SNMP (v1 mostly) MIB intro

Network Mgmt/Sec.

Jim Binkley

Outline

- MIB-2 and subgroups
  - introduction
  - system, ip, etc.
  - snmpwalk handout
- MIB-2 extensions
- reality checks

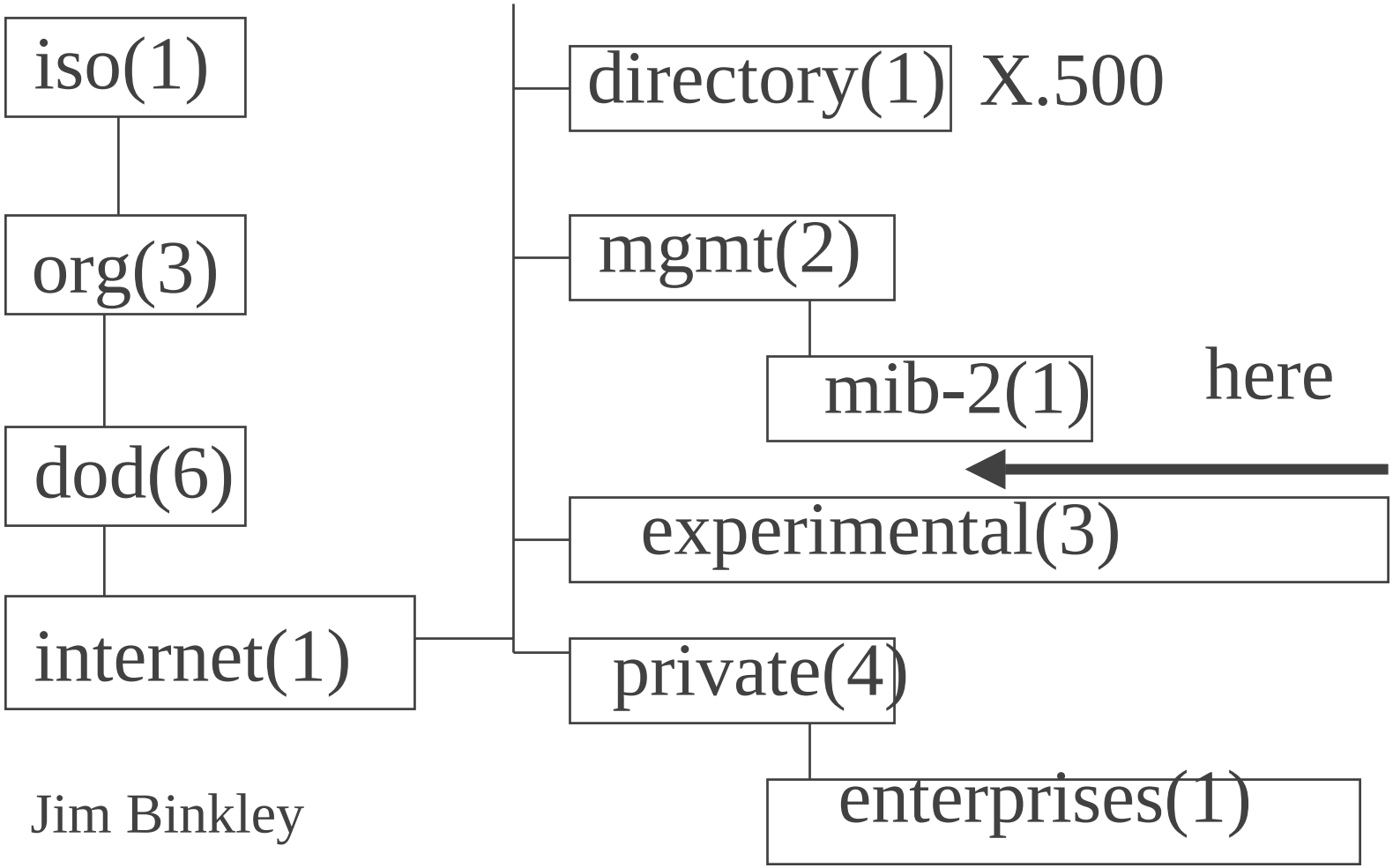
MIBS/rfcs of interest

- **MIB-II RFC 1213 -** defines several hundred basic agent objects, system, interfaces, ip, etc (snmp v1 too)
- transmission Ethernet Interface MIB - **EtherLike MIB - RFC 1643**
  - defines ethernet-like data-link layer objects
- note snmp v2 mib **RFC 1573** redefined interfaces group

re MIB-2, and SNMP v2 changes

- split MIB-2 up
- system, now in SNMPv2-MIB, 1907
- interfaces, replaced by IF-MIB, 1573
- ip/icmp, IP-MIB, and IP Forwarding MIB, 1354, 2011
- tcp, TCP-MIB 2012, udp, UDP-MIB 2013
- transmission, no change
- snmp, SNMPv2-MIB, 1908
- **and note that lots of implementations don’t care**

top part of OID tree



note: at least 3 areas of interest

- MIB-2 snmp v1 itself (**evolved in snmp v2**)
  - system - 1.3.6.1.2.1.1 { mib-2 1 }
  - interfaces
  - ...
  - snmp { mib-2 11 }
- extensions to MIB-2 (next slide)
- enterprise/private MIBS themselves
  - cisco universe of its own

MIB-2/v1 basic sub-trees (1--11)

- **system** - { mib-2 1 } (who am i ... and where am i)
- **interfaces** - { mib-2 2 } (caveat emptor, v2)
- at - { mib-2 3 } (toast)
- **ip** - { mib-2 4 } (addresses, stats, arp, route tables)
- **icmp** - { mib-2 5 }
- **tcp** - { mib-2 6 } (note connections)
- **udp** - { mib-2 7 }
- egp - { mib-2 8 } (egp is history, now bgp)
- cmot - { mib-2 9 } (composted toast)

MIB-2/v1 basic sub-trees (1--11)

- **transmission** { mib-2 10 } (ethernet stats)
- **snmp** - { mib-2 11 }

interesting idea/s ...

- as we look at these think about them from both:
  - admin POV (“how can I learn useful facts about the system”)
  - defender POV (“what can I learn about intrusions ...”)
- and think about how they are implemented
  - how can snmp mib itself be implemented?

IETF extensions to MIB-2 tree

- many kinds of standard/noble attempt objects have been added over the years to the MIB-2 line of IETF objects
- may or may not be implemented
- e.g.,
  - appletalk
  - OSPF
  - RMON (a universe in a few mibs)
  - IF-MIB

MIB-2 IETF extensions table

- appletalk, mib-2 13, rfc1243, rfc1742
- ospf , mib-2 14, rfc1253, rfc1850
- bgp, mib-2 15, rfc1269, rfc1657 (bgp-4)
- rmon, mib-2 16, rfc1271, rfc1757
- bridge (dot1), mib-2 17, rfc1493
- decnet, mib-2 18, rfc1289
- character, mib-2 19, rfc1316
- repeater, mib-2 22, rfc1516
- rip-2, mib-2 23, rfc1389
- ident, mib-2 24, rfc1414

MIB-2 IETF extensions table

- host resources, mib-2 25, rfc1514
- 802.3 mau, mib-2 26, rfc1515
- if-mib, mib-2 31, rfc1573
- dns server, mib-2 32, rfc1611
- ups, mib-2, mib-2 33, rfc1628
- sna-nau, mib-2 34, rfc1666
- etherlike, mib-2 35, rfc1650
- atm, mib-2 37, rfc1695
- modem, mib-2 38, rfc1696
- printer, mib-2 43, rfc1759

MIB-2, v2 mib extensions exist

- (later) ...
- but e.g.,
  - IP-MIB, mib-2 48, rfc2011
  - tcp-mib, mib-2 49, rfc2012
  - udp-mib, mib-2 50, rfc2013
  - entity-mib, mib-2 47, rfc2037

MIB check on cisco router

- basic MIB-2 mibs exist plus
- 13 - appletalk
- 14 - ospf
- 17 - bridge (dot1)
- 31 - if MIB
- 34 - sna
- 37 - atm
- 47 - entity mib

MIB-mining

- looking in extended mib-2 mibs OR
- enterprise mibs
- for genuinely useful objects ...
  - examples:
  - cisco traffic meter on cisco switches
  - cisco temperatures in enviornmental mibs
  - cisco router/load average values
  - interfaces, if names to snmp port numbers

MIB-2 design criteria (rfc 1213)

- objects must be essential
- only “weak” control objects are allowed
  - no object called “reboot” (catch on fire)
  - lots of RO objects, but can you cause a TCP disconnection or remove a RT entry?
- avoid duplication in objects
- nothing system specific (no BSD unix)
  - note: ucd/unix and hp/unix host mibs (sequent too)
- avoid heavy instrumentation of critical code areas

system, mib-2 1

- overall info about the system
- sysDescr - 1 - string - RO
- sysObjectID - 2 - OID - RO
- sysUpTime - 3 - TimeTicks - RO
- sysContact - 4 - string - RW
- sysName - 5 - string - RW
- sysLocation - 6 - string RW
- sysServices - 7 - integer - RO

thou shalt set

- contact
- name
- location
- do this for the next person or the vacation replacement ...
- this is one reason repeaters should be managed (where the heck is it?)

interfaces

- basically a table of entries per interface
- basic facts and counters for input/output of packets, all RO
- note some parts are “deprecated”
  - ifOutQLen (not widely implemented)
  - ifInNucastPkts, ifOutNucastPkts
  - ifSpecific (ignored is more like it)

interfaces, mib-2 2

- ifNumber (1)
- ifTable (2) (the table)
  - ifEntry (1) (the row)
    - » ifIndex (1) - snmp interface index elsewhere too
    - » ifDescr (2) - hopefully manu. interface name
    - » ifType (3) - integer, int/string table lookup exists
    - » ifMtu(4) - max packet size
    - » ifSpeed(5) - gauge
    - » ifPhysAddress(6) - MAC

interfaces.ifTable.ifEntry, cont.

- ifAdminStatus(7), up/down/testing (configured)
- ifOperStatus(8), up/down/testing, hw reality
- ifLastChange(9), in TimeTicks, when i/f entered op state
- ifInOctets(10) - packet count in IN BYTES (MRTG)
- ifInUcastPkts(11) - unicast packet count in
- ifInNUcastPkts(12) - broadcast + multicast pkts in
- ifInDiscards(13) - no errors, but had to discard (overflow?)
- ifInErrors(14) - errors therefore tossed
- ifInUnknownProtos(15) - no network-layer protocol
- ifOutOctets(16) - bytes sent out interface (MRTG)

cont.

- ifOutUcastPkts(17)
- ifOutNUcastPkts(18)
- ifOutDiscards(19)
- ifOutErrors(20)
- ifOutQLen(21) - length of output pkt queue
- ifSpecific(22) - OID

notes:

- 1. snmp port mapping to interface name (presumably ifEntry) terribly important
  - need it for MRTG
  - not necessarily a sensible integer order
  - beware dynamic interfaces
  - new modules that cause all ports to move
- bugs certainly exist here (or discrepencies)
  - Sun/solaris ifInOctets/ifOutOctets notorious
  - cisco ifInNUcastPkts,ifOutNUcastPkts

network interface types (some)

- values in ifType are integer
- known string mappings include:
  - other 1
  - ethernetCsmacd 6
  - iso88025TokenRing 8
  - fddi 15
  - ppp 23
  - ds3 30
  - hssi 46

ip, mib-2 4

- what was in at table (arp ...) moved here
- boolean + counters + three tables
- tables include:
  - **ipAddrTable -** ip unicast/bcast/netmask per i/f
  - **ipRouteTable -** dest,nexthop,mask, metrics, type
  - **ipNetToMediaTable** (it’s the arp table, jim)

ip mib-2 4, start with counters

- ipForwarding(1), RW, 1 == router, 2 == not a router
- ipDefaultTTL(2), RW
- ipInReceives(3), includes errors
- ipInHdrErrors(4)
- ipInAddrErrors(5), ip dst on recv invalid
- ipForwDatagrams(6), # of forwarded datagrams
- ipInUnknownProtos(7), recv packet but no protocol
- ipIndiscards(8), lack of buffer space, of interest in router?
- ipInDelivers(9), # of packets sent to tcp/udp upstairs
- ipOutRequests(10), transport pkts delivered down to us

more ip

- ipOutDiscards(11) - pkts tossed due to lack of buffer space
- ipOutNoRoutes - pkts tossed due to no route
- ipReasmTimeout - ip reassembly timeout failure
- ipReasmRedqs - fragments that needed reassembly
- ipReasmOKS - # of packets reassembled
- ipReasmFails - # of reassembly failures
- ipFragsOK - # of packets fragmented ok
- ipFragsFails - pkts discarded due to DONTFRAGMENT
- ipFragsCreates - # of fragments created
- ipAddrTable, ipRouteTable, ipNetToMediaTable
- ipRoutingDiscards(23), tossed routing table entries (?)

ipAddrTable (index in bold)

- informally == UNIX # ifconfig -a
- ipAddrTable - it is all READONLY
  - ipAddrEntry
    - » **ipAdEntAddr**, IP address (192.1.2.3)
    - » ipAdEntIfIndex - snmp interface index (2)
    - » ipAdEntNetMask, IP address (255.255.255.0)
    - » ipAdEntBcastAddr, INTEGER, how many bits to bcast (least significant bit count)
    - » ipAdEntReasmMaxSize - biggest packet can reassemble

? can this capture

- > 1 IP addresses on the same physical interface???
- sometimes called helper address or IP alias or virtual address
- one to many with one physical i/f, many IP addresses

ipRouteTable, curiously R/W

- ipRouteEntry (row)
  - **ipRouteDest, IpAddress - dest ip**
  - ipRouteIfIndex - interface index
  - ifRouteMetric1-5, INTEGER, meaning depends on protocol type (hop count, etc)
  - ipRouteNextHop, IpAddress (gateway)
  - ipRouteType, integer, note can mark invalid
  - ipRouteProto, integer, RO
  - ipRouteAge, how old route is in seconds
  - ipRouteMask, IpAddress
  - ipRouteInfo, OID, RO

functional equivalent

- WNT in dos box, netstat -rn
- UNIX, almost universal, netstat -rn
- Cisco, show ip route

ipRouteType values

- other(1) - none of the following
- invalid(2) - route marked invalid
- direct(3) - destination is on directly connected subnet
- indirect(4) - destination is across next-hop router

ipRouteProto

- other - none of the following
- local - manually configured
- netmgmt - network management protocol
- icmp - icmp redirect
- rip
- ciscoIgrp
- ospf
- bgp

some comments on this

- one destination may not be enough
  - ipForward, RFC 1354 tries to replace
  - ipForward is { ip 24 }, comes after ipRoutingDiscards
- ipForwardNumber introduced to count # of entries (I wish ...)
- overall similar but index now is 4-tupe
  - dest/policy (tos)/nexthop/protocol

can we delete a routing table entry?

- in general, hard to predict what can be done about deleting a row
- however routing table and arp table both have invalid values
  - ipRouteType set to invalid
  - ipNetToMediaType set to invalid
- result is implementation specific
- consider security DOS consequences

ipNetToMediaTable

- arp table equivalent
- arp -a
- however index is 2-tuple
  - ipNetToMediaIfIndex, INTEGER
  - ipNetToMediaNetAddress
  - hopefully this serves as clue to which way (in terms of multi-homed home) ip X/MAC X can be found

arp table

- ipNetToMediaIfIndex, INTEGER
- ipNetToMediaPhysAddress, PhysAddress
- ipNetToMediaNetAddress, IpAddress
- ipNetToMediaType, INTEGER
  - other(1), invalid(2)
  - dynamic(3), ARP ... or whatever
  - static(4), “published”, proxy arp possible

icmp, mib-2 5 - RO

- just counters for inbound and outbound traffic
- icmpInMsgs(1)
- icmpInErrors(2)
- **icmpInDestUnreachs**(3) - host getting dest. unreachables
- icmpInTimeExcds(4)
- icmpInParmProbs(5)
- icmpInSrcQuenches(6)
- icmpInRedirects(7)

ping, ping, ping, etc...

- icmpInEchos(8) - # received of “pings”
- icmpInEchoReps(9) - # received of ping replies
- icmpInTimestamps(10)
- icmpInTimestampReps(11)
- icmpInAddrMasks(12)
- icmpInAddrMaskReps(13)
- icmpOutMsgs(14)
- icmpOutErrors(15) - msgs not sent due to errors
- **icmpOutDestUnreachs**(16) - router # of lack of routes
- **icmpOutTimeExcds(**17) - router # of traceroutes?

more icmp

- icmpOutParmProbs(18) - hah ...
- icmpOutSrcQuenches(19)
- icmpOutRedirects(20)
- icmpOutEchos(21)
- icmpOutEchoReps(22)
- icmpOutTimestamps(23)
- icmpOutTimestampReps(24)
- icmpOutAddrMasks(25)
- icmpOutAddrMaskReps(26)

tcp, mib-2 6

- all RO except for table column entry tcpConnState, which is RW integer
- tcpRtoAlgorithm(1) - how retransmit timer works
- tcpRtoMin(2) - min value for retransmit timer
- tcpRtoMax(3) - max value for retransmit timer
- tcpMaxConn(4) - max # of total tcp connections
- tcpActiveOpens(5), counter, count of active opens so far
- tcpPassiveOpens(6), counter, count of passive opens so far
- tcpAttemptFails(7), counter, failed conn. requests

notes

- active open is actually “transition to the SYN-SENT state”
- passive open is actually “transition to the SYN-RCVD state”

tcp, more

- tcpEstabResets(8), # of resets recv in established state
- tcpCurrEstab(9), or CLOSE-WAIT, # open now
- tcpInSegs(10) - packet count in, includes errors
- tcpOutSegs(11)
- tcpRetransSegs(12) - total # of retransmitted segments
- tcpConnTable(13) - index is 4-tuple (tcp socket)
  - tcpConnEntry (row) (next slide for column entries)
- tcpInErrors(14) - total number of pkts with errors
- tcpOutRsts(15) - # of resets sent

connection table is RW!

- tcpConnState, INTEGER, RW
- tcpConnLocalAddress, IpAddress
- tcpConnLocalPort, INTEGER
- tcpConnRemoteAddress, IpAddress
- tcpConnRemotePort, INTEGER
- index == ip addresses + ports (all 4)

tcp connection state values

- closed(1)
- listen(2)
- synSent(3)
- synReceived(4)
- established(5)
- finWait1(6)
- finWait2(7)
- closeWait(8)
- lastAck(9)
- closing(10)
- timeWait(11)
- **deleteTCB(12)**

udp, mib-2 7, all RO

- udpInDatagrams(1) - total # pkts upstairs
- udpNoPorts(2) - recv. pkts but no port
- udpInErrors(3) - errors other than NoPorts
- udpOutDatagrams(4) - # sent
- udpTable(5), index is both column objects
  - udpEntry(1)
    - » udpLocalAddress(1), IpAddress (listener)
    - » udpLocalPort(2), listener port #

udp note

- no attempt to track opposite udp “talker” in terms of HER port number
- 1 -N mapping
- actual input count is sum of 1st three counters
  - udpInDatagrams + udpNoPorts + udpInErrors

note host/router dichotomy

- ip/icmp end to end at home, and probably mostly a matter of forwarding at router
- udp/tcp end to end by definition, wouldn’t mean much at a router (except for attack exposure...)

transmission

- interface mib doesn’t go far enough in terms of link-layer stats
  - e.g., no collisions, “runts” for ethernet
- must be link-layer specific
- therefore RFC1643, Etherlike-MIB
  - never mind the Etherlike part, it’s ethernet Jim
- aka dot3 (802.3)

Gaul has 2 tables this time

- dot3 (transmission 7)
  - dot3StatsTable(2)
    - » dot3StatsEntry
  - dot3CollTable(5)
    - » dot3CollEntry (indexing affected by SNMPv2)
  - dot3Tests(6) - not a table, but a non-leaf node
  - dot3Errors(7) - not a table, but a non-leaf node
    - » errors that may occur during test

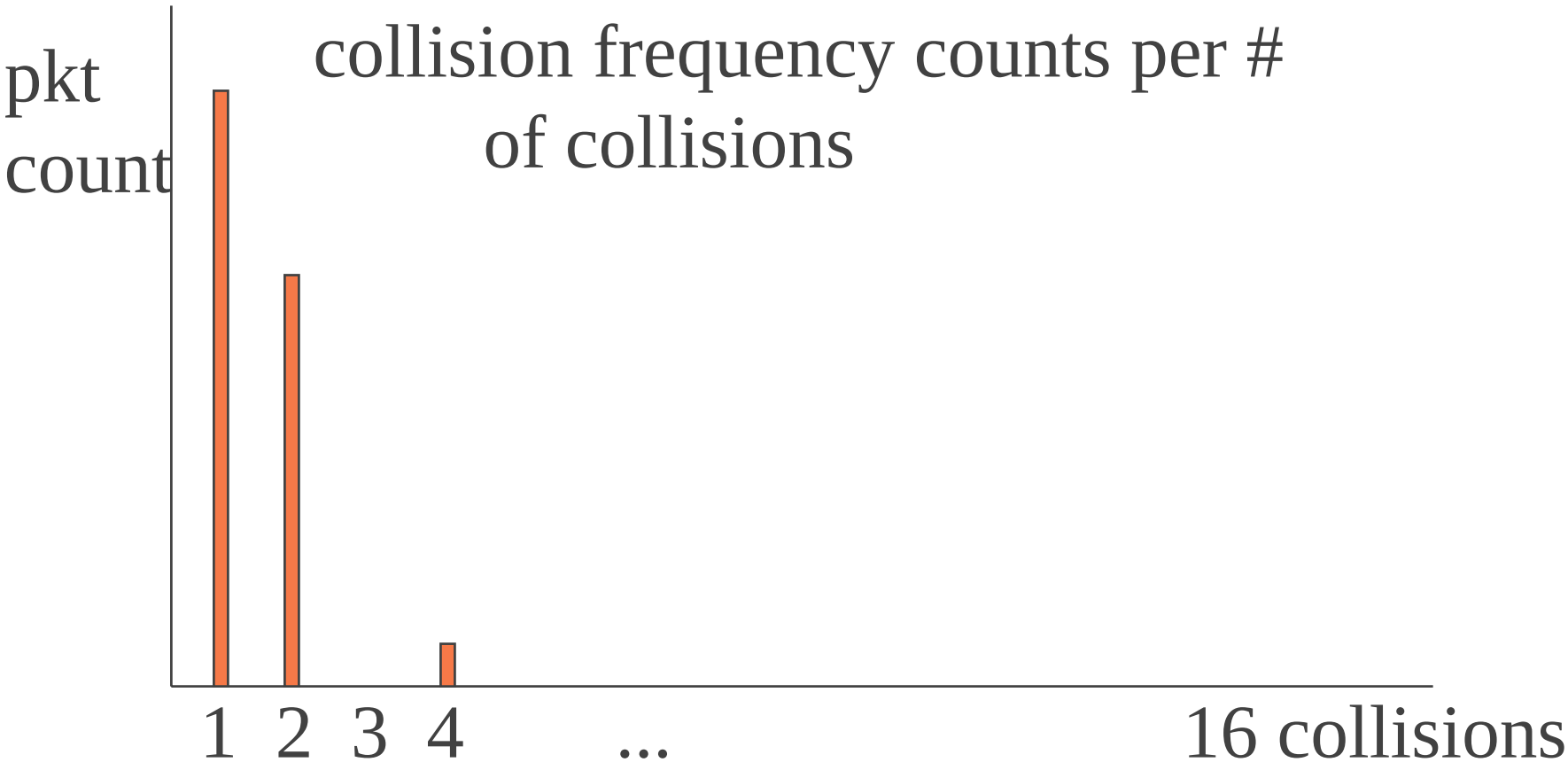
dot3StatsTable/Entry - all RO

- dot3StatsIndex, INTEGER - index, same as ifIndex in interfaces group
- dot3StatsAlignmentErrors - alignment errors
- dot3StatsFCSErrors - checksum errors
- dot3StatsSingleCollisionFrames - sent OK, 1 collision
- dot3STatsMultiplecollisionFrames - sent OK, > 1 coll.
- dot3StatsSQETestErrors
- dot3StatsDeferredTransmissions - 1st attempt, had to wait
- dot3StatsExcessiveCollisions - failed, too many collisions

dot3

- dot3StatsInternalMacTransmitErrors, never mind
- dot3StatsCarrierSenseErrors
- dot3StatsFrameTooLongs (“giants”), collision evidence
- dot3StatsInternalMacReceiveErrors
- dot3StatsEtherChipSet OID
- dot3CollTable - index is interface value + CollCount
  - dot3CollEntry
    - » dot3CollCount (1..16) X axis
    - » dot3CollFrequencies, counter, Y axis
- dot3Tests - neglect

dot3 collision table/per i/f



snmp mib itself

- from host POV, application layer
- input/output counts
  - gets/sets/traps, etc. as basic counts
- note errors, including:
  - too big
  - bad community names
- note packets coming in can be responses

snmp, mib-2 11, almost all RO

- snmpInPkts(1)
- snmpOutPkts(2)
- snmpInBadVersions(3)
- snmpInBadCommunityNames(4) - comm. string wrong
- snmpInBadCommunityUses(5) - e.g. write with no rights
- snmpInASNParseErrs(6)
- snmpInTooBigs(8) - response with error too big
- snmpInNoSuchNames(9) - response with error no such ...
- snmpInBadValues(10) - response
- snmpInReadOnlys(11) - response is readOnly

snmp mib cont.

- snmpInGenErrs(12) - response with general error
- snmpInTotalReqVars - total OK OIDS retrieved
- snmpInTotalSetVars - total OIDS set by remote manager
- snmpInGetRequests - input get requests
- snmpInGetNexts
- snmpInSetRequests
- snmpInGetResponses - total get-response recv.
- snmpInTraps - total traps coming in to us
- snmpOutTooBigs - response with too big sent
- snmpOutNoSuchNames(21) - response with no name sent

snmp mib, cont.

- snmpOutBadValues - response sent
- snmpOutGenErrs - response sent
- snmpOutGetRequests
- snmpOutGetNexts
- snmpOutSetRequests
- snmpOutGetResponses
- snmpOutTraps - trap messages sent by us
- snmpEnableAuthenTraps - only RW - send authent. traps
  - enabled(1), disable(2)

snmp v1 criticisms

- security is poor
- danger of too much overhead if large network
  - each table column entry is one get/response pair
  - MRTG/HPOV cycle-times should be observed and made larger if necessary
- traps may be lost due to use of UDP
  - important reason to get manager close to core infrastructure
- basic MIBS may have implementation holes or holes like multicast info is lacking

some virtues

- network structure may be automatically discovered and displayed
  - as opposed to keeping-up by hand drawings
  - or nothing at all
- information can be USEFUL (in the extreme)
  - tell two HPOV stories ... and one MRTG story
  - especially if devices are managed