

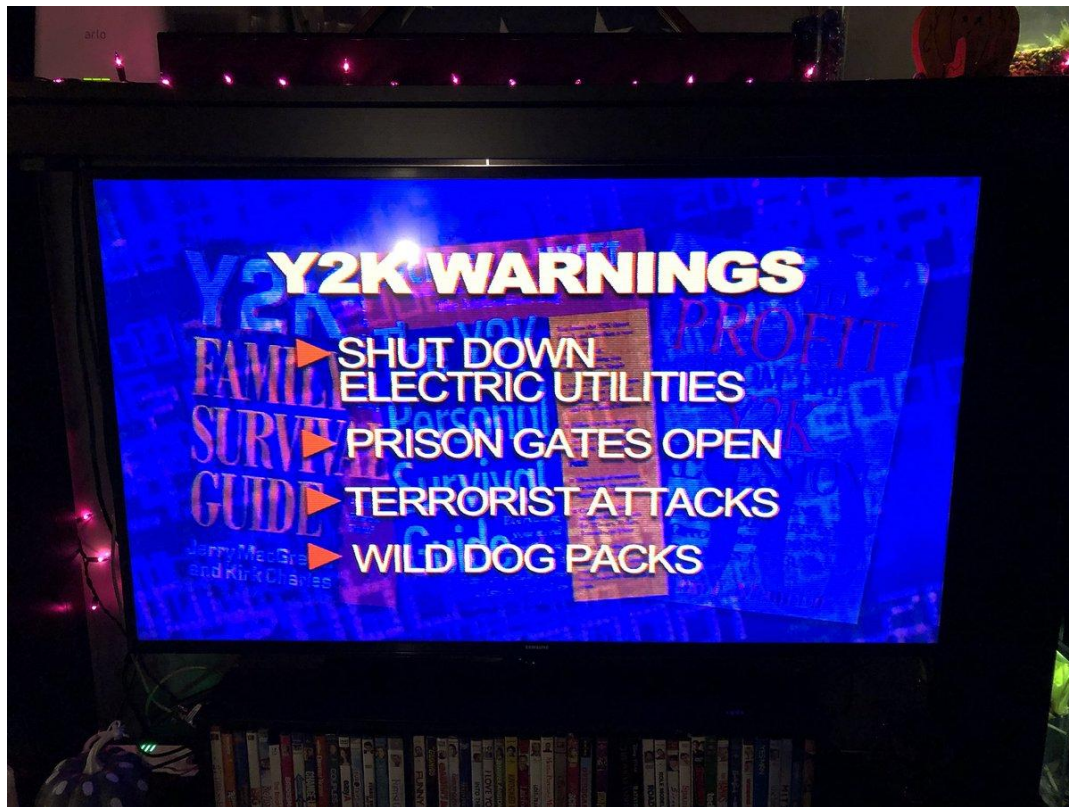
IAB Workshop on Y2038(*)

Wes Hardaker

(high level “should we” slides – full proposal offered if interest)

Table 5: (PAST) Rollover Event Timeline

Date	Event	Type
Monday 1 January 1900 00:00:00 UTC	NTP epoch established & 2-digit year encoding epoch established	Reference point
Thursday 1 January 1970 00:00:00 UTC	Unix epoch established	Reference point
Sunday 6 January 1980 00:00:00 UTC	GPS epoch established	Reference point
Saturday 21 August 1999	GPS week-number rollover (era 0 $\rightarrow$ era 1)	Historical reference; documented operational consequences (see §6.3 Table 4)
Saturday 1 January 2000	Y2K calendar transition (two-digit year representation rollover)	Rollover (mitigated)
Saturday 6 April 2019	GPS week-number rollover (era 1 $\rightarrow$ era 2)	Rollover (mitigated, partial)



Coming up next: so much worse!

Table 5: (FUTURE) Rollover Event Timeline

Date	Event	Type
Thursday 7 February 2036 06:28:16 UTC	NTP era-1 rollover (32-bit unsigned seconds counter from 1900 epoch)	Rollover (pending)
Tuesday 19 January 2038 03:14:07 UTC	Unix 32-bit signed time_t rollover (from 1970 epoch)	Rollover (pending)
Saturday 20 November 2038 approximately 23:59:42 UTC (assuming GPS-UTC offset remains +18 seconds; offset is subject to further leap-second insertion or omission)	GPS week-number rollover (era 2 → era 3)	Rollover (pending)
Sunday 7 February 2106 06:28:16 UTC	Unix 32-bit unsigned rollover (from 1970 epoch)	Rollover (long-tail)
Approximately 292 billion years after the Unix epoch	Unix 64-bit signed time_t rollover (from 1970 epoch)	Rollover (long-tail; cosmological timescale)

Note: these are collectively all being referred to as
Y2038

Work being done in the ITU-T

- Trey Darling started in the ITU-T:
 - “Technical Report on Global Coordination Requirements for 2038-class rollover events”
- Editors:
 - Trey Darley
 - Pedro Umbelino
 - Arnaud Taddei
 - Gianpaolo Angelo Scalone
 - Lorenzo Dina
 - Wes Hardaker
 - *Plus many more technical contributors*

ITU-T sectors being considered

- Telecommunications Networks
- Financial Systems
- Defense and Munitions
- Electric and Connected Vehicles
- Implantable and Hospital Medical Devices
- CSIRT/PSIRT/CERT Coordination
- Emergency Services Operations
- CS/OT Environments
- Cloud and ICT Infrastructure
- GNSS-Dependent Services
- Space Infrastructure
- Broadcasting and Emergency Alerting
- Government Services and Public Administration
- Insurance and Reinsurance
- Epistemic Coordination Layer
- Buildings and Built Environment
- Manufacturing and Industrial Production
- Logistics and Distribution
- Mining and Extraction
- Water and Wastewater
- Food and Farming
- Oil and Gas
- Power Plants
- Power Grid
- Railroad
- Bridges and Tunnels
- Aviation
- Shipping
- Wastes and Recycling
- Biological Research and Pharmaceuticals

ITU-T sectors being considered

- Telecommunications Networks
- Financial Systems
- Defense and Munitions
- Electric and Connected Vehicles
- Implantable and Hospital Medical Devices
- CSIRT/PSIRT/CERT Coordination
- Emergency Services Operations
- CS/OT Environments
- Cloud and ICT Infrastructure
- GNSS-Dependent Services
- Space Infrastructure
- Broadcasting and Emergency Alerting
- Government Services and Public Administration
- Insurance and Reinsurance
- Epistemic Coordination Layer
- Buildings and Built Environment
- Manufacturing and Industrial Production
- Logistics and Distribution
- Mining and Extraction
- Water and Wastewater
- Food and Farming
- Oil and Gas
- Power Plants
- Power Grid
- Railroad
- Bridges and Tunnels
- Aviation
- Shipping
- Wastes and Recycling
- Biological Research and Pharmaceuticals

What IETF Protocols or encodings will be impacted?

- NTP obviously – they'll respond to the ITU-T liaison statement (#2205)
- Protocols
 - Protocol fields
 - Protocols encodings
 - Implementations of otherwise conforming protocols
- Data Models – many of these contain default timestamp data, but could have unique instances
 - MIB modules?
 - YANG modules?
 - JSON models?
- Interoperability
 - Testing – HOW?
 - Libraries vs Applications
 - Explosive permutation counts

Encoding Example: BER

In BER and DER, integers are encoded as TLV binary sequences

1 = 0x02 01 01

127 = 0x02 01 7f

128 = 0x02 02 00 80

$2^{31} - 1$ = 0x02 04 7F FF FF FF

2^{31} = 0x02 05 00 80 00 00 00

$2^{32} - 1$ = 0x02 05 00 FF FF FF FF

2^{32} = 0x02 01 00

Very random protocol examples

- DNS: SOA serial number is often encoded as a unix timestamp
 - **GOOD:** RFC1982 covers how to deal with this when wraps occur
 - **GOOD:** pretty sure most DNS software handles this
 - **UNKNOWN:** What do implementations actually do when copying a 64 bit number?
- IP timestamp option (RFC781):
 - Timestamp:
“a right-justified, 32-bit timestamp in milliseconds modulo 24 hours from midnight UT.”
 - What happens at:
Midnight $= \text{int}(2^{31} - 3 \cdot 3600 - 14 \cdot 60 - 8)$
Now $= \text{int}(2^{32} + 10)$
 - Will all code behave the same way?
- E-Mail: what happens when a system gets a

Implementation scenario complexities

- One side supports 64 bit timestamps, the other **wraps** at 2^{32}
- One side supports 64 bit timestamps, the other **sends negative** past 2^{31}
- Both sides support 64 bit timestamps, but protocols **fail transmitting** them
- Both sides support 64 bit timestamps, but one field in a **data model is 32 bits**
- Either side implements something as an **int** instead of an **unsigned int**
- ...

How do we test all implementations of every protocol?

Workshop Topic Examples

- What **protocols** have potential concerns?
- What are common **implementation** mistakes known and unknown?
- How can **interoperability testing** be performed?
- How can implementations be resilient against **bad values received**?
- What are **lessons learned** from past events that might be helpful for the future?
- ...