Poster: Reliable Data Collection in Challenged Networks using Unmanned Aircraft

Andrew Jenkins, Daniel Henkel, Timothy X Brown
Electrical and Computer Engineering and Interdisciplinary Telecommunications Program
University of Colorado at Boulder, Boulder, CO, U.S.A.
{jenkinaj,henk,timxb}@colorado.edu

ABSTRACT

Sensor data collection can be facilitated by radio nodes in unmanned aircraft. They can provide communication to large geographic areas and have high mobility. However, long distances and aircraft mobility may cause episodic connectivity or require data to be physically carried. Such challenged communication necessitates a delay-tolerant approach to networking. We designed and implemented a delay-tolerant network architecture for the sensor data collection problem that provides unicast and multicast connectivity across heterogeneous networks without requiring global knowledge. Our implementation utilizes a modular approach enabling unique networks to build on our testbed. We motivate our design choices, describe the implementation of a real-world testbed, and evaluate network performance using a novel DTN visualization tool.

1. INTRODUCTION

In a wireless sensor network, the basic communication problem is to deliver sensor data (which we denote *events*) to sensor monitoring stations (SMS) that store or process the events. In turn, the SMS may want to send *commands* back to individual sensors to control their operation. We term this class of communication sensor data collection (SDC).

Our work addresses the challenge of sensor networks located remotely from monitoring stations, a common challenge when sensing in large geographical areas [2]. We use unmanned aircraft to deliver events and commands. In some modes, the unmanned aircraft could be simple store-and-forward routers, but as the communication grows more challenged, a delay-tolerant approach is needed. The aircraft accept custody of events when near the sensor networks and then move to the sensor monitoring stations to deliver them.

A testbed system has been constructed and outdoor experiments verify our approach.

CHANTS '07 Montreal, Quebec, Canada (Poster).

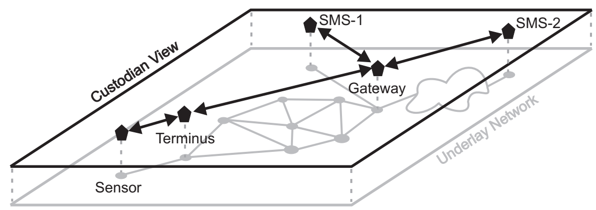

Figure 1: Sensor Data Collection architecture, with an overlay custodian network shown on the elevated plane.

2. ARCHITECTURE AND DESIGN

We construct this delay-tolerant network as an overlay above sensor networks, mobile ad-hoc networks (MANET), and the public Internet, as shown in Fig. 1. Special DTN routers in each participating network implement delay tolerance. The "terminus" accepts sensor data from any sensor node associated with it, which keeps processing and power requirements low in the sensor. Special gateway nodes located in the MANET serve data to sensor monitoring stations (SMS), which can be located anywhere on the Internet. In our scenario, unmanned aircraft (UA) perform as mobile gateways which actively seek out the widely spaced terminus to collect data.

Data forwarding is accomplished in stages through possibly unreliable and opportunistic links, for example in MANETS, as shown in Fig. 2. The DTN routers act as *custodians* of messages, enabling data transfer in networks with no contemporaneous end-to-end paths. This *custody transfer* operates in both directions, multicasting sensor messages (*events*) to all active SMS and unicasting control messages (*commands*) from an SMS to a sensor.

To support multicasting, a custodian replicates the received event message and forwards a copy to active custodians in the next stage. An *equivalence class* concept is implemented to reduce the number of message replications required.

Discovering active custodians at neighboring stages

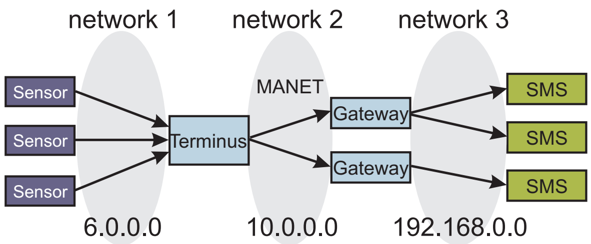

Figure 2: Staged delivery through heterogenous networks.

is done through two kinds of *service discovery* protocols. Gateways that are currently connected to an SMS actively announce their presence to all Terminus through an *advertising protocol*. The Gateways learn about the link status to previously known SMS through a *probing protocol*.

Nodes with high mobility (such as our UA) provide links that are very unreliable and intermittent, and which exhibit significant frame errors, collisions, and congestion. A new transport-layer protocol was devised called *Reliable Packet Forwarding (RPF)* that supports flow control and custody transfer. This development was prompted by observations that TCP performs poorly over multi-hop MANETs. RPF is a sliding-window protocol under development.

Addressing adheres to the common IPv4 scheme and network address and port translation (NAPT) has been implemented in each DTN router to connect the stages of this heterogeneous, possibly disconnected and uncoordinated network.

3. IMPLEMENTATION

The protocol elements are implemented as modules within the Click Modular Router architecture [1]. Our modular approach means that protocol elements can be mixed and matched, and the design of the network expanded or contracted as needed. For instance, powerful sensors can implement the sensor and terminus functions directly. Multiple stages of gateways are also possible with this modular approach.

We constructed the network using these software modules on Soekris single board computers running customized Gentoo Linux from CompactFlash storage. Each node has a miniPCI 802.11 WiFi card, 1 Watt amplifier and two Fast Ethernet interfaces, and can localize via GPS.

4. VISUALIZING PERFORMANCE

In network tests it is useful to be able to simply "observe" the operation of the network and quickly spot anomalies. We have designed a tool called the

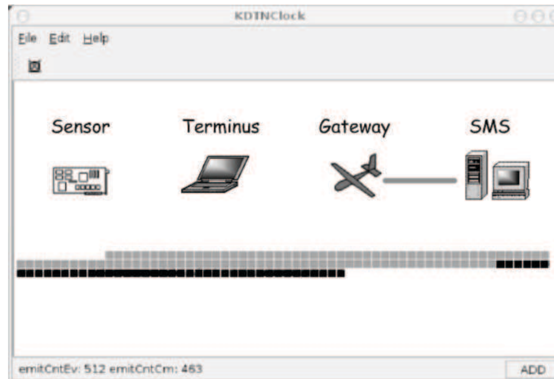

Figure 3: DTNClock: Intuitive network performance visualization.

"DTNClock", which tracks the operation of any DTN and displays it graphically. That makes it possible to observe a bottom-line network performance, to quickly identify network problems, and to verify protocols and implementation during active research. Fig. 3 shows the implementation of our tool as a grid of 60 by 60 squares, each square representing one second of the clock.

In our network scenario, the tool runs on a sensor and verifies connectivity to an SMS. It sends out one "measurement packet" each second and paints a new red square. The SMS at the other end of the network runs a simple application which responds to all measurement packets with an ACK back to the original sender. Upon correct reception of this reply, the DTNClock in the sensor paints the corresponding square green.

The time it takes for a red square to become green constitutes a full round-trip time. With network performance degrading, e.g., due to unreliable links and multiple retransmissions, this time gap widens. In extreme cases, such as data ferrying, one would see groups of red squares accumulating before being acknowledged, i.e., painted green, in bursts. This is shown in the upper part of Fig. 3.

5. POSTER DEMO

We provide a tabletop demonstration of an end-to-end four-node network, using the DTNClock to display the connectedness of the network, data ferrying, and recovery time upon reconnection.

6. REFERENCES

[1] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, August 2000.

[2] M. Martonosi. Embedded systems in the wild: Zebranet software, hardware, and deployment experiences. *ACM SIGPLAN Not.*, 41(7):1–1, 2006.

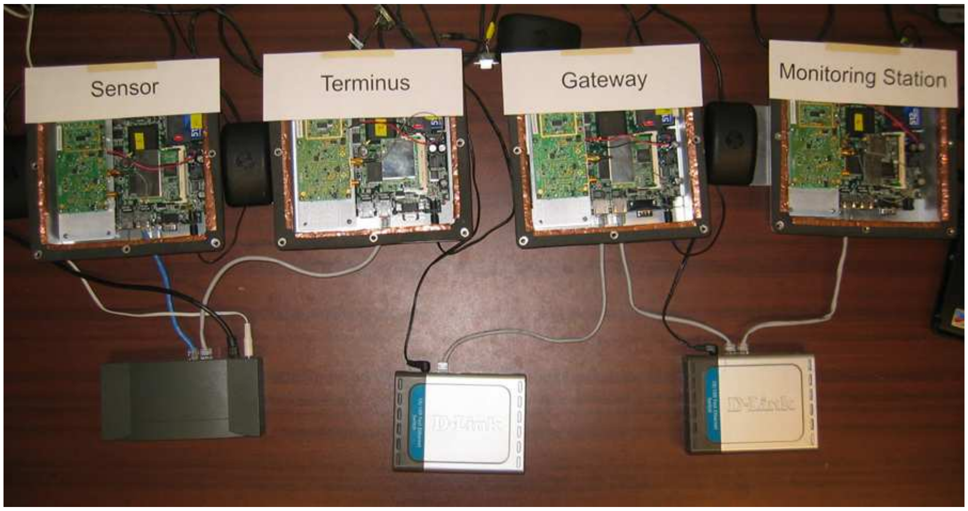

Figure 4: Sensor Data Collection Tabletop Demo Setup. The demo consists of four single-board computers.

DELAY TOLERANT NETWORK DEMO

Our sensor data collection system reliably connects sparsely distributed sensors over unreliable and intermittently connected links. In a well-connected state, the system adds a small amount of delay due to the overhead of custody transfers. As the links experience high error rates or bursty errors, average delay increases further since each packet may require multiple transmissions over lossy links for successful custody transfer. Finally, as the network grows even more challenged, opportunistic ferrying may be required.

Our tabletop demonstration allows users to apply these three conditions to the network (well-connected, lossy links, and ferrying) and observe the impact of these conditions on delay. We will construct a four-node system of a sensor, a terminus, a gateway, and a monitoring station, connected via wired links to allow users control over connectivity. This setup is shown in Fig. 4. The DTNClock provides real-time visual feedback on the state of the network, as described in the abstract and shown in Fig. 3. The sensor generates dummy sensor data (events); these are marked with a unique number and forwarded through terminus and gateway to the SMS. As each event is generated, the DTNClock will paint a red square. Once the event reaches the SMS, it responds with a dummy command containing the same unique number. This command is forwarded through the gateway and terminus to the sensor. Upon arrival at the sensor, the red square is repainted green.

In the well-connected state, users observe that every second, a red square is painted and then after a delay of milliseconds, repainted green. If links are broken, red squares continue to be drawn every second but are never painted green. Upon restoring the link, the custody transfers will be retried successfully and the squares painted green. Users can add random or bursty lossiness to the links and observe more outstanding red squares that are eventually painted green. Finally, by breaking links in order, users can simulate ferrying and observe that data is delivered end-to-end even though the network is never fully connected.

The demonstration requires sufficient table space for 4 of our single board computers and a laptop (1.5m by 1m), and 150W of 110VAC. We will provide the single board computers, laptop, cabling, power supplies and power strip.