Event TLVs,

Another Idea

Brian Arnold

Cisco Systems

May 6, 2003

Modified by Kevin Daines during OAM STF meeting in Seoul

World Wide Packets

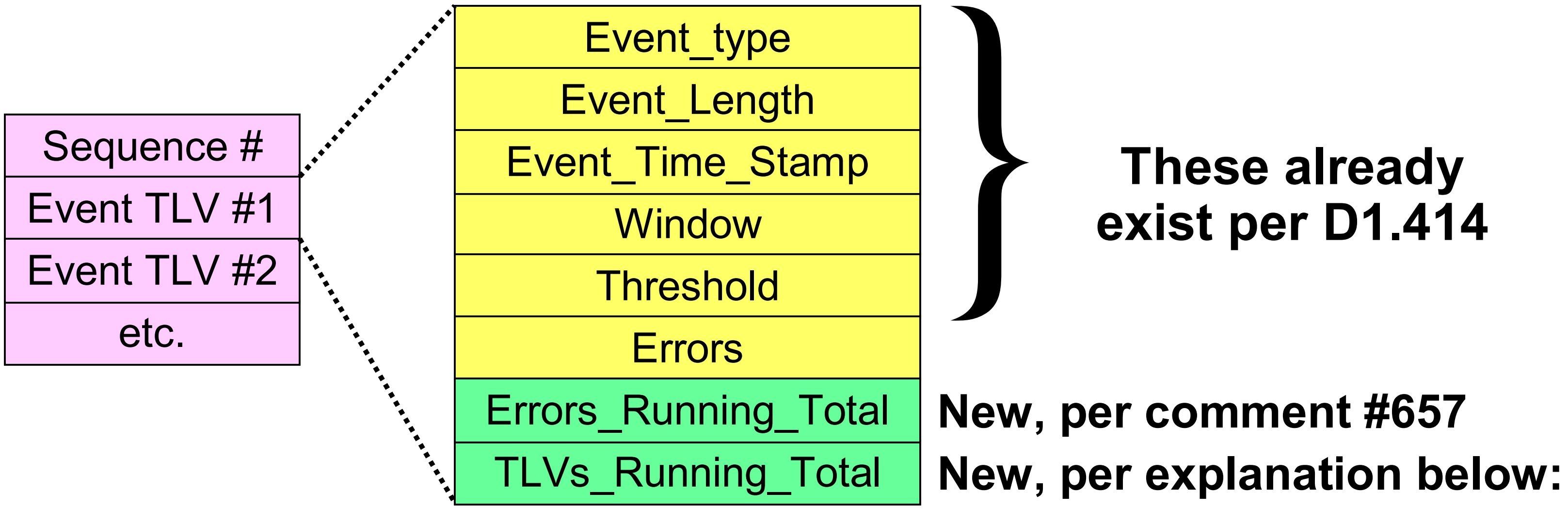
May 12, 2003

What to do about Event PDUs? Various recent issues on reflector, conf calls, and comments on D1.414:

- Timestamps
- Last seen vs. running count
- Counters of TLVs
- Counters of actual number of errors
- C30 interaction

One idea is to add two more fields to Event TLVs

Possible Event TLV structure



Sequence #

Event TLV #1

Event TLV #2

etc.

Event_type

Event_Length

Event_Time_Stamp

Window

Threshold

Errors

Errors_Running_Total

TLVs_Running_Total

These already exist per D1.414

New, per comment #657

New, per explanation below:

TLVs_Running_Total: A count maintained by the generator/sender of the event PDU, that is incremented by one each time a unique TLV of this event type is generated and sent. (“Unique” to eliminate counting duplicates of same info).

Primary purpose might be to allow system to count number of times a particular threshold was exceeded by counting TLVs, rather than looking at the counter values themselves.

Event TLV Q&A

Q: What good does this do?

A: Allows the system to accurately know how many times a particular event TLV was sent. For some systems it might be more useful to know how many times a threshold was exceeded (causing an event TLV to be generated), rather than knowing how many actual errors were counted.

Q: Why not allow the receiver to just count the number of PDUs/TLVs received?

A: OAMPDUs are not guaranteed reliable. Carrying a running count in the TLV nearly guarantees that if a receiver eventually receives an event PDU (with the running total of TLVs sent), that the receiver will be brought up to date with no loss of information, even if there have been lost PDUs (counter rollover notwithstanding). Counting only received PDU TLVs *guarantees* loss of or ambiguous information if PDU (and duplicate) is lost.

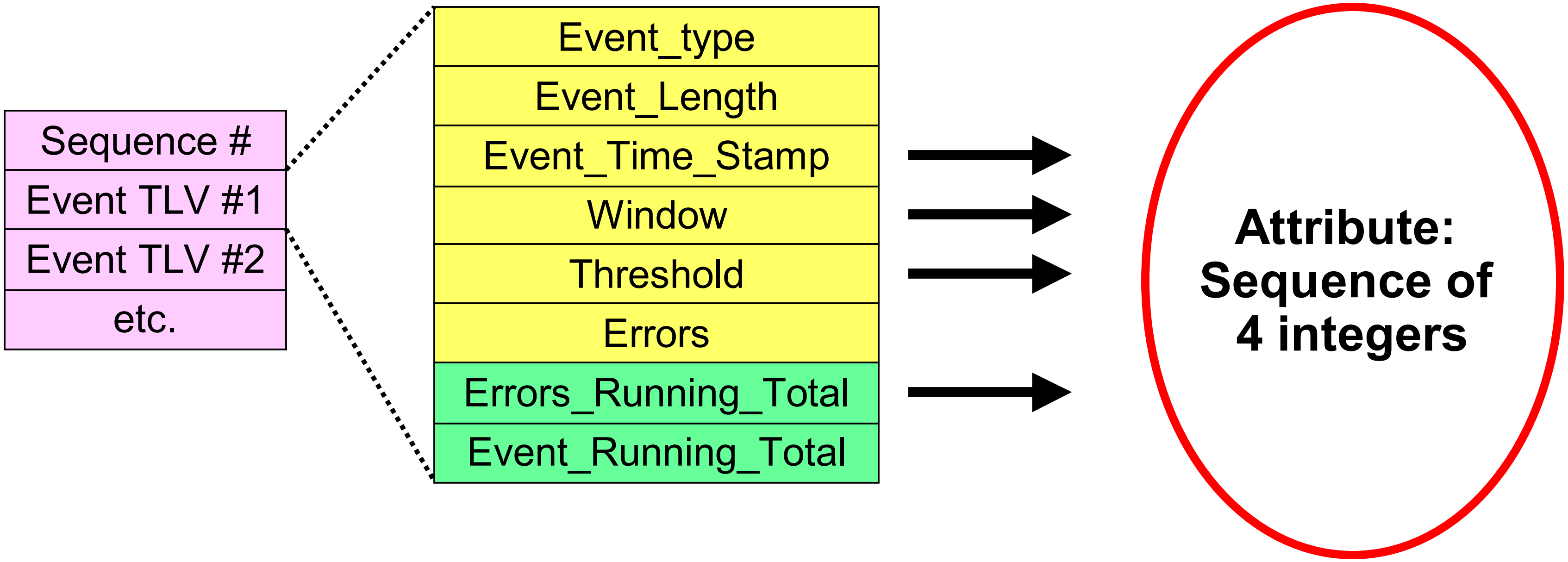
Q: What else?

Allows the running counts to be contained within the “last seen” Event TLV information.

Q: What else?

Might be appropriate to add running total field to “Summary” TLV, too. TBD?

What goes in C30?



Sequence #

Event TLV #1

Event TLV #2

etc.

Event_type

Event_Length

Event_Time_Stamp

Window

Threshold

Errors

Errors_Running_Total

Event_Running_Total

Attribute: Sequence of 4 integers

aOAMRemoteErrSymPeriodEvent (30.11.1.1.41)

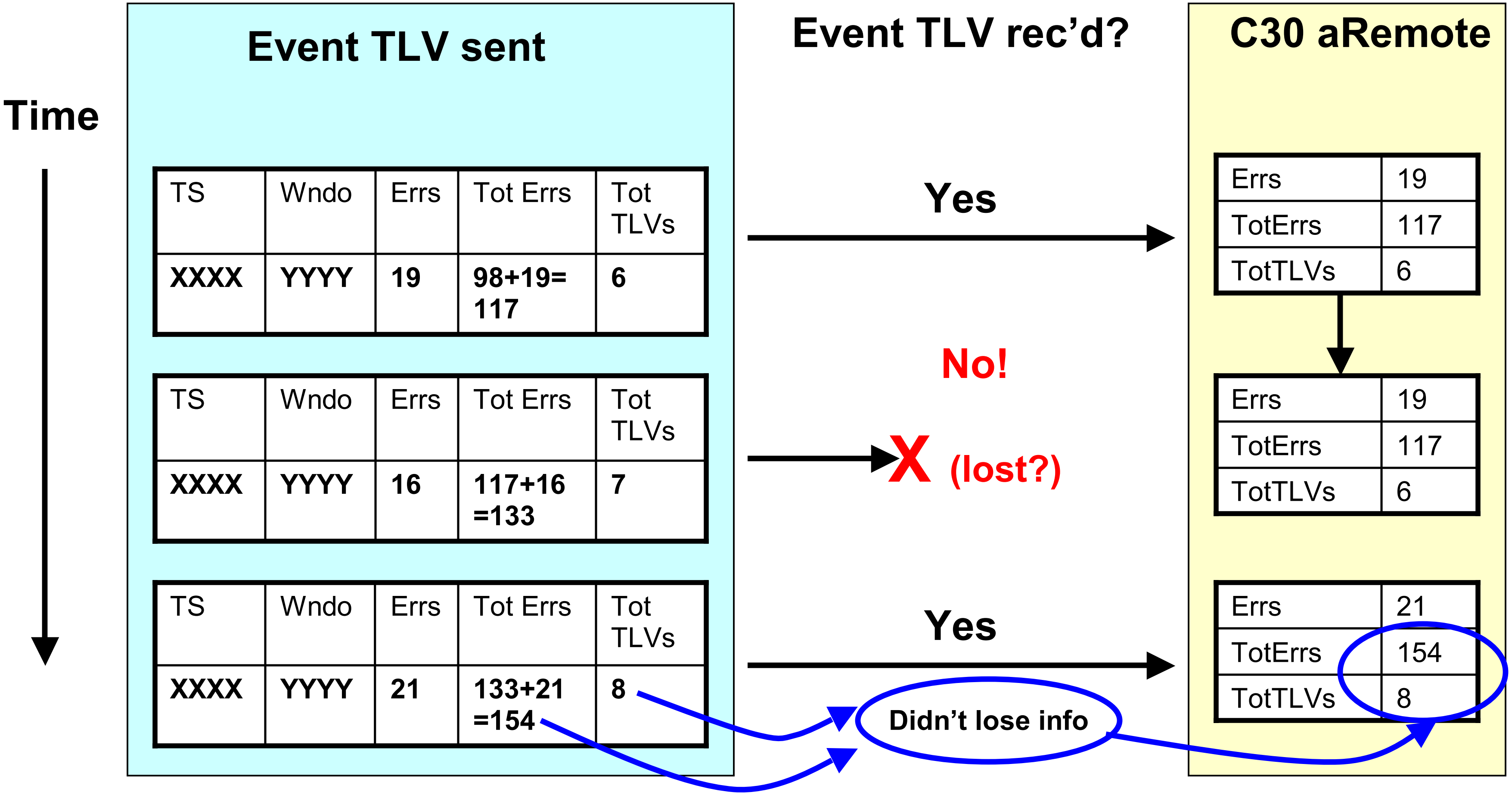
aOAMRemoteErrFrameSecsEvent (30.11.1.1.42)

aOAMRemoteErrFramePeriodEvent (30.11.1.1.43)

All three objects would be a sequence of *five integers* as indicated above, not *three* as in D1.414.

Example showing one event type

Assume: steady state operation; local threshold = 15; 5 TLVs of this type have already been generated/sent; total error count contained within TLVs sent = 98; example shows only some of TLV fields and some of the C30 aRemote object integers



Time

Event TLV sent

TS	Wndo	Errs	Tot Errs	Tot TLVs
XXXX	**YYYY**	**19**	**98+19= 117**	**6**

TS	Wndo	Errs	Tot Errs	Tot TLVs
XXXX	**YYYY**	**16**	**117+16 =133**	**7**

TS	Wndo	Errs	Tot Errs	Tot TLVs
XXXX	**YYYY**	**21**	**133+21 =154**	**8**

Event TLV rec'd?

Yes

No!

X (lost?)

Yes

Didn't lose info

C30 aRemote

Errs	19
TotErrs	117
TotTLVs	6

Errs	19
TotErrs	117
TotTLVs	6

Errs	21
TotErrs	154
TotTLVs	8

Pros and Cons

Pros: Allows accurate reporting and accumulation of both number of error TLVs sent (due to threshold crossings), and the summed count of errors that were contained within those TLVs.

Pros: Information not lost if PDU is lost since the sender maintains the counts.

Pros: C30 object contains both “last seen” info and running count, which reduces or eliminates penalty on systems that can’t watch C30 continuously.

Cons: Feature creep. Each event TLV would now contain one ‘T’, one ‘L’, and *six* ‘V’s. Ahem.

Cons: Déjà vu? Doesn’t seem like we’ve considered something similar before? Is it better this time?