# Data routing and transmission protocols

Dr Lloyd Wood

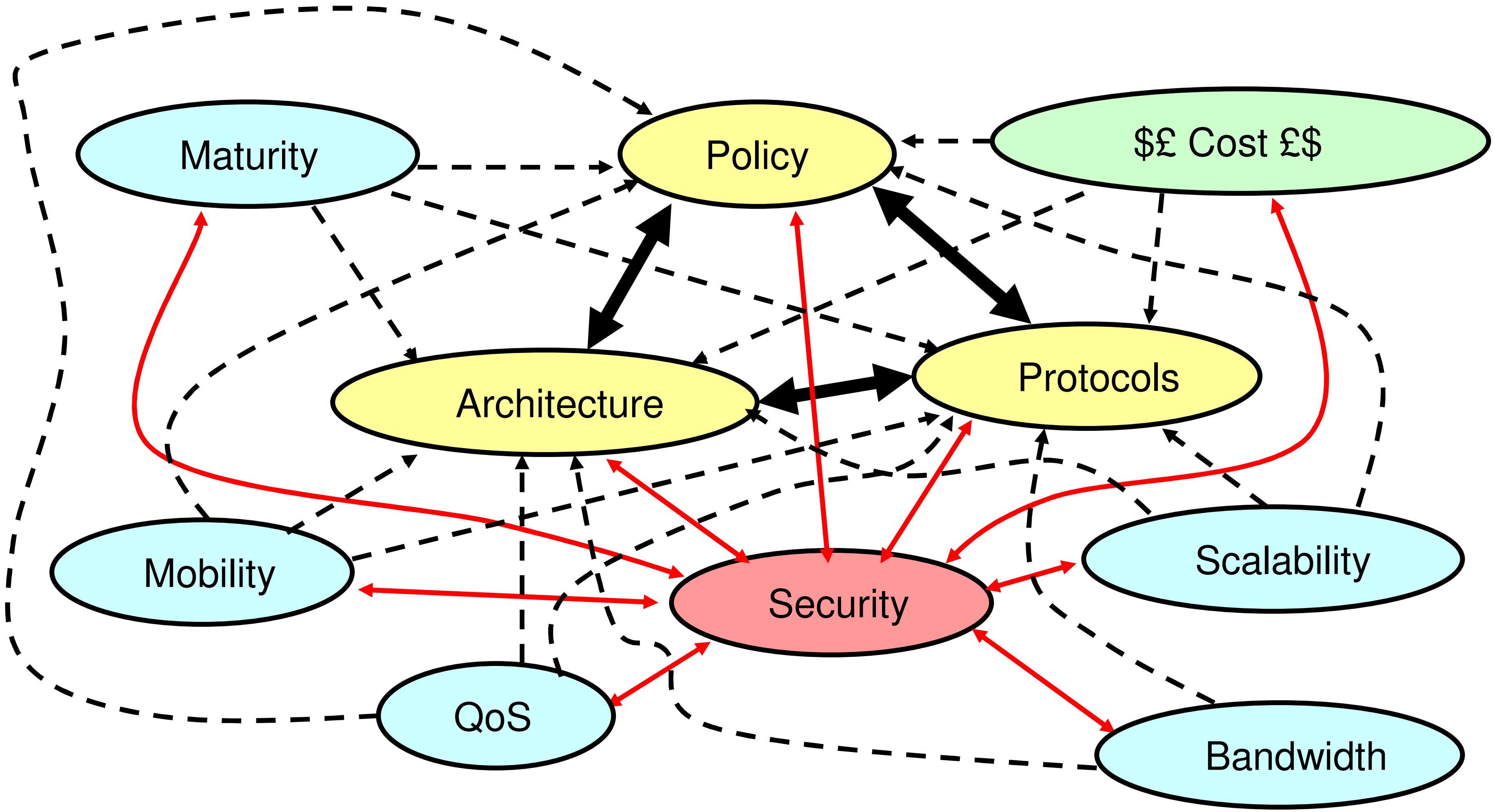
Global Defense, Space and Security, Cisco Systems

http://www.cisco.com/go/space

DARPA Fractionated Spacecraft Workshop, Colorado Springs, 3 August 2006.

# Network Design Triangle

## really choices from pre-existing designs…



# Protocol stack layering

| OSI Model<br>classic theory | TCP/IP stack<br>actual practice | |
|---|---|---|
| application | application<br>(presentation?)<br>(session?) | Email, FTP, ssh,<br>web browsing… |
| presentation | | |
| session | | |
| transport | (session?)<br>transport | TCP(+http?), UDP(+RTP),<br>SCTP… |
| network | network | IPv4 (dominant)<br>IPv6 (on the up) |
| data link | link/MAC | IEEE 802, SONET,<br>Frame Relay (HDLC), etc… |
| physical | physical | channel coding;<br>Wire, Fiber, RF, etc… |

# Protocol layers

- Why have these separate layers? It's good engineering.
  - flexibility and extensibility.
  - clear separation with standard interfaces between layers.
  - divide and conquer – modularity.
  - customise layer to meet a need, without changing entire stack.
- Transport layer
  - Really protocols to *transfer* data between end systems.
- Network layer
  - End-to-end addressing, to allow routing and forwarding.
- Data link layer
  - point-to-point *link-local* addressing so edges can communicate.
  - maps network layer onto different physical layers.
  - provides medium access control (CDMA, TDMA, FDMA, Collision Sense Multiple Access, Aloha, etc.)
  - engineering channel characteristics (IP wants bit error rate $< 10^{-6}$); the link ARQ/channel FEC tradeoff.

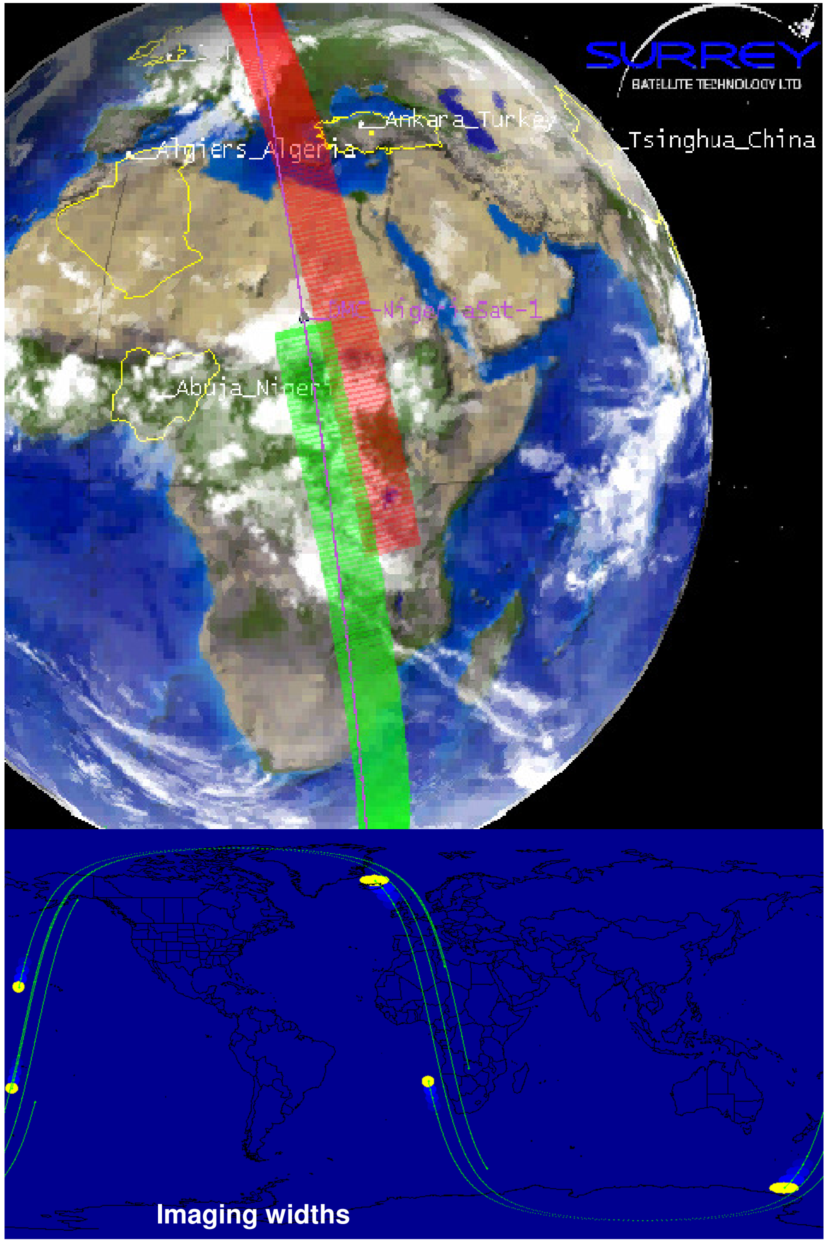
# IP and the Internet are *not* TCP

- Internet has hundreds of protocols running over IP. TCP is just one protocol; many others (DNS, ssh, streaming video) use UDP instead.
- TCP performs poorly over satellite. So?
- TCP's operating assumptions: Competition; loss is congestion. Backoff ensures fairness.
- Once outside our shared terrestrial Internet, TCP's assumptions become less useful.
- Other protocols don't share TCP's design assumptions; have different delay limitations.

# Scheduling model is different

- If you operate all the payloads, they don't have to compete; you can schedule them one after another to use the dedicated link.
- So TCP's congestion control doesn't help you; it just gets in the way.
- Coarse-grained scheduling model and shared ownership *vs* fine-grained and competition between different owners on the ground.
- UDP via static route from Pluto? Why not?

# Example of *not* using TCP #1




- SSTL's DMC – Disaster Monitoring Constellation
- Transfer stored images from LEO satellite payload to ground station during pass using Internet Protocol.
- 10-minute pass window, 8Mbps S-band downlink.
- Could download up to a gigabyte of data in that time during each pass, if link filled back-to-back.

# Example of *not* using TCP #2

- Avoided TCP – too slow and chatty (uplink bottleneck); only *one* payload talking at a time to the ground station, not beyond. No congestion!
- Implemented CCSDS CFDP (based on UDP) – quite complex, and still not fast enough.
- So wrote own very simple custom UDP-based transfer protocol with NACKs (called it *Saratoga*) to fill their link with packets all the time.
- Performance improvement, but still using Internet Protocol. Engineering optimisation in *one* protocol layer to suit the problem.

# Delay budgets

- Need to think about delay requirements of protocols and dimension for them – just like doing link budget for signal power levels.
- You can oversaturate a modem with a too-powerful too-close source, or a signal from a far-away source can be too weak to be picked up at all, as it's below the noise floor.
- The modem has a *dynamic range*.
- Similarly, each protocol has a dynamic range in tolerated delays for its timers. TCP is weak over satellite. Or for supercomputing. Those applications lie either side of TCP's range; performance falls off.

# Routing #1

- Routing protocols also have dynamic ranges, and state machines with complex timing interactions.
- Pick the routing protocol based on engineering fitness for purpose, matching the problem domain.
- If you’ve got a small cluster of satellites flying in formation, and they all need to know what each other knows, why not just bridge over shared wireless?
  Routing could be overkill in this scenario.

# Routing #2

- Routing is just another protocol
- Types/Terms:
  – Not limited by timers:
    - Default route (route of last resort)
    - Static route (predetermined, fixed)
      – These are useful! If your spacecraft only has one preferred link…
  – Affected by timer limitations
    - Dynamic
    - Ad-hoc

# Routing #3

- Dynamic routing
  - Interior Protocols, inside your network
    - Distance-vector vs link-state protocols – link state generally performs better, but more complex.
    - Routing Information Protocol (RIP) – legacy these days?
      - » Small networks
    - Open Shortest Path First (OSPF) – popular.
      - » Many additional metrics available
      - » Widely used in large networks (e.g. DoD, Businesses, Universities)
  - Exterior Gateway Protocols, to other networks
    - Border Gateway Protocol (BGP)
      - » Used in the Big Internet between separately-owned *autonomous systems* such as ISPs (Internet Service Providers). Usually policy-, not efficiency-, -driven.

# Routing #4

- Ad-hoc routing protocols
  - MANETs - Mobile Ad hoc NETworks
    - Self-configuring and self-organizing network of mobile nodes usually connected via wireless links
    - *Proactive* routing protocols: Optimized Link State Routing (OLSR), Open Shortest Path First (OSPF) extensions
      - Useful in relatively stable networks
      - Suitable for large and dense networks
    - *Reactive* routing protocols: Ad Hoc On-Demand Distance Vector (AODV), Dynamic Source Routing protocol (DSR), Dynamic MANET On-demand (DYMO)
      - Applicable to highly dynamic networks
- Pick according to fitness for purpose.

# Mobility

- IP was designed for a fixed computer network. Real mobility wasn't considered.
- **Mobile IP** is really to make roaming devices outside your network think they're in your home network for admin purposes. Hides handovers from protocols that can't cope with address changes – such as TCP!
- Mobile IP doesn't include security as-is. That's why we 'VPN in' instead.
- DHCP takes care of giving your roaming laptop connectivity. No Mobile IP needed.

# Mobility #2

- Mobility can be handled in many ways by the routing protocol or at a higher layer (dynamic DNS, even?); it’s just indirection with unique identification of endhosts. Modern transfer protocols like SCTP cope with multihoming/handovers themselves.
- Like routing, there are lots of pre-existing design solutions that you can adopt.
- Example of *not* using Mobile IP: Boeing’s *Connexion* adds and withdraws BGP routes as planes move between satellite footprints.

# Security

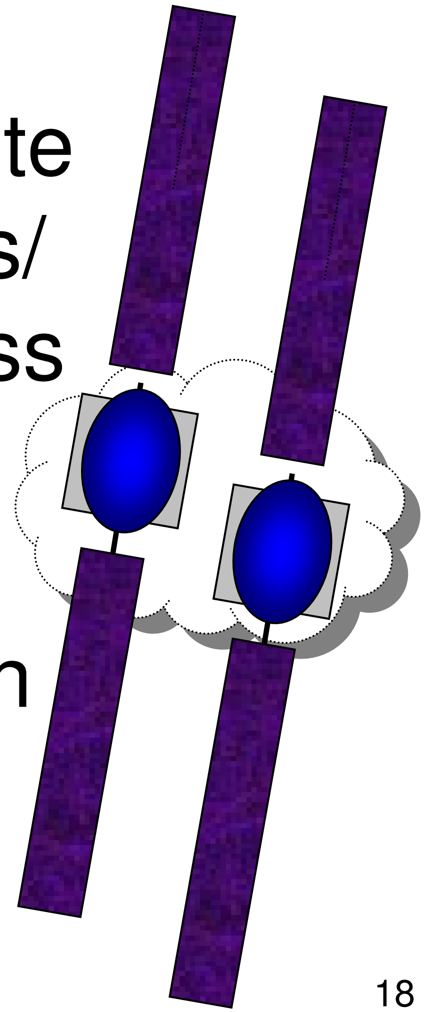
- IPSec works well for securing/authenticating endhosts and transfers of data.
- HAIPE adopts a very IPSec-like approach (apparently)
- Not sufficient for the (justly) paranoid; can still do traffic analysis of who's talking to whom.
- So you want link crypto, too. And sending out null frames to mask traffic patterns…
- …on a limited spacecraft power/CPU budget?
- Engineering again:
  how much security is considered enough?

# Three views of fractionation

1. Fractionate communications – split computing power between spacecraft for survivability and redundancy, use intersatellite links in clusters that communicate in formation. Discussed for small interlinked geostationary satellites – the *slot clouds* concept.
2. Fractionating within the spacecraft – making it modular with plug and play payloads (the approach taken by SSTL; how the Cisco router got into space so quickly.)
3. And fractionate your protocol stack into layers!

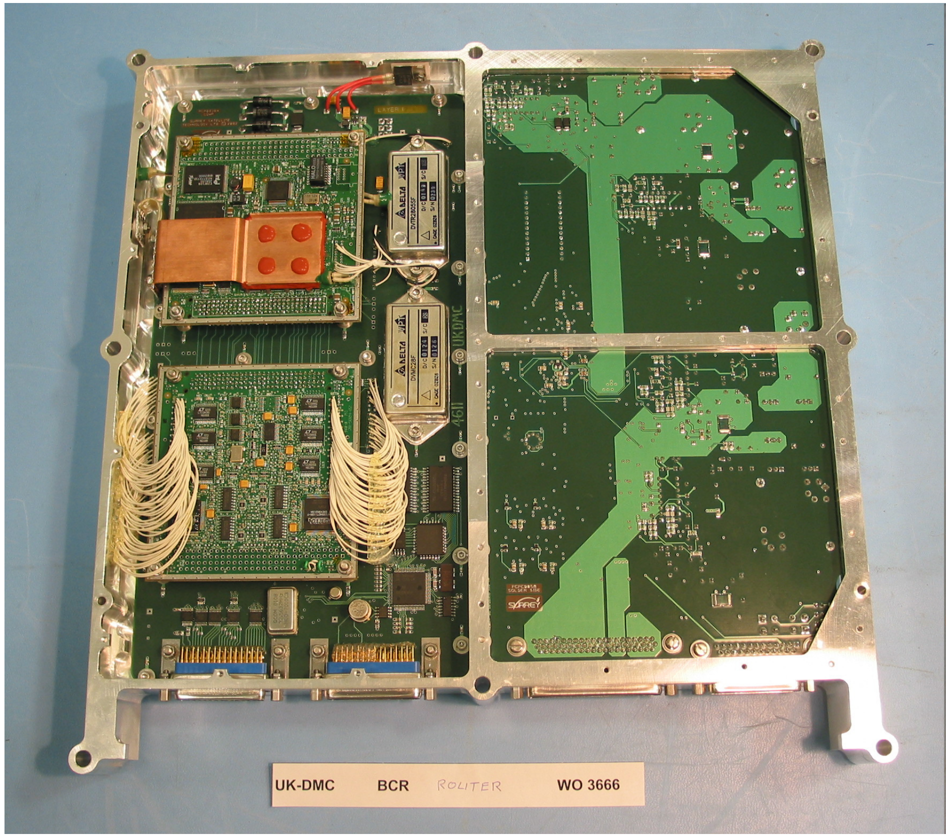
# 1. *Slot clouds*

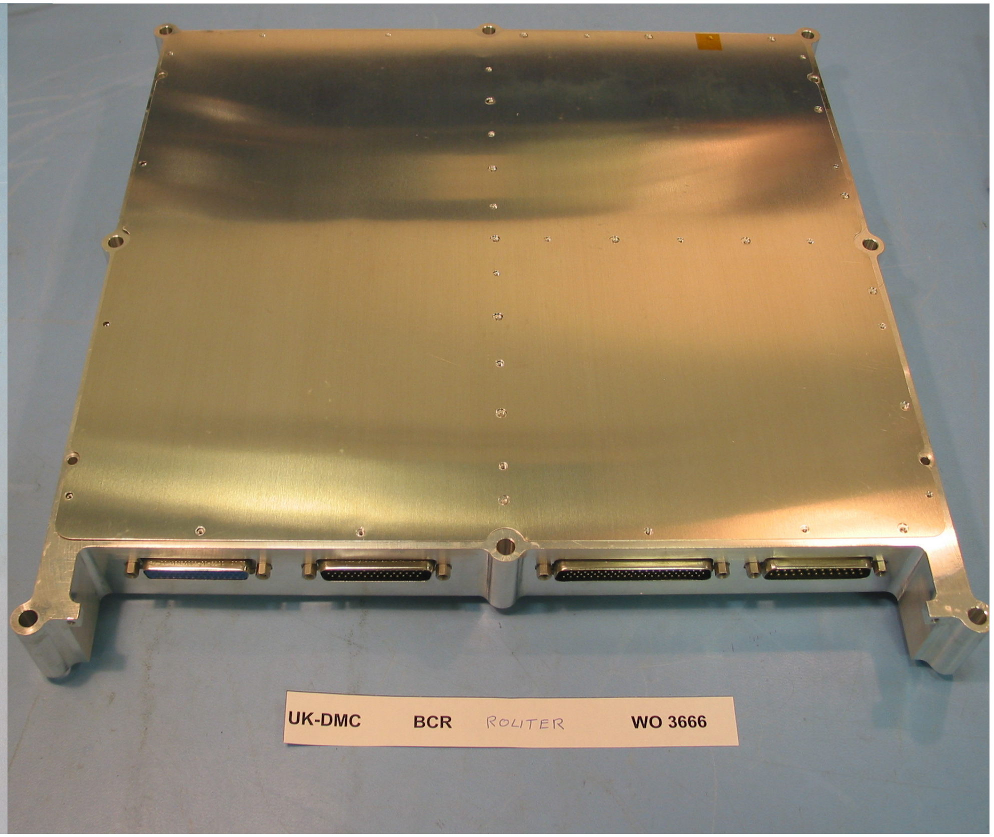
- Idea of small geostationary satellites, stationkeeping in formation.
- Communicating via short intersatellite links, at far higher rates than uplinks/ downlinks. (Reuse terrestrial wireless technologies where the ranges are similar?)
- Creates a larger *virtual* satellite, with capabilities added to over time.
- Computing cluster in space, really.



# 2. Modular spacecraft

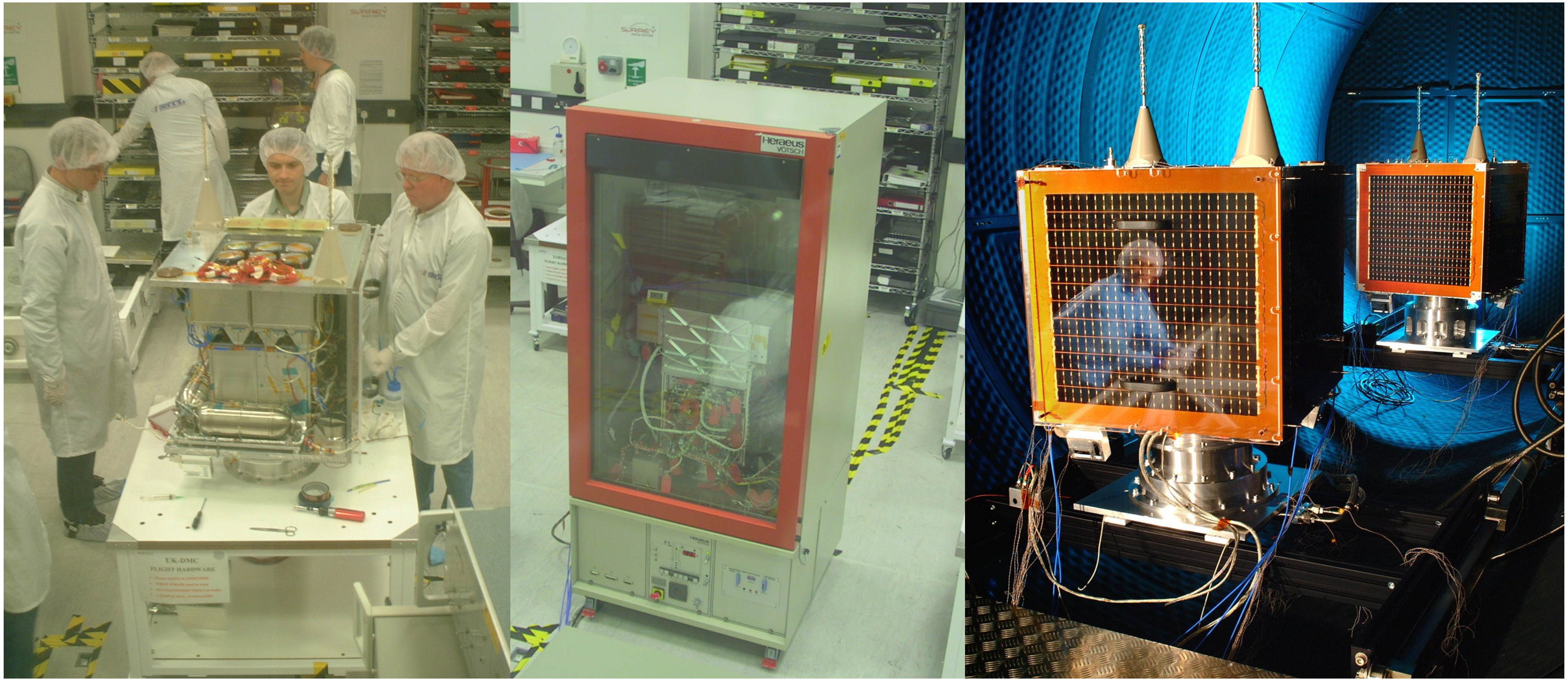
- Routers usually live in racks in airconditioned rooms – the Cisco router in orbit just lives in half of a different kind of rack onboard the UK-DMC satellite.
- Faster modular development and payload integration.
- Smart satellite buses.



# 3. Fractionating the protocol stack

- Where we came in.
- Like fractionating the satellite, it's just good modular engineering.



**Questions?**

thankyou

http://www.cisco.com/go/space