Tips, techniques and tools for remote monitoring

Michael Sallaway

Engineer at Opengear

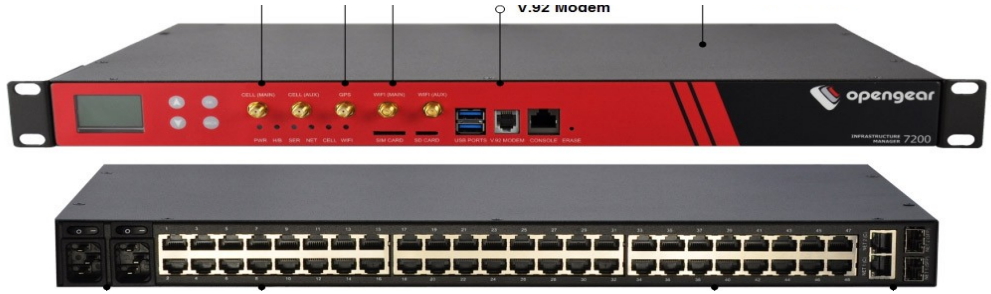
michael.sallaway@opengear.com

Who am I?

- Embedded Linux software engineer (~10 years)
- Snapgear: VPN routers & firewalls
- Opengear: remote out-of-band management
- Tinkerer / hobbyist / jack-of-all-trades

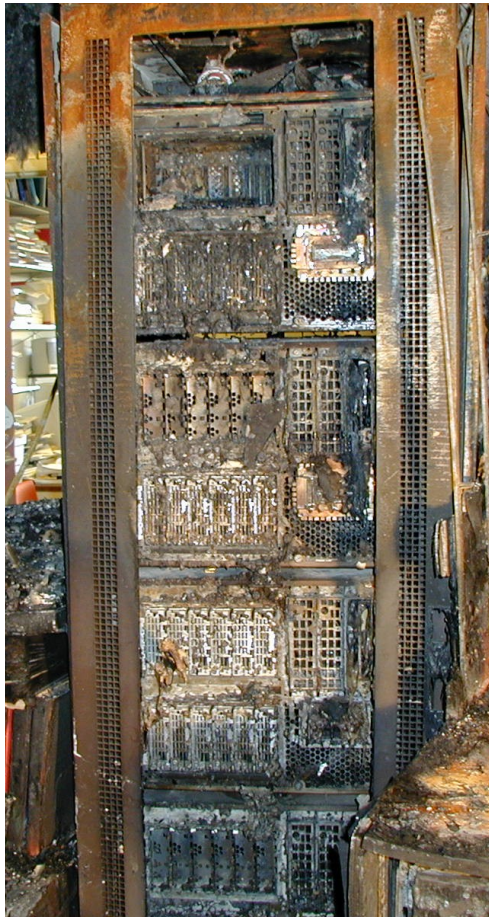
What do we do?

- Embedded Linux appliances
- Out-of-band & remote access
- OSS-friendly, hacker-friendly
  - Providing custom development kits
  - Support custom scripts & functionality



Monitoring: Why?

- Detecting Very Bad Stuff™
  - with remote access, fixing it
- Determine standard conditions
  - Pre-empt Very Bad Stuff™
- SLAs & Reporting: metric bounds

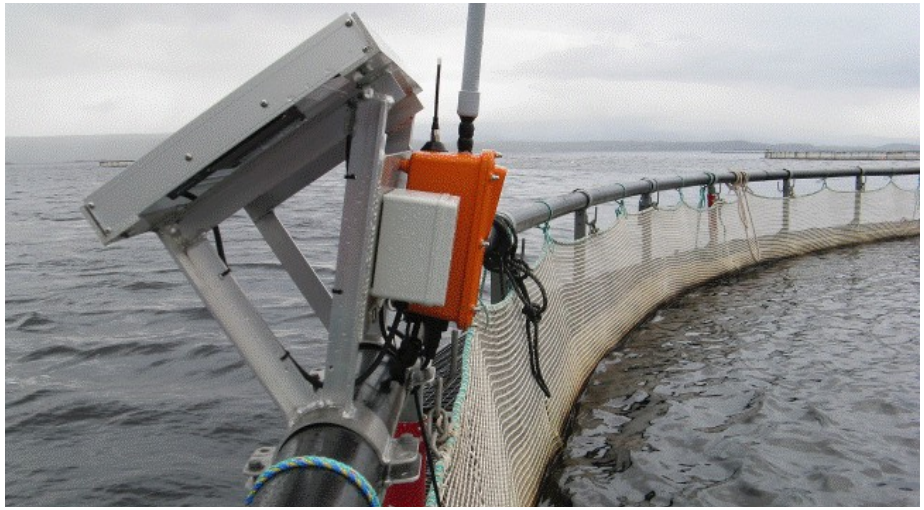


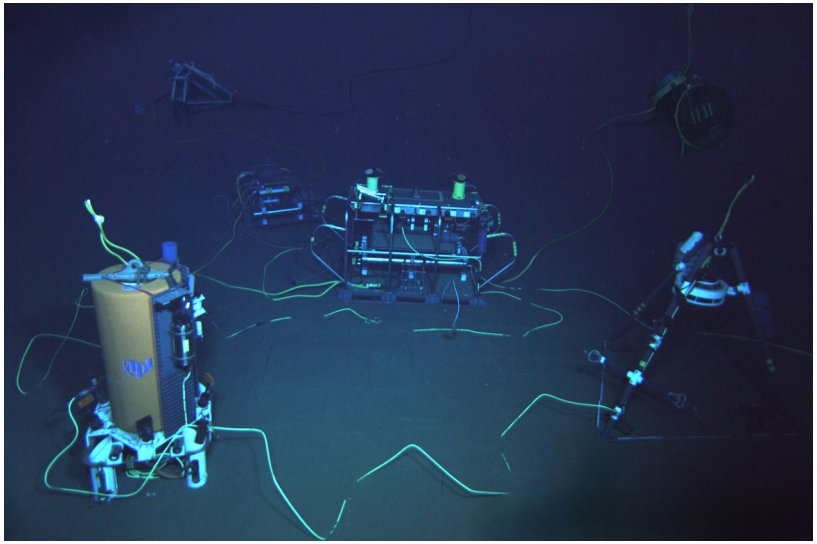
Monitoring: Where?

- Data centres
- Wiring closets
- Remote locations

Monitoring: Where?

- **Really** remote locations...





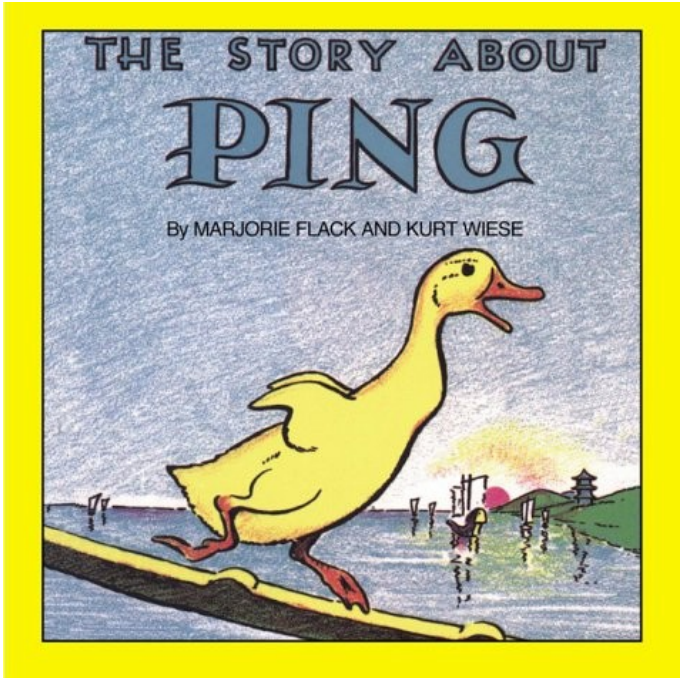
Monitoring: Where?

- Mobile locations
  - Buses
  - Gas tankers
  - Industrial vehicles
  - Internet of Things

- Each dictates different monitoring & access needs & methods

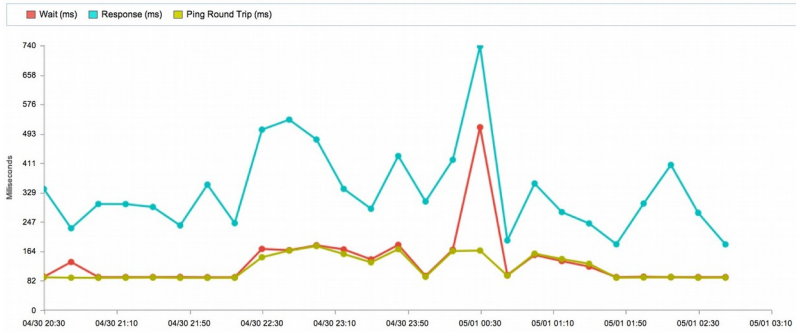
Monitoring: What?

- Basic:
  - Devices are up: ping
  - Network is working: interfaces, routes
  - Services available: HTTP, DNS, SSH



Monitoring: What?

- More complex examples:
  - Application statistics: load, performance
  - SSL certificate validity, web APIs
  - Network performance, latency, data usage



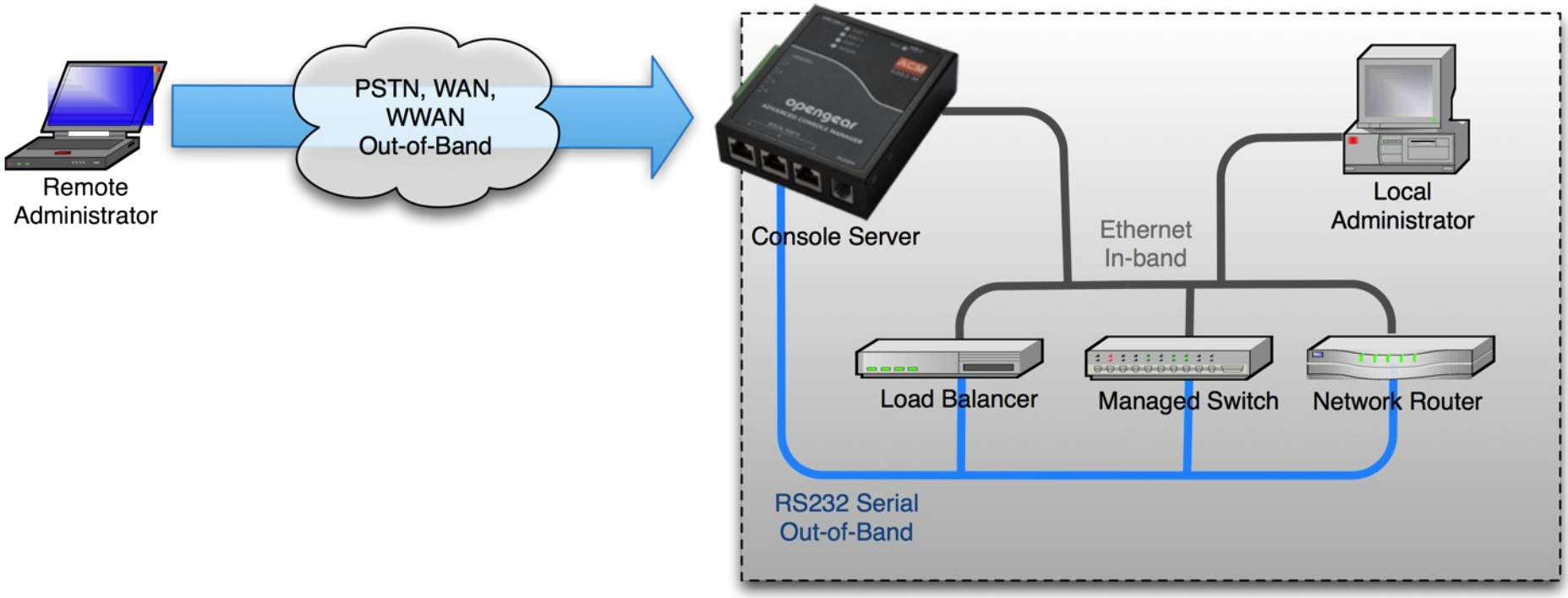
Monitoring: What?

- Environmental:
  - Power supply, load, and UPS
  - Temperature & humidity
  - TTL I/O: alarms, contact sensors
  - GPS (mobile installations)

Monitoring: How (access)?

- Primary connection:
  - Can go down
  - May not be present
- Out-of-band: independent channel
  - When primary network is down
  - When provisioning/troubleshooting remote hardware
- PSTN modem, Wi-fi, Cellular (3G/4G)

Out-of-band

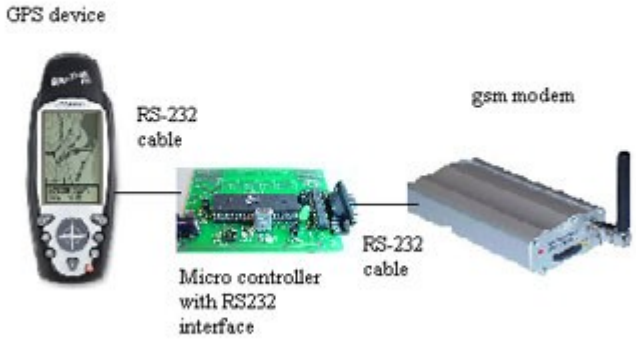

Remote Administrator
PSTN, WAN, WWAN Out-of-Band
Console Server
Ethernet In-band
Local Administrator
Load Balancer
Managed Switch
Network Router
RS232 Serial Out-of-Band

Cellular Out-of-band

- Network independent from primary carrier
- Pervasive: straight out of the oven
- Quick to provision
- Fast (remote access, site access)
- SMS for portable notifications/control

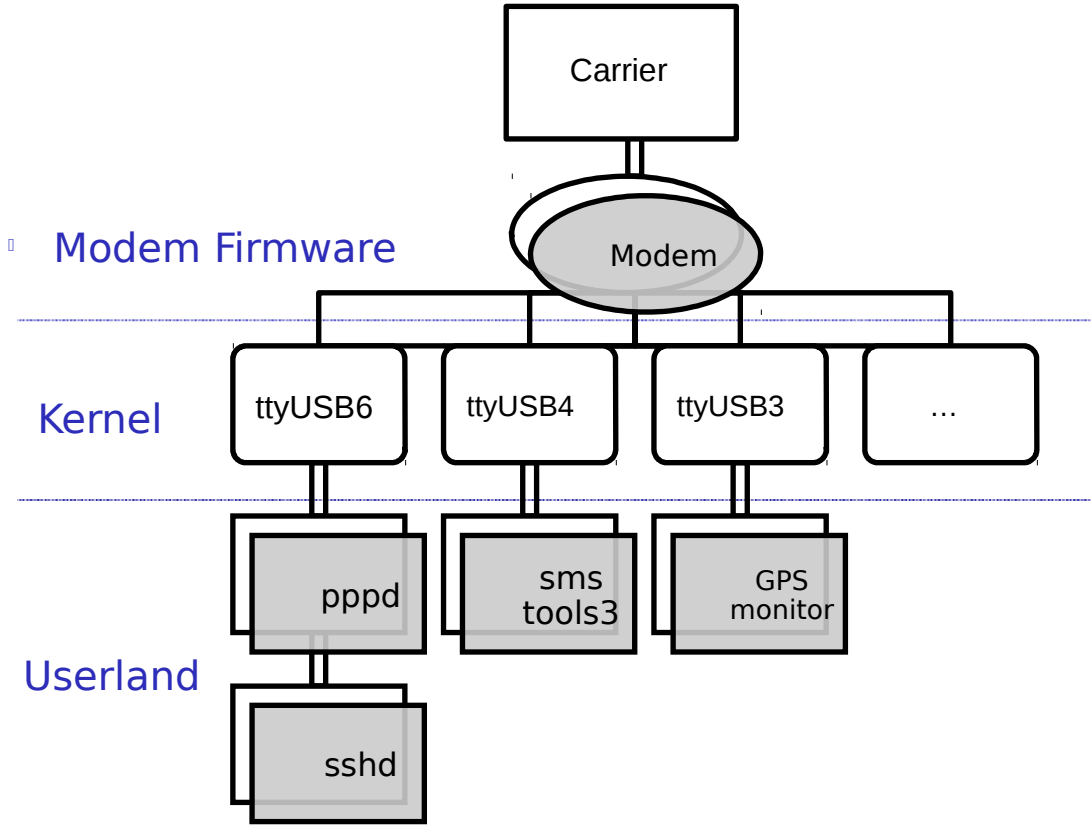
Cellular Linux support

- Originally RS232 serial, later USB CDC ACM serial
- AT commands with GSM extensions (APN, RSSI)
- PPP to tunnel IP over serial



Cellular Linux support

- Became multiple serial endpoints
  - (command, data, GPS, …)
- Proprietary SDKs to access and manage interfaces



Cellular Linux support

- 4G / LTE evolving to USB ethernet
  - (some serial channels for control)
- Interfaces are becoming more standardized
  - Sierra directIP
  - Qualcomm QMI / GOBI modems

libqmi

Cellular: what can go wrong?

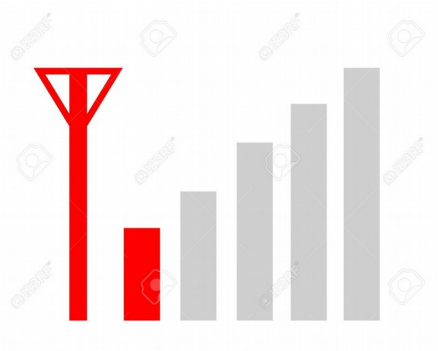
- Dropped Connections
  - Idle or unresponsive connections
  - Overloaded base station
- Hardware unresponsive
  - Firmware bugs, edge case triggers
  - Marginal reception, high USB power draw

Cellular: what can go wrong?

- Kernel/driver/SDK problems
  - Memory leaks, OOM-killer failures
  - Kernel panics
  - TTY interface endpoint problems
- Modem config changes (APN, auth, etc.)
- Unauthorized SMS commands

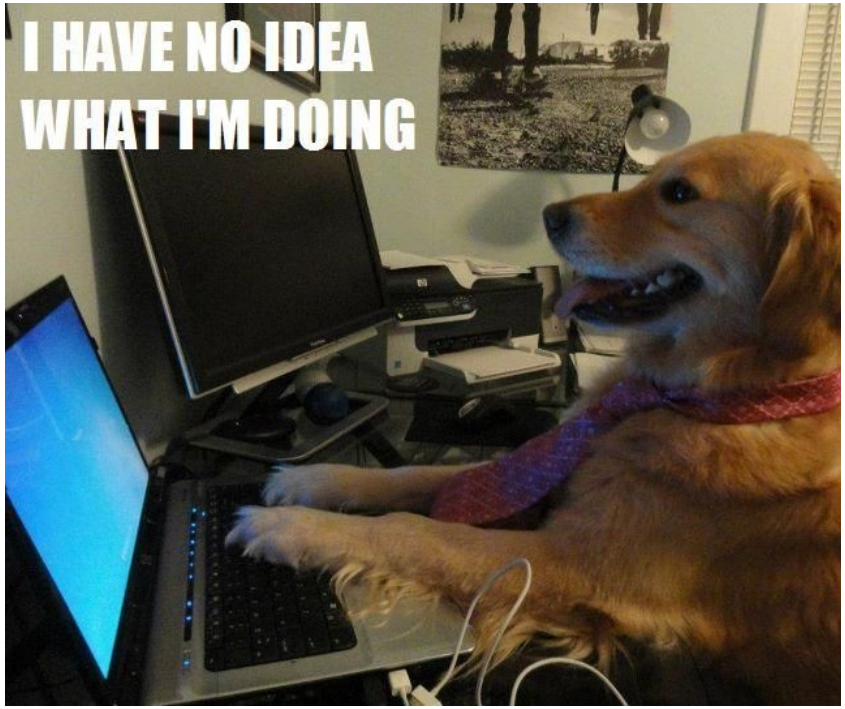
Cellular: how can we handle that?

- Upgrading modem firmware and SW support
- Monitor RSSI for good antenna placement
- Multiple carriers (dual SIM)
- Restrict SMS: trusted #'s
- Wrappers: eg. handle TTY loss
- Watchdogs. Lots of watchdogs.



Cellular: Watchdogs

- Network connectivity:
  - Periodic ping test
  - Periodic DNS lookup
  - Restart PPP or interface if suspicious
  - Helps as an ISP keepalive

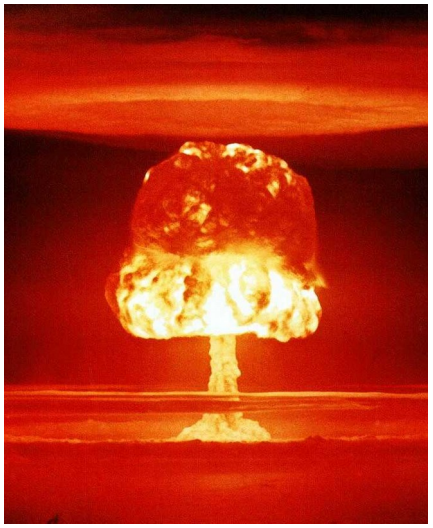


Cellular: Watchdogs

- Software interface:
  - Monitor device (AT commands)
  - Monitor TTY endpoints
  - Restart software stack
  - Reset modem (power cycle USB)

Cellular: Watchdogs

- Nuclear option:
  - “Something's wrong”: reboot after long periods of failed connectivity
  - “Something's really wrong”: hardware watchdog for kernel panics, hard locks



- With multiple levels of watchdogs and checks, we will have a reliable OOB connection
- What do we do with it?

Monitoring: How (software)?

- Central framework (Zenoss, Solarwinds, Zabbix, Nagios)
- Custom scripts and binaries
  - Digital I/O
  - Environmental
- OSS tools
  - NUT: 'Network UPS Tools': power management
  - Powerman: power outlets
  - smstools3: SMS spooler over cell TTY
  - SNMP polling of other devices

SMS Server Tools 3

Monitoring: Nagios

- Widely used OSS infrastructure monitoring
- Based on 'checks'
  - Program/script that runs to test something, reports status
- Active & passive checks (NRPE, NSCA)
- Integrates into other commercial systems

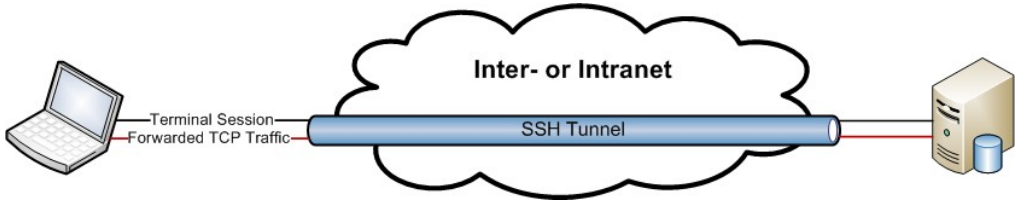
Nagios®

Nagios over Cellular

- Mobile plans usually NAT'ed; inbound needed:
  - Active checks (NRPE)
  - Remote access & troubleshooting
- Remote initiated tunnel or VPN to central server
  - SSH
  - stunnel
  - OpenVPN

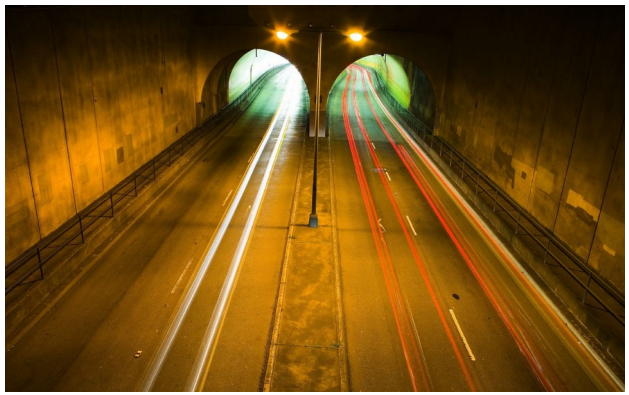
SSH over Cellular (“Call Home”)

- Remote device initiates SSH, with port forward:
  - *ssh -N -R 2222:localhost:22 tunnel@central.nagios.server*
- Keepalives to avoid carrier timeouts:
  - *-o ServerAliveInterval=30 –o ServerAliveCountMax=3*
- Restart if the port forward fails:
  - *-o ExitOnForwardFailure=yes*



SSH over Cellular (“Call Home”)

- Use reciprocal key auth for user on both sides
  - Check fingerprints (on both sides!)
  - Consider restricting user's shell (eg. GNU rush)
- Will have automatic tunnel to remote devices:
  - *ssh tunnel@localhost:2222*
- Can proxy tunnels on server:
  - *Match User tunnel*
    - *GatewayPorts yes*



SSH over Cellular: caveats

- **Data usage!**
  - Can minimize data usage
  - Pooled business data plans
  - Use in-band when available
- Alternative carrier networks:
  - Public IPs (can be hard to get)
  - IPv6 (ensure software support)
  - Carrier-managed VPN networks

Data usage tips and tricks

- Enable SSH compression
- Tunnel passive Nagios checks:
  - *ssh -C -N -L 5667:localhost:5667 ...*
- Tune Nagios checks:
  - Increase *check_interval* for regular checks
  - Decrease *max_check_attempts* before alerting
  - Use *volatile & passive* checks where possible

Data usage: monitoring

- ISP web API
- iptables accounting rules
- Polling procfs interface statistics
- OSS packages: vnStat, ntop, MRTG, etc.

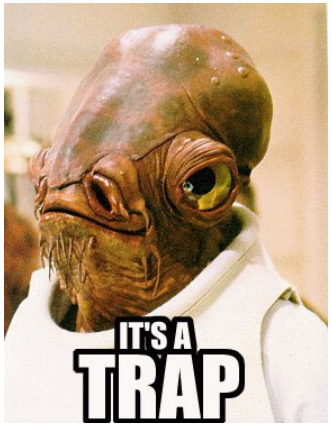
vnStat ntop MRTG MULTI ROUTER TRAFFIC GRAPHER

Data usage: complement with in-band

- Bring up cellular dynamically
  - Monitoring script
  - Connection management daemon
  - Manually with SMS command
- SSH tunnel restarts/reconnects automatically
- Smart connection manager: dependencies, failover groups, smart checking

Bringing it all together

- Monitoring & remote access:
- Out-of-band access is essential
  - Cellular is an excellent candidate network
  - Data usage is a big trap
  - Defensive design: assume it will fail, and always try to recover



- Remote → home VPN gives central access

Questions?