Review of: *IP is Dead, Long Live IP for Wireless Sensor Networks*

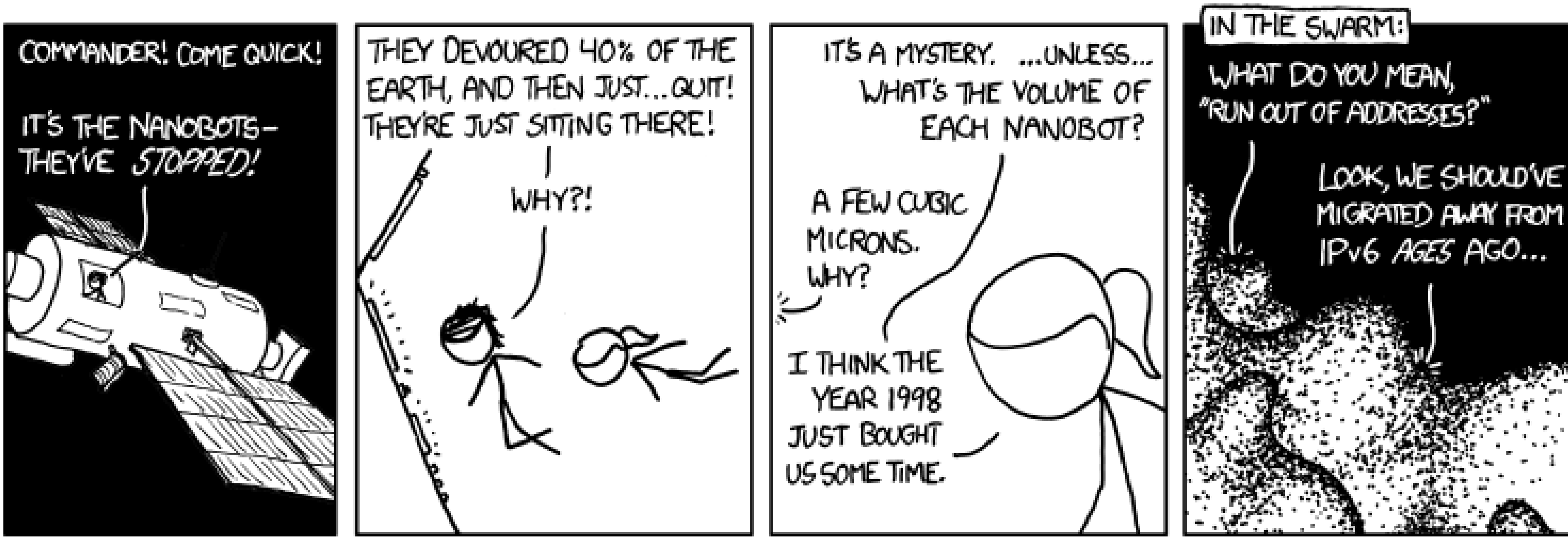
Authors: Jonathan Hui &
David Culler

By Larry Walters
600.450.01 Fall 2011

Overview

- IPv6 based architecture for wireless sensor networks (WSNs)
- Detail the mechanisms they had to implement
- Present results from test network and models of performance
- The Point: IP based architecture for WSNs is superior to existing systems without a particular architecture, and provides good framework going forward

IPv6 (1)



http://xkcd.com/865/

IPv6 (2)

- Hopeful successor to IPv4
- “Fixes” issue of limited address space
- Allows networked communication between source and destination
- Authors claim it is better suited to WSNs than IPv4
  - Can utilize mechanisms already developed for WSNs, cause of inherent generality and extensibility of IPv6
  - Allows for more efficient implementations

History

- Both WSNs and IP have advanced over the years
- Most protocols for WSNs are for link layer, not network
- Now have IEEE 802.15.4
  - For lower speed & power networks
  - IPv6 and associated protocols can be used with it in WSNs

Claims of the Paper

- "Develop a complete IPv6-based network architecture for WSNs that allows end-to-end communication between nodes and any IP device at the network layer"
- "Develop a software architecture that preserves IP's layered protocol model, defining the services, interfaces, and their interactions that can incorporate many of the techniques developed within the WSN community"
- "Present the implementation of a complete, efficient, and production-quality IPv6 solution for WSNs and show that this general network architecture can outperform existing systems that do not adhere to any particular standard or architecture"

Network Structure

- Each WSN node is like an IP router
  - Each has a common prefix
  - WSN on the edge of other network(s)
- Border routers interface WSN with other IP network(s)
- IP link is 1 hop

We’ll say how it’s done

- IP links need to support “best-effort” reliable transport mechanisms
- But IP standard doesn’t say what those mechanisms have to be
- So can use methodologies developed in the WSN world that are common place

Link Layer (1)

- No standard for duty-cycled link layer protocol in multi-hop network
  - So we'll make our own
- Media Management Control (MMC)
  - Coordination of receiver-transmitter schedules
- Medium Access Control (MAC)
  - Arbitrates access transmitters have to channel(s)
- Always-On link
  - Nodes communicate without any link layer state (i.e. no synchronization)

Link Layer (2) – Media Management Control (1)

- Want a duty-cycled link that is:
  - Always-On
  - With low latency
  - Broadcast capable
  - Using synchronous ACKs
- Elements of B-MAC, WiseMAC, and X-MAC
- Sampled Listening
  - “Chirp” (the WSN version of tweeting)
  - Contains destination address and time till actual data comes
  - Allows recipient and non-recipients to act in energy saving and efficient manner

Link Layer (3) – Media Management Control (2)

- Synchronous ACKs
  - Use regular IEEE 802.15.4 data packet, that way can put in address and payload
- Scheduling
  - Put sample period and phase in payload of ACKs, so neighbors can sync to each other (local scheduling)
- Streaming
  - But wait! There's more! Indicate in packet if there's more to come, so don't have to chirp and wakeup as often

Link Layer (4)

- Point of all that was to:
  - Optimize link layer performance
  - Enable network layer to support efficient data delivery
- Neighbor Table
- Provide Received Signal Strength Indication (RSSI) for all traffic
  - Use, with link success rates, as quality info

Header Compression

- IPv6 is too fat for IEEE 802.15.4
- Have to fragment packets, so try to compress as much as possible
- Make assumptions about common values and remove redundant info across layers
  - Can represent values with 1 or 2 bit flags
  - Shorten addresses by dividing namespace
- Header and next header compression
- Can compress 48B header to 6B – 25B depending on type of packet

Configuration & Management

- Use ICMPv6 framework
- Neighbor Discovery
  - Find other nodes
  - Determine link-layer addresses
  - Find routers
  - Configure network parameters
- Router Ads (RAs) tell where the routers are and propagate network parameters
- Use Trickle algorithm (remember CTP) to control when to get updates
- Auto-Configuration
  - Stateless: info to all nodes
  - DHCPv6: info to specific nodes

Forwarding (1)

- Forwarder gets packets, looks up the next hop, then sends data on its way
- Want forwarder to be:
  - Energy Efficient
  - Reliable (high end-to-end success rate)
- Has a forwarding table, but the router handles the entries in it

Forwarding (2) – Unicast (1)

- Hop-by-hop recovery
  - Link layer ACKs indicate if network layer was able to accept message
  - Put network layer info with link layer: don’t have to do own management, and it’s still effective
  - Next hop lookup done for every transmit, allows to quickly adapt to forwarding table changes
- Streaming

Forwarding (3) – Unicast (2)

- Congestion Control
  - Occurs when queue is full; wait that long to get max benefit from streaming
  - Uses feedback to adjust transmission rates
- Quality of Service (QoS)
  - Can specify which is more important: energy efficiency or latency
  - Tagging
    - Latency Tolerant – need so can use energy saving
    - Low and High priority
  - Different traffic classes

Routing (1)

- Can’t use routing protocols developed for traditional networks
  - Might require too much memory or need complex synchronization
- No strictly defined topology
  - Must do routing with partial info
- Configure default routes to the border routers
- Border routers have route to every node
  - Routing concentrated at the borders
  - Minimal control message overhead
  - Allows for optimal routes when talking with other networks

Routing (2)

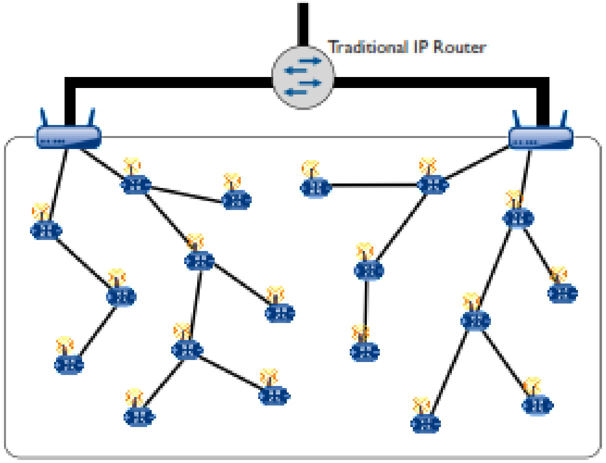
- Discovering Routes
  - Routing table not the same as forwarding table
    - Use routing table to evaluate routes
  - Use RAs to determine the routes
  - Get the sender's distance in hops and ETX relative to nearest border router
- Managing Routing Table
  - Puts potential routes in table, wants to select a default one
  - Use Packet Reception Ratio (PRR) for link quality estimator; gets more accurate over time
  - Sort table by path cost and confidence in quality estimate
  - Remove entry if estimate gets too low

Routing (3)

- Select Default Routes
  - Usually use top entry (sorted)
  - If your draft pick isn't what you thought, try some other routes
  - May pick another route just to get a more accurate quality estimate
  - Use regular data packets to get link quality estimates; less control overhead
- Maintain Route Consistency
  - Put expected hop count and ETX of next hop in transmissions
  - Hop count good for detecting loops
  - If ETX received is significantly different than what is anticipated, then could have inefficient routes

Routing (4)

- WSN nodes have default route
- Low Overhead
  - Occasional broadcasts and unicasts
  - Can using Trickle and existing data
    - Put routing info in RAs



Evaluation (1)

- Implemented all functionality talked about
- Use TelosB and TinyOS (seems like everyone does)
- Developed power cost models
  - Link Energy
    - Predicts within 2%
  - Network Energy
  - Application Energy

Evaluation (2)

- Models made from empirical data
- Tested for a month in real WSN
  - High success rate (avg. 99.98% per node)
    - Data delivered successfully
- Throughput lower than theoretical cause they used AES encryption
- Overall code and energy cost low
  - Some header overhead
  - 24KB code size, uses 3.6KB of RAM
- Bottom Line: Better than those other guys

Evaluation (3)

Deployment	Year	RP (m)	DC	Latency (s)	DRR
GDI [43]	2003	20	2.2%	0.54-1.085	28%
Redwoods [48]	2004	5	1.3%	300	49%
FireWxNet [23]	2005	15	6.7%	900	40%
WiSe [44]	2006	30	1.6%	60	33%
Dozer [7]	2007	2	1.67%	15	98.8%
SensorScope [4]	2008	2	1.11%	120	95%
IPv6	2008	1	0.65%	0.125	99.98%

Observations (i.e. why IP rocks)

- IPv6 and WSNs are highly complementary
  - Requirements of WSNs are well served by IPv6
- IP layering is great, with it got:
  - A low power system
  - Small footprint
  - Good throughput
  - Low latency
  - High reliability
- ICMPv6 good for dealing with lots of unattended nodes that need to be configured
  - Better than anything proposed for WSNs so far
- Provides invaluable interoperability with other IP devices
- Introduces beneficial framework that's been missing until now

References

- J. Hui, D. Culler. *IP is Dead, Long Live IP for Wireless Sensor Networks.* Raleigh, NC, USA. SenSys'08
- Wikipedia
- XKCD