Delay-Tolerant Networking for Challenged Internets

Kevin Fall

Intel Research

Berkeley, CA

kfall@intel.com

http://www.intel-research.net http://www.dtnrg.org

Apr 20, 2004 – Sprint Labs

intel®

RFC1149 : A Challenged Internet

- "…encapsulation of IP datagrams in avian carriers" (i.e. birds, esp carrier pigeons)
- Delivery of datagram:
  - Printed on scroll of paper in hexadecimal
  - Paper affixed to AC by duct tape
  - On receipt, process is reversed, paper is scanned in via OCR

Implementation of RFC1149

CPIP: Carrier Pigeon Internet Protocol

- See http://www.blug.linux.no/rfc1149/

Ping Results

```
Script started on Sat Apr 28 11:24:09 2001
vegard@gyversalen:~$ /sbin/ifconfig tun0
tun0      Link encap:Point-to-Point Protocol
          inet addr:10.0.3.2  P-t-P:10.0.3.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:150  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:88 (88.0 b)  TX bytes:168 (168.0

vegard@gyversalen:~$ ping -i 900 10.0.3.1
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms

--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss
round-trip min/avg/max = 3211900.8/5222806.6/6388671.9 ms
vegard@gyversalen:~$ exit

Script done on Sat Apr 28 14:14:28 2001
```

Private Addresses

About 1.5 Hrs

High Loss

Network Intermittency

- *Intermittency* – the inability to establish and maintain a contemporaneous e2e association
  - Causes: interference, power failure, mis-configuration
  - Observation: potentially much cheaper than ‘persistent’
- Applications and networking layer should gracefully accommodate network outages
  - Planned or not
  - And continue using whatever technology is available
- Networking should be ***Delay Tolerant***

Example: Developing Regions

- Lots of projects to get the *Web* to 3rd world:
  - But not all applications require the Web
    - Web does not equal "The Internet"
    - (e.g. e-mail = most popular Internet application)
  - 'Always on' networking may be hard
    - High installation and operational costs
    - Poor connectivity reflected in poor application performance

- Assuming *network intermittency* may be better…

Unstated Internet Assumptions

- End-to-end RTT is not terribly large
  - A few seconds at the very most [typ < 500ms]
  - (reactive window-based flow/congestion control works)
- Some path exists between endpoints
  - Routing finds single "best" existing route
    - [ECMP is an exception]
- E2E Reliability using ARQ works well
  - True for low loss rates (under 2% or so)
- Packet switching is the right abstraction
  - Internet/IP makes packet switching interoperable

Non-Internet-Like Networks

- Stochastic mobility
  - Mesh networks
  - Mobile routers w/disconnection (e.g. ZebraNet)
- Periodic/predictable mobility
  - Spacecraft communications
  - Busses, mail trucks, police cars, etc (InfoStations)
- “Exotic” links
  - Deep space [40+ min Mars RTT; episodic connectivity]
  - Underwater [acoustics; low rate; high error; latency]

New challenges…

- Very Large Delays
  - Natural prop delay could be seconds to minutes
  - If disconnected, may be much longer
- Intermittent/Scheduled/Opportunistic Links
  - Scheduled transfers can save power and help congestion; scheduling required for rare link assets
- High Link Error Rates / Low Capacity
  - RF noise, light or acoustic interference, LPI/LPD concerns
- Different Network Architectures
  - Many specialized networks won't/can't ever run IP

What to Do?

- Some problems surmountable in Internet/IP
  - 'cover up' the link problems using PEPs
  - Mostly used at "edges," not so much for transit
- Performance Enhancing Proxies (PEPs):
  - Do "something" in the data stream causing endpoint TCP/IP systems to not notice there is a problem
  - Lots of issues with transparency: security, operation with asymmetric routing, etc
- Some environments *never* have an e2e path…
  - Yet still want *eventual delivery* with high probability

Delay-Tolerant Networking Architecture

- Goals
  - Internetwork(s) supporting interoperability across ‘**radically heterogeneous**’ networks
  - Acceptable performance in high loss/delay/error environments
  - Decent performance for low loss/delay/errors
- Components
  - Flexible Naming Scheme with late binding
  - Message Overlay Abstraction and API
  - Routing and link/contact scheduling w/CoS
  - Per-hop Authentication and Reliability

Naming

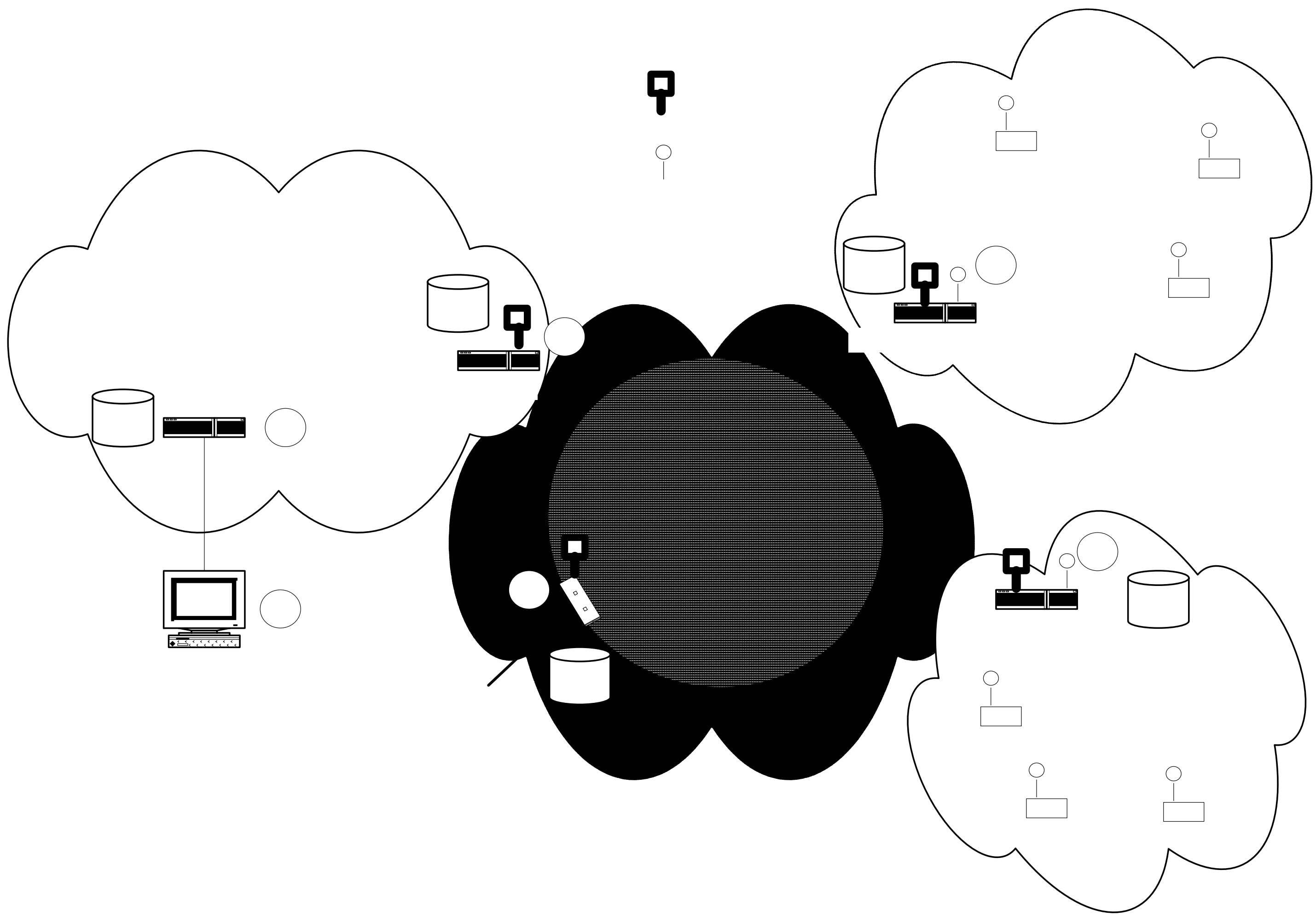
- Support 'radical heterogeneity' using ***regions***:
  - Instance of an internet, not so radical inside a region
  - Common naming and protocol conventions
- Endpoint Name: ordered name pair **`{R,L}`**
  - **`R`**: routing region [globally valid]
  - **`L`**: region-specific, opaque outside region **`R`**
- **Late binding** of **`L`** permits naming flexibility:
  - Associative or location-oriented names [URN vs URL]
  - May encompass esoteric routing [e.g. diffusion]
  - Perhaps an Internet-style URI [see RFC2396]
- *To do*: make **`R,L`** compressible in transit networks

Naming Challenges

- Structure of R (region name)
  - Variable length, hierarchical, centrally? allocated
  - Could likely use DNS namespace w/out mechanism
- How does a sender know/learn destination's R?
  - "just does" (like well-known port)
  - Some centralized or distributed service – TBD (hard)
- What semantic rules really apply to L?
  - Associative and location-based names seem useful
    - Associative – "send to Kevin's pager" [who looks up?]
    - Location – "send to pager [addr: p103x] via Inet gw"
- Associative naming requires *indirection*
  - Unworkable in high-delay/disconn environment

Example Regions

(with Sensor Networks)



Data

Home Base

Reliable Message Overlay

- End-to-End Reliable Message Service: "Bundles"
  - "postal-like" message delivery over regional transports with coarse-grained CoS [4 classes]
  - *Options*: return receipt, "traceroute"-like function, alternative reply-to field, custody transfer
  - Supportable on nearly any type of network
- Applications send/receive bundles
  - "Application data units" of possibly-large size
  - May require framing above some transport protocols
  - Arrange for responses to be processed long after request was sent (application *re-animation*)

Is this just e-mail?

	naming/ late binding	routing	flow contrl	multi-app API	security	reliable delivery	priority
e-mail	Y	N (weak)	Y (weak)	N	Y (opt)	Y (weak)	Y (weak)
DTN	Y	Y	Y	Y	Y (opt)	opt	Y

- Many similarities to e-mail service interface
- Primary difference involves *routing*
- E-mail depends on an underlying layer's routing:
  - Cannot generally move messages closer to their destinations in a partitioned network
  - In the Internet (SMTP) case, not delay tolerant or efficient for long RTTs due to "chattiness"
- E-mail security authenticates only user-to-user

Routing Graph Dynamics

- Topology dynamics may be *predictable*
  - "Scheduled Links"
    - May be direction specific [e.g. ISP dialup]
  - "Opportunistic Links"
    - Unscheduled, unexpected availability
  - "Predicted Links"
    - Learn from history or other non-perfect info
- Link ``*Predictability continuum*''
  - S/O: extreme cases of expected utility/avail of a route
  - Represent by a entropy-like measure (?)
  - Relationship to epidemic routing + erasure coding

Scheduled Link Routing

- *Inputs*: topology graph, vertex buffer limits, contact set, prioritized message demand matrix
- A ***contact*** is an opportunity to communicate:
  - One-way: (t_s, t_e, S, D, C, D)
  - (t_s, t_e): contact start and end times
  - (S, D): source/destination ordered pair
  - C: capacity (rate; assume const over [*s*..*e*]); D: delay
- Vertices have buffer limits; edges in G if ever in any contact
- *Problem*: Compute the "best" set of paths for all messages so as to minimize total delivery time [or something else]

Store and Forward

- Bundle routers generally have persistent storage
  - May offer *custody transfer* "service" if requested
  - Will try "very hard" to not discard messages for which it has accepted custody
  - Accepting custody for a bundle may involve a significant allocation of resources at a bundle router
- Some questions:
  - What do questions of flow and congestion control look like in one of these environment?
  - When should a bundle router avoid taking custody?
  - Given the hop-by-hop nature, if congestion control is figured out, does this also solve flow control?

Fragmentation & Replication

- Fragmentation: dice a large message up
- Replication: copy fragments to enhance delivery probability
- *Proactive* fragmentation
  - Achieve multi-path routing by splitting messages
  - Tricky relationship to custody transfer
- *Reactive* fragmentation
  - Make use of partially-received messages arriving at next hop
  - Effectively makes a fragment out of recv'd msg
  - Unpleasant issue for digital signatures

Security Concerns

- Infrastructure protection
  - Deny data forwarding to unauthorized users
- DTN Security Requirements
  - Authentication of overlay forwarders
  - (optional) authentication/privacy for end users
  - Support for access control list methods
  - Operation in primarily-disconnected environments

Security Concerns (2)

- Compromise for scalability
  - ACLs and user keys contained at firs-hop ‘edge’ routers
  - Edge routers authenticate and re-sign messages in their own keys
  - Next-hop routers need only check keys of its O(log n) [or maybe O(1)] neighbors

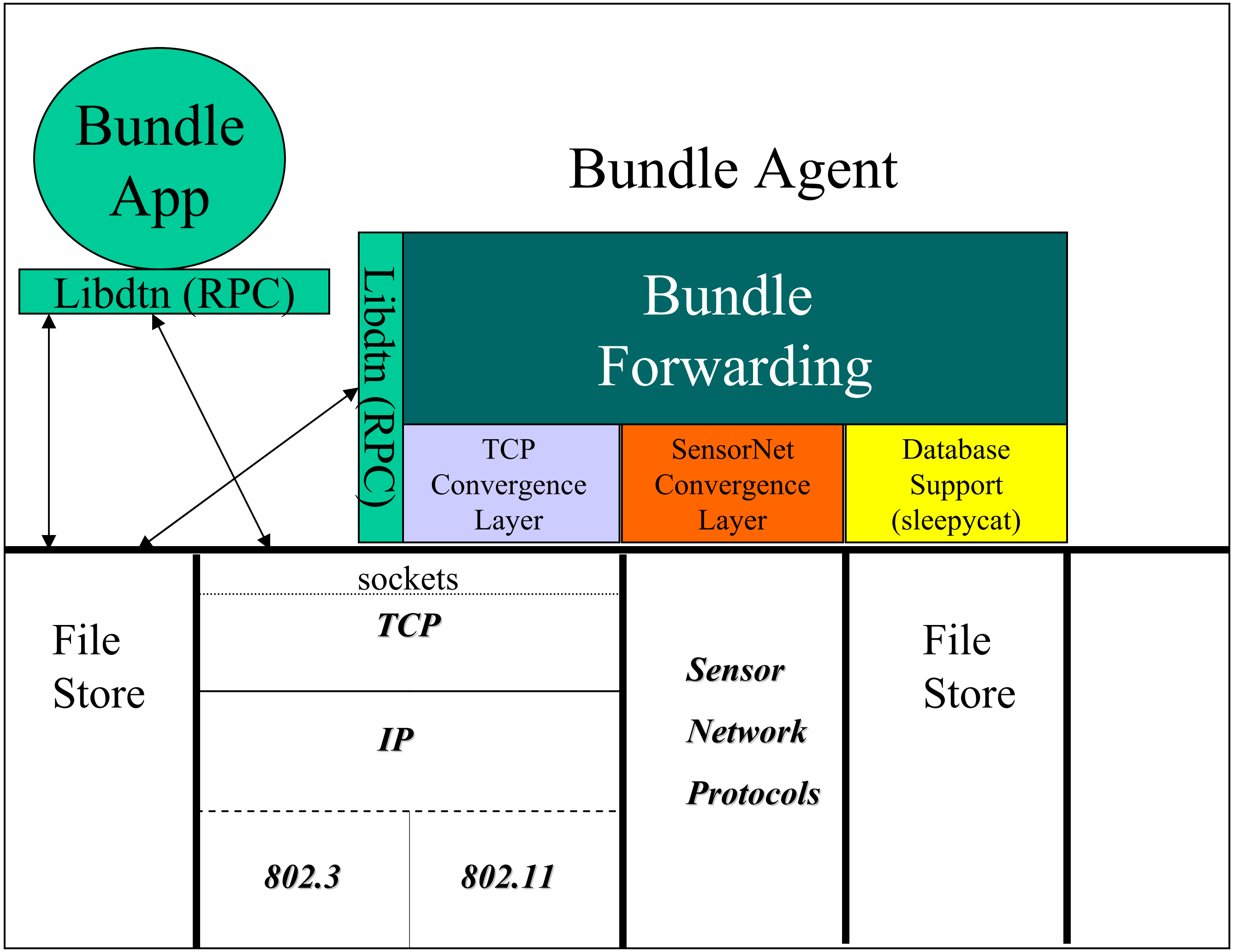
Security Issue Details

- Effect of a router compromise:
  - Router compromise could result in traffic being carried from that point onward
  - Router cannot completely masquerade as sender
    - Sending user still has its own private/public pair
- Identity-Based Cryptography (IBC)
  - Asymmetric scheme based on ECC (Weil/Tate pairing)
  - Allows a form of 'on the fly' generation of public keys
    - No public key storage
    - No sending copy of sender's public keys
    - Can do credentials as well
    - (New--Cryptanalysis may still find issues)

Application Interface

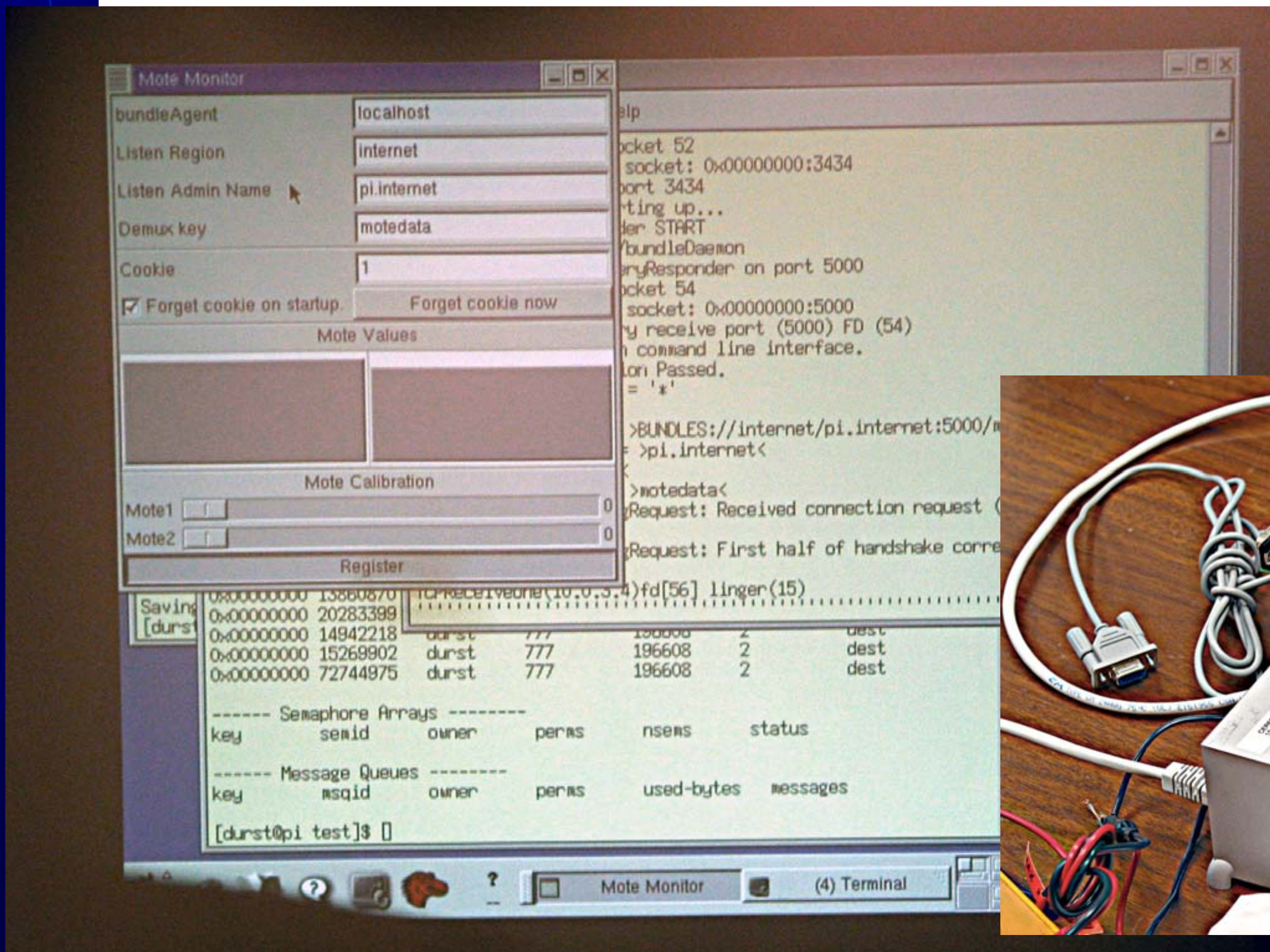
- API is Asynchronous [“split-phase”]
  - Callback registrations are persistent
  - Methods to ‘re-animate’ programs no long running
  - Implemented as RPC-based shared lib (Linux)
- Application interface details
  - Send options similar to postal system
    - Return-receipt, 4 priorities, ‘traceroute’
  - Query options indicate whether message is likely to be offloaded from local node
    - Can be used for user interface and cache control
  - Status results to sender or 3rd party

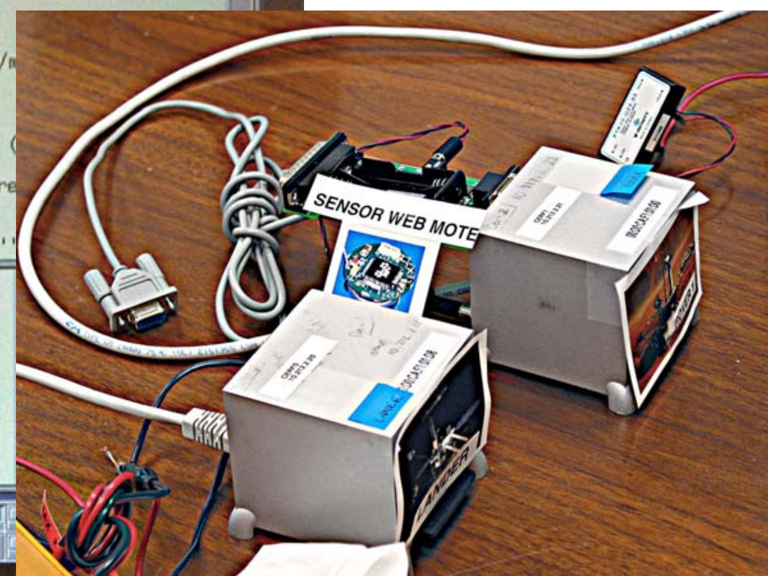
Bundle Forwarder



Demo (1)

Demo (2)


Mote Monitor
bundleAgent
localhost
Listen Region
internet
Listen Admin Name
pi.internet
Demux key
motedata
Cookie
1
Forget cookie on startup
Forget cookie now
Mote Values
Mote Calibration
Mote1
Mote2
Register
socket: 0x00000000:3434
socket: 0x00000000:5000
>BUNDLES://internet/pi.internet:5000/
>pi.internet<
>motedata<
Request: Received connection request
Request: First half of handshake corre
------ Semaphore Arrays --------
key semid owner perms nsems status
------ Message Queues --------
key msqid owner perms used-bytes messages
[durst@pi test]$
Mote Monitor
(4) Terminal


SENSOR WEB MOTE
LANDER

DTN Project Status

- IETF/IRTF DTNRG formed end of 2002
  - See http://www.dtnrg.org
- DTN Agent Source code released 3/2003
  - Available via CVS (see web page)
- Several available documents (currently ID's):
  - DTNRG Architecture document
  - Bundle specification
  - Application of DTN in the IPN
- Spawned new program at DARPA
  - See http://www.darpa.mil/ato/solicit/DTN/

DTN for Developing Regions of the World

ICT4B – Information and Communication Technologies for Billions (NSF: UCB/Intel + HP)

- Systems approach to sustainable development
- Engage and educate researchers
- Harness talents of native peoples

ICT Uses in Developing Regions

- Healthcare
- Government
- Trade
- Communications

System Architecture

- Clusters (service providers)
- Village Kiosks (cache, comms)
- End-user devices and sensors
- Intermittent Networking (DTN)

Delay Tolerant Networking

- Network/applications tolerant to disruption/disconnection, heterogeneity
  - Power, interference
  - Mobility
  - Vastly cheaper infrastructure if real-time not required

ICT4B Project Status

- ICT4B NSF ITR funded 10/2003 (5yr)
- DTN forwarding layer and early apps being tested (code released 3/2003)
- Joint UCB/Intel attendance at ‘ICT for Sustainable Development’ conference Jan 2004/Bangalore; ‘Bridging the Divide’ conference Mar 2004/Berkeley; ‘Digital Rally’ Apr 2004/San Jose
- Fellow travelers: HP Labs India, IIT Bombay/Kanpur/Madras, Univ. of Washington, MITRE, DARPA, CMU, UCLA, JPL, U Waterloo

Acknowledgements

– Bob Durst, Keith Scott (MITRE)
– Vint Cerf (MCI)
– Scott Burleigh, Adrian Hooke (NASA/JPL)
– Juan Alonso (SICS)
– Howard Weiss (SPARTA)
– Forrest Warthman (Warththman)
– Stephen Farrell (Trinity College, Dublin/Ireland)
– Sushant Jain (Univ of Washington)
– Eric Brewer, Sergiu Nadevschi, Mike Demmer, Rabin Patra (UCB)
– Melissa Ho (Intel Research)
– Jordan Hayes (Thinkbank)
– The *dtn-interest* mailing list

For more Information

- Delay Tolerant Networking Research Group
  - **http://www.dtnrg.org**
- Intel Research
  - **http://www.intel-research.net**
- Technologies/Infrastructure for Developing Regions:
  - **http://tier.cs.berkeley.edu**

kfall@intel-research.net

Thank you…