Stream: Low Overhead Wireless Reprogramming for Sensor Networks

IEEE INFOCOM 2007

Presented by: Chih-Kai Wang
Ontline

• Introduction
• Existing protocol: Deluge
• Stream design
• Stream analysis
• Experiments and result
• Conclusion
Introduction

• Existing protocol bundling the reprogramming protocol and application as one program image
• Stream: to minimize the size of the part which is need to bound with application.
Deluge

- Using monotonically increase version number for images.
- Segments the binary code image into pages, and pipelines the different pages across the network.
- Three-way handshake: advertisement, request, broadcast.
Node 1: advertisement images meta-data

Node 2: image from node 1 newer then my own.

Request image from node 1

Continue advertisement

Node 5: I have a identical meta-data

Suppress advertisement
Deluge – Transmitting Images

DelugeBasic (Preinstalled)

Image: User application + Reprogramming protocol

DelugeBasic (Preinstalled)

Image: User application + Reprogramming protocol
Stream Design

- StreamReprogrammingSupport(Stream-RS) as image 0.
- StreamApplicationSupport(Stream-AS) attached to user’s application.
- Only transfer user application and Stream-AS.
- Stream-AS provides a little reprogramming functionality of reprogramming protocol.
Stream – Transmitting Images

Stream-RS (Preinstalled)

Image: User application + Stream-AS

Stream-RS (Preinstalled)

Image: User application + Stream-AS
Stream: Protocol design

Reboot from image 0 (Stream-RS)

Transfer image

Reboot command

Transfer image

Reboot command

Transfer image

A new node was added

Request new image

Send new image

Reboot from image 0 (Stream-RS)

Running Stream-RS

Running user program and Stream-AS
Stream-AS

• Attached to user application
• Provide the functionality to reboot from image 0
Stream-AS

• Like Deluge, two lines of nesC code must be added:
  – Components StreamASC;
  – Main.StdControl->StreamASC;
Stream-AS

• Steady-state behavior
  – Does not advertise code image meta-data
  – Uses less RAM for Stream than for Deluge
Stream-RS

- Preinstalled in all nodes as node 0
- Executed during reprogramming phase
- Transfer image
Stream-RS

- Three-way handshake
Stream-RS

S={3,4}

advertisement

S={4}
Stream analysis-Energy Cost

• Assume retransmission of packet are independent
• The energy cost of transfer a image to a h hop node

\[ E_h = \frac{h \times N_p \times N_{pkt} \times C}{P_s} \]

– Ps = the probability of successful transmission
– Np = number of pages
– Npkt = number of packets in each page
– C = energy cost of transmitting one packet
Stream analysis-Energy Cost

• For total energy cost

\[ E = \sum_{h=1}^{h_{\text{max}}} \left[ N_{Nh} \times \frac{h \times N_p \times N_{\text{pkt}} \times C}{P_s} \right] \]

- \( N_{Nh} \) = Number of nodes at hop h
Stream analysis-Energy Cost

A standalone application (which does not perform radio communication)

- Deluge: increase 20 pages
- Stream: increase 10 pages

Figure 3: Total energy consumed in the 10x10 grid topology with standalone applications
Stream analysis-Energy Cost

Figure 4: Total energy consumed in the 10x10 grid topology with communicating applications

GenericComm component (provided by TinyOS)

- Deluge: increase 11 page
- Stream: increase 1 pages
Stream analysis-Convergence Time

• The expected convergence time is

\[ T_{\text{conv}} = T_r \times (N_p + h_{\text{max}} - 1) \times \sum_{i=1}^{\infty} [1 - (1 - (1 - P_s)^{i-1})^{N_{\text{pkt}}}] \]

- \( T_r \) = round time
- \( h_{\text{max}} \) = maximum number of hops
Stream analysis-Energy Cost

A standalone application (which does not perform radio communication)

- Deluge: increase 20 page
- Stream: increase 10 pages

Figure 5: Convergence time for 10x10 grid topology with standalone applications
Stream analysis-Energy Cost

Figure 6: Convergence time for 10×10 grid topology with communicating applications

GenericComm component (provided by TinyOS)

• Deluge: increase 11 page
• Stream: increase 1 pages
Experiments

• Mica2
  – 7.37 MHz, 8-bit microcontroller
  – 128KB program memory
  – 4KB RAM
  – 512KB external flash
  – Communicate via 916MHz radio transceiver

• Topologies:
  – Grid
  – Linear
Experiments - Grid

- Application: 11 pages
- Deluge: add 22 pages
- Stream: add 12 pages

Figure 7: Reprogramming time for grid networks

Figure 8: Number of bytes transmitted in the network during reprogramming for grid networks
Experiments - Linear

- Application: 11 pages
- Deluge: add 22 pages
- Stream: add 12 pages
Simulation

• Use TOSSIM
• Larger grid
Simulation

Figure 11: Reprogramming time for nxn grids

Figure 12: Number of bytes transmitted in the network during reprogramming for nxn grids

Figure 13: Reprogramming time (different node densities)

Figure 14: Number of bytes transmitted in the network during reprogramming for different node densities
Conclusion

• Reduce transmitted bit
• Reduce programming time, energy costs and program memory
• Improve the protocol for a new node to get image from network