SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks

Laura Galluccio , Sebastiano Milardo†, Giacomo Morabito , Sergio Palazzo University of Catania, Catania, Italy

†CNIT Research Unit Catania, Catania, Italy

2015 IEEE Conference on Computer Communications (INFOCOM)

Introduction

- The reasons of the slow commercial take off of WSNs
  - WSNs are characterized by different requirements depending on the specific application and deployment scenario
- Make WSNs *programmable*
  - Tightly related to the operating system, requiring the application developers to focus on intensive low-level details rather than on the application logic.

Introduction (cont.)

- SDN and OpenFlow have been recently proposed to solve analogous issues in the wired domain
- [4], [5] are the works extend the SDN concepts to WSNs and W-PANs, but still have some shortcoming
  - Protocol details
  - No performance evaluations
- This paper overcome the above problems and define a stateful SDN solution for WSNs called SDN-WISE

[4] T. Luo, H.-P. Tan, and T. Q. S. Quek. Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks. *IEEE Communications Letter*. Vol. 16, No. 11, pp: 1896–1899. November 2012.
[5] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo. Software Defined Wireless Networks: Unbridling SDNs. *In Proc. of EWSDN 2012*. October 2012.

Introduction

- SDN-WISE introduces a software layer, allows several virtual networks to run on the same physical wireless sensor or WPAN network
- Make sensor nodes programmable as finite state machines
  - Reduce the signaling between nodes and Controller
- SDN-WISE provides tools for running a real Controller in an OMNeT++ simulated network

Related Work

- Differently from traditional OpenFlow, Sensor OpenFlow supports in-network packet processing and various types of addressing defined for WSNs [4]
- Compared to Sensor OpenFlow, SDWN offers a more flexible specification of the rules to classify packets, i.e., flow matching can consider any part of the packet, and supports the use of duty cycle to achieve energy efficiency in WSNs [5]

SDN-WISE Overview

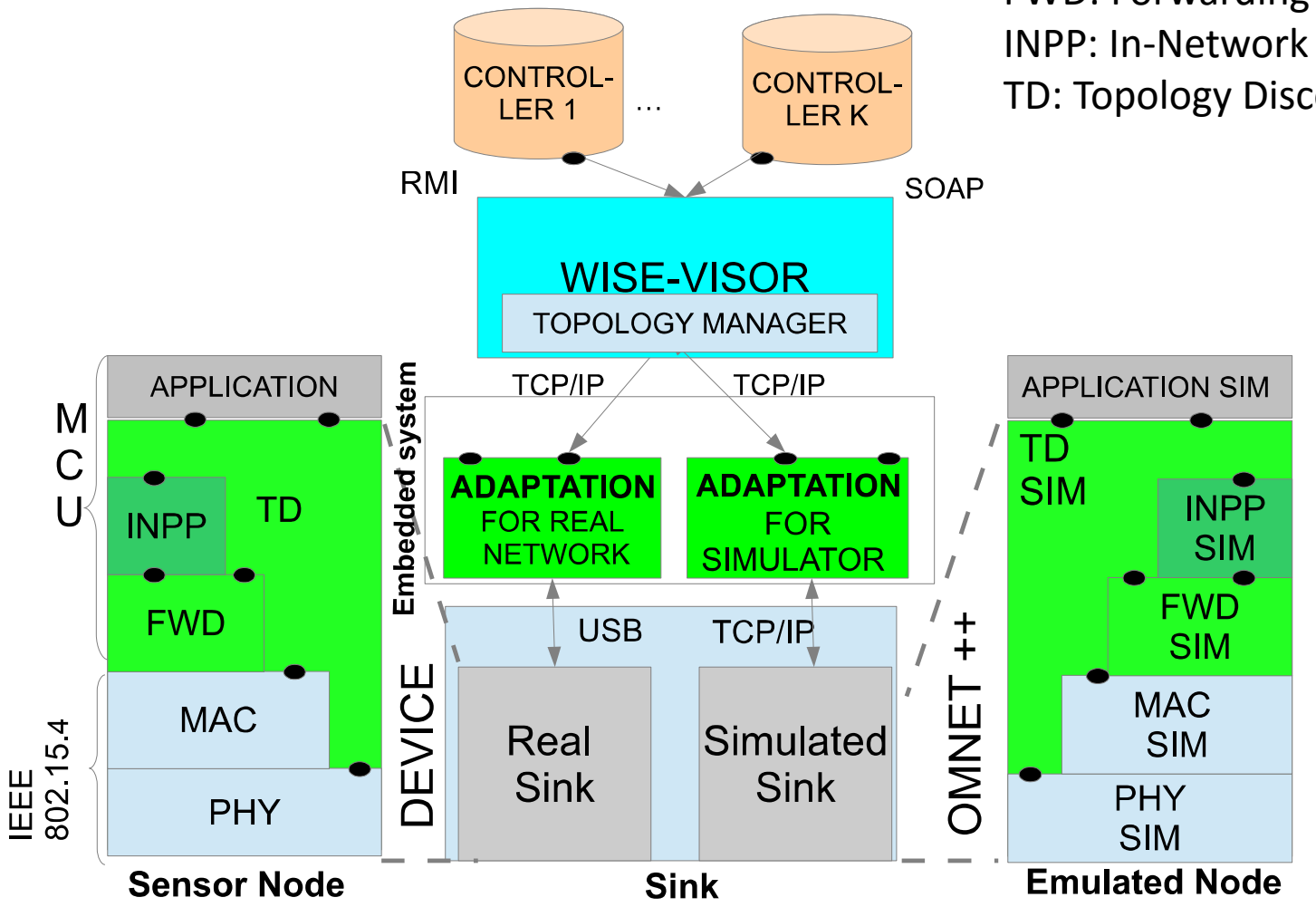
- WSNs are characterized by low capabilities in terms of memory, processing, and energy availability
- Requirements different from wired networks
  - Energy efficient - SDN-WISE support duty cycle and data aggregation
  - SDN-WISE nodes can handle packets based on the content stored in their header and payload
  - Packet classification can be done based more complex relational operators such as *higher than and different from*.

SDN-WISE Approach

- 3 data structures
  - *The WISE States Array*
  - The *Accepted IDs Array*
  - The *WISE Flow Table*
    - Matching Rules
    - Best next hop towards the sink

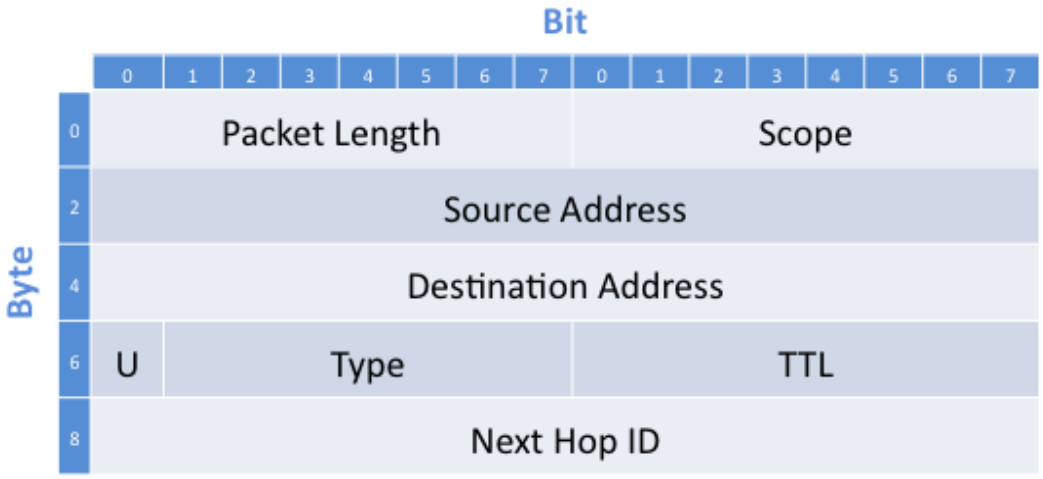
SDN-WISE Protocol Stack

FWD: Forwarding
INPP: In-Network Packet Processing
TD: Topology Discovery



WISE Protocol Details - Forwarding

- WISE packet header
  - The *Scope* identifies a group of Controllers that have expressed interest in the content of the packet
  - flag *U* is used to mark packets that must be delivered to the closest sink



WISE Flow Table

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Op.	Size	S	Addr.	Value	Op.	Size	S	Addr.	Value	Op.	Size	S	Addr.	Value	Type	M	S	Addr.	Value	TTL	Counter
=	2	0	2	B	>	2	0	10	x_{Thr}	=	1	1	0	0	Modify	1	1	0	1	122	23
=	2	0	2	B	≤	2	0	10	x_{Thr}	=	1	1	0	1	Modify	1	1	0	0	122	120
=	2	0	2	B	-	0	-	-	-	-	0	-	-	-	Forward	0	0	0	D	122	143
=	2	0	2	A	=	1	1	0	0	-	0	-	-	-	Drop	0	0	-	-	100	42
=	2	0	2	A	=	1	1	0	1	-	0	-	-	-	Forward	0	0	0	D	100	32

- Matching Rules,
  - Up to 3 conditions of Matching Rules
  - S=0: the packet regards to the current packet, S=1: regards to the state
  - *Operator* field gives the relational operator to be checked against the *Value*

WISE Flow Table (cont.)

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Op.	Size	S	Addr.	Value	Op.	Size	S	Addr.	Value	Op.	Size	S	Addr.	Value	Type	M	S	Addr.	Value	TTL	Counter
=	2	0	2	B	>	2	0	10	x_{Thr}	=	1	1	0	0	Modify	1	1	0	1	122	23
=	2	0	2	B	≤	2	0	10	x_{Thr}	=	1	1	0	1	Modify	1	1	0	0	122	120
=	2	0	2	B	-	0	-	-	-	-	0	-	-	-	Forward	0	0	0	D	122	143
=	2	0	2	A	=	1	1	0	0	-	0	-	-	-	Drop	0	0	-	-	100	42
=	2	0	2	A	=	1	1	0	1	-	0	-	-	-	Forward	0	0	0	D	100	32

- Actions
  - Type field can be “Forward to”, “Drop”, “Modify”, “Send to INPP”, “Turn off radio”
  - *M* specifies whether the entry is exclusive (M = 0) or not (M = 1)
  - The field of Addr. and Value depend on the type of action
  - S: in case of “Modify”, specifies whether executed on the packet or the state.
- Statistics
  - Same as OpenFlow

An Example

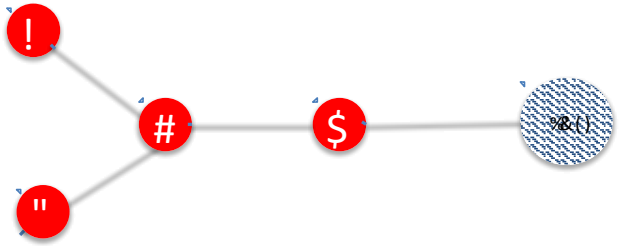


Fig. 4: Exemplary topology.

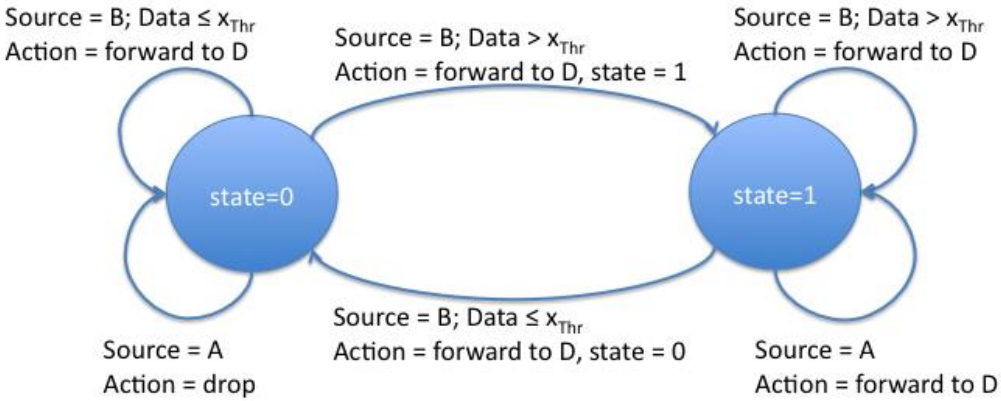
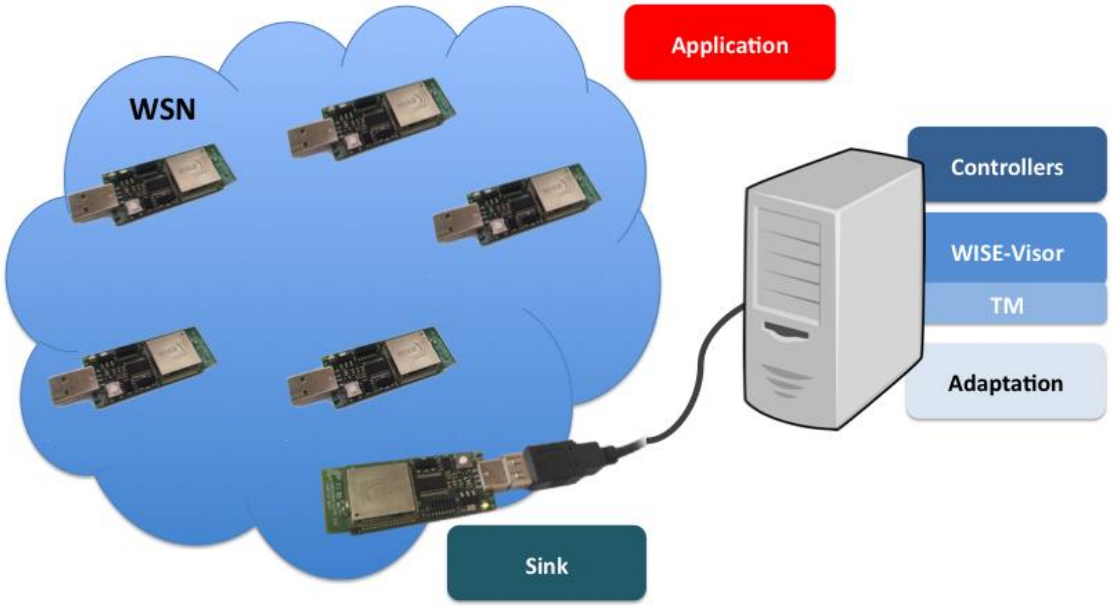


Fig. 5: Finite state machine implementing a policy such that packets generated by A are dropped if the last data measured by B is lower than (or equal to) x_{Thr}.

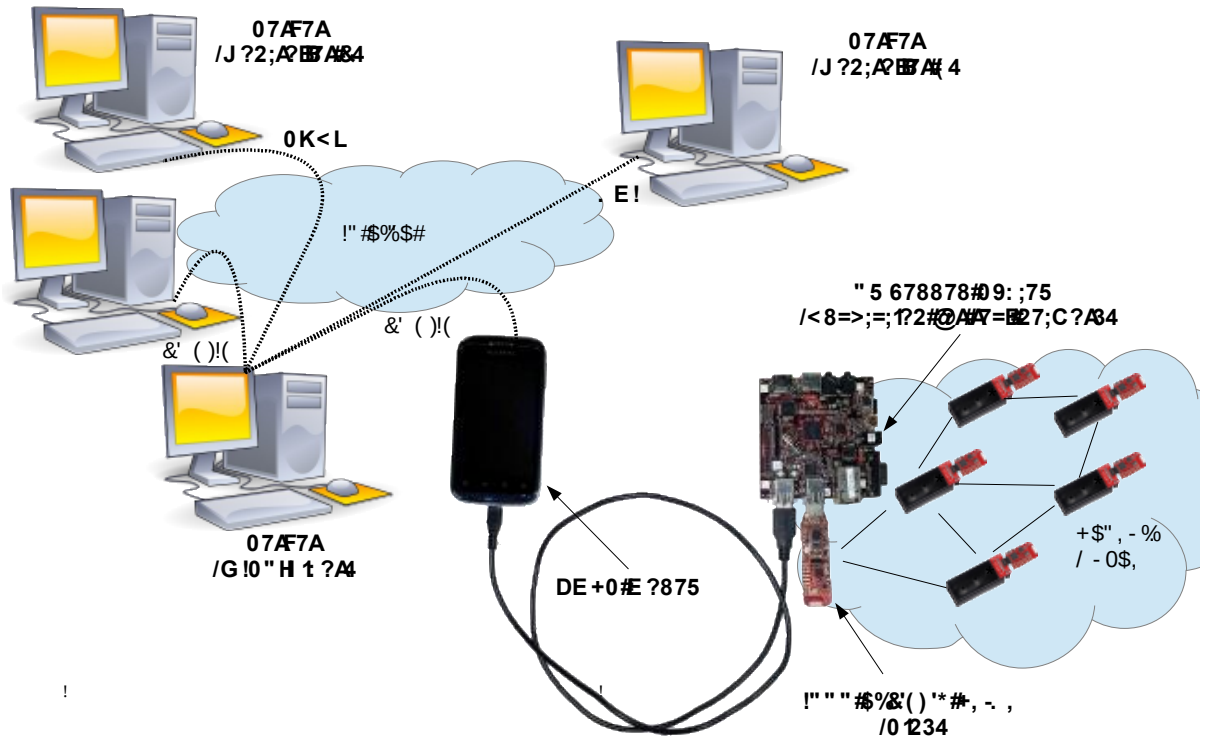
Prototype and Testbed

- Sensor nodes: EMB-Z2530PA is a wireless module developed by Embit for LR-WPAN applications



(a) Simplest deployment option.

A Distributed Deployment



(b) Distributed deployment option.

Performance Evaluation

- Testbed setup
  - 6 nodes (5 sensor nodes and a sink)
  - In each measurement campaign 5000 data packets have been sent, each every 15 seconds
  - Different payload sizes have been considered for such packets (10, 20 and 30 bytes)

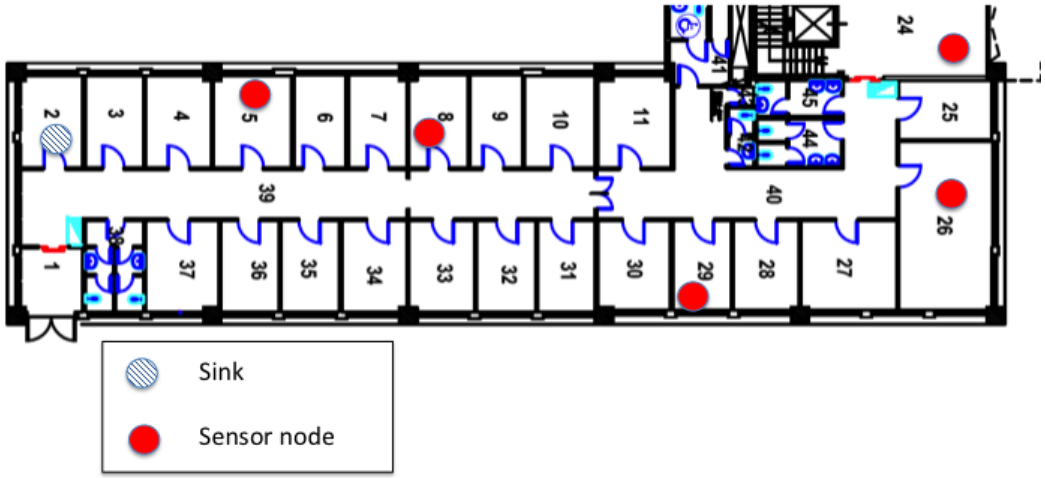


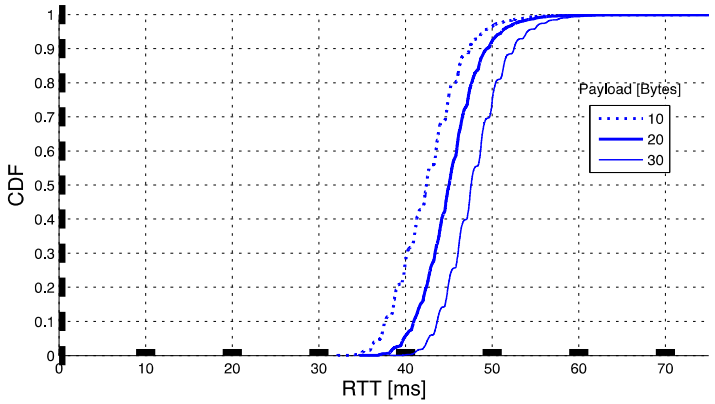
Fig. 8: Nodes deployment.

Performance Metrics

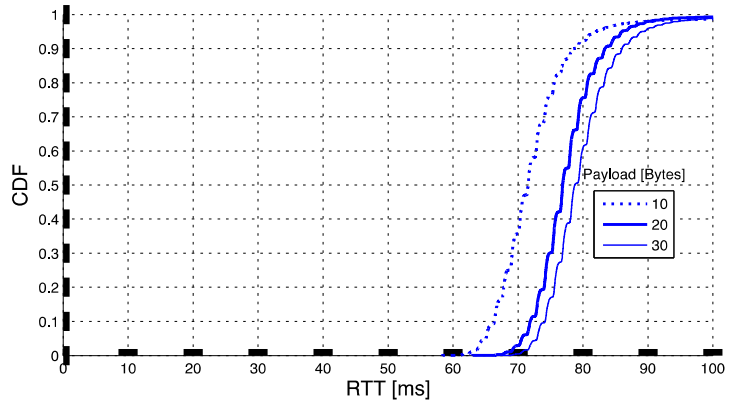
- Round Trip Time (RTT)
  - the time interval between the generation of a data packet and the acknowledgment
- Efficiency
  - the ratio between the number of payload bytes received by the destinations and the overall number of bytes circulating in the network
- Controller response time
  - the duration from the Controller receives a request to send the corresponding entry

Result - RTT

- CDF of RTT in different payload
- RTT increases as the distance and the payload increase



(a) Number of hops = 3.



(b) Number of hops = 5

Fig. 9: CDFs of the RTT for different payload sizes and different distances between the source and destination node.

Result - Efficiency

- The efficiency vs. the payload size for different values of the interval between consecutive transmissions of the TD packets, T

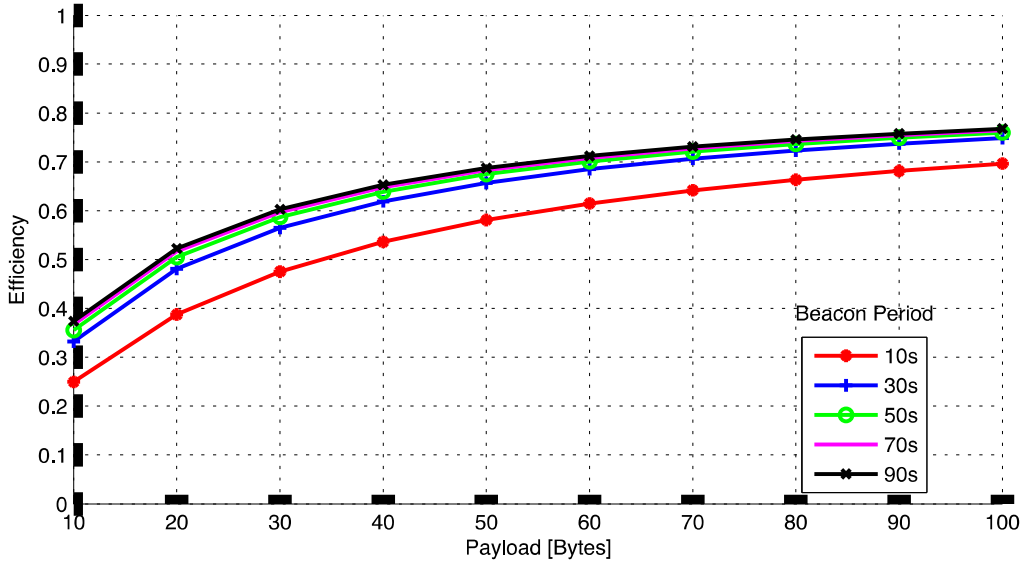
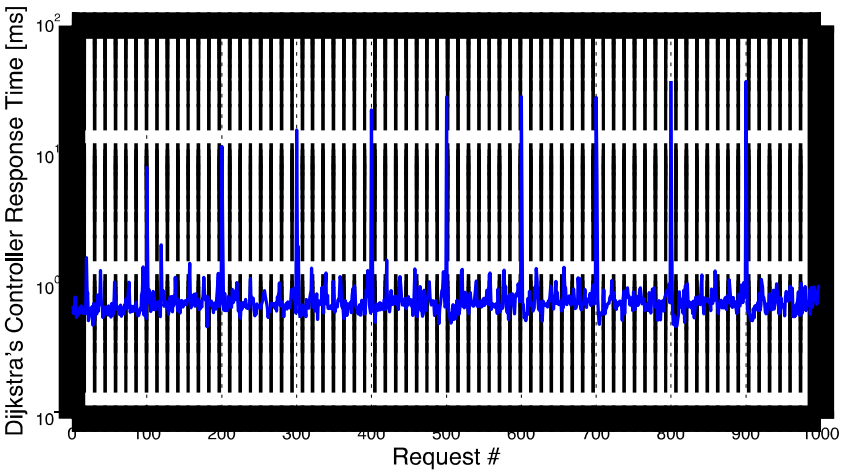


Fig. 14: Efficiency for different values of beacon sending period.

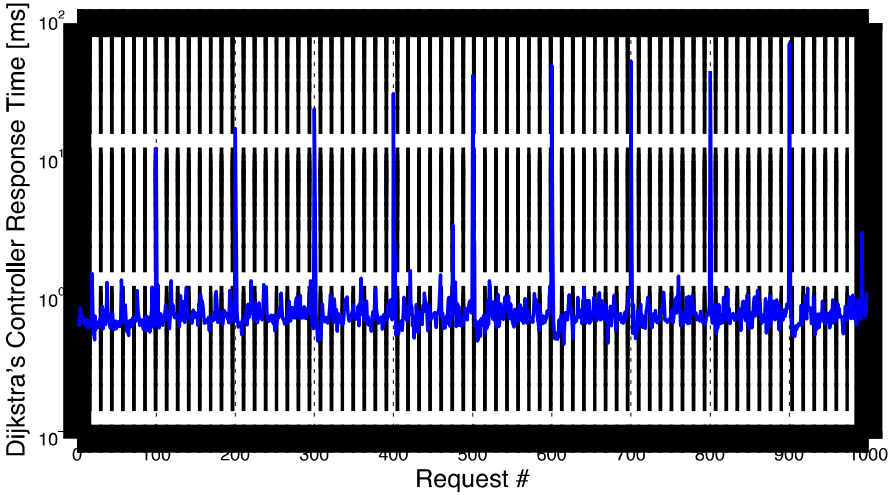
Most of the inefficiency is due to the high ratio between the header size and the payload size

Result – Controller Response Time

- Simulate the process of request generation by the nodes modeling a network consisting of 50, 60, and 70 nodes



(a) 50 Nodes.



(c) 70 Nodes.

Conclusion

- SDN-WISE is stateful and aimed at reducing the amount of information exchanged between sensors and SDN controllers
- Details on the SDN-WISE protocol stack are provided as well as results obtained from extensive measures in a physical testbed

Future Work

- Make the SDN-WISE network resilient to intentional attacks and bugs in the Controller software
- Network coding in SDN-WISE