Center for Embedded Networked Sensing

Disruption Tolerant Shell

Martin Lukac, Lewis Girod, Deborah Estrin
CENS System Lab – http://research.cens.ucla.edu

Introduction: Data Collection and System Management in Challenged Networks

Meso American Subduction Experiment

- **Extensive**: 500 Km from Acapulco through Mexico City to Tampico
- **Dense**: 1 sensor every 5-10 Km
- **High bandwidth**: data acquisition rate: 3 x 24 bit channels at 100Hz each
- **Online and reliable**: semi real-time (on the order of days), reliable data delivery to UCLA for analysis
- **Online system management**: query state, change configuration, update binaries
- **Application driven topology**: application determines sensor placement. Infrastructure does not

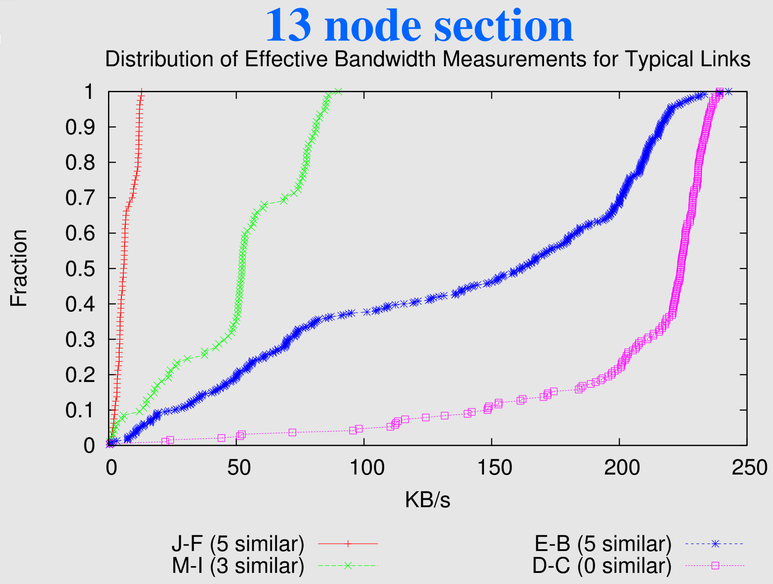
Software Requirements

- **Data delivery** – Bandwidth driven
  - –Bandwidth: 20-40 of MB per day per station
  - –Latency: get the data eventually, but reliably
  - –Many to one routing
- **System management** – Latency driven
  - –Bandwidth: usually less than 10's of KB's
  - –Latency: as fast as possible
  - –One to all routing and back

Problem Description: End-to-End Tools Fail at Critical Times

- **Frequent unpredictable disconnections**
  - –Rainy season: sites flood (some 24x7) and trees grow
  - –Wind/weather: misaligned antennas
  - –Equipment malfunction: amps burn, voltage regulators break
- **Poor and unstable links**
  - –Connectivity is a secondary concern for site selection
  - –Stretched links highly susceptible to weather and environment
- **Human effort is a critical resource**
  - –Installation, maintenance, protection

Bandwidth Variability
13 node section



- Data delivery and system management techniques designed for wired or always-on-wireless do not work well
  - –Typical tools use TCP to create and maintain an end to end session to deliver a stream of data over multiple hops
  - –These tools expect reliable links with low latencies
- Patterns of poor links, disconnections, and disruptions
  - –Difficult to obtain and maintain end-to-end connections
  - –Intermittent end-to-end connections insufficient for required bandwidth and latency

Proposed Solution: Disruption Tolerant Shell

Data Delivery: DTN

- **Use Delay Tolerant Networking techniques**
- **Buffer data into hour long bundles (1-3 MB)**
- **Deliberate one hop bundle transfer**
- **Path to sink determined by best ETX**
- **Improvement over end-to-end**
  - –Not affected by path disconnections
  - –Keeps retrying on single link instead of full path
  - –Continual 'progress' being made towards sink
  - –More efficient use of bandwidth in face of disconnections and bottlenecks

System Management: DTS

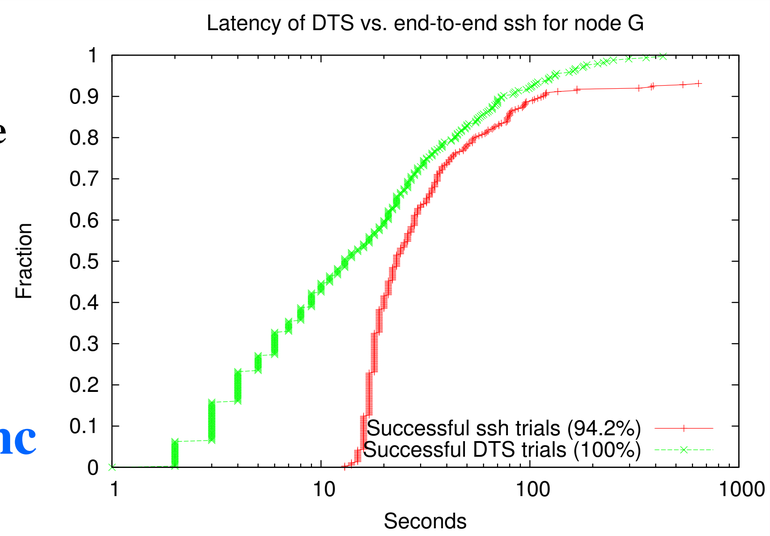
- **Existing management tool: remote shell (ssh)**
- **Modified management tool: Disruption Tolerant Shell (DTS)**
  - –Asynchronous remote shell to all nodes in network simultaneously
  - –Provides node management capabilities when end-to-end connections are unavailable or fail
  - –Ensures that commands will succeed: as long as there is eventually a connection between a node and any other node that already has the command

DTS features

- **Guaranteed in order execution from source node**
- **Safe recovery from reboots and crashes**
- **Implicit feed back on nodes and links: spot bottlenecks, dead nodes**
- **Execute a command on individual nodes**
- **Push a file to all nodes**
  - –Distribute new script or component

DTS Results - Cuernavaca

- **Compared latency of DTS to parallel ssh**
- **DTS is faster 90% of the time, comparable the rest of the time**
- **DTS reaches 100% of nodes**
  - –ssh requires retries from the source node
- **Latency can vary by day, but DTS always faster or comparable to ssh**



DTS Network Service: StateSync

- **StateSync: Reliable and efficient publish-subscribe mechanism**
- **Implements a broadcast dissemination protocol**
  - –Published data is hop-scoped
  - –DTS publishes commands and responses one hop
- **Works well for applications that require:**
  - –Reliable delivery
  - –Have a few Kbytes of data to share
  - –Data lifetime is long compared to system latency requirements
  - –Suitable for DTN since it does not use end-to-end connections
- **StateSync data model: tables of key value pairs**
  - –DTS has a command table and response table
- **Logging mechanism**
  - –Do not republish whole table: only send changes to tables
  - –More efficient use of bandwidth in face of disconnections
- **Retransmission protocol**
  - –Keeps retrying on individual links
  - –Not affected by path disconnections
  - –No overhead of end-to-end connection