IP is Dead, Long Live IP for Wireless Sensor Networks

Authors: Jonathan W. Hui and David E. Culler

Presenter: Zoltán Kincses

Introduction

- *„While many of the lessons learnt from Internet and mobile network design will be applicable to design wireless sensor networks"*

- *„Sensor networks have different enough requirements to warrant reconsidering the structure of application and services"*

- The internet architecture was denounced:
  - resource constraints may cause us to give up the layered architecture
  - the sheer number of devices and the unattended deployment preclude the broadcast communication or configuration
  - localized algorithms and in-network processing will be required (robustness, scalability)
  - sensor node may not need identity
  - WSNs will be tailored to sensing task at hand

- In addition:
  - traditional interfaces and layers should not be used
  - protocols developed to operate at link layer rather than network layer
  - the basic organization of the WSNs is similar to the IrDA and USB

- This paper provide three primary contribution
  - developed a complete IPv6-based network architecture for WSNs
  - developed a software architecture
    - layered architecture
    - services, interfaces and interactions
  - present the implementation of a complete, efficient and production-quality IPv6 solution for WSNs

Related Works

- Numerous IPv4, IPv6 stacks designed for limited memory and computation capabilities
- uIP changed the perception
  - RFC IP stack applicable on embedded devices
  - uIP include low power link built on IEEE 802.15.4
- IEFT formed the 6LoWPAN working group
  - RFC 4944
    - specifies how IPv6 datagram are carried in IEEE 802.15.4 frame
    - fragmentation, header compression

- MSRLab6, NanoStack validate feasibility of RFC 4944 in WSN
- IPv6 provides a communication architecture for WSNs
  - layering
  - addressing
  - header formats
  - configuration
  - management
  - routing and forwarding

An IPv6 Architecture

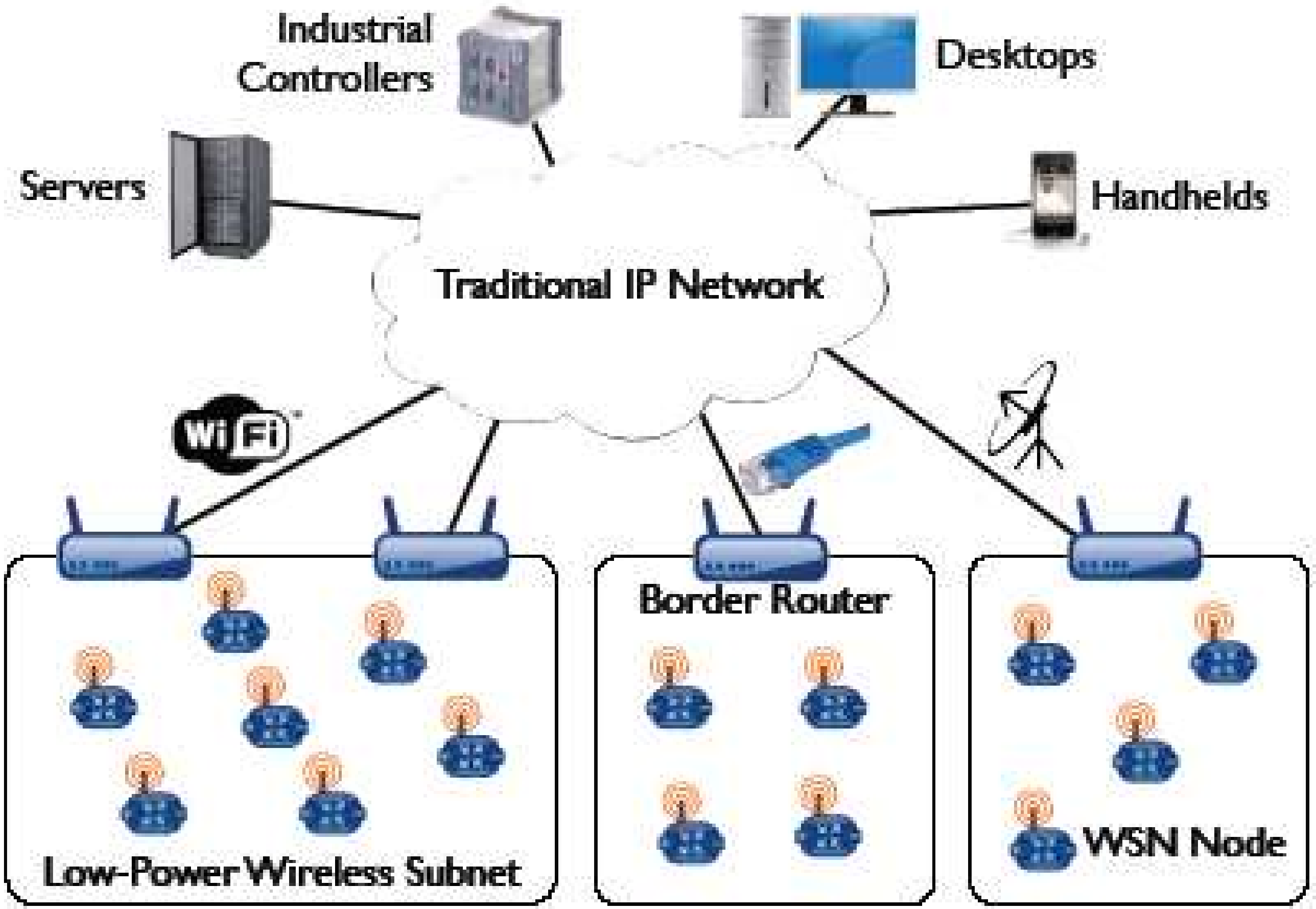
- IPv6 is the designated successor of IPv4
  - scalability is a primary goal
  - address space is much larger
  - autoconfiguration
  - various layer two protocols (ARP, DHCP)
- Support richer set of communication paradigm
- Increase MTU requirement to 1280 bytes

IPv6 is better suited to the needs of WSNs than IPv4 in every dimension?

- Yes:
  - trickle-based dissemination
  - hop-by-hop feedback
  - collection routing
  - inclusion of necessary functionality (DHCP)
  - Autoconf ICMPv6 (scalability, visibility, unattended operation)

Architectural Overview

- WSN network organization
  - collection of low-power wireless nodes
  - require multi-hop to reach each other
  - each WSN nodes serves as an IP router
  - WSNs operate on the edge of IP networks
  - nodes generally remain within the WSN
  - WSN connected to other IP network through a boarder router (Ethernet, WiFi, GPRS, Satellite)


Industrial Controllers
Desktops
Servers
Handhelds
Traditional IP Network
Wi Fi
Border Router
Low-Power Wireless Subnet
WSN Node

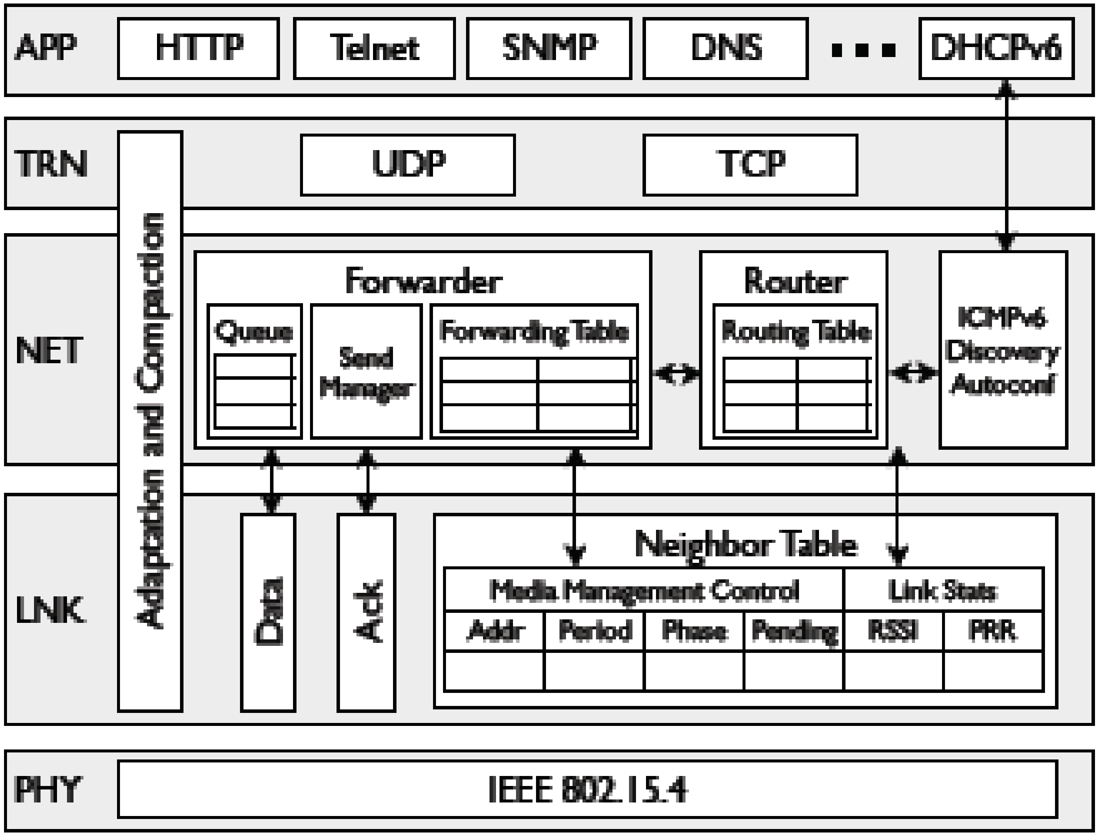


Figure 2: Software Architecture. *Each node implements a full network stack, respecting IP's layered model while using the proper mechanisms to support efficient communication in WSNs.*

Avoiding IP Link Emulation

- IP-based protocols generally assume three operation
  - always-on
    - the IP link provides connectionless communication services
  - best-effort reliability:
    - the link must allow the network layer to achieve high „best-effort” datagram delivery
  - single broadcast domain
    - the IP link provides transitive reachability

- Equate an IP link to those neighbors reachable within a single radio transmission
- WSN composed of overlapping link local scopes
  - gives the necessary visibility
  - expose the unreliable nature of wireless communication

Link Layer

- Developed a duty-cycled link protocol
  - to reduce idle-listening cost
  - this requires scheduling
  - MMC (Media Management Control)
    - Coordination of receiver-transmitter schedules
  - MAC (Medium Access Control)
    - Arbitrates the access to the media

Emulating an Always-On Link

- Two mechanism emerged
  - sampled listening
    - monitors the channel periodically
    - lengthening the transmission
  - scheduling
    - time schedules across the nodes
    - no need to lengthen transmissions
    - cost of establishing the schedules

- Goal to design a low power duty-cycled link
  - always-on
    - nodes should be able to communicate without establishing a connection
  - low latency
    - transition delay to any neighboring node should be low
  - broadcast capable
    - nodes should be able to broadcast frames
  - synchronous acks
    - the link should allow IP to achieve high „best-effort” datagram delivery

Media Management Control

- MMC builds on B-MAC and WiseMAC
  - sampled listening
    - chirp frames (destination address, rendezvous time)
    - channel sample
  - synchronous acks
    - must be used if loss grater then 10%
    - define a new ack frame
  - scheduling
    - sample period and phase in the ack
  - streaming capabilities
    - increase throughput and energy efficiency (Frame Pending bit)

Link Software Abstraction

- Link layer maintains a neighbor table
  - link-specific states
    - addresses
    - schedules
    - frame pending indicator
    - link-quality statistic (RSSI, success rates)
    - LRU (Least Recently Used) policy when inserting new neighbors

Adaptation and Compression

- IEEE 802.15.4 supports
  - 127 bytes payload best case
  - 80 bytes payload in worst case
- IPv6
  - header 40 bytes
  - MTU 1280 bytes

Adaptation layer for frame compression

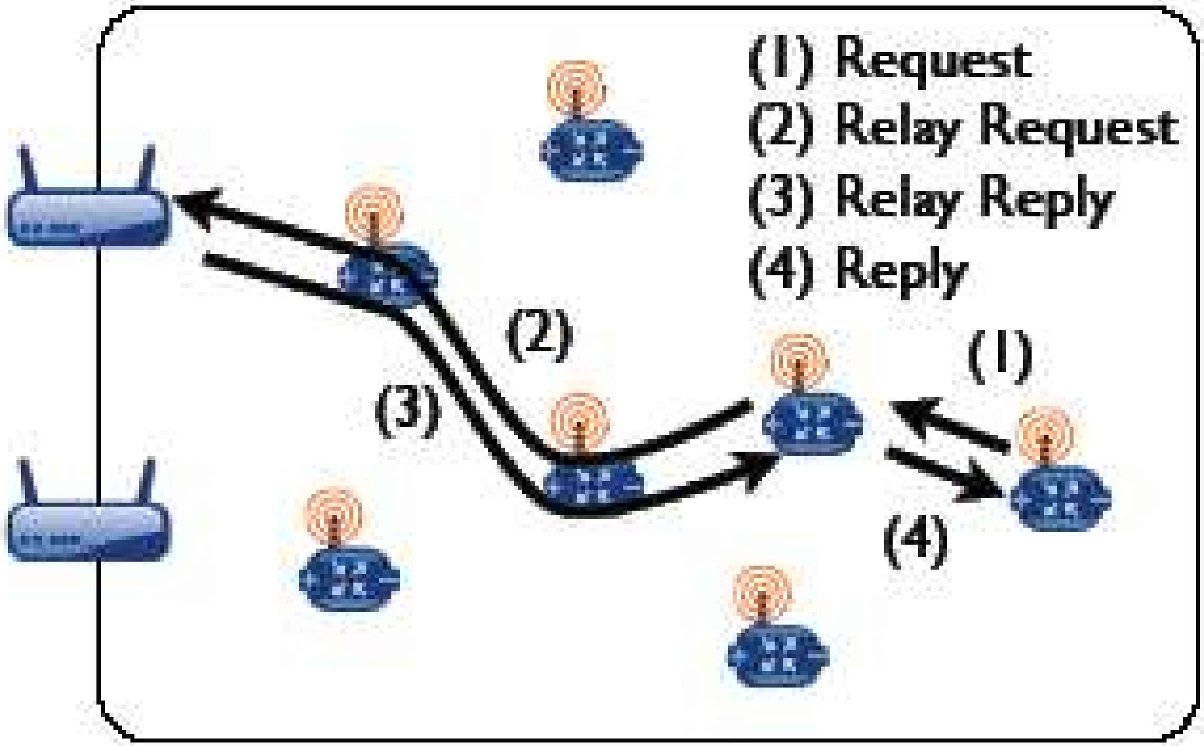
Header Compression

- RFC 4944 compress headers in two ways
  - making assumption about common values
    - IPv6 header compression (6LP_IPHC)
  - removing redundant information across layers
    - IPv6 next header compression (6LP_NHC)
- Compression efficiency
  - 48-byte UDP/IPv6 header
    - 6 bytes (local link unicast)
    - 8 bytes (local link multicast)
    - 25 bytes (communication arbitrary IP devices)

ICMPV6, Discovery and Autoconf

- Neighbor discovery (ND)
  - discover each other, link-layer addresses, find routes, configure network parameters
  - periodically multicast RA (Router Advertisements)
  - boarder routers are the entry points of WSNs
  - use Trickle to maintain network parameters
    - trickle period resets when new parameters are discovered or receive RS (Router Solicitation) messages
    - extended RA including freshness of the information

- Autoconfiguration
  - stateless autoconf methods
    - disseminates parameters to all nodes
    - generates IPv6 addresses by concatenating a prefix with IID
    - challenge to ensure the address is unique
  - DHCPv6 autoconf method
    - assigns parameter to individual nodes
    - directly leverage the IPv6 infrastructure
    - trivially ensure the uniqueness of the addresses
    - every WSN operates as a DHCPv6 Relay Agent


(1) Request
(2) Relay Request
(3) Relay Reply
(4) Reply
(2)
(1)
(3)
(4)

Forwarding

- The IP architecture separates forwarding from routing
  - forwarder responsible for receiving datagram and performing next hop lookups
  - router managing entries in the forwarding table
- IP network layer must provide best-effort datagram delivery
- The primary goal of our forwarder design
  - energy efficiency
  - high end-to-end success rates

Unicast Forwarder

- The used mechanisms
  - hop-by-hop recovery
  - streaming
  - congestion control
  - quality of services

- Hop-by-hop recovery
  - increase energy efficiency and end-to-end delivery rates
  - network layer responsible to retransmit datagrams and allows the re-routing
  - forwarder performs net-hop lookup, because the link quality may be highly variable
- Streaming
  - decrease average transmission costs
  - forwarder indicates whether other packets for the same next-hop destination will follow

- Congestion control
  - detect and mitigate congestion
    - queues become full
    - decrease energy efficiency (forwarding failures)
    - congestion detected when the queue is full
- Quality of services
  - three mechanism
    - upper layers must tag datagrams as latency-tolerant
    - upper layers must tag datagrams as high-priority
    - forwarder permits queue reservations for different traffic classes

Multicast Forwarder

- The multicast forwarder implements a simple controlled flood using Trickle
- Nodes buffer a single datagram for continuous retransmissions until the maximum transmission period is reached

Routing

- Reachability, forming paths, minimize some routing metric
- Ad-hoc wireless networks make the routing challenging
- Limited resources further increase the challenge
- WSN routing protocols based on the link quality
  - RSSI, LQI to compute PRR
  - send control messages to compute PRR

- Because of limited resources
  - nodes have next-hop information for a limited set of destination
  - default route for all others
- Routers configure default routes towards a boarder router
- Board routers maintain host routes to every nodes in the WSN

Default Routes

- Four main tasks
  - discovering routes
  - managing the routing table
  - selecting default routes
  - maintaining route consistency

- Discovering Routes
  - the default router maintains a routing table
  - router uses RA messages to discover routes
  - routing information includes the sender distance in hops and the ETX (estimated transmissions)
- Managing the routing table
  - router inserts potential routes into the routing table based on the quality information
  - sorting the path cost and confidence in the link quality estimate

- Selecting default routes
  - default route: top entry in the routing table
  - router may choose to deviate from this
    - re-route when a few consecutive transmission fails
    - my try another route that potentially provide a better route
  - link quality probes using existing data traffic
- Maintaining a route
  - inconsistent routing information may causes loops
  - tags each datagram with the expected hop count and the EXT
  - in case of inefficient routes RA Trickle timer is restarted

Host Routes

- Boarder routers maintain host routes to every node in the WSN
- route datagrams to WSN nodes by inserting an IPv6 Routing Header (contains path to the destination)
- boarder routers can easily generate host routes, by learning the default route graph and reversing it links
- Recorded Route Option contains a list of addresses identifying the hosts that have forwarded the datagram

Routing Overhead

- Communication overhead
  - broadcasts from the routers
  - unicast transmittions from the leaf nodes to the router
  - overhead reduced by piggybacking
  - worst case routing stretch: 2D

Transport Layer

- Transport layer provides end-to-end communication
- Must implement UDP/TCP
- This allows WSN nodes to communicate unmodified IP devices

Evaluation

- TinyOS 2.x and TelosB platform
- TelosB
  - 16-bit TI MSP430 MCU
  - 48KB ROM, 10KB RAM
  - 2.4 GHz, 250 kbps TI CC2420 IEEE 802.15.4 radio
  - AES-128 authentication

ROM and RAM requirement

Component	ROM	RAM
CC2420 Driver	3149	272
802.15.4 Encryption	1194	101
Media Access Control	330	9
Media Management Control	1348	20
6LoWPAN + IPv6	2550	0
Checksums	134	0
SLAAC	216	32
DHCPv6 Client	212	3
DHCPv6 Proxy	104	0
ICMPv6	522	0
Unicast Forwarder	1158	315
Multicast Forwarder	352	4
Message Buffers	0	2048
Router	2050	64
UDP	450	6
TCP	1674	48

Link energy cost

$$P_{total} = P_{listen} + P_{rx} + P_{tx}$$

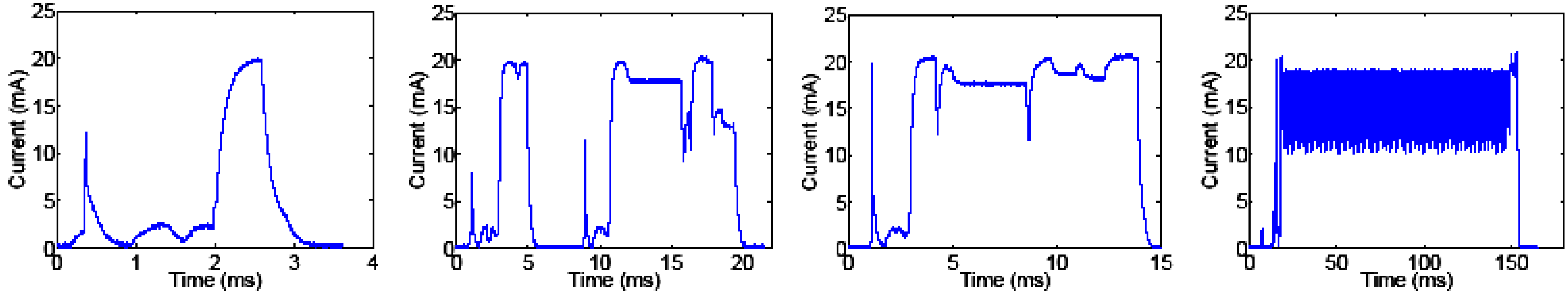
$$P_{listen} = P_{sleep} + f_{sample} \cdot E_{sample}$$

$$P_{rx} = f_{rx} \cdot E_{rx}$$

$$P_{txu} = f_{txu} \cdot \left(E_{tx} + E_{cb} + E_{cd} \cdot \left(2 + \frac{f_{\square}}{f_{txu}} \right) \right)$$

$$P_{txb} = f_{txb} \cdot \left(E_{tx} + E_{cb} + E_{cd} \cdot \frac{1}{f_{sample}} \right)$$

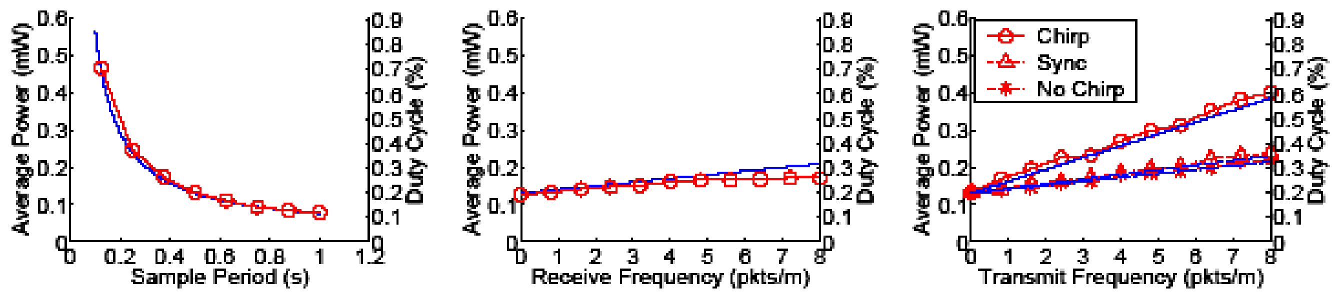
Link power model



(a) Channel Sample (b) Receive Len=127B (c) Transmit Len=127B (d) Transmit Chirp=125ms

Primitive	Cost (uJ)
Sample (E_{sample})	54
Overhear (E_{oh})	108
Receive (E_{rx})	593
TX No Chirp (E_{tx})	630
TX 125ms Chirp (E_{ctx})	6670
Chirp Base (E_{cb})	119
Chirp Delta (E_{cd})	46

(e) Energy Cost of Primitives



(f) Listen (g) Receive (h) Transmit

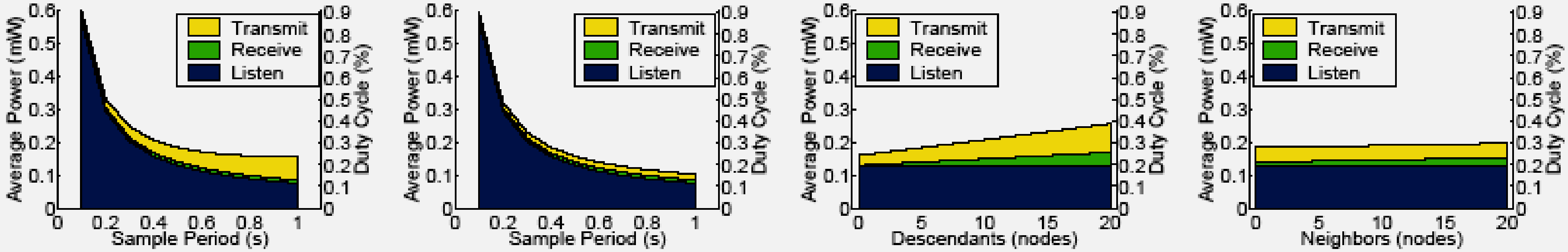
Network Energy Cost

$$f_{rx} = N \cdot f_{ra} + D \cdot f_{rr}$$

$$f_{txb} = f_{ra}$$

$$f_{txu} = \left(1 + D\right) \cdot f_{rr}$$

Network Maintance Model



(a) Sample Period (Router) (b) Sample Period (Host) (c) Descendants (d) Density

Application Energy Cost

$$f_{rx} = N \cdot f_{ra} + D \cdot \left(f_{rr} + f_{app} \right)$$

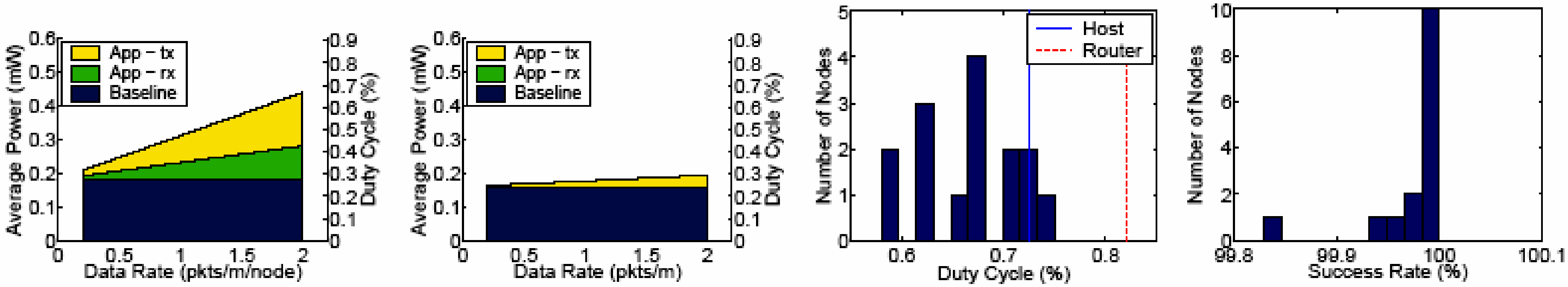
$$f_{txb} = f_{ra}$$

$$f_{txu} = \left(1 + D\right) \cdot \left(f_{rr} + f_{app} \right)$$

Performance of prior WSN deployment

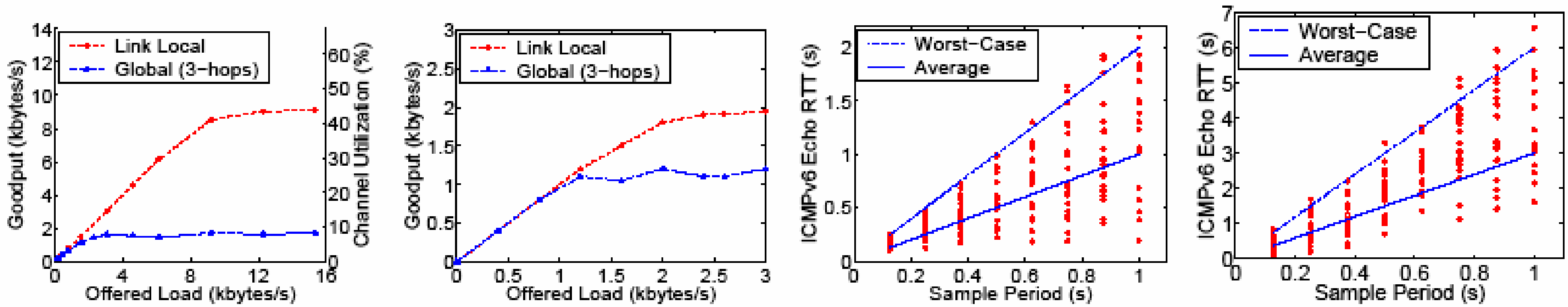
Deployment	Year	RP (m)	DC	Latency (s)	DRR
GDI [43]	2003	20	2.2%	0.54-1.085	28%
Redwoods [48]	2004	5	1.3%	300	49%
FireWxNet [23]	2005	15	6.7%	900	40%
WiSe [44]	2006	30	1.6%	60	33%
Dozer [7]	2007	2	1.67%	15	98.8%
SensorScope [4]	2008	2	1.11%	120	95%
IPv6	2008	1	0.65%	0.125	99.98%

Application Power Consumption



(a) Router (b) Host-Only (c) Duty Cycle (d) Success Rate

Goodput and latency



(a) UDP Goodput (b) TCP Goodput (c) Link Local Ping (d) Global Ping

Thank you for your kind attention