Networking for Nanosats

Luke Stras

www.stras-space.com
A Step Back

• old-school design – centralized CPU
• works great, but...
  - massive wire harnesses
    • complex, heavy, inflexible
  - harder to do interactive integration
  - centralized point of failure
  - people step on each other's toes
    • and everyone wants hardware to play with
Building a Harness
So, Distribute

- put CPUs and I/O where they're needed
- hardly revolutionary
  - MIL-STD 1553, 1773
  - Hubble, SMEX, MIDEX, SAMPEX, ISS
    - 1980s-vintage Battlestars
- but Moore's Law helps
  - MOST
    - late 1990s 50 kg microsat
  - SNAP-1
    - 2000 6.5 kg nanosat
  - SpaceQuest Distiller
    - 2006 0.5 kg
Why?

• smaller wiring harness
  - just power + data
    • though still need “last-inch” I/O
  - smaller connectors

• improved robustness
  - node crash won't cascade
  - quasi-redundant (with careful partitioning)
  - decrease MTTF, but increase availability
Still Why?

• easy to partition work
  - each sub-system gets their own S/W and H/W
  - easier to distribute geographically

• easier integration
  - plug-and-play
  - pull bad nodes
  - use surrogate nodes for unfinished hardware
  - piecewise integration
How To Do It

I. pick a technology
II. implement it
III. done!
Standards Are Great

• ... because there's so many to choose from
• look at some factors
  - speed
  - power consumption
  - physical layer
  - link layer
  - network layer
• but don't make your own
  - unless that's your mission
Physical Stuff

• speed: 100 kbps to 400 Mbps
• power consumption
  - speed is power
  - consider steady-state power vs. E/bit
• topology
  - hubs need space, power, reliability
  - buses are slower, have funny connectors
• physical medium
  - pickier for faster networks
Software and Protocols

- often coupled to physical network
  - TCP over Ethernet, CANOpen over CAN
  - but IP over ATM, too
- standards are great
  - often have subtle design features
- simple terminal software is good
  - leverage existing technology
- think this through!
- good toolkits are worth their weight in bits
## Some Practical Standards

<table>
<thead>
<tr>
<th>Std</th>
<th>Speed</th>
<th>Topo*</th>
<th>Pwr [mW]</th>
<th>E/bit [µJ/bit]</th>
<th>Proto?**</th>
<th>Complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS-232</td>
<td>100k</td>
<td>PtP peer</td>
<td>45</td>
<td>0.45</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>RS-485</td>
<td>100k</td>
<td>bus peer</td>
<td>45</td>
<td>0.45</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>I2C/SPI</td>
<td>100k</td>
<td>bus M/S</td>
<td>13</td>
<td>0.13</td>
<td>Some</td>
<td>Low</td>
</tr>
<tr>
<td>CAN</td>
<td>125k</td>
<td>bus peer</td>
<td>100</td>
<td>0.86</td>
<td>Yes</td>
<td>Med</td>
</tr>
<tr>
<td>Ethernet</td>
<td>100M</td>
<td>PtP peer</td>
<td>260</td>
<td>0.0026</td>
<td>Yes</td>
<td>Med/Hi</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10M</td>
<td>PtP peer</td>
<td>125</td>
<td>0.013</td>
<td>Yes</td>
<td>Med/Hi</td>
</tr>
<tr>
<td>USB</td>
<td>12M</td>
<td>PtP M/S</td>
<td>6</td>
<td>0.0005</td>
<td>Yes</td>
<td>Hi</td>
</tr>
<tr>
<td>FireWire</td>
<td>400M</td>
<td>PtP peer</td>
<td>930</td>
<td>0.002</td>
<td>Yes</td>
<td>Hi</td>
</tr>
</tbody>
</table>

* Network topology
  - PtP = point-to-point
  - bus = common bus
  - peer = nodes are equivalent peers
  - M/S = nodes are in master/slave relationship

** Higher-level protocols defined?
No Silver Bullets

- more power
- more area
- funny connections (maybe)
- more pieces to test
- more software to write
  - remember Brooks' Law
Conclusions

• small satellites don't have to be bespoke
  - getting less so every year
• easy to partition work
  - especially for multi-year projects with high staff turnover
• sometimes, you plug it all in, and it just works
  - but when it doesn't, you can isolate broken bits
“Classic” wiring harness

Network cabling