop AS
  - ipCidrRouteTable (RFC 2096 1/97) replaced
    - ipForwardTable (RFC 1354 7/92) replaced
    - ipRouteTable (RFC 1213 3/91)

VRRP Sample Network



VRRP Counters - RFC 2787

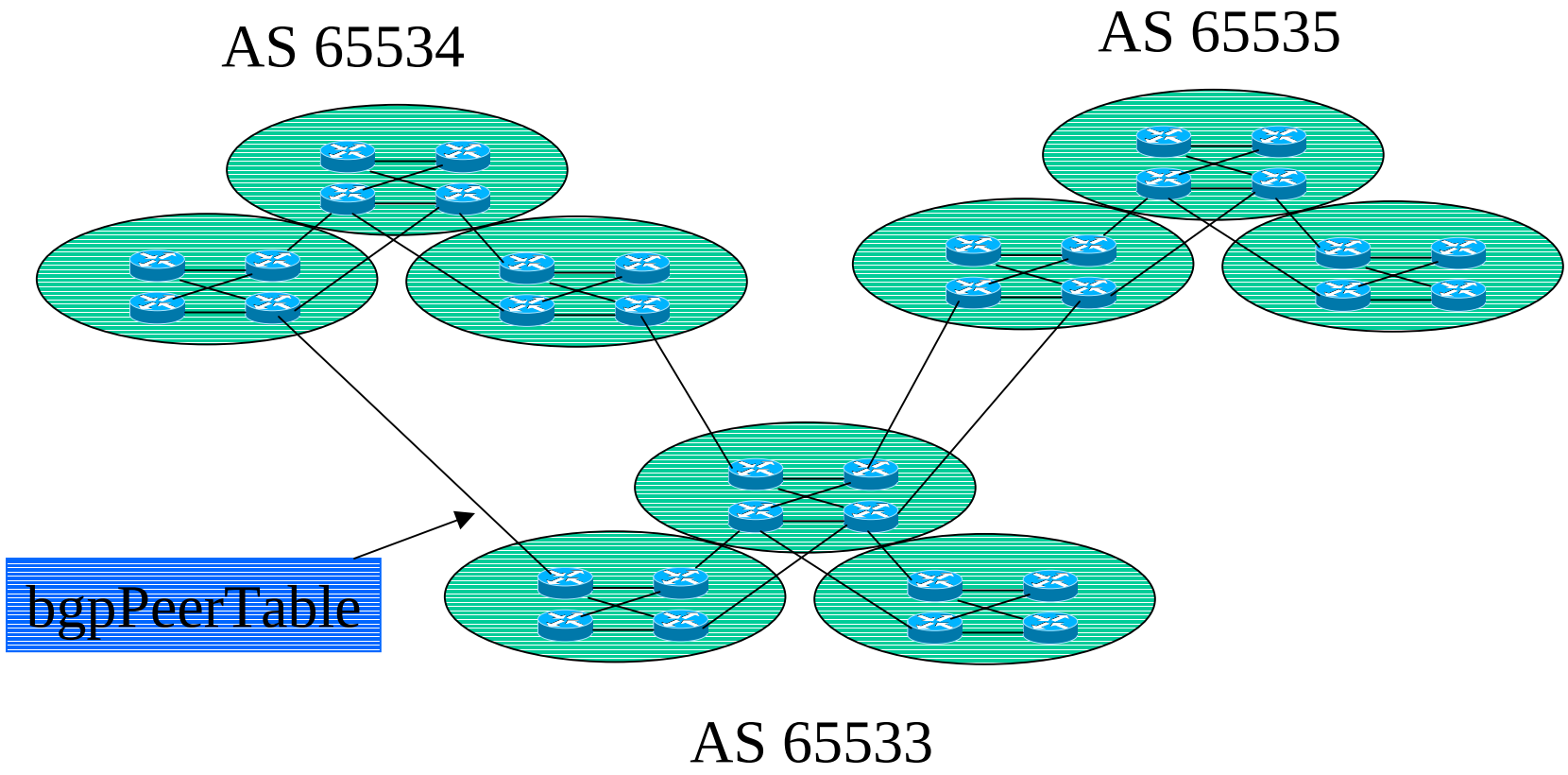
- Global counters
  - vrrpRouterChecksumErrors
  - vrrpRouterVersionErrors
  - vrrpRouterVrIdErrors
- vrrpRouterStatsTable per VrID
  - vrrpStatsBecomeMaster
  - vrrpStatsAdvertiseRcvd
  - vrrpStatsAdvertiseIntervalErrors
  - vrrpStatsIpTtlErrors

BGP

- BGP-MIB RFC 1657
  - Provides current status of eBGP, iBGP peers

bgpPeerTable
index: bgpPeerRemoteAddr (IPv4 addr)
- bgpPeerInUpdates
- bgpPeerOutUpdates
- bgpPeerInTotalMessages
- bgpPeerOutTotalMessages
- bgpPeerLastError
- bgpPeerFsmEstablishedTransitions

BGP Sample Network



BGP

- TCP Status per BGP session is also available per RFC 2012 in
  - tcpConnTable
    - Indexed by: tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort
- Missing all functionality of follow on BGP RFCs since RFC 1654.
  - Cisco BGP Accounting MIB Module useful for tracking traffic aggregates on per AS level

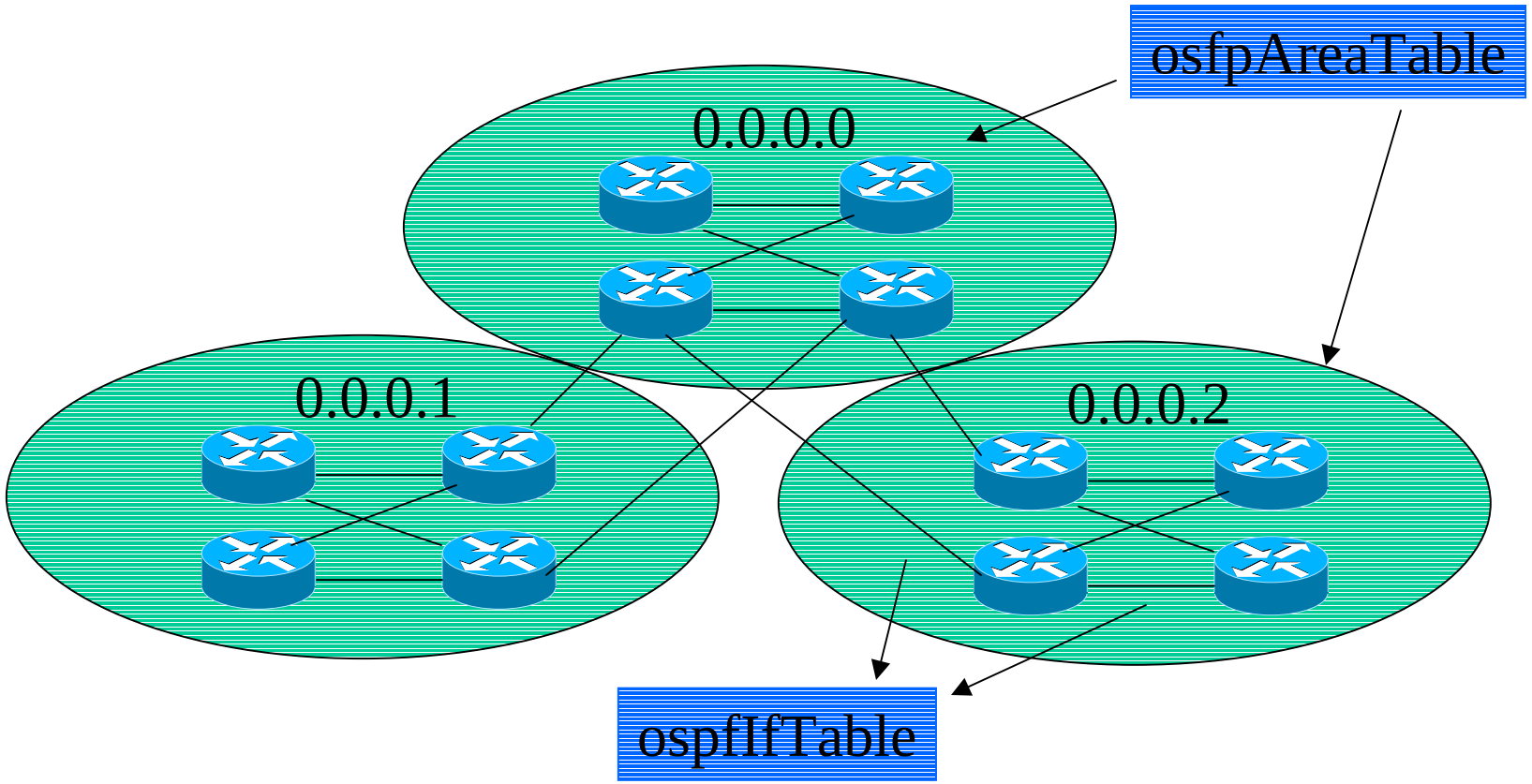
OSPF

- RFC 1850 defines some useful objects:

ospfAreaTable
indexed by Area ID
ospfSpfRuns - number of times Djkstra was run

ospfIfTable
indexed by: IP Address, ifIndex or Zero
ospfIfEvents - number of time state changed

OSPF Sample Network



OSPF MIB Module

- RFC 1850 does not keep a counter for LSDB overflows, it does have a notification for such an event though: ospfLsdbOverflow
- There exist additional counters for:
  - Virtual Links
  - Neighbors

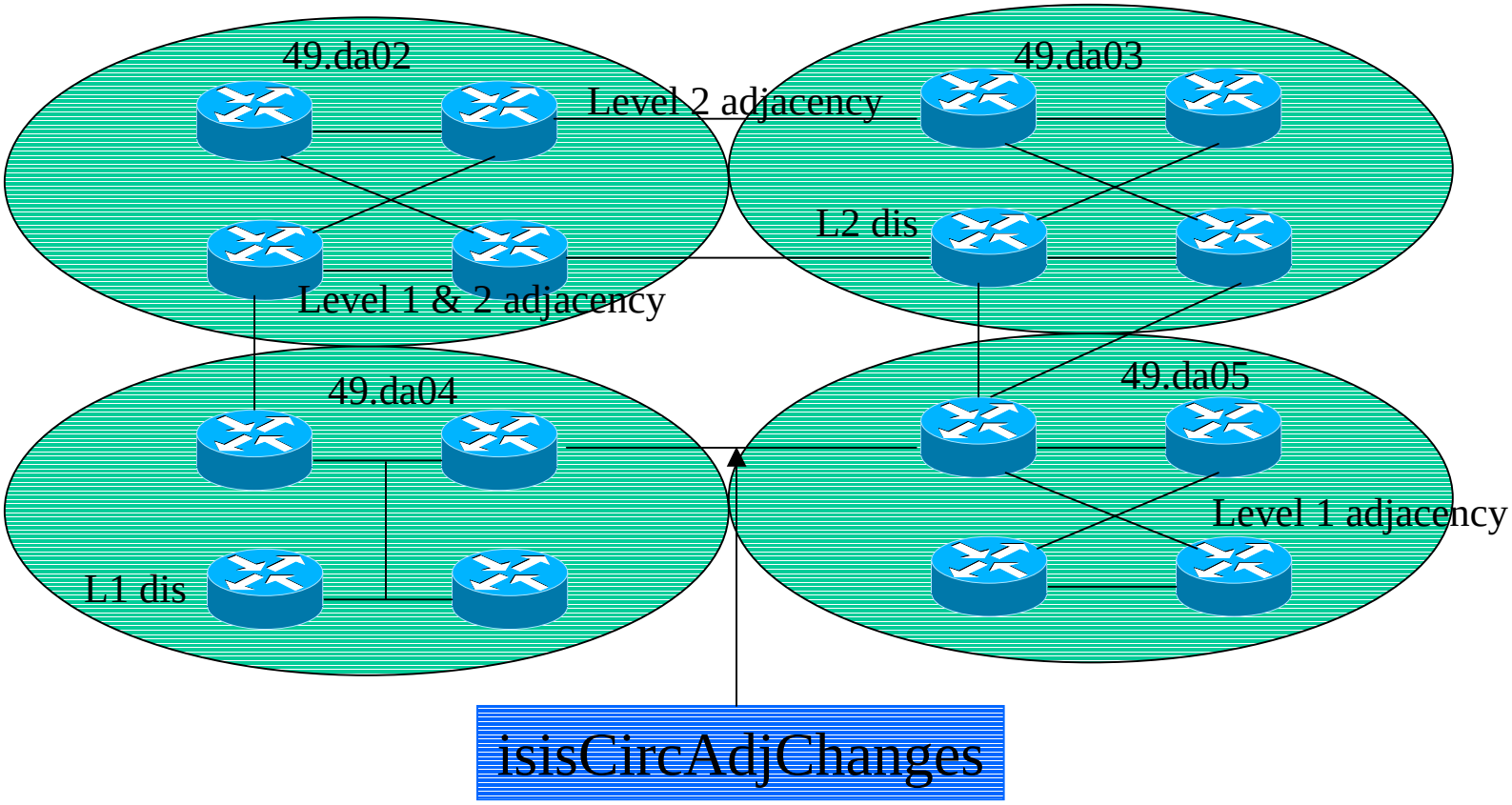
IS-IS MIB Module

- Internet Draft in development
  - draft-ietf-isis-wg-mib-03.txt

isisSysTable
indexed by:
instance of protocol (sysInst)
Eleven counters
isisSysLogAdjacencyChanges

isisCircTable
indexed by
sysInst, circInst
Eight Counters including
Changes in Adjacency:
isisCircAdjChanges

IS-IS Sample Network



RMON 2

- RFC 2021 monitors network/applications
  - Network Layer
    - nlHost - bytes, packets
    - nlMatrix - bytes, packets Src->Dest, Dest->Src
  - Application Layer
    - alHost
    - alMatrix
    - Uses an “ifIndex” as a dataSource
  - Per Protocol Distributions
    - per port, based on protocol dictionary

RMON 2

- Uses counters that start from zero
  - Per RFC, useful if counters are in rows in tables that come and go or if table is indexed using TimeFilter
  - ProtocolDirStatsTable
    - Indexed by:
      - protocolDistControlIndex,
        - » control table that created this row
      - protocolDirLocalIndex (PDLI)
        - » pointer to protocol that this collection represents
    - protocolDistStatsPkts
    - protocolDistStatsOctets

RMON 2

```
RS8-3# rmon show protocol-distribution et.5.5
RMON II Protocol Distribution Table

Index: 506, Port: et.5.5, Owner: monitor
      Pkts     Octets   Protocol
      ----     ------   --------
        19       1586   ether2
        19       1586   *ether2.ip-v4
         2        192   *ether2.ip-v4.icmp
        17       1394   *ether2.ip-v4.tcp
        17       1394   *ether2.ip-v4.tcp.www-http
```

Acknowledgments

- Many thanks to the following reviewers
  - Premal Ashar, Riverstone
  - Joe Clarke, Cisco Systems
  - Greg Hankins, Riverstone
  - Marc Lasserre, Riverstone
  - Greg Mintel, Cisco Systems
  - David Perkins, SNMPinfo
  - Dave Pothier, Cisco Systems
  - Daniel Vltavsky, Riverstone
- Thanks to the talented and dedicated authors of all Standards Track MIB Modules