Wireless ACK Collisions Not Considered Harmful

Prabal Dutta†, Răzvan Musăloiu-E.‡, Ion Stoica†, Andrea Terzis‡

Computer Science Division†
University of California, Berkeley
{prabal,istoica}@cs.berkeley.edu

Computer Science Department‡
Johns Hopkins University
{razvanm,terzis}@cs.jhu.edu

HotNets-VII - Calgary, Alberta, Canada - Oct. 6-7, 2008
Energy is *the* critical resource in wireless embedded systems

Therefore, the radio is kept mostly off and the trend is to keep it off more. And there’s growing interest in operation at 0.1% or 0.01%.

And the radio dominates the power budget even at low duty cycles of 1-2%.

---

Asynchronous network wakeup

P. Dutta, D. Culler, S. Shenker, “Procrastination Might Lead to a Longer and More Useful Life”, HotNets-VI, November, 2007, Atlanta, GA
Wait, not so fast! Node 5 might receive a response

- **Power capture**: One response has a sufficiently higher power than the sum of the others...
  
  *and it arrives first*

- **Delay capture**: One response frame arrives some time before the remaining ones...
  
  *and its power is higher than the sum of the others*

- **Message-in-Message capture**: Like power capture, but the highest power frame arrives in the middle of another frame transmission and radio detects elevated energy...
  
  *and the radio does continuous preamble detection*

- These are a lot of caveats...
The Query/Response exchange can be mapped to “acknowledged anycast” semantics

- A node
  - Broadcasts (or multicasts) a query packet
  - Waits to see whether the packet is ACK’ed
  - By at least one of its neighboring nodes
  - And stays awake if an ACK is received
  - But goes to sleep otherwise

- And the frame collision is like a multicast “ACK implosion”
Outline

- Introduction
- Implementing “acknowledged anycast”
- How well does it work?
- How much does it cost?
- How could it be used?
- What are its limitations?
Backcast, an “acknowledged anycast” implementation

- Backcast is a link-layer frame exchange
  - A single radio DATA frame transmission
  - Triggers zero or more *identical* ACK frames
  - Transmitted with tight timing tolerance
  - So there is minimum inter-symbol interference
  - And where ACKs collide non-destructively at the initiator

You should be skeptical that this idea might work
Many wireless standards use link-layer automatic ACKs
- IEEE 802.11 a/b/g
- IEEE 802.15.4
The DATA/ACK frame turnaround time is tightly-controlled
Creates conditions favorable for power and delay capture
But what should the destination address be?
- 802.11 a/b/g: DON’T ACK broadcast or multicast frames
- 802.15.4: Standard is silent on ACK-ing broadcast frames
We play with the SRC and DST addresses to get around this
Outline

• Introduction
• Implementing acknowledged anycast
• How well does it work?
  - Under carefully-controlled conditions?
  - Under typical operating conditions?
• How much does it cost?
• How could it be used?
• What are its limitations?
Carefully-controlled experiment with equal-power, equal-path delay

**Setup**

- Platform: Telos mote
- Protocol: IEEE 802.15.4
- Radio: Texas Instruments CC2420
- Experiment
  - 8 responder nodes
  - Connect with splitter/attenuator
  - Turned on sequentially
  - Transmit 100 packets
  - 125 ms inter-packet interval
  - Log
    - RSSI: received signal strength
    - LQI: chip correlation rate
    - ARR: ACK reception rate

**Methodology**

- Platform: Telos mote
- Protocol: IEEE 802.15.4
- Radio: Texas Instruments CC2420
- Experiment
  - 8 responder nodes
  - Connect with splitter/attenuator
  - Turned on sequentially
  - Transmit 100 packets
  - 125 ms inter-packet interval
  - Log
    - RSSI: received signal strength
    - LQI: chip correlation rate
    - ARR: ACK reception rate
As the number of colliding ACKs goes from one to eight...

But, for two nodes, LQI exhibits outliers and a lower median
More realistic deployment environment

Setup

Methodology

- Platform: Telos mote
- Protocol: IEEE 802.15.4
- Radio: Texas Instruments CC2420
- Experiment
  - 12 responder nodes
  - Located in CS Dept testbed
  - Turned on sequentially
  - Transmit 500 packets
  - 125 ms inter-packet interval
  - Vary DATA-ACK turnout time
  - Vary # of preamble bytes
  - Log
    - RSSI: received signal strength
    - LQI: chip correlation rate
    - ARR: ACK reception rate
As the number of colliding ACKs go from one to twelve...

Chip errors increase slightly

ACK reception rate remains at 100%

Chip errors increase dramatically

ACK reception rate starts at 100% and then falls quickly

Chip errors are quite volatile

ACK reception rate is variable and volatile

Since experiments are identical except for the ACK timing, these results suggest more than just capture is at play.
Outline

- Introduction
- Implementing acknowledge anycast
- How well does it work?
- How much does it cost?
- How could it be used?
- What are its limitations?
The cost of a Backcast can be less than 1 ms for 802.15.4

RXTX turnaround time: 192 µs

Max data packet 4.256 ms

Responder

Initiator

ACK transmission time 352 µs
Outline

- Introduction
- How does it work?
- How well does it work?
- How much does it cost?

- How could it be used?
  - Asynchronous network wakeup
  - Low-Power Unicast
  - Robust Pollcast

- What are its limitations?
Asynchronous network wakeup: a node might broadcast a wakeup Query and continue to Listen if it receives a Response.

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Listen</th>
<th>Q</th>
<th>A</th>
<th>Listen</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Node 2</th>
<th>Q</th>
<th>A</th>
<th>Listen</th>
<th>Q</th>
<th>A</th>
<th>Listen</th>
<th>Q</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>DST=0xFFFF SRC=0x0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node 3</th>
<th>Q</th>
<th>A</th>
<th>Listen</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Node 4</th>
<th>Q</th>
<th>A</th>
<th>Listen</th>
<th>Q</th>
<th>A</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Node 5</th>
<th>Q</th>
<th>A</th>
</tr>
</thead>
</table>

P. Dutta, D. Culler, S. Shenker, “Procrastination Might Lead to a Longer and More Useful Life”, HotNets-VI, November, 2007, Atlanta, GA
Low-Power Unicast Communications

Node 1 (Sender)

Node 2 (Receiver)

Node 3 (Sender)

Backcast

frame collision
Robust Pollcast

Node 1
(Sender)

Node 2
(Receiver)

Node 3
(Sender)

MAC=0x8765

MAC=0x8765

DST=0xFFFF
SRC=0x0002
PRED=elephant

MAC=0x8765

DST=0x8765

M. Demirbas, O. Soysal, and M. Hussain,
INFOCOM’08, April, 2008, Phoenix, AZ
Outline

• Introduction
• Implementing acknowledged anycast
• How well does it work?
• How much does it cost?
• How could it be used?
• What are its limitations?
  - Protocol support
  - Radio support
  - Other factors
Limitations

- **Protocol support**
  - Superposition semantically valid for modulation scheme
  - Auto ACK with tight timing
  - ACK frames are identical

- **Radio support**
  - Broadcast auto ACKs
  - Multicast auto ACKs
  - Multiple MAC addresses for interface
  - Auto ACKs based on SRC address filtering
  - Auto ACKs based on DST address filtering

- **Other factors**
  - Propagation delay $\rightarrow$ $\Delta$RTT < $\frac{1}{2}$ symbol time
  - ACK frames do not cancel at PHY layer
  - Security $\rightarrow$ Easy to spoof
Backcast

• Provides acknowledged anycast semantics
  - Quickly
  - Efficiently
  - More robustly than app-level query/response
  - Avoids ACK implosion problem

• Abstraction enables better implementations
  - Asynchronous neighbor discovery
  - Low-power unicast communications
  - Robust pollcast
  - ...
Discussion