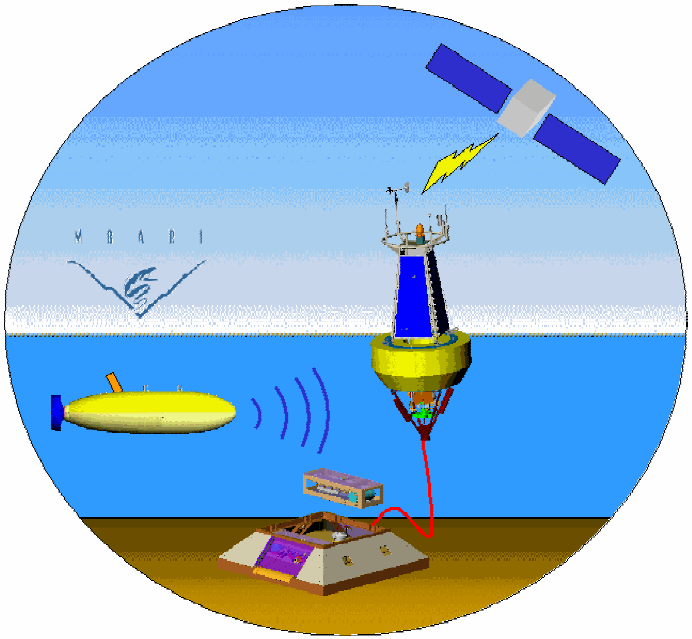
Power Management



Mike Risi

M B A R I

Primary requirement sources

- *MOOS Deployed Subsystem Software Requirements Specification:*
  - 3.2.3, 3.2.13, 3.2.13.1, 3.2.13.2

- *Power Management Software for MOOS Functional Requirements*

Power Management

- Requirements
  - definitions
    - mechanism and policy
    - system and application
- Implementation
  - Phase 1, *Static Power Management*
  - Phase 2, *Power Prediction and Adaptive Power Management*

Power Management Requirements

- Requirements definitions
  - Mechanism - What needs to be done
  - Policy - How that capability is used
  - System - what needs to be done at the Linux kernel/driver level
  - Application - what needs to be done by the SIAM application

Power Management Mechanism

- System
  - running
  - idle
  - sleeping
- Application
  - switching between running and sleeping mode
  - enable and disable instrument power

Power Management Policy

- System
  - How does Linux handle services (e.g. telnetd, ftpd, etc.)

Power Management Policy

- Application - Static Power Management
  - support instrument POWER_ALWAYS, POWER_WHEN_SAMPLING, POWER_NEVER policies
  - enter and exit sleep mode at appropriate time

Power Management Policy

- Application - Power Predication and Adaptive Management
  - interfaces to power subsystems
  - prioritization of energy usage
  - event response power allocation

Power Management - Phase 1

- Implementation of *Static Power Management*
  - System policies
  - SleepManager
  - RMI protection mechanisms
  - InstrumentService
  - PowerManager
  - CommsLeaseListener

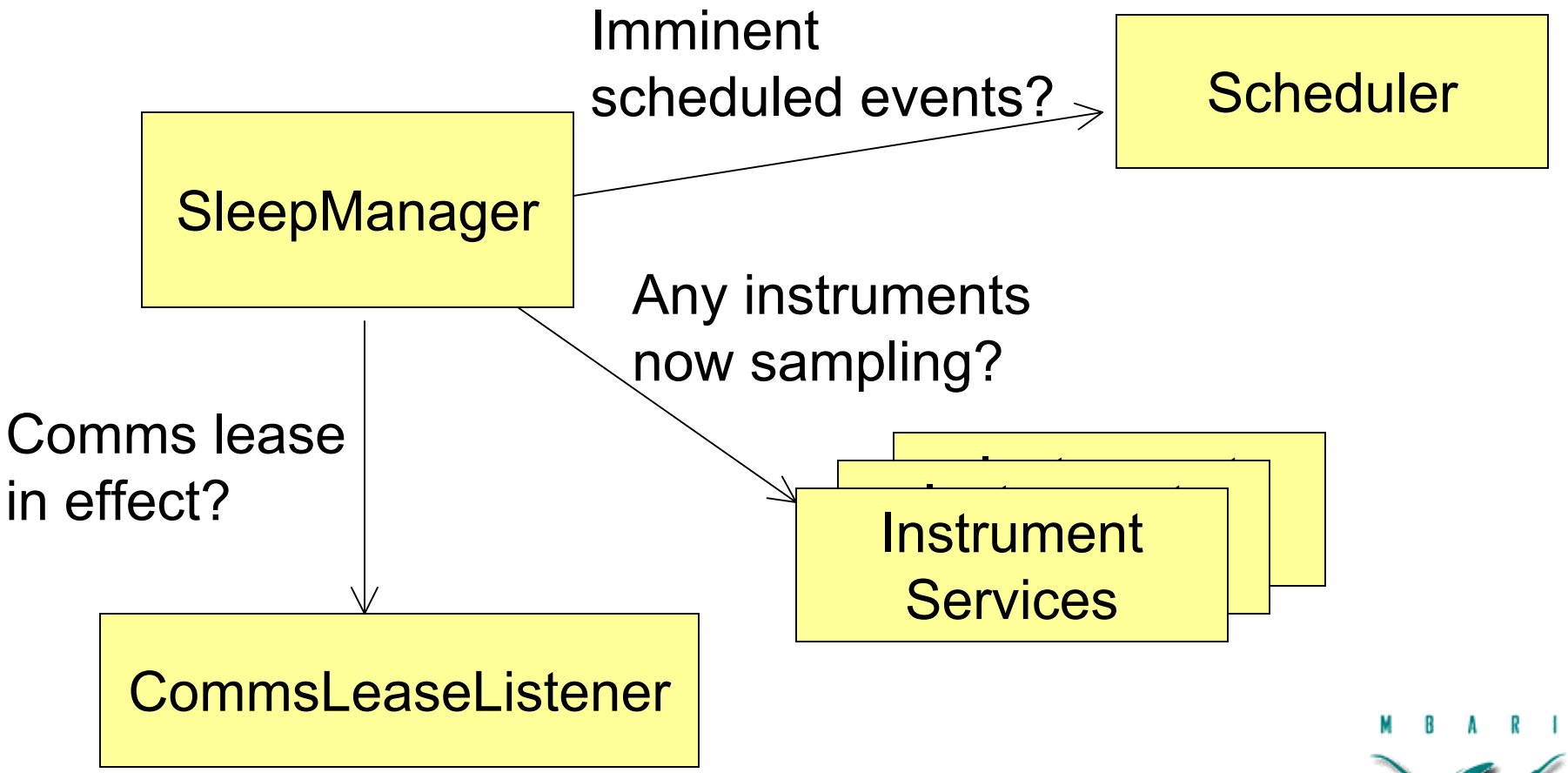
Power Management - Phase 1

- System Policies
  - All normal operations use RMI
  - Linux facilities ftp, ssh, telnet, etc. are not used during normal operation

Power Management - Phase 1

- SleepManager queries subsystems to determine if it is appropriate to go to sleep
  - Scheduler
  - CommsLeaseListener
  - InstrumentService
  - PowerManagedSockets

Power Management - Phase 1



Power Management - Phase 1

- RMI protection mechanisms
  - PowerManagedRMISocketFactory
  - Leasing

Power Management - Phase 1

- InstrumentService
  - instrument power modes
    - POWER_ALWAYS
    - POWER_WHEN_SAMPLING
    - POWER_NEVER
  - InstrumentService status indicates whether it's safe to sleep

Power Management - Phase 1

- PowerManager
  - monitors node power usage
  - grants request to InstrumentService for power
- CommsLeaseListener
  - power manages communication resources

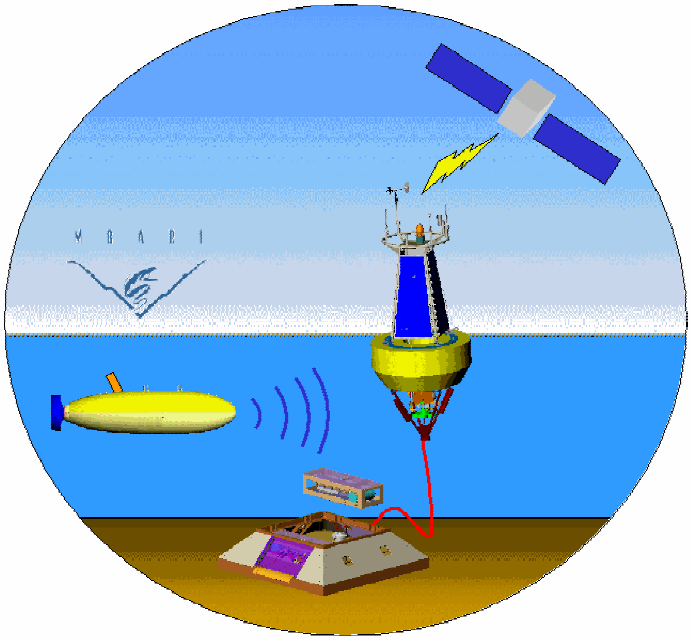
Power Management - Phase 1

- Issues
  - httpd is necessary for RMI codebase
  - Power managed RMI sockets versus leasing
  - MSP430 sleep/wakeup interface
  - ethernet interface
  - spi driver

Power Management - Phase 2

- Implementation of power predication and adaptive management
  - power subsystems
  - prioritization of energy usage
  - event response power allocation

Event Propagation and Response



Mike Risi

M B A R I

Primary requirement sources

- *MOOS Deployed Subsystem Software Requirements Specification:*
  - 3.2.5.4, 3.2.10, 3.2.11, 3.2.12.2, 3.2.17.1, 3.3.5

- *SIAM User Story:*
  - 8

Event propagation

- Definition of events
- Distributed events versus local events
- Events embedded in device data streams
- RMI callbacks
- Jini distributed events

What is an event?

- Event is notification of state change

- Types of events
  - asynchronous events
  - autonomous event response
  - event logs

Distributed versus local events

- Distributed events timely delivery not guaranteed
- Distributed events may arrive out of order
- Distributed events may be lost due network failure

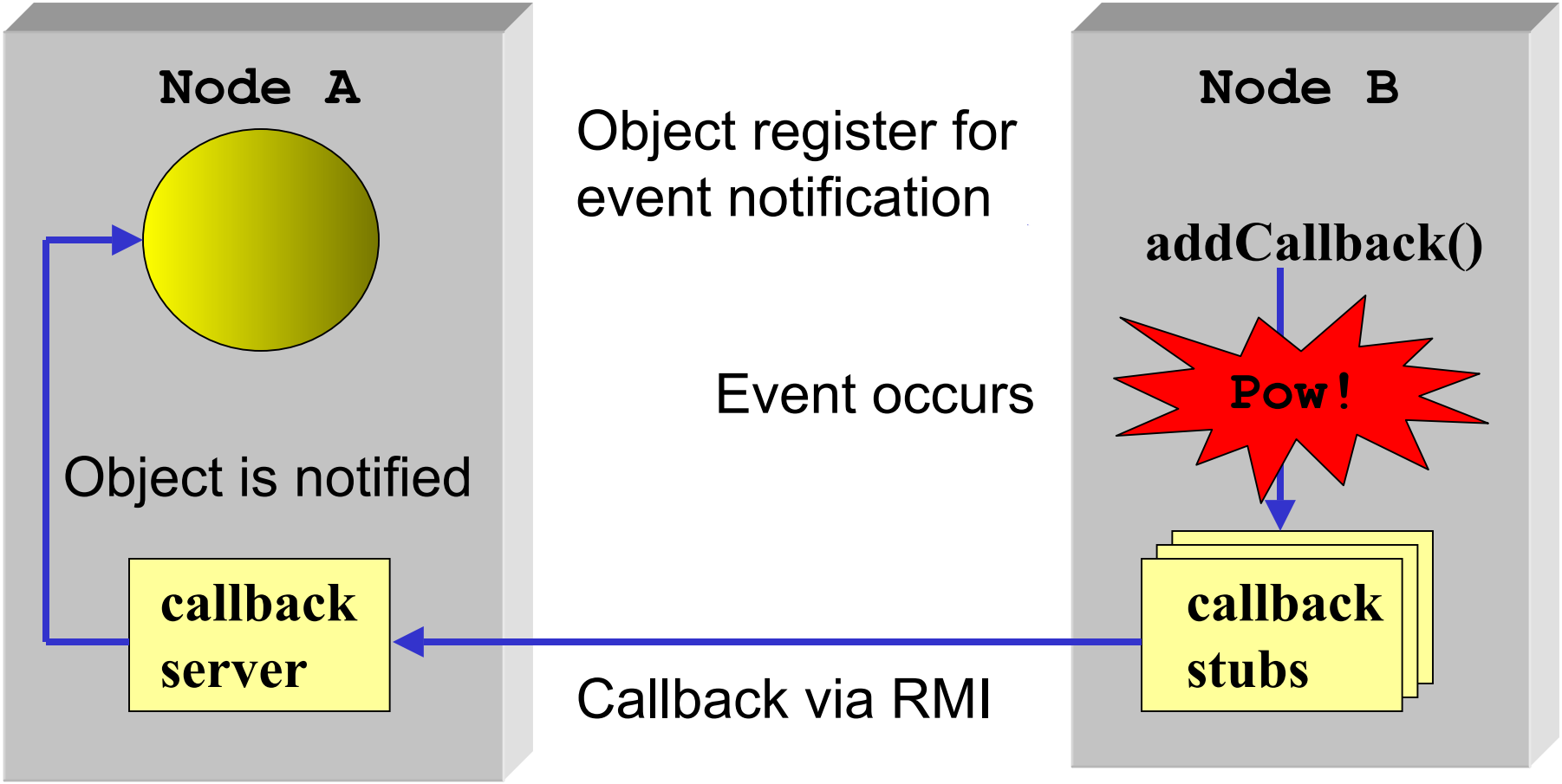
Events embedded in device data streams

- Metadata packets
  - instruments service starting
  - user changes service property
  - service may autonomously generate metadata
- Message packets
  - port annotation
  - certain types of errors

RMI Callbacks

- RMI callbacks can be used for distributed events
- Nodes already use RMI as standard interface
- Prototyped for GUI notification of instrument state change

RMI Callbacks



RMI Callback Issues

- Subscribed node is sleeping during event
- Network failure occurs during event propagation
- May consume large amounts of bandwidth

Jini Distributed Event Model

- RemoteEventListener uses RMI
- RemoteEvent is small
  - event ID
  - sequence number
  - hand-back object, supplied by consumer may be null
  - reference to event producer

Jini Distributed Event Model

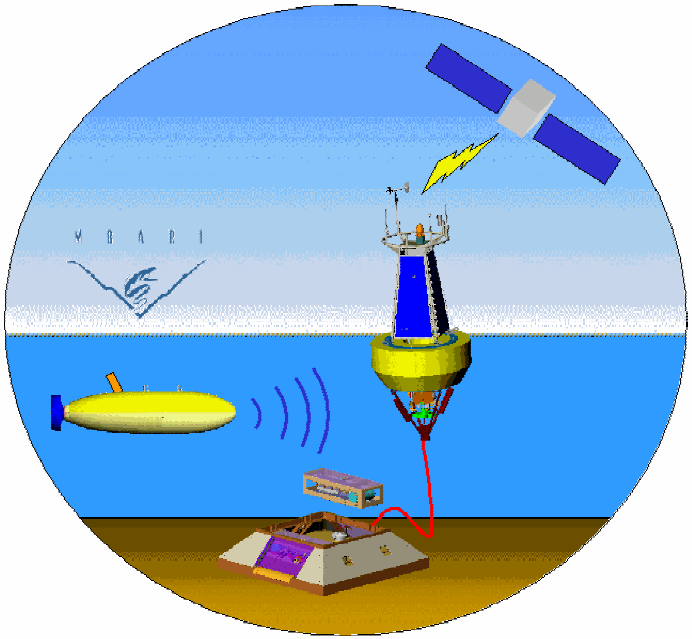
- Third party *registrant*
  - may reside on event consumer, producer, or some other node
  - filters events
  - stores events and forward events
    - addresses network failures
    - addresses sleeping nodes

Jini Distributed Event Model

- While all of Jini may not be appropriate some concepts may be useful

- May be able to use net.jini.core.event.* package unmodified

Multi-node Moored Networks

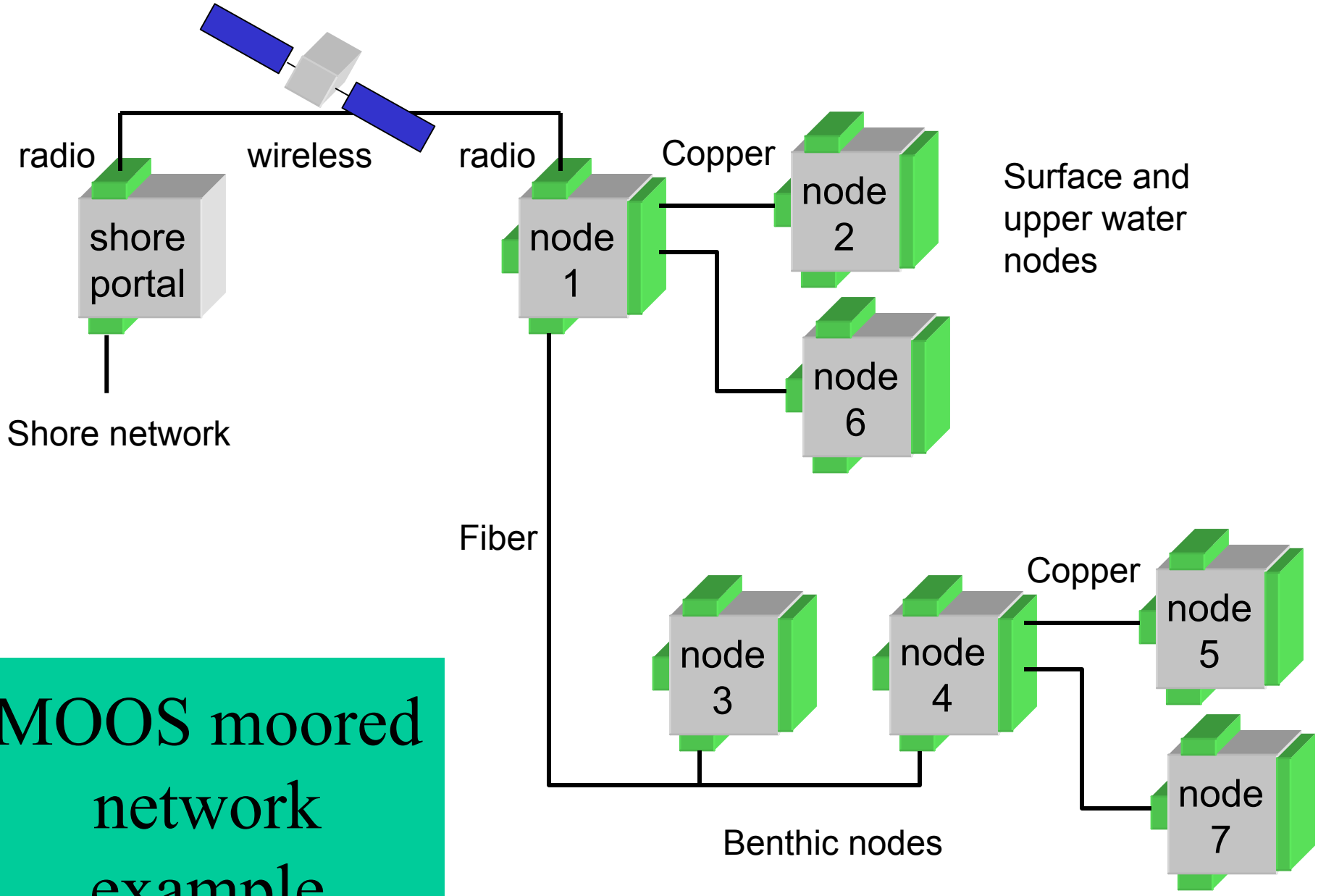


Tom O'Reilly

M B A R I

Primary requirement sources

- *MOOS Deployed Subsystem Software Requirements Specification*
  - 3.2.14, 3.2.19

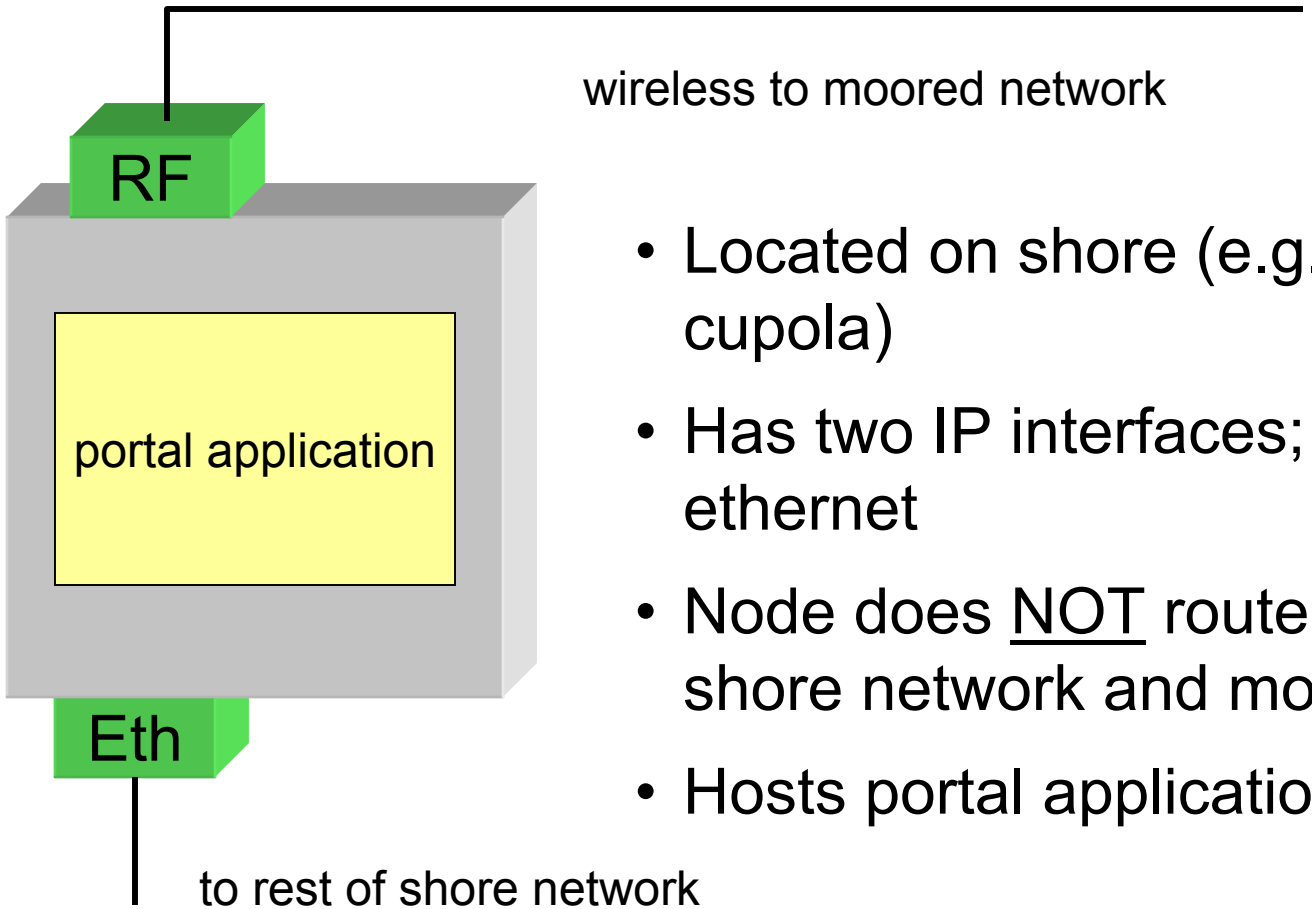


MOOS moored network example

MOOS portal

- Shore-based store-and-forward service over low-bandwidth wireless link
- When RF link is up, forwards commands to moored network, and retrieves data from network
- In general, all traffic between mooring and shore goes through portal

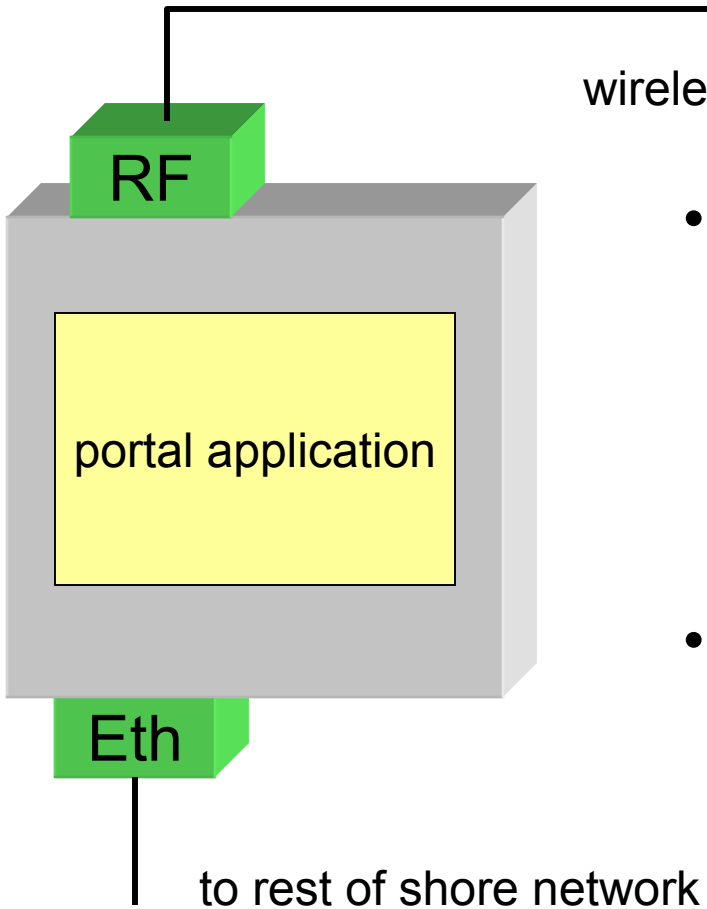
Portal node



- Located on shore (e.g. in MBARI cupola)
- Has two IP interfaces; RF/ppp and ethernet
- Node does NOT route between shore network and moored network
- Hosts portal application

MBARI

Portal application

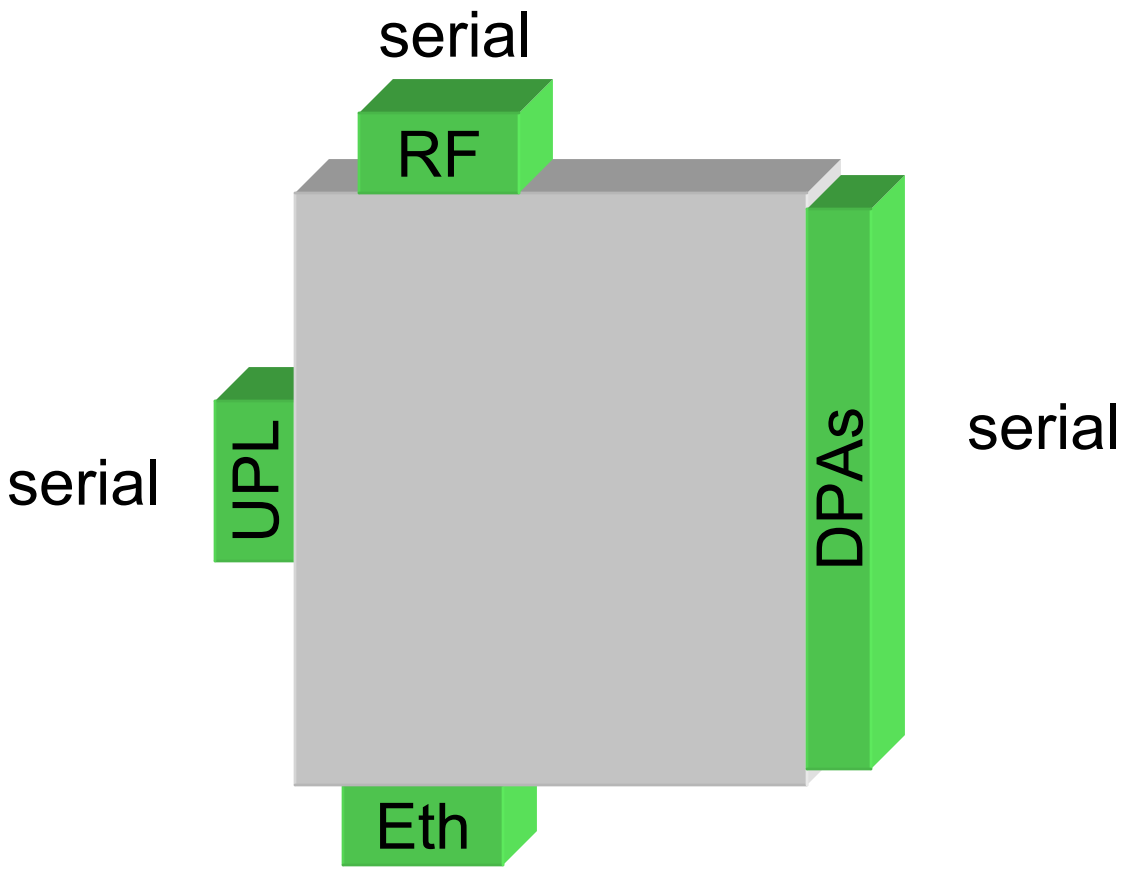


- When wireless link is up, sends commands to mooring nodes, retrieves telemetry from mooring nodes via Java *Remote Method Invocation (RMI)*
- Forwards telemetry to *Shore Side Data System (SSDS)* via *Java Messaging Service (JMS)*

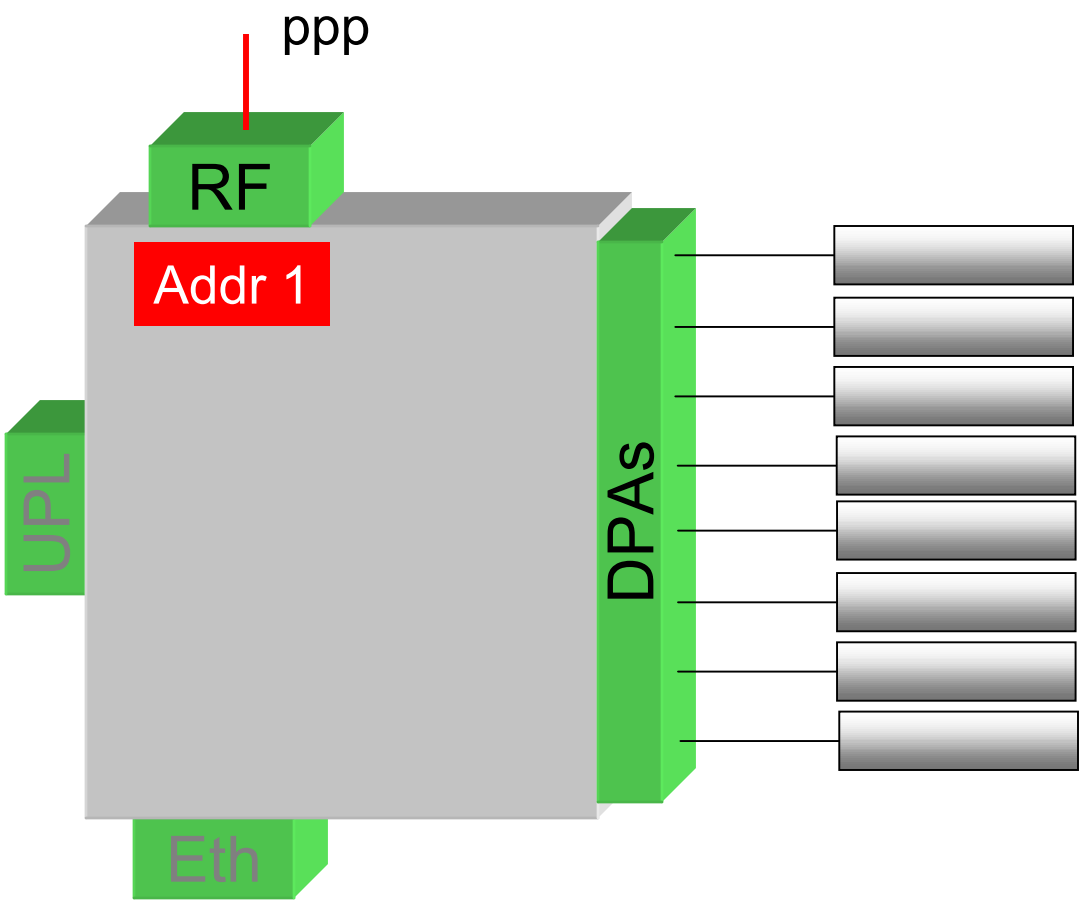
MMC power management

- Mooring power is limited resource
- MMC alternates between *running* and *sleep* states
- Goes to sleep when nothing to do for at least 60 seconds
- Awakened by serial or ethernet traffic, environmental alarm, or at scheduled acquisition time

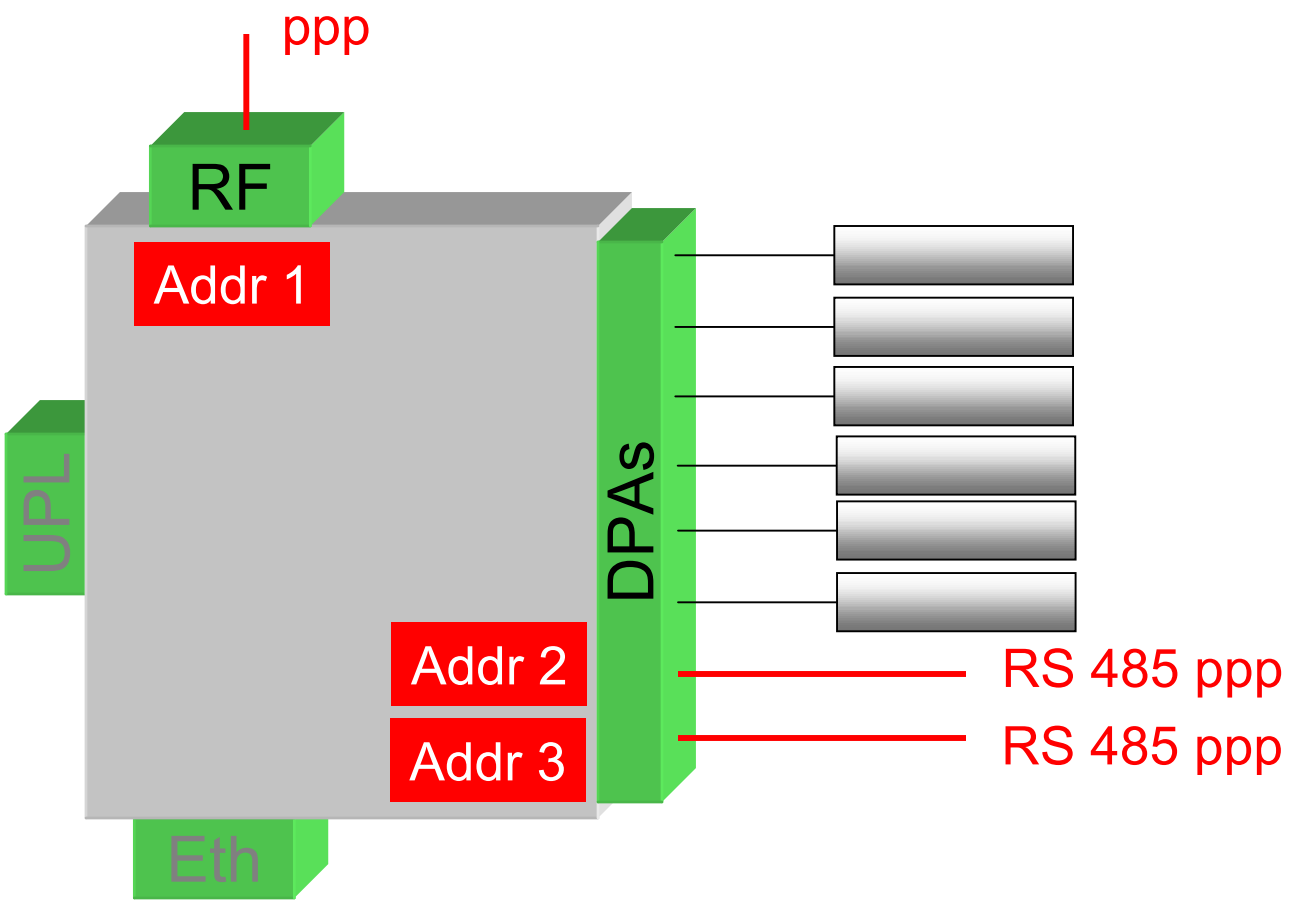
MMC network interfaces



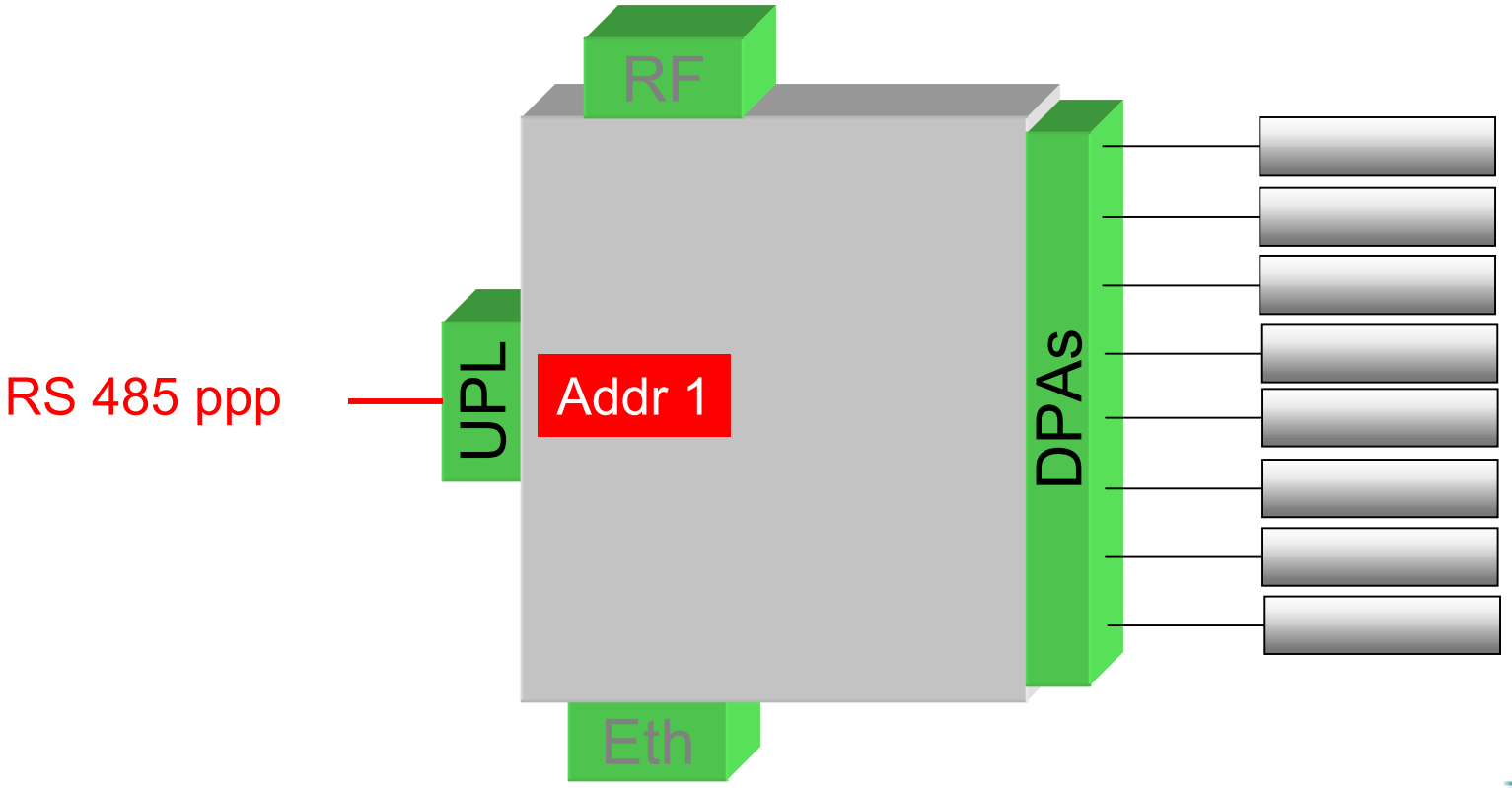
MOOS Lite, single-node



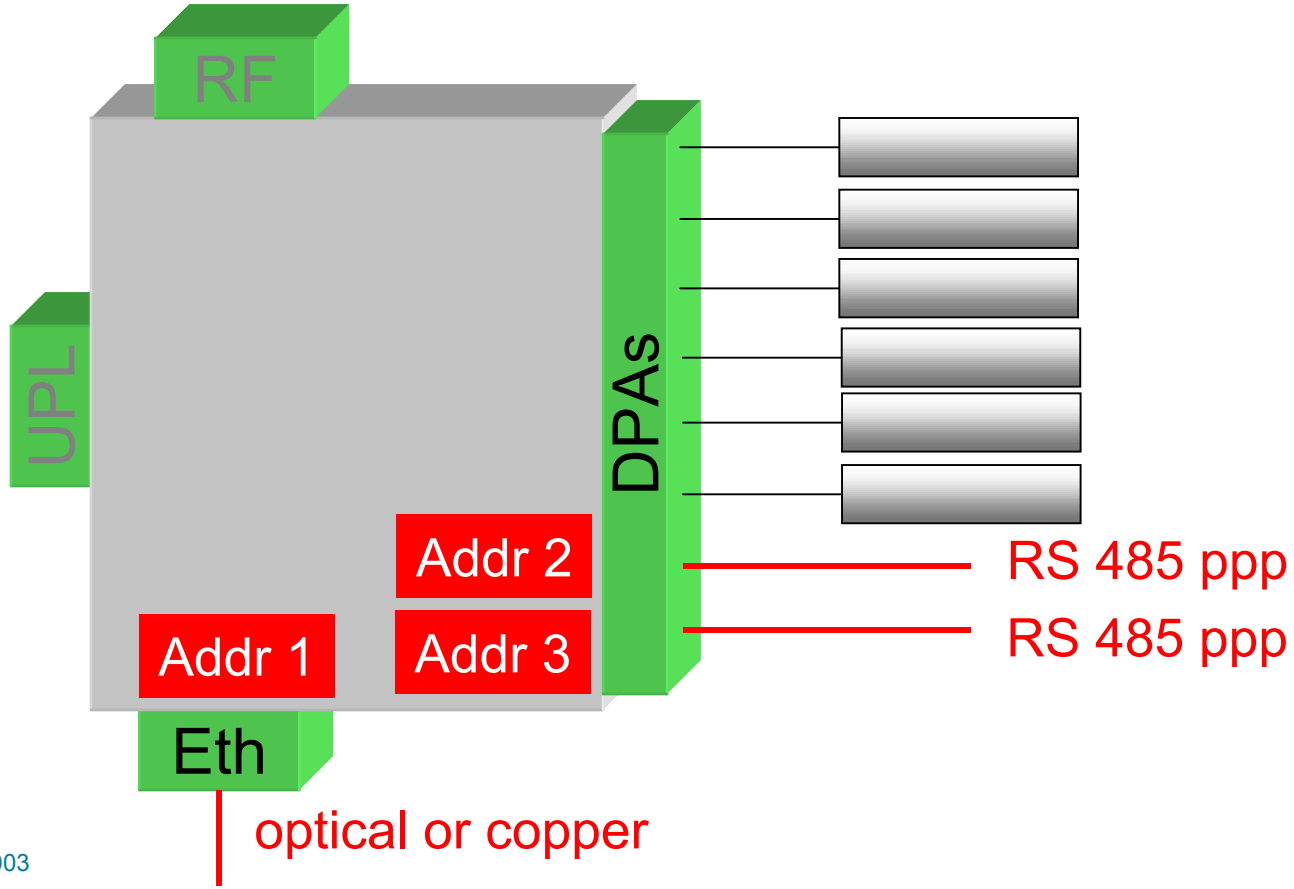
MOOS Lite, surface node



MOOS RS-485 node

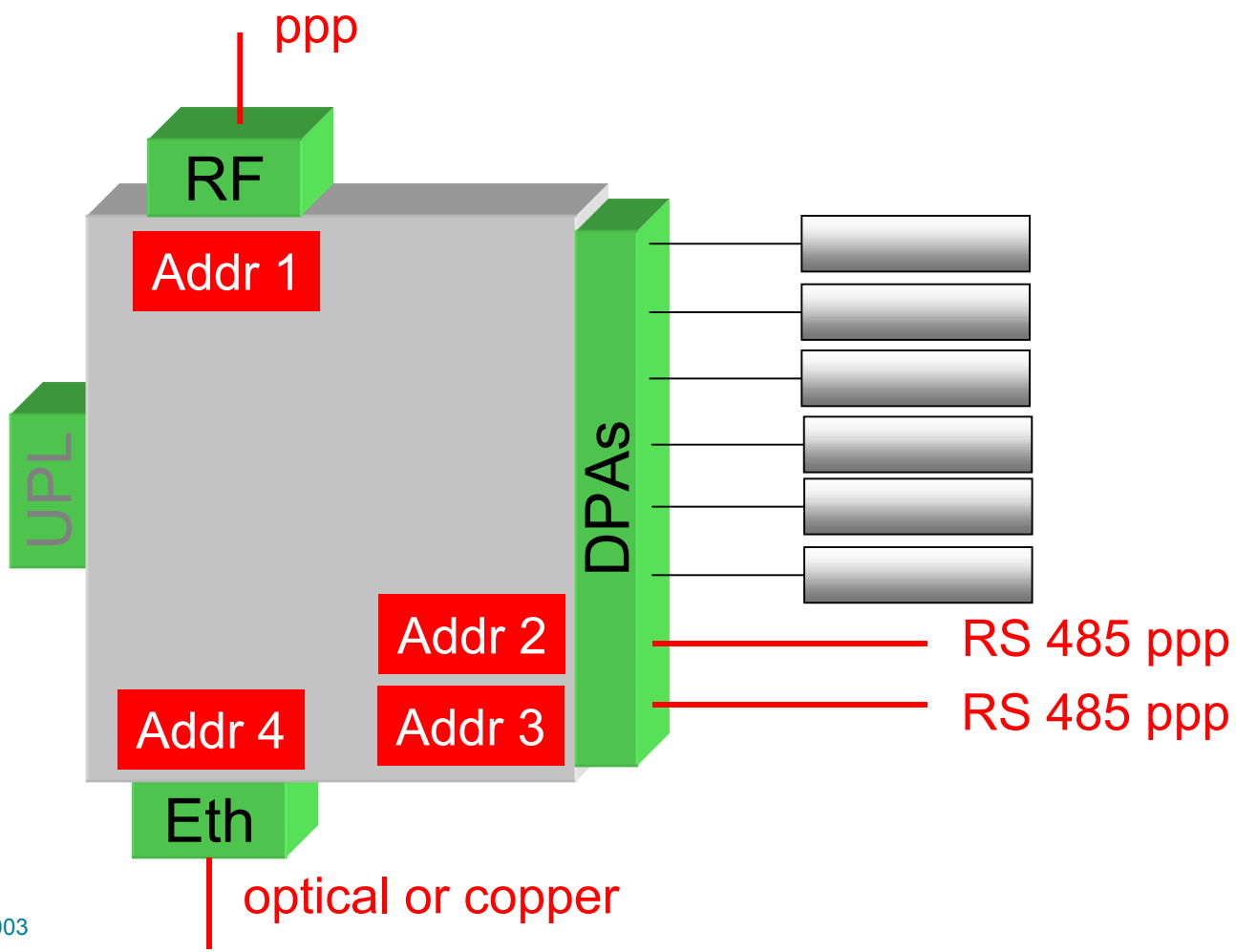


MOOS benthic node



MBARI

MOOS surface node

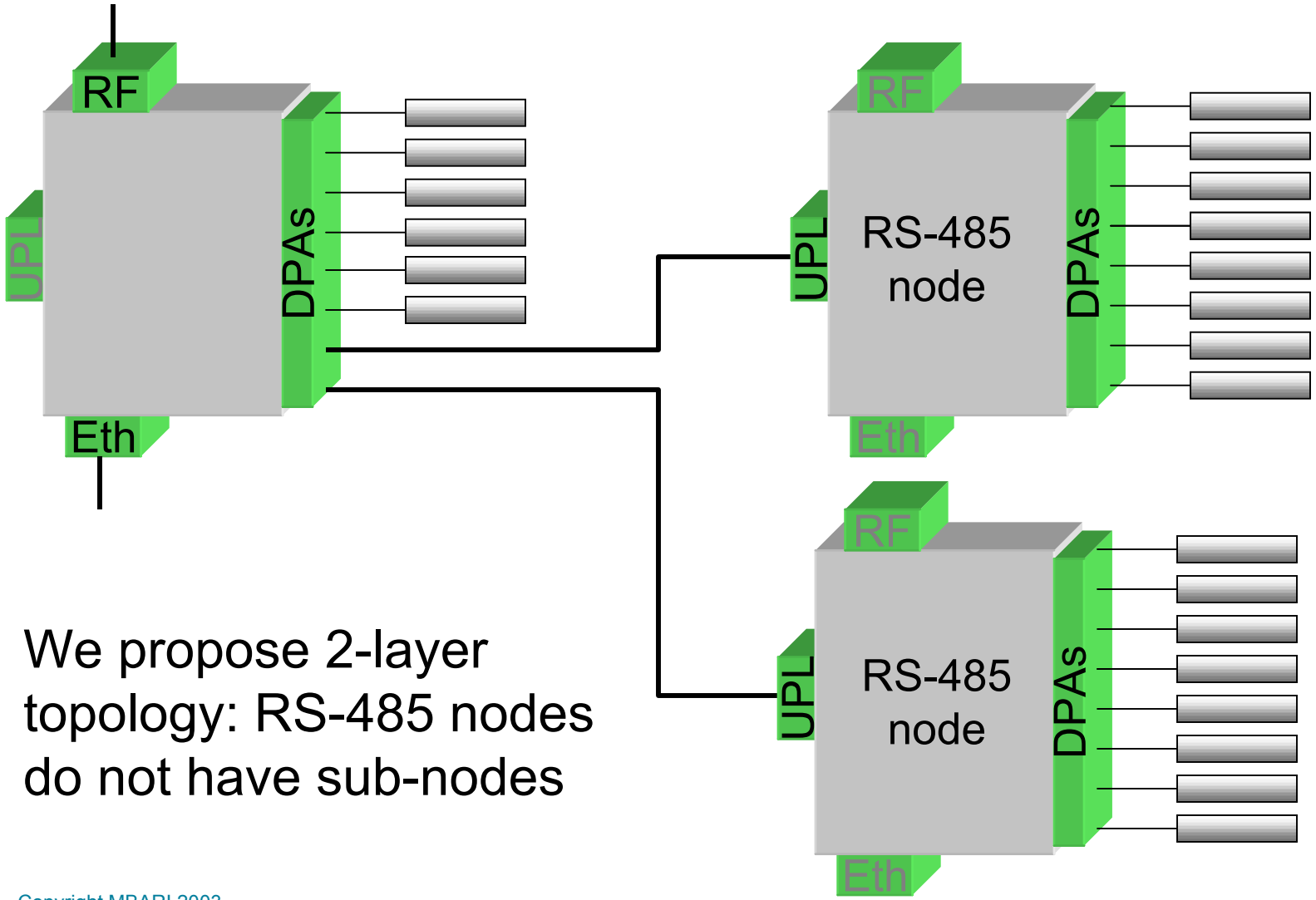


MBARI

Multi-homed nodes

- Some nodes have multiple IP addresses
- Implications for
  - Network configuration and setup, routing
  - RMI configuration

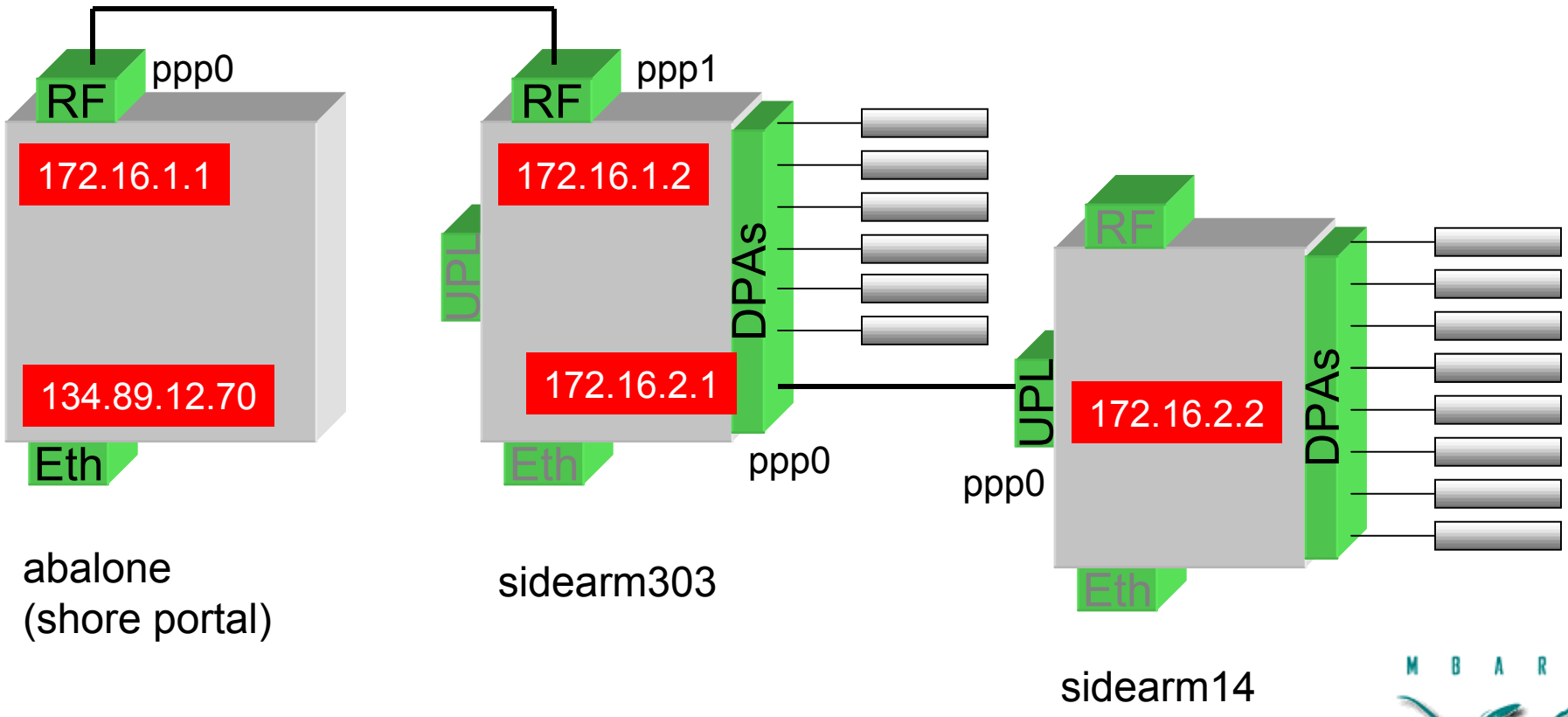
RS-485 node topology

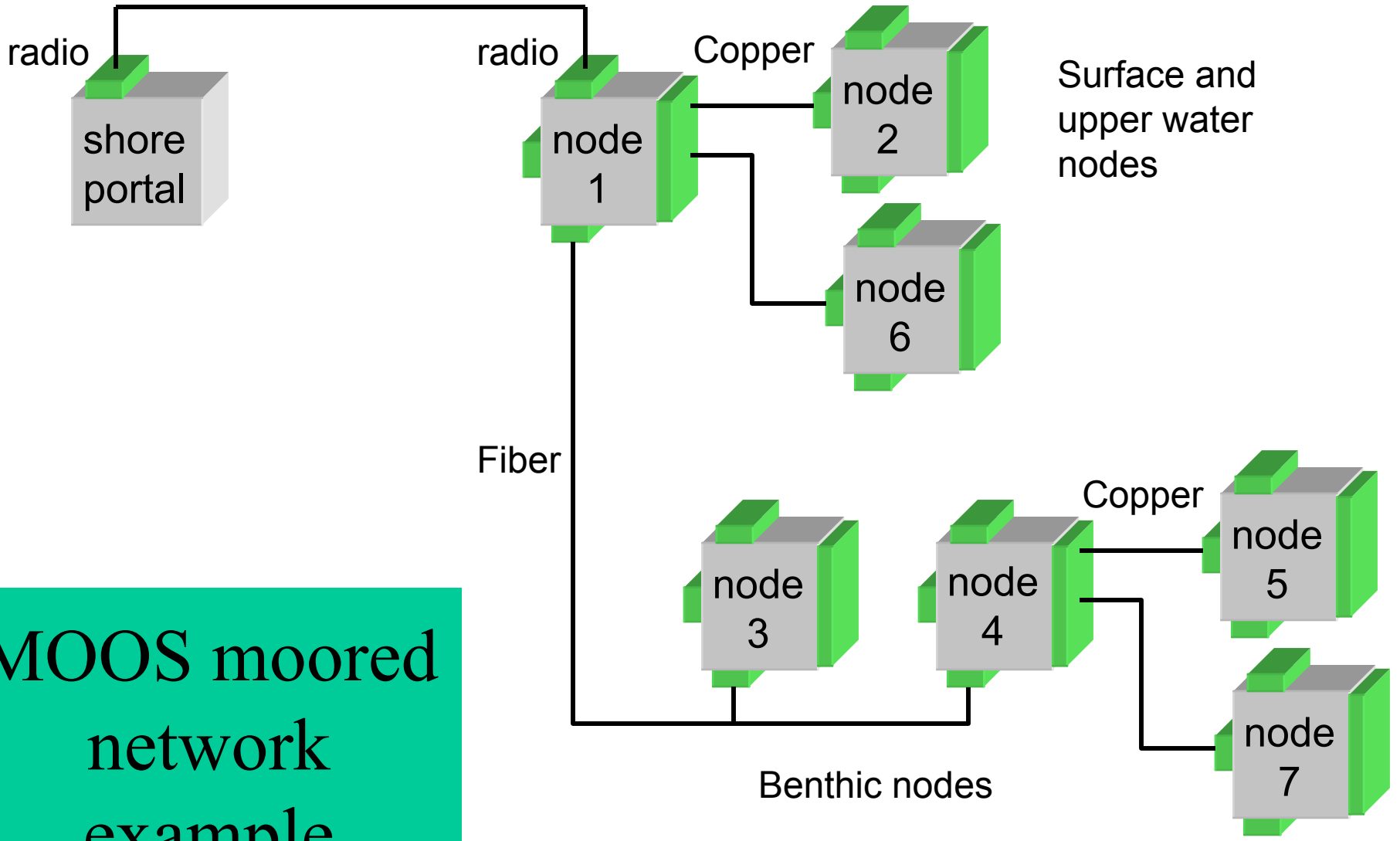


- We propose 2-layer topology: RS-485 nodes do not have sub-nodes

MBARI

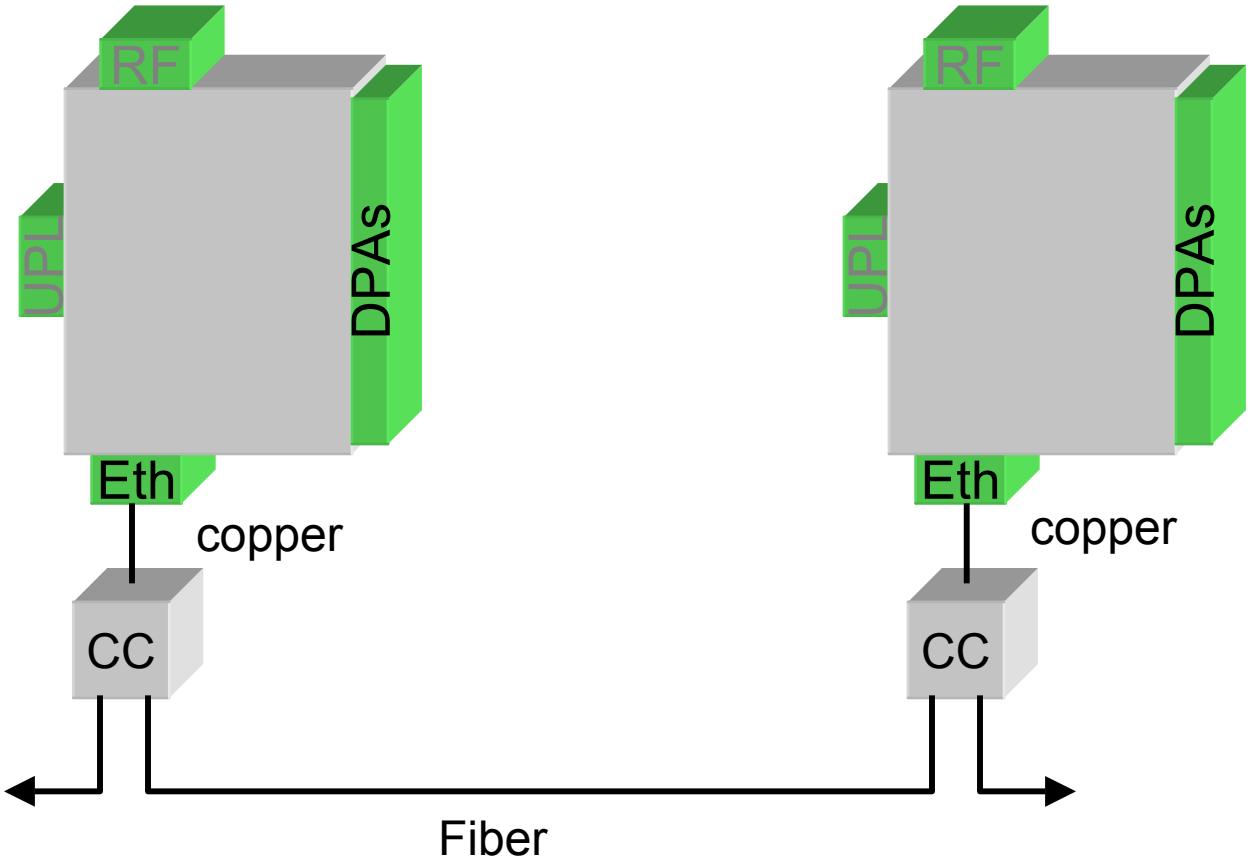
RS-485 node example





MOOS moored network example

Communication controller



Communication controller

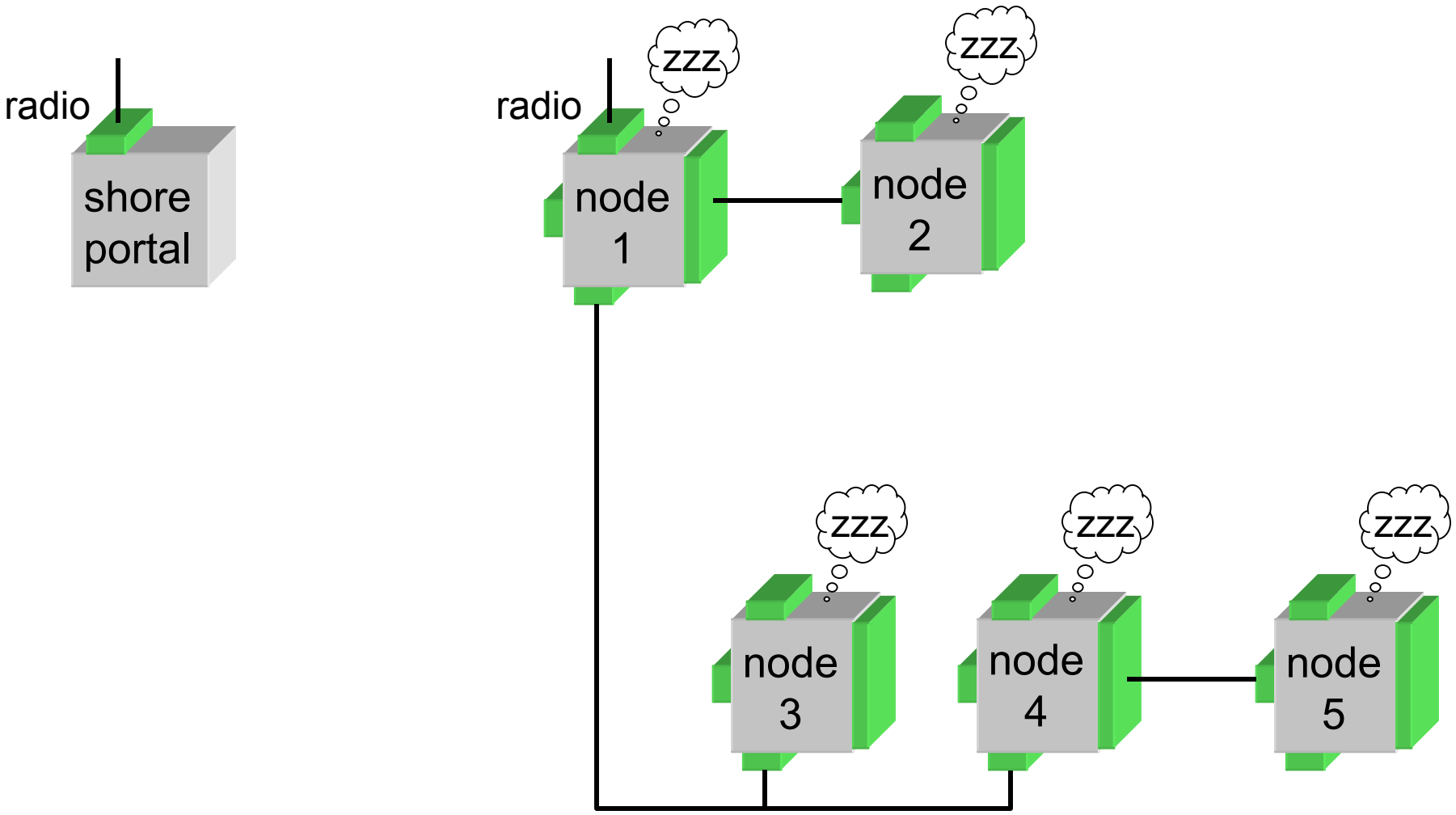
- Reception of multicast or broadcast packet always wakes up SideARM
- Reception of unicast packet wakes up destination SideARM

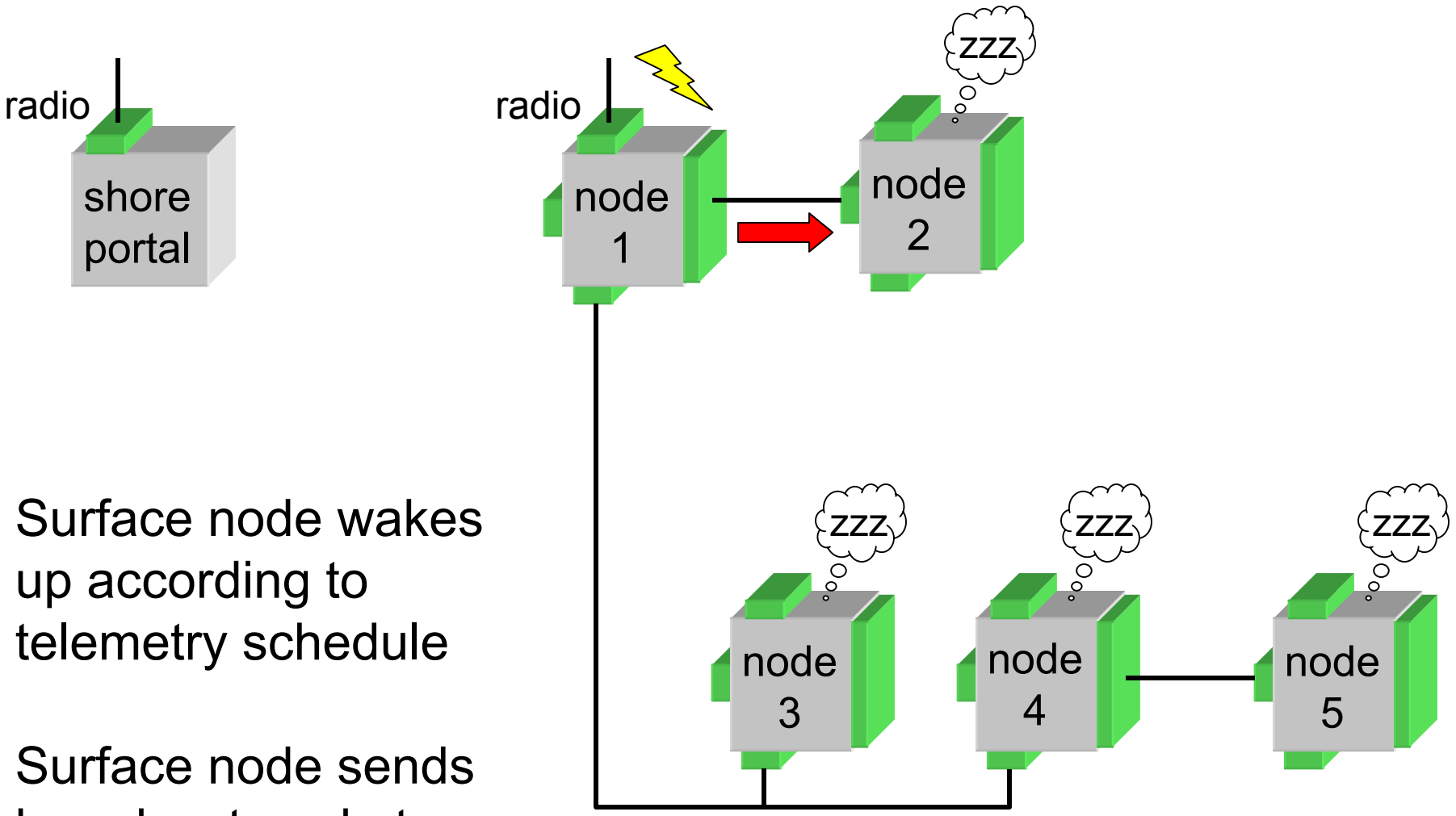
Zeroconf

- Zero-configuration networking
  - Apple has *Rendezvous* implementation
  - Utilizes existing DNS protocols
- IP address assignment, name-to-address translation
  - Applicable to link-local addresses only: no inter-network communications
- Service discovery

Telemetry management

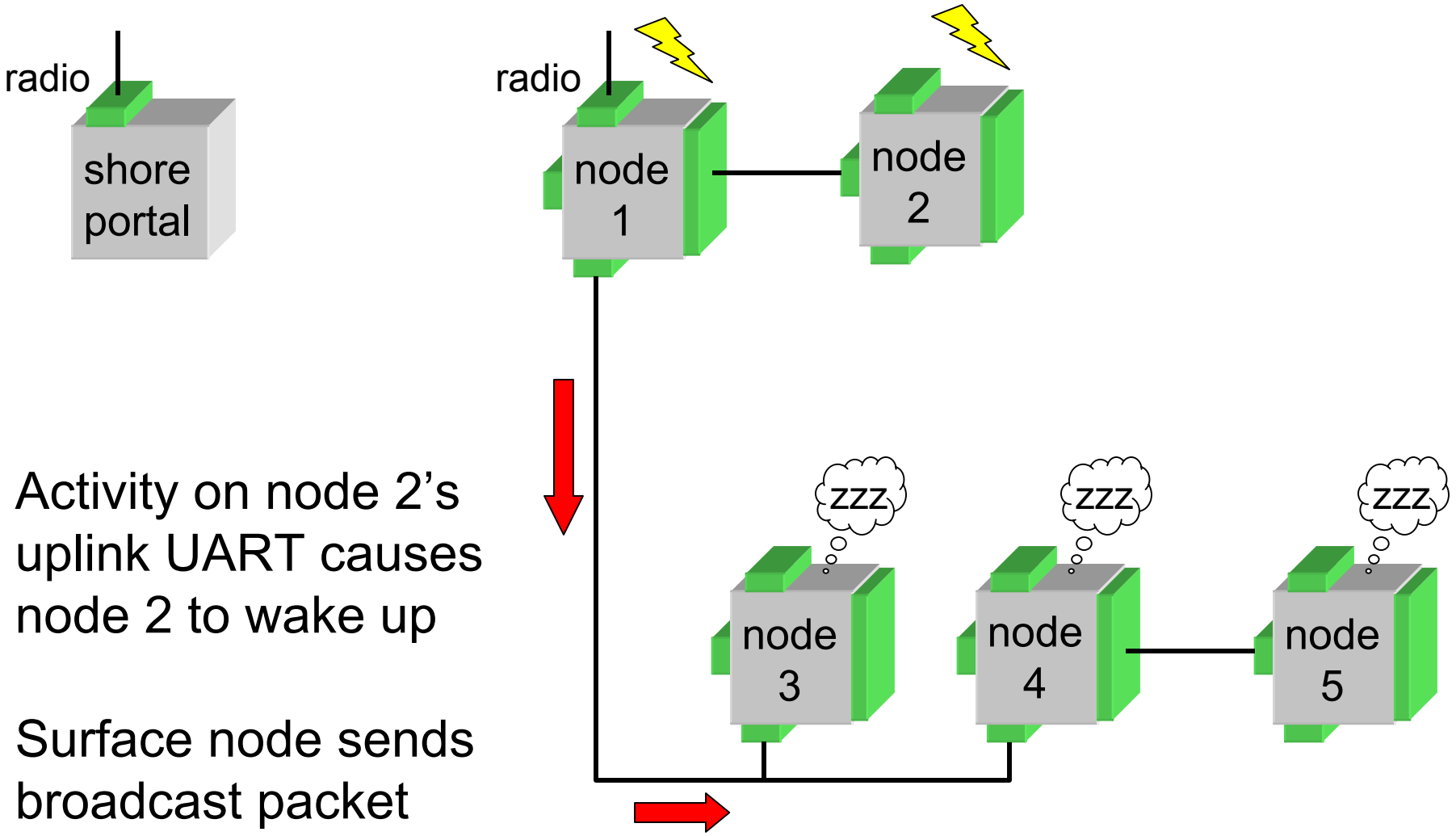
- Portal needn't have *a priori* knowledge of:
  - Moored network node addresses
  - Node power or communications schedules
  - Instruments present on each node



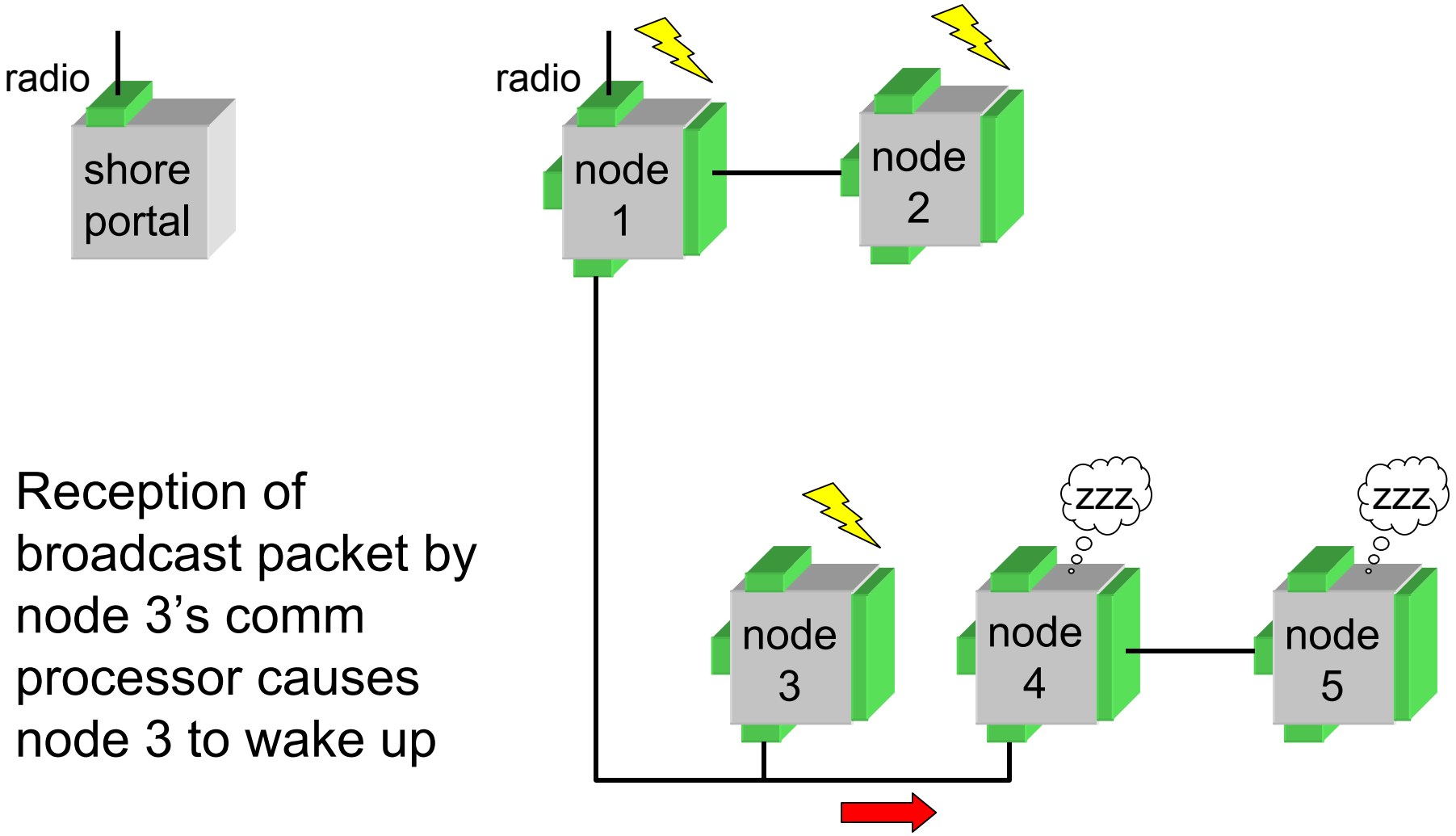


- Surface node wakes up according to telemetry schedule
- Surface node sends broadcast packet through each DPA network interface

MBARI

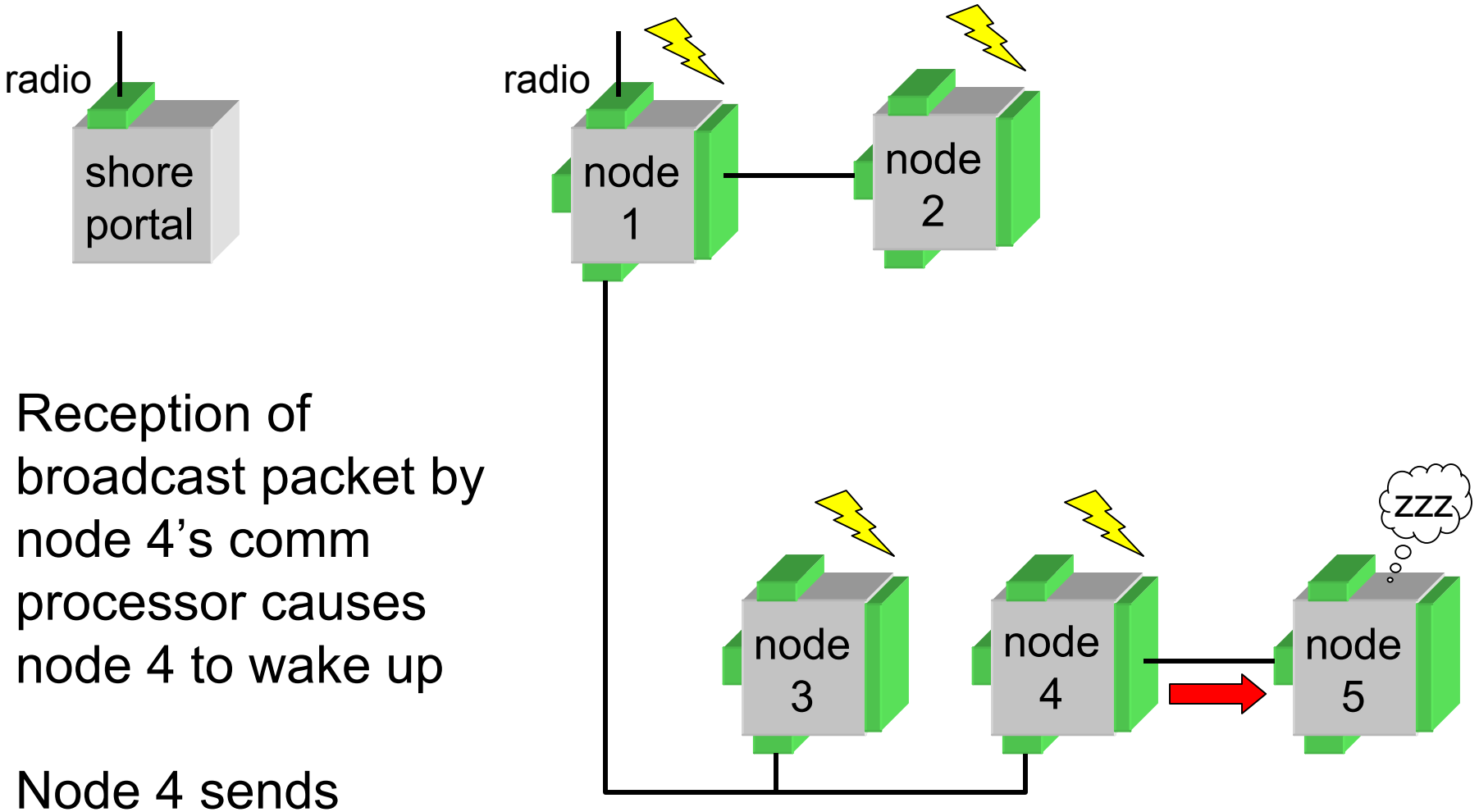


- Activity on node 2's uplink UART causes node 2 to wake up
- Surface node sends broadcast packet through Ethernet interface

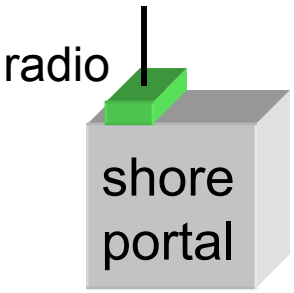


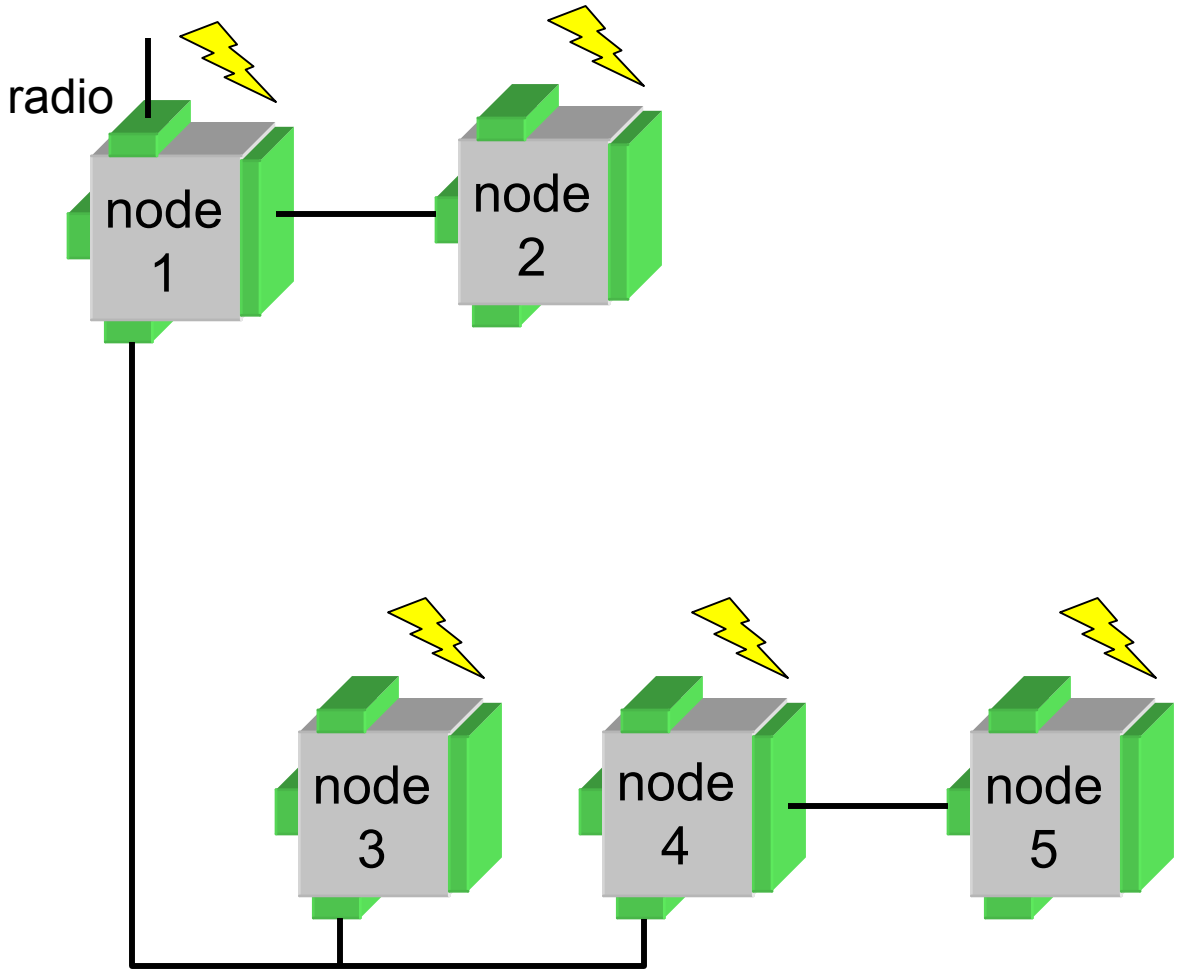
- Reception of broadcast packet by node 3's comm processor causes node 3 to wake up
- Node 3's comm processor forwards broadcast packet to node 4

MBARI



- Reception of broadcast packet by node 4's comm processor causes node 4 to wake up
- Node 4 sends broadcast packet through each DPA network interface



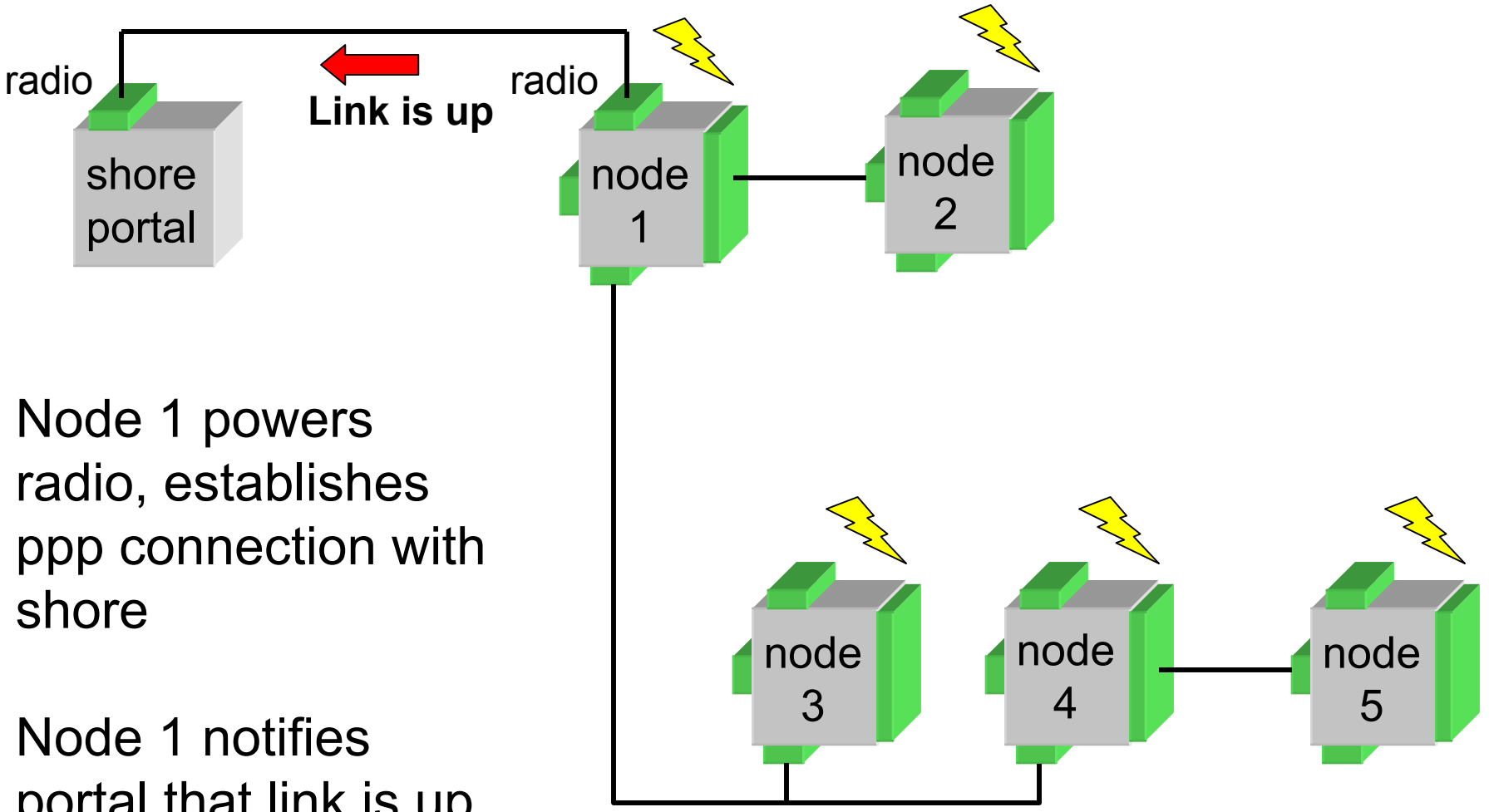


- All nodes now running

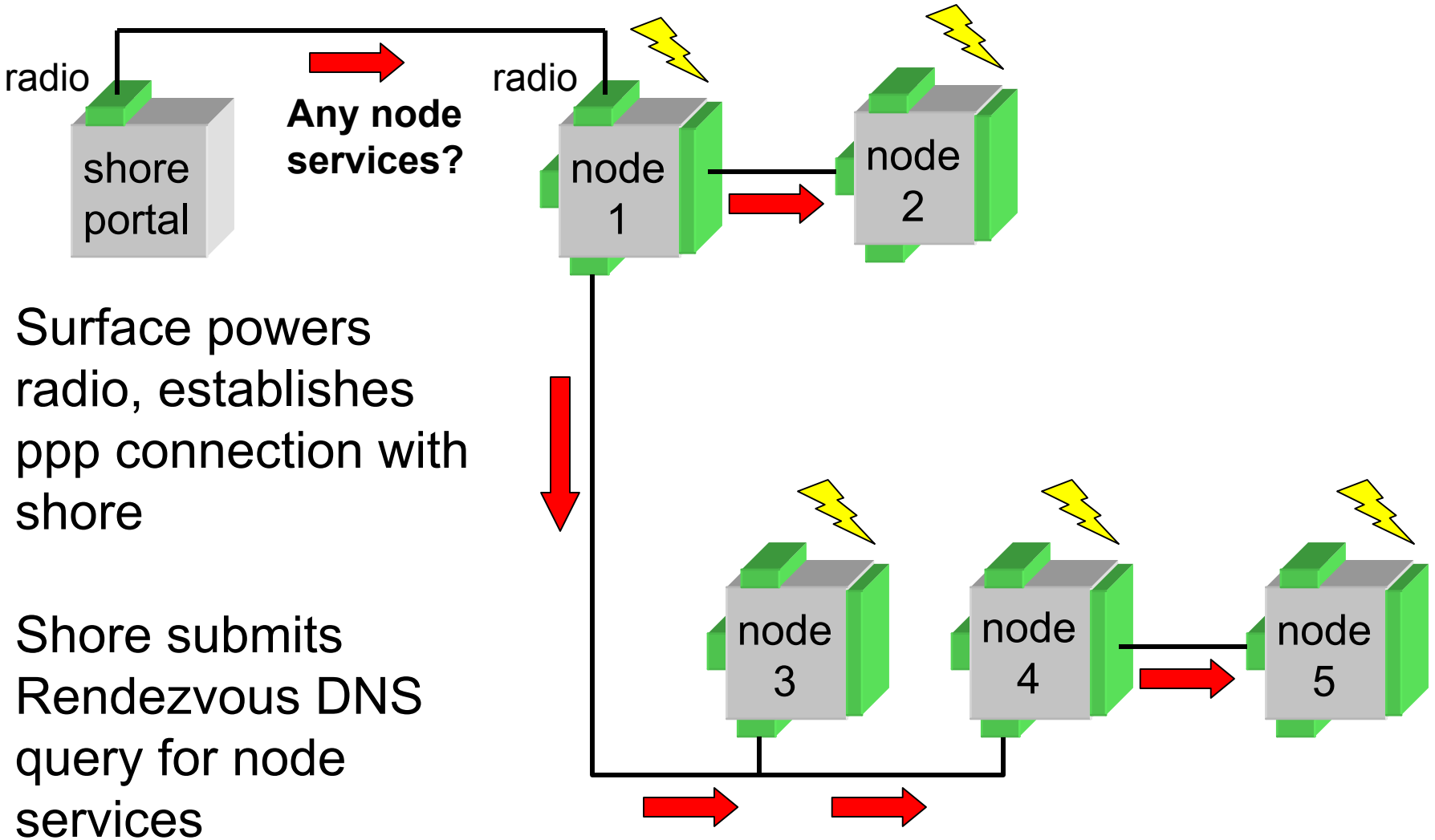
MBARI

This strategy...

- Nodes don't need to know IP addresses or power schedules of other nodes
- But all nodes get powered on at once for node service discovery and telemetry download

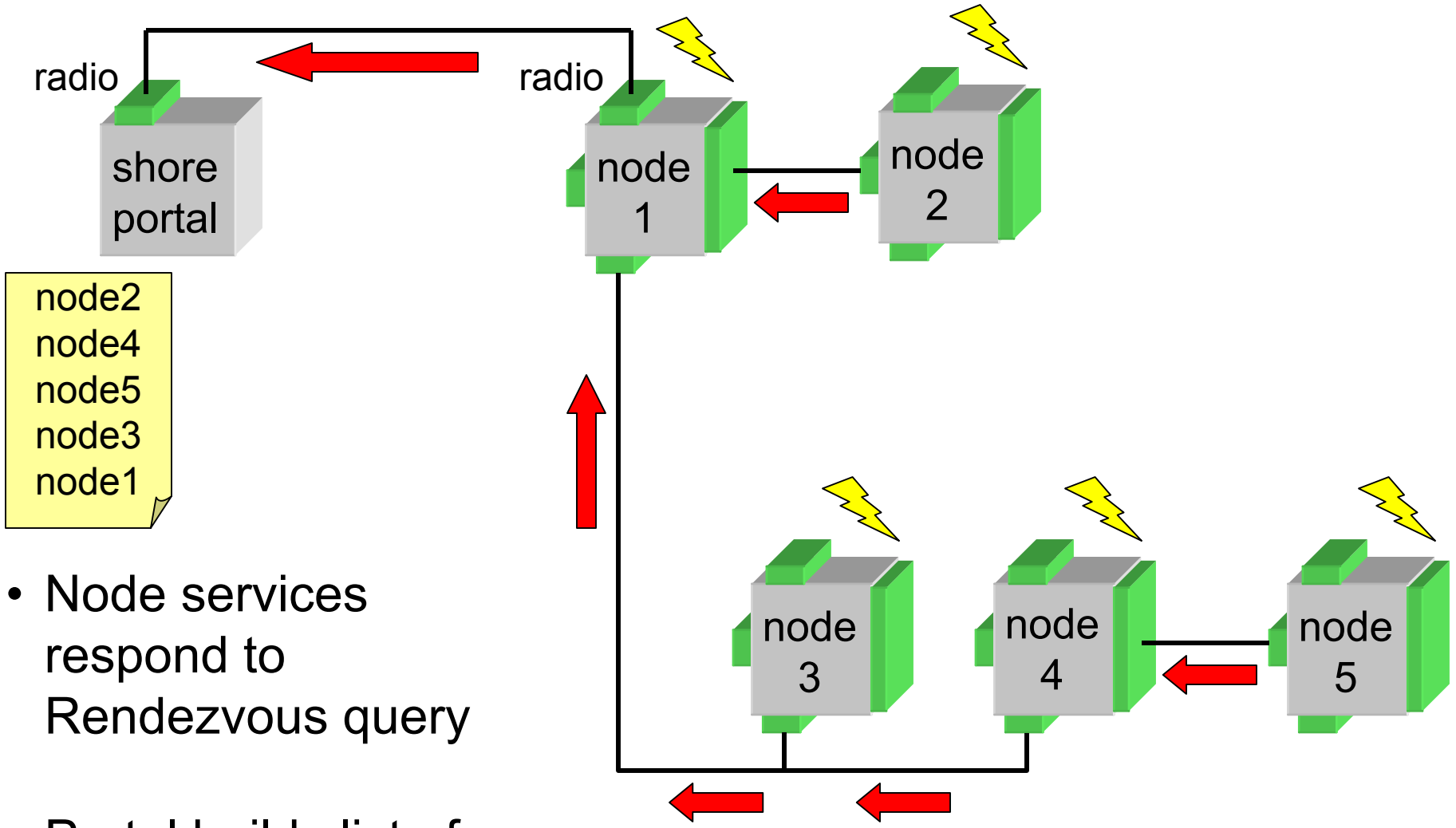


- Node 1 powers radio, establishes ppp connection with shore
- Node 1 notifies portal that link is up



- Surface powers radio, establishes ppp connection with shore
- Shore submits Rendezvous DNS query for node services
- Multicast propagates through network

MBARI



- Node services respond to Rendezvous query
- Portal builds list of node services

Tools needed for...

- RS-485/ppp link setup
- IP address assignment
- Packet routing

IP address assignment: Ethernet interfaces

- DHCP, automatic address allocation
  - DHCP server assigns client node an address which is valid until client reboots

DHCP

- *Automatic address allocation*
  - DHCP server assigns client node an address which is valid until client reboots

- We don't need the more complex *dynamic address allocation*

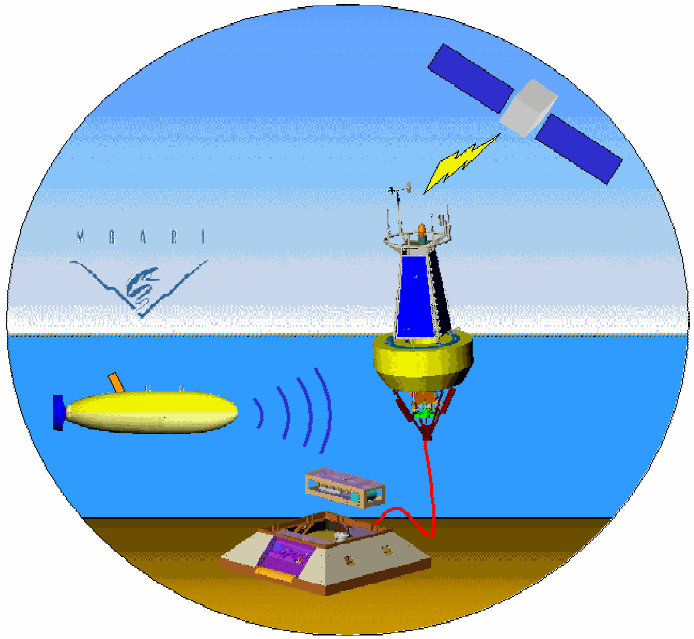
DHCP: interaction with power management

- Sleeping DHCP server: client request via *limited broadcast* will wake it
- Sleeping client: Clients should use *automatic allocation*, which does not require a lease
- Client requests address via broadcast protocol; new client will wake all nodes on network

IP address assignment: serial interfaces

- PPP endpoint assignment
  - Preconfigured before deployment?

Security



Kent Headley

M B A R I

Security

- Requirements
  - Current status
- Issues
  - General concerns
- Plan
  - How to proceed

Primary requirement source(s)

- *SIAM DeployedSoftwareRequirements*
  - *3.7*
- *SIAM OperationsSoftwareRequirements*
  - *2.2.6*
- *Note: both referenced sections are empty!*

Security Requirements

SIAM security requirements

- Not well defined
- No users have set requirements, but there may be implicit assumptions :
  - Science users
    - data privacy
    - administrative privileges
  - Operations users
    - access control
  - Information systems
- Defining requirements will require input from experts

Security Issues

- Wireless access
  - New vulnerabilities
  - Level of importance
- Access control
  - Policies and mechanisms
- Data privacy
  - Protecting ‘proprietary’ data
- Control of bandwidth use
  - Intentional and inadvertent misuse

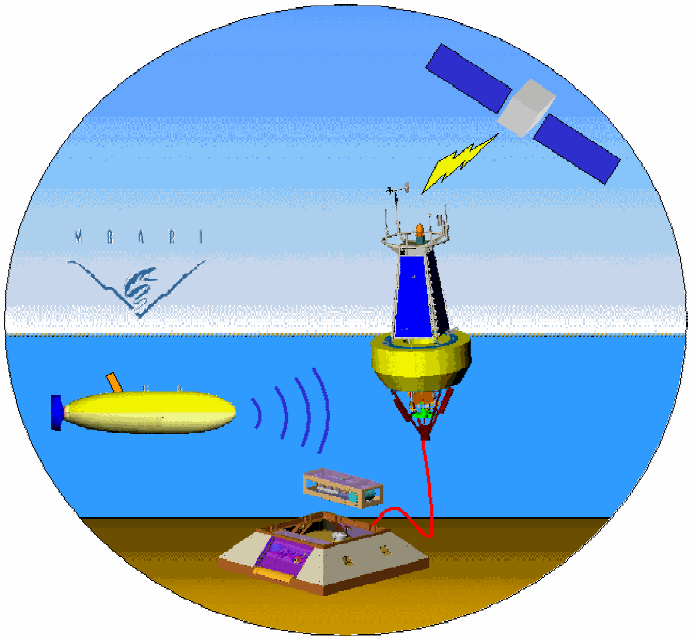
Security Issues

- Protecting access points
  - Balancing security and operational simplicity
- System activity monitoring
  - How much is needed
- Protocol selection
  - device discovery, event propagation, general networking
- Others…

Security Plan

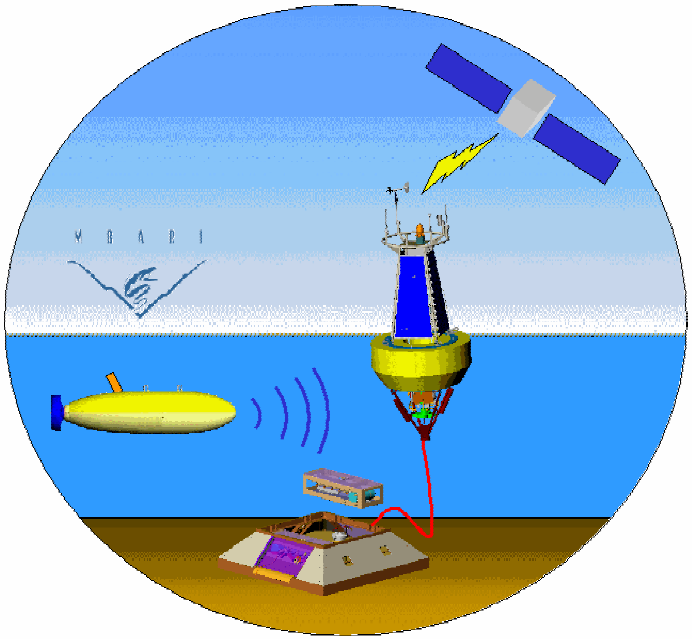
- Gather good requirements
  - Users (including as yet unidentified users)
  - Consult experts
- Construct security policies
- Examine architectural issues
  - Understand state of the art
  - Trade studies
  - Policy vs technology
- Develop design and phased implementation plan

Break



M B A R I

Fault Tolerance



Tom O'Reilly

M B A R I

Primary requirement sources

- *SIAM Behaviors Required by SSDS:*
  - numerous requirements
- *MOOS Deployed Subsystem Software Requirements Specification*
  - numerous requirements
- *MOOS Operations Subsystem Software Requirements Specification*
  - numerous requirements

Deployed file system fault-tolerance

- Onboard FLASH utilizes journaling JFFS2 file system
  - Less prone to corruption
- Compact FLASH file system could be upgraded to EXT3 (journaling)

Network fault-tolerance

- RMI distributed objects
  - Underlying TCP-IP includes retry mechanisms
  - RMI exception mechanisms allow applications to handle network failures
  - If RMI method call completes, very high confidence that data was transferred properly

Network fault-tolerance

- Portal
  - Keeps track of retrieved packet time-stamps; “picks up where it left off” each time RF link is restored
  - Network exception handling
  - Stores retrieved packets as well as distributes to SSDS

Network fault-tolerance

- Service discovery enables applications to respond to network failures
- Investigating event propagation mechanisms for intermittent networks (e.g. Jini events)

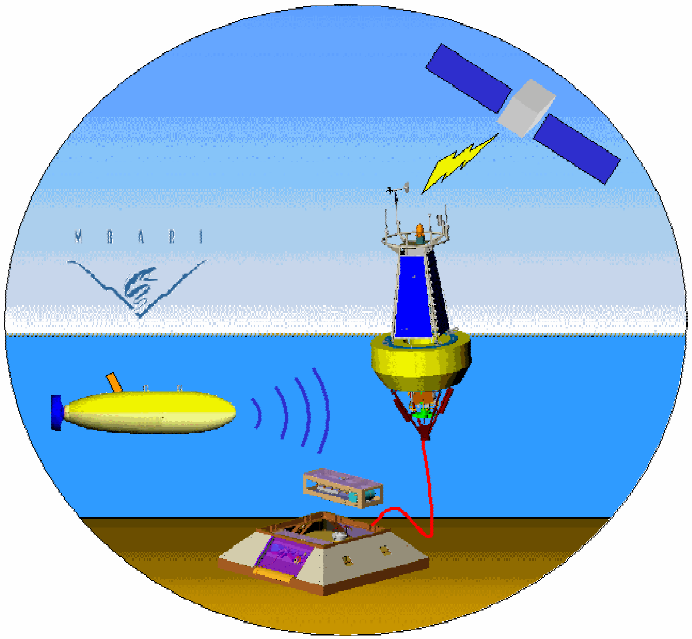
Issues

- What are requirements for automatic handling of environmental alarms and ground faults?

Issues

- Watchdogs
  - Software watchdog
  - RF communications watchdog
  - DPA safety watchdog
  - Others?

Revision Control, Bug Tracking



Mike Risi

M B A R I

Bug tracking and revision control

- Bug tracking system
- Revision control
- Debug message logging

Bug tracking system

- phpBugTracker
  - uses php scripting language
  - freely available at http://phpbt.sourceforge.net/
  - users and developers can create account and enter bugs

CVS revision control

- Use CVS for software revision control
  - developers have access to all modules
  - anyone on MBARI network can view code at anytime via web interface
  - user meeting and deployment releases are tagged
  - many clients available
    - WinCVS
    - gCVS
    - MacCVS
    - CLI
    - editor support (e.g. Visual SlickEdit)

Log4j message logging

- Log4j logging module used for all system error and information debug messages
- Messages can be directed to different message consumers via appenders
  - console
  - sockets
  - files

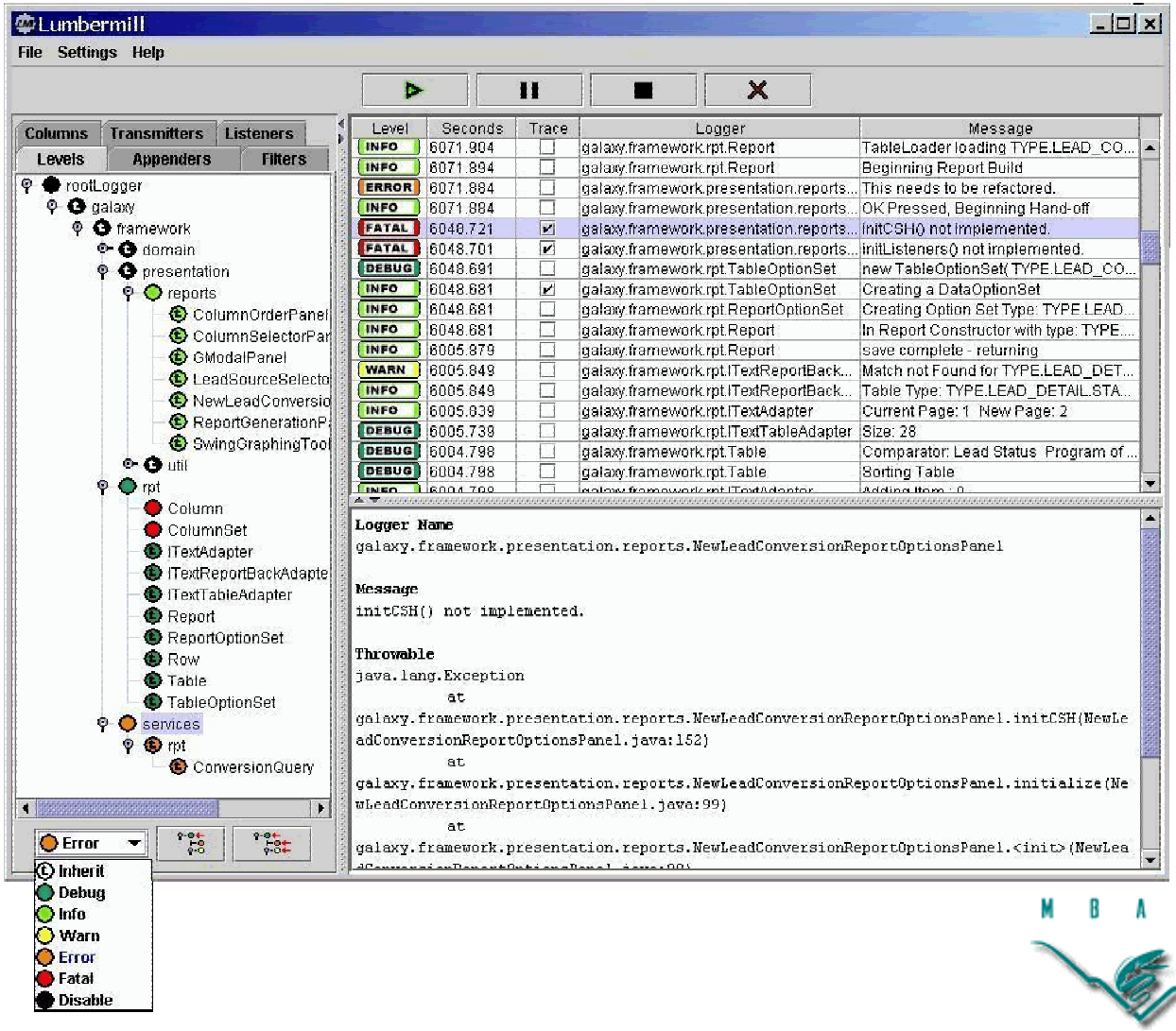
Log4j message logging

- Messages can be logged at different levels and filtered at message producer or consumer
  - ALL
  - DEBUG
  - INFO
  - WARN
  - ERROR
  - FATAL
  - OFF

Log4j message logging

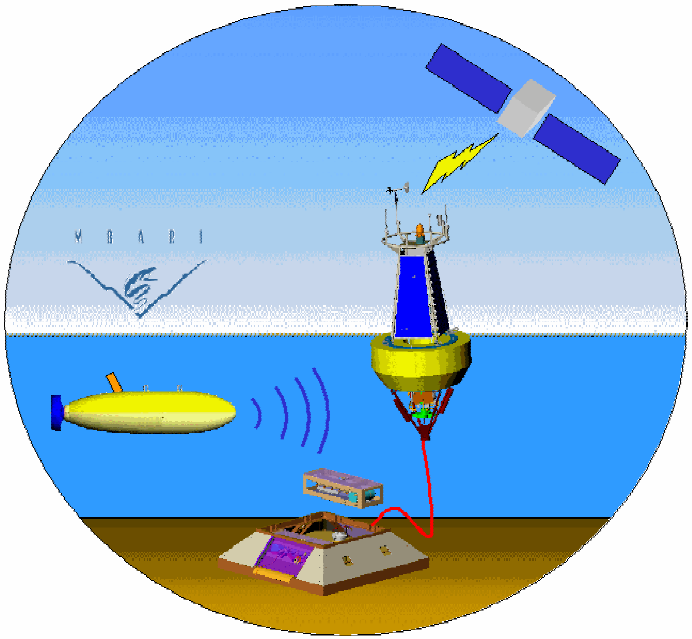
- Socket appender client GUI
  - displays message from socket appender
  - filters messages by package, class, and level
  - free GPL license

http://traxel.com/lumbermill/



MBARI

Discussion



M B A R I