# SIS-AMS Working Group

Colorado Springs, USA

January 16-20 2007

# Agenda: Tuesday

- Tuesday 16 January:
  - 10:00 Overview of progress since June.
  - 10:30 Results of RAMS tree-structured network testing.
  - 11:30 Results of real-time message exchange testing.
    - VxWorks port, message queues.
  - 12:30 Adjourn.

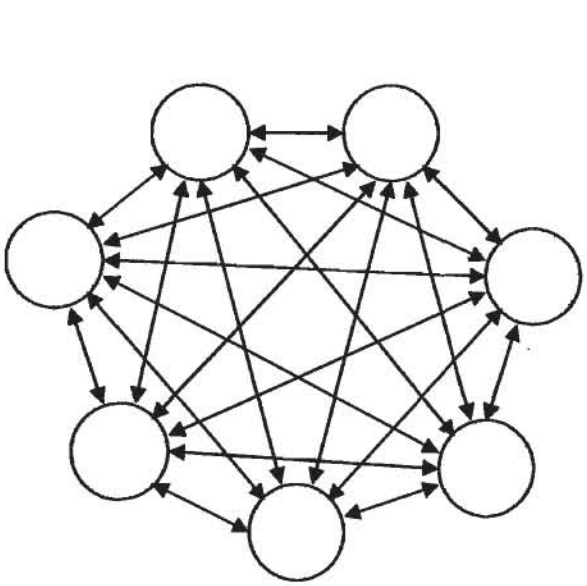
# Overview of Progress Since June

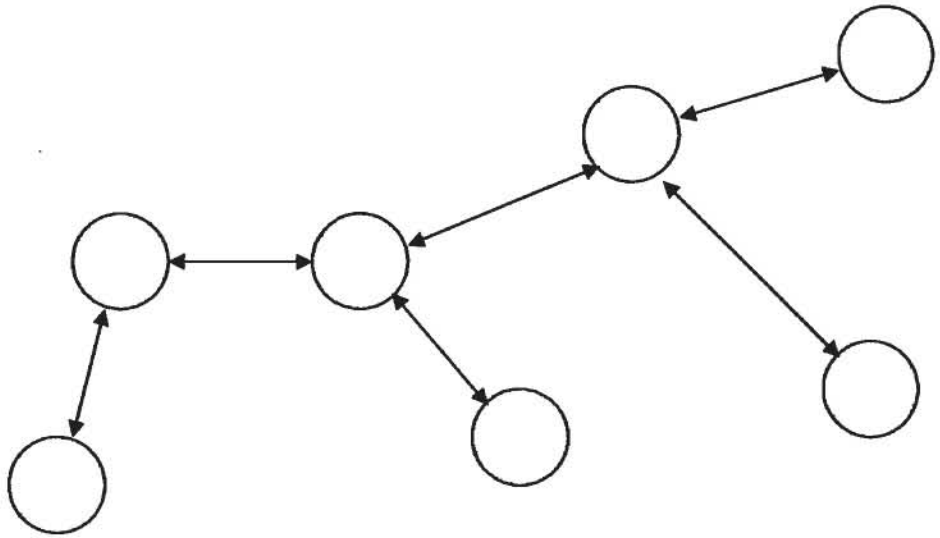
- Upgraded reference implementation per changes to draft Red Book (from June meetings).
- Upgraded and clarified Remote AMS specification.
  - Added support for RAMS networks that are configured as trees (as well as meshes).
- Successfully ported AMS to VxWorks, benchmarked real-time messaging performance.

# Remote AMS Progress

- Original RAMS design supported only RAMS networks configured as meshes (every gateway is directly connected to every other).
  - Simple and fast, but not scalable: too much overhead at each gateway as the number of gateways increases.
- Upgraded RAMS design supports option of configuring RAMS network as a tree.
  - End-to-end message latency may be greater but number of neighbors at each gateway can be kept small, enabling the network to scale up to large numbers of gateways.

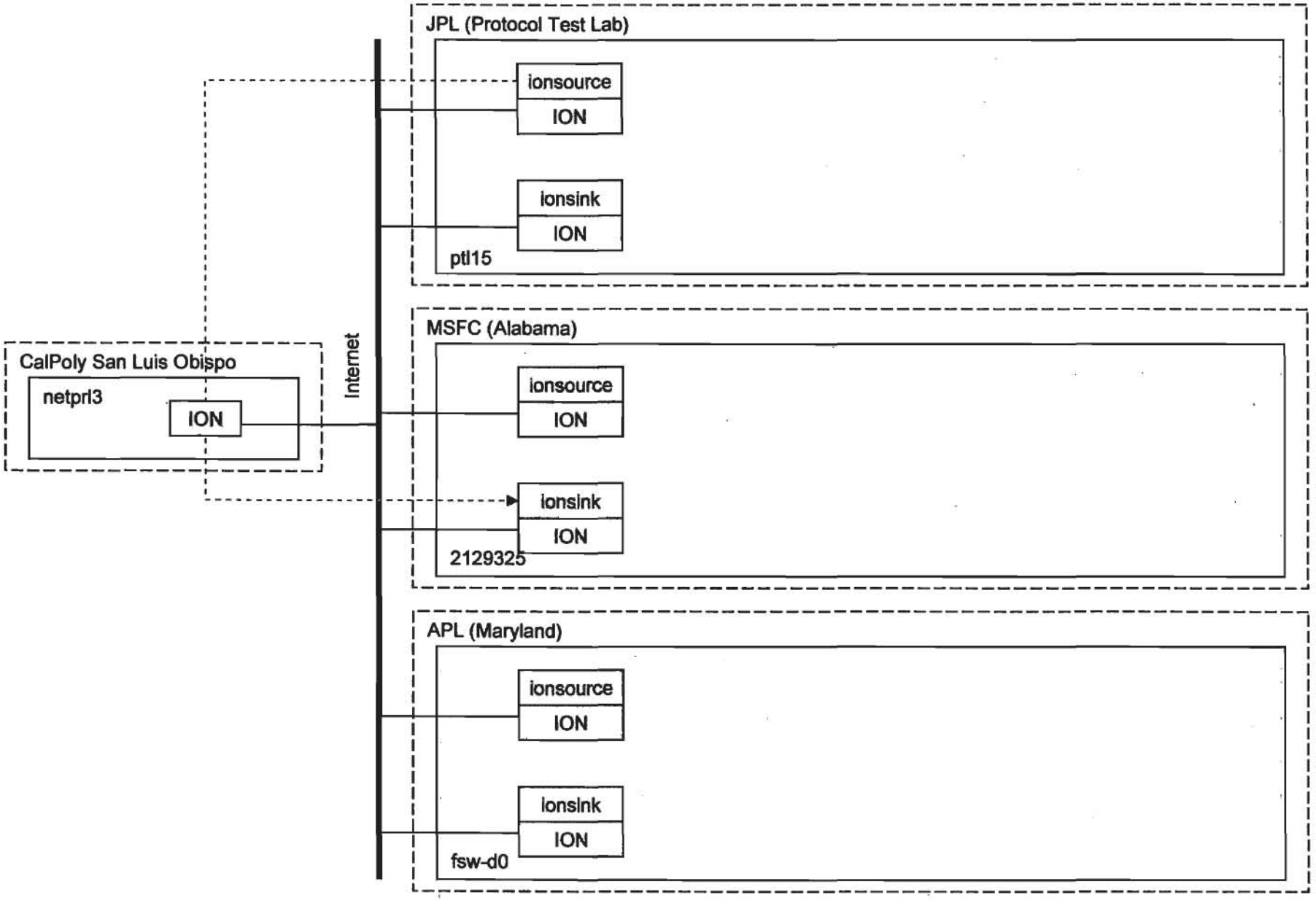
# RAMS Mesh vs Tree

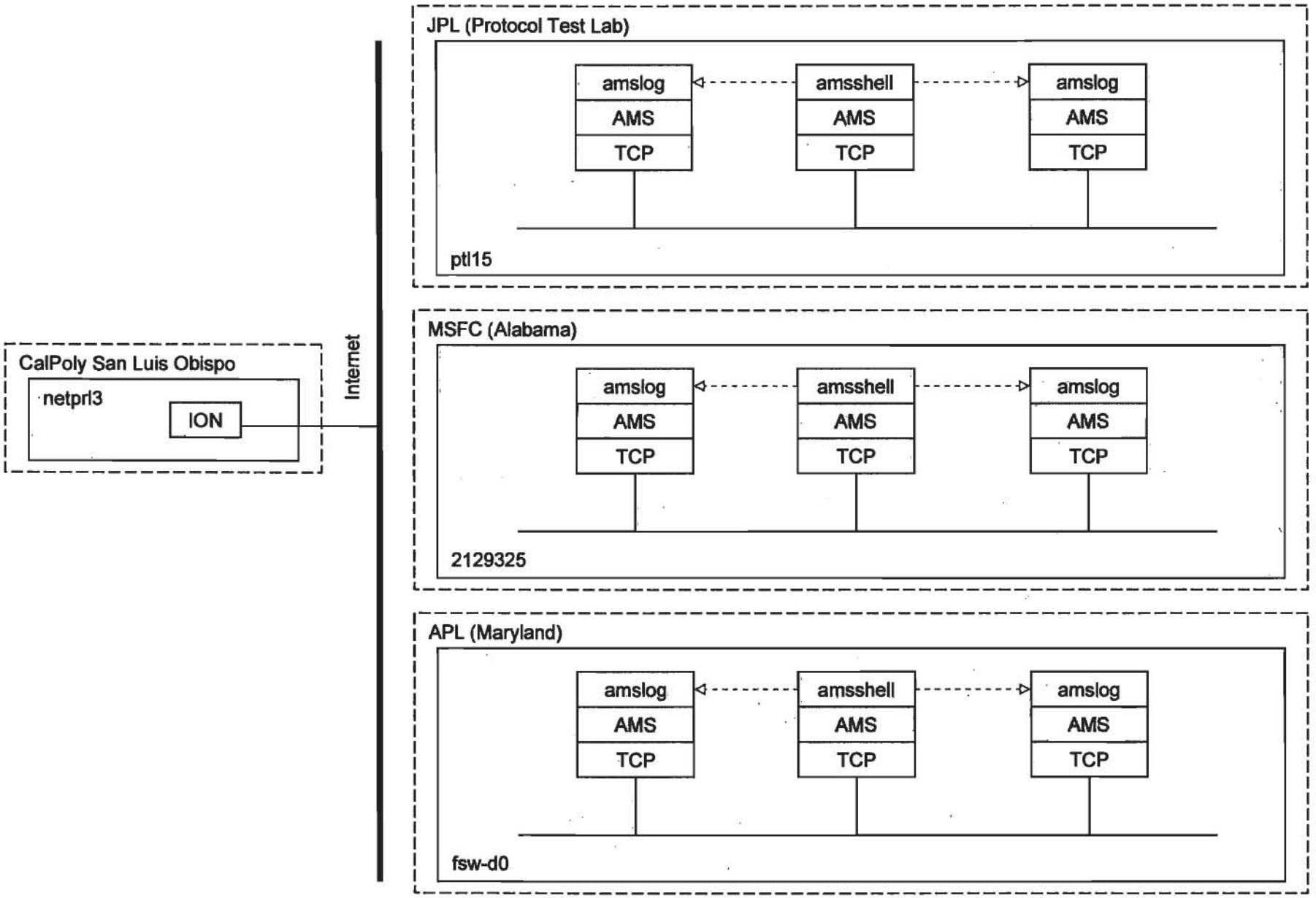


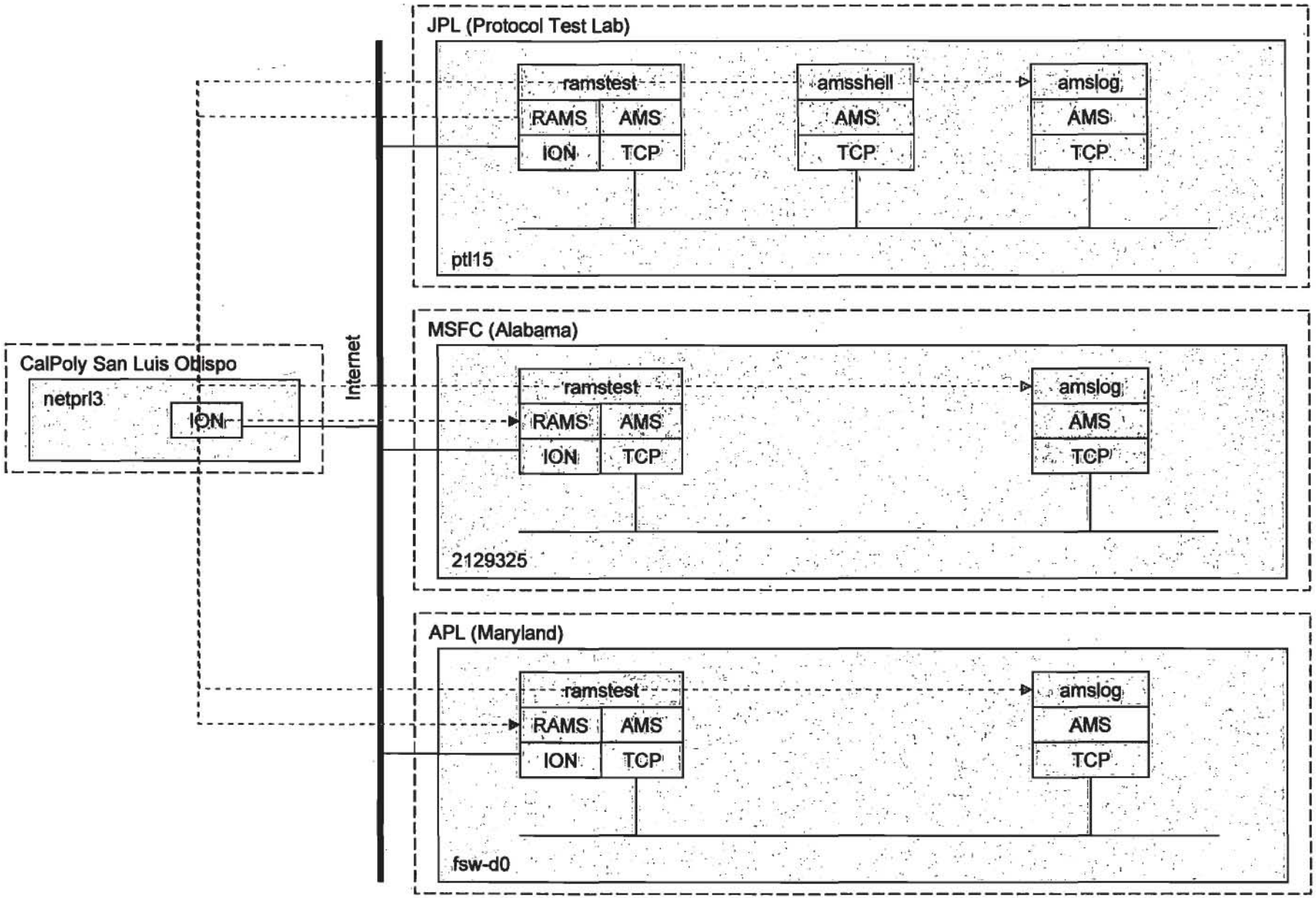


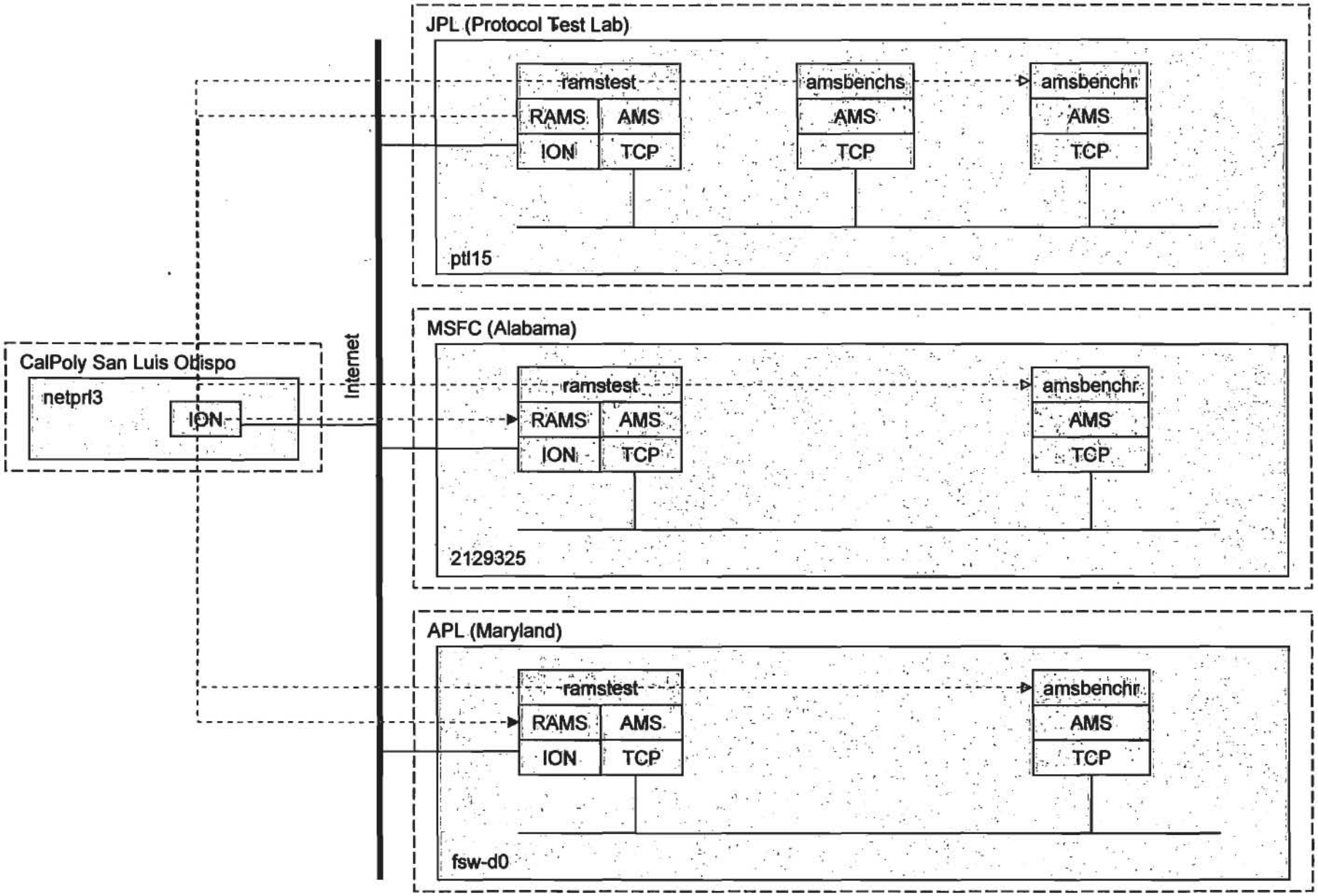
# RAMS testing exercise

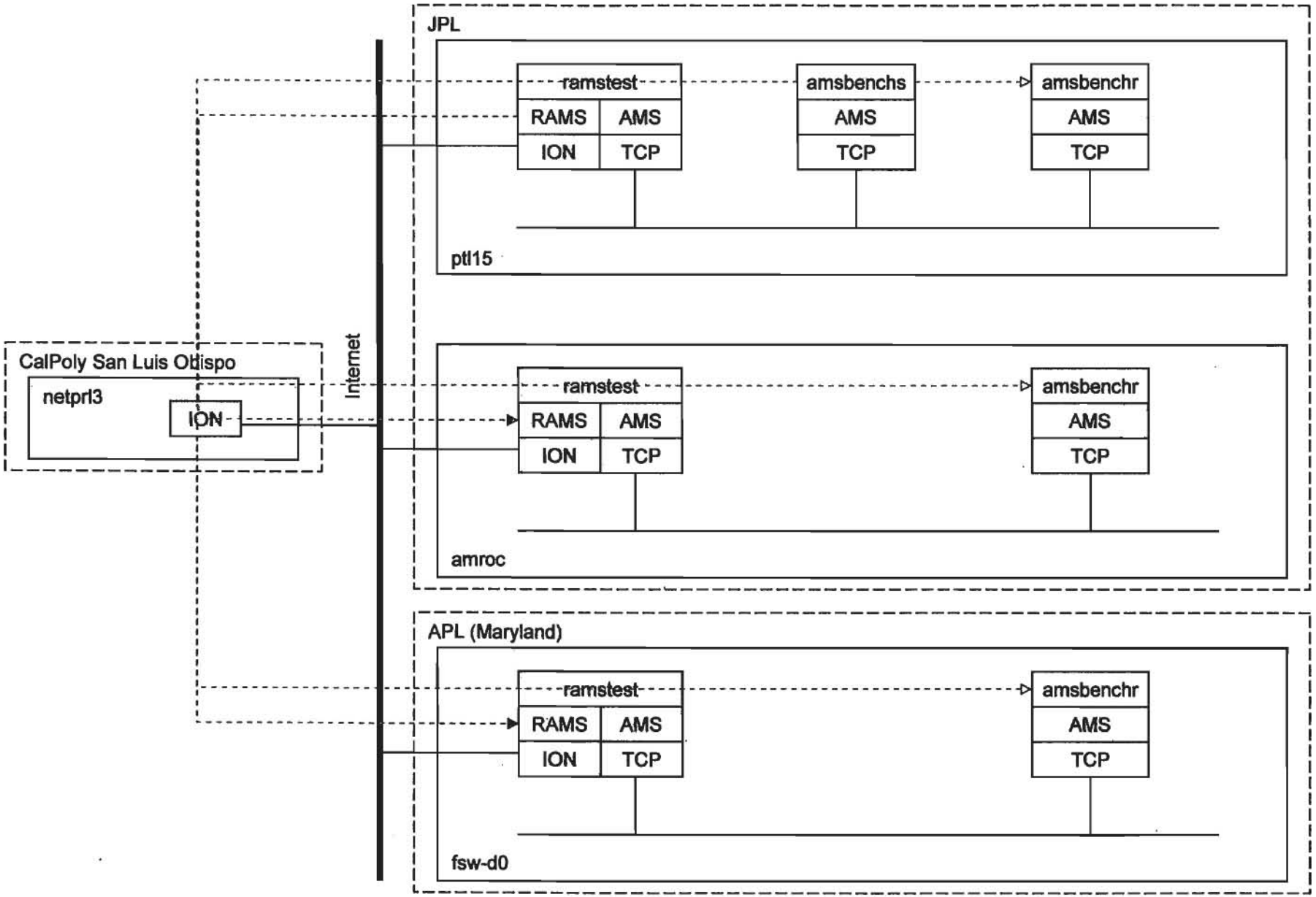
- Objective: demonstrate AMS message exchange over a wide-area network (the Internet), characterize performance.
- Method:
  - Operate AMS continua at JPL, APL, Marshall SFC.
  - Delay over the Internet is probably low enough to enable all three centers to be in a single continuum; to make RAMS necessary, did no firewall modification at any center.
  - Therefore all traffic had to travel through a third-party routing point at Cal Poly San Luis Obispo – a star-shaped overlay network.
  - Needed an overlay network protocol for this, so used JPL's DTN Bundle Protocol implementation ("ION").


JPL (Protocol Test Lab)
ionsource
ION
ionsink
ION
ptl15
MSFC (Alabama)
ionsource
ION
ionsink
ION
2129325
APL (Maryland)
ionsource
ION
ionsink
ION
fsw-d0
Internet
CalPoly San Luis Obispo
netprl3
ION


JPL (Protocol Test Lab)
amslog
AMS
TCP
amsshell
AMS
TCP
amslog
AMS
TCP
ptl15
MSFC (Alabama)
amslog
AMS
TCP
amsshell
AMS
TCP
amslog
AMS
TCP
2129325
APL (Maryland)
amslog
AMS
TCP
amsshell
AMS
TCP
amslog
AMS
TCP
fsw-d0
Internet
CalPoly San Luis Obispo
netprl3
ION


JPL (Protocol Test Lab)
ramstest
RAMS
AMS
ION
TCP
amsshell
AMS
TCP
amslog
AMS
TCP
ptl15
MSFC (Alabama)
ramstest
RAMS
AMS
ION
TCP
amslog
AMS
TCP
2129325
APL (Maryland)
ramstest
RAMS
AMS
ION
TCP
amslog
AMS
TCP
fsw-d0
Internet
CalPoly San Luis Obispo
netprl3
ION


JPL (Protocol Test Lab)
ramstest
RAMS
AMS
ION
TCP
amsbenchs
AMS
TCP
amsbenchr
AMS
TCP
ptl15
MSFC (Alabama)
ramstest
RAMS
AMS
ION
TCP
amsbenchr
AMS
TCP
2129325
APL (Maryland)
ramstest
RAMS
AMS
ION
TCP
amsbenchr
AMS
TCP
fsw-d0
Internet
CalPoly San Luis Obispo
netprl3
ION


JPL
ramstest
RAMS
AMS
ION
TCP
amsbenchs
AMS
TCP
amsbenchr
AMS
TCP
ptl15
CalPoly San Luis Obispo
netprl3
ION
Internet
ramstest
RAMS
AMS
ION
TCP
amsbenchr
AMS
TCP
amroc
APL (Maryland)
ramstest
RAMS
AMS
ION
TCP
amsbenchr
AMS
TCP
fsw-d0

# Real-Time Message Exchange

- Presentation by Amalaye Oyake.

# Agenda: Thursday

- Thursday 18 January:
  - 8:30 Use of AMS in wireless avionics testbed.
  - 9:30 Prospects for real-time message exchange over ring bus.
  - 10:00 Use of AMS in support of SharedNet.
  - 10:30 Open discussion of other applications and ops concepts.
  - 12:30 Break for lunch.
  - 13:30 Review of revisions to AMS Red Book, discussion of other potential revisions.
  - 17:30 Adjourn.

# AMS in Wireless Avionics

- Presentation by Amalaye Oyake.

# Real-time AMS Over Ring Bus

- Presentation by Amalaye Oyake.

# DARPA Content-Based Routing

1. Want to push data, so that it's there as soon as it's needed.
   - Minimize latency.
2. But don't want to push everything to everybody; targets must pull what they need.
   - Minimize bandwidth consumption.
3. Must retain data until user is ready to use it, and not decrypt until then.
   - Minimize need for repeated transmission.
   - Minimize security exposure.

# AMS/SharedNet Concept

- For secure data retention within the network fabric:
  DTN Bundle Protocol (BP).
- To strike a balance between data "push" and "pull":
  AMS publish/subscribe functionality.
- For secure data cache at the edge of the network:
  Sharednet relevant common operational picture.
- Sharednet client registers as an AMS node.
  - Uses AMS to subscribe on user's behalf.
  - Receives remotely published data via RAMS gateway.
  - Caches data locally, securely, pending retrieval by the user.

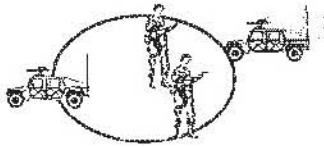
# Overview of Sharednet

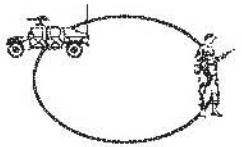
- Middleware developed by JPL for the US Navy and Marine Corps.
- Objective:
  - Distribute the C2 information needed to assemble a *relevant* common operational picture (COP).
  - Provide the distribution framework required to facilitate collaborative planning and increased situational awareness (i.e. storage, translators, and agents).
- Design Principles
  - Consumers, not producers, drive information flow.
  - Operating environment failures are inevitable and must be gracefully handled.
    - Communication sessions must survive network outages and loss of nodes.

# SharedNet – AMS – DTN

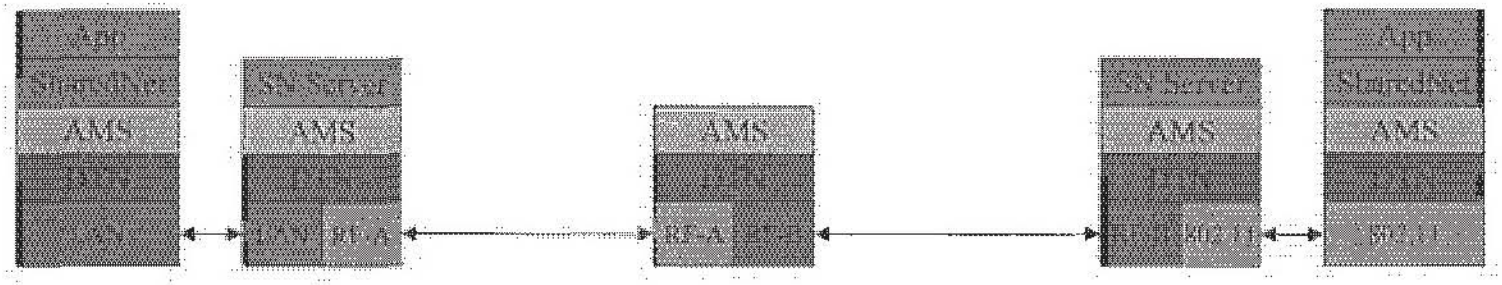
Mission Operations at Sea HQ

Airborne & Orbital Relays





UAVs, people & sensors

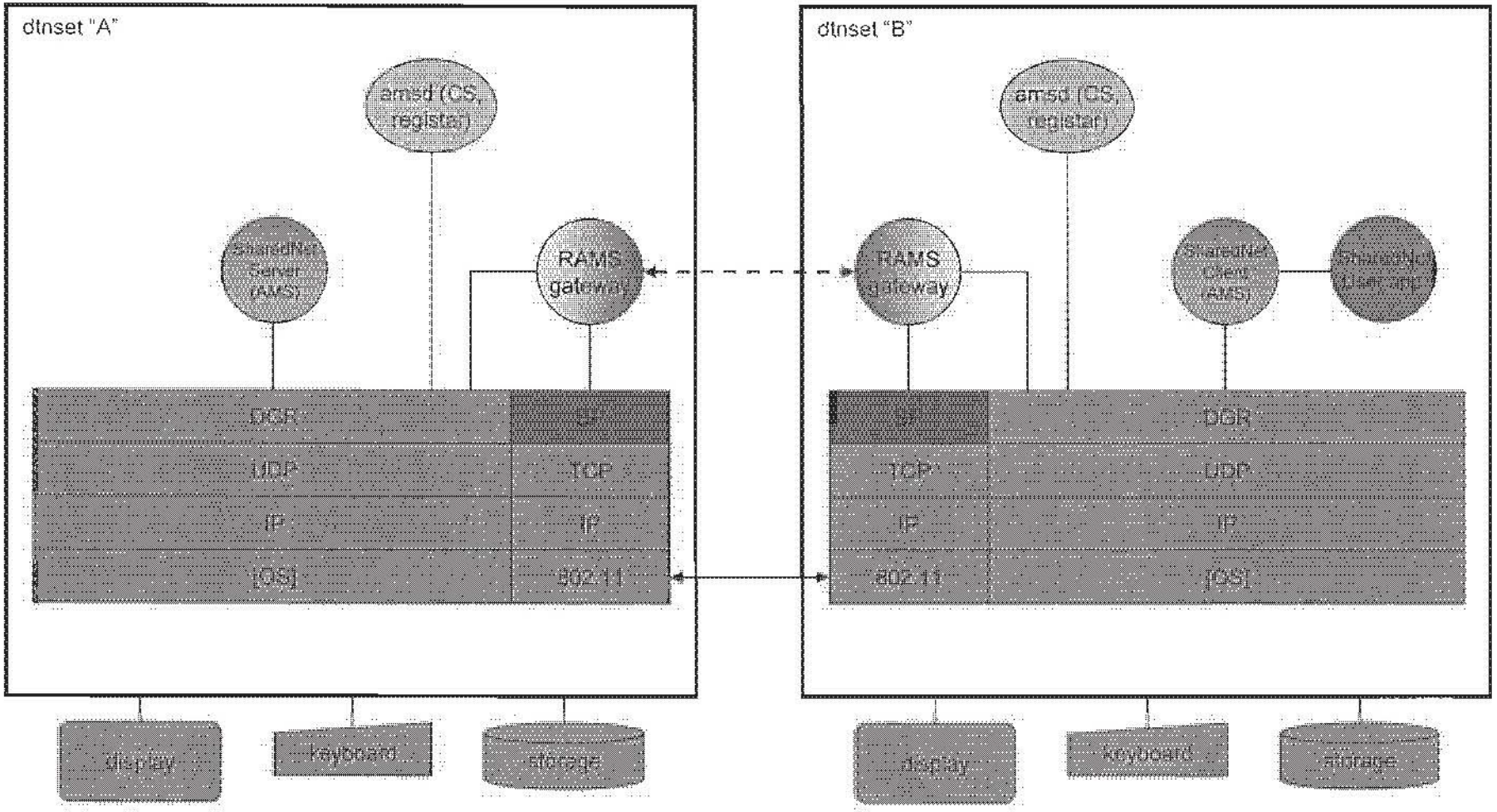



- Servers with well-connected clients
- Reachback to other information sources (e.g. through GIG)
- Interface with other systems via translators and web services

- DTN nodes enable reliable transmission and routing over transient and variable relay platforms.

- Local servers automatically synchronize with remote servers according to subscriptions on behalf of end users.
- End users can communicate within local groups even if link to HQ is down.
- All information produced/consumed by end users is stored on their devices and on local server.

# Detailed Structure



# Red Book Revisions

- Substantial change to the discussion of Remote AMS.
  - Still somewhat in flux as we implement and test the protocol.
  - But no known problems at this time.
- No other significant changes to the AMS red book.

# Agenda: Friday

- Friday 19 January:
  - 8:30 Discussion of plans for development of a second AMS implementation.
  - 9:00 Review of reference implementation, discussion of implementation issues.
  - 12:30 Break for lunch.
  - 13:30 Continue review of reference implementation and discussion of implementation issues.
  - 17:30 Adjourn.

# Reference Implementation

- JPL's implementation of AMS has the following components:
  - Daemon process (**amsd**) that can function as:
    - Configuration server
    - Registrar for a single unit of a single message space
    - Or both, depending on the values of command-line parameters.
  - Library (**libams**) that implements AMS application programming interface.
    - This library is typically provided as a "shared object" that is dynamically linked with application code at run time.
- All code written in C.

# Reference Implementation (2)

- Common infrastructure library ("ICI")
  - Portability layer ("platform") simplifies compilation and execution of common code in multiple operating environments.
    - Linux
    - VxWorks
    - Interix (Windows Services for Unix)
    - Future candidates: Solaris, RTEMS.
  - Dynamic memory management system ("PSM") provides dynamic management of a privately allocated block of memory. (Optional)

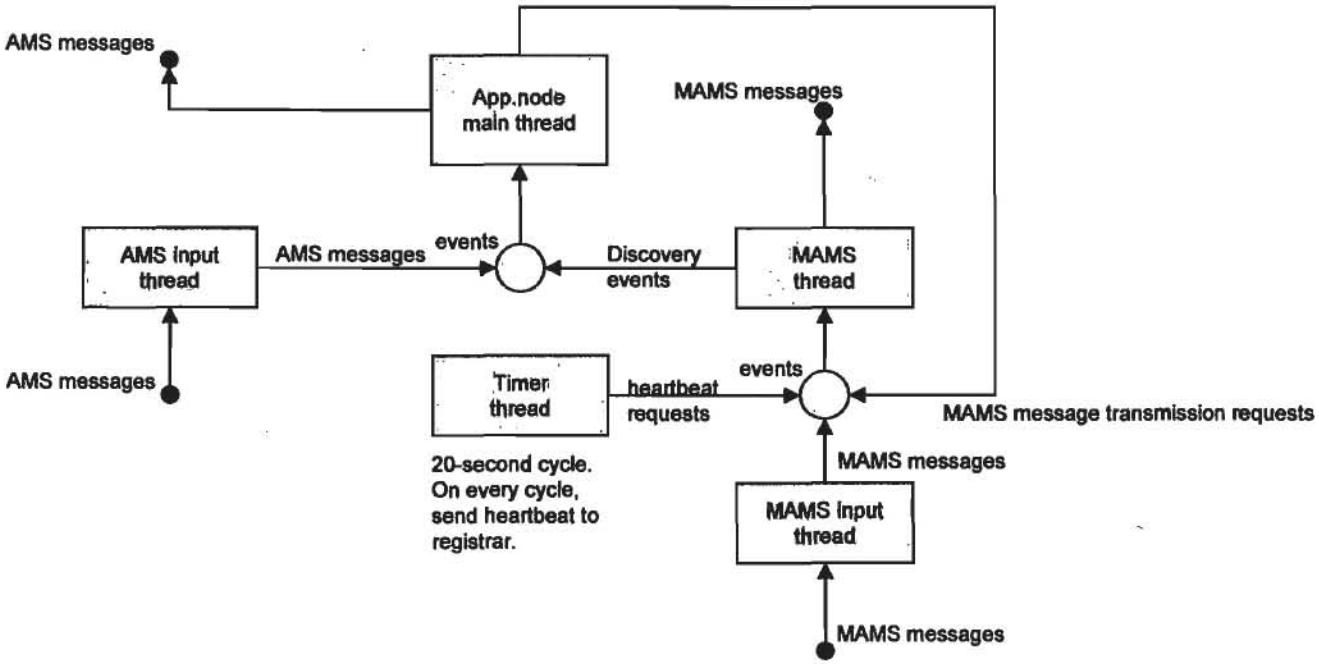
# Reference Implementation (3)

- An AMS application program, linked with **libams**, uses the *ams_register* function to instantiate an AMS node registered within a specified unit of a specified message space.
- Once registration is accomplished, the application may commence *inviting*, *subscribing* to, *publishing*, *announcing*, *sending*, and *replying* to messages.

# Reference Implementation (4)

- The implementation is multi-threaded.
  - Registration causes two POSIX threads (pthreads) to be started, to handle timing and MAMS events in the background.
  - One more pthread is started to handle MAMS messages.
    - Receives MAMS messages via the primary transport service.
    - Inserts them into the MAMS event queue (merging them with MAMS message transmission requests).
  - One additional pthread is started for each transport service on which the node is able to receive AMS messages.
    - Receives AMS messages via this transport service.
    - Inserts them into the AMS event queue (merging them with "discovery" events inserted by the MAMS event handling thread).

# Elements of a single AMS node



# Report of AMS WG Meetings

16-20 January 2007

# AMS Working Group Meetings

- Met with MOIMS/SM&C WG to discuss use of AMS for SM&C.
  - Reviewed RAMS "tree" option, for enhanced scalability.
  - Agreed to identify two "service classes" of AMS.
    - Class 1: Implementation of MAMS and AMS protocols only.
    - Class 2: Implementation of RAMS as well as MAMS and AMS protocols.
  - Agreed to add illustrative sequence diagrams to section 4 of the AMS specification and to document the "wire" representation of data items marshaled by AMS.
  - Agreed to use AMS as the standard end-to-end underlying messaging service for SMCP.

# AMS Working Group Meetings (2)

- Met with SOIS/TCOAS WG members to discuss suitability of AMS for real-time on-board messaging.
  - Reviewed results of real-time AMS message exchange testing on VxWorks platform at JPL.
  - Reviewed use of AMS in wireless avionics testbed at JPL.
  - Reviewed prospects for real-time message exchange over on-board ring bus.
  - Reviewed use cases and message transfer requirements developed by SOIS since June 2006 meetings.
  - Agreed that AMS is suitable for real-time on-board message exchange as characterized by the SOIS Message Transfer Service specification.

# AMS Working Group Meetings (3)

- Discussed plans for developing second implementation of AMS.
  - NASA/GSFC will begin this development in February of 2007.
- Briefly discussed protocol enhancements as reflected in first edition of AMS Red Book.
  - No problems identified.
- Plan for between now and next working group meeting:
  - Complete second implementation of AMS (service class 2).
  - Begin interoperability testing between the two implementations of AMS.
  - Publish second edition of AMS Red Book reflecting changes agreed on at WG meetings and modifications emerging from interoperability testing.

# End of File