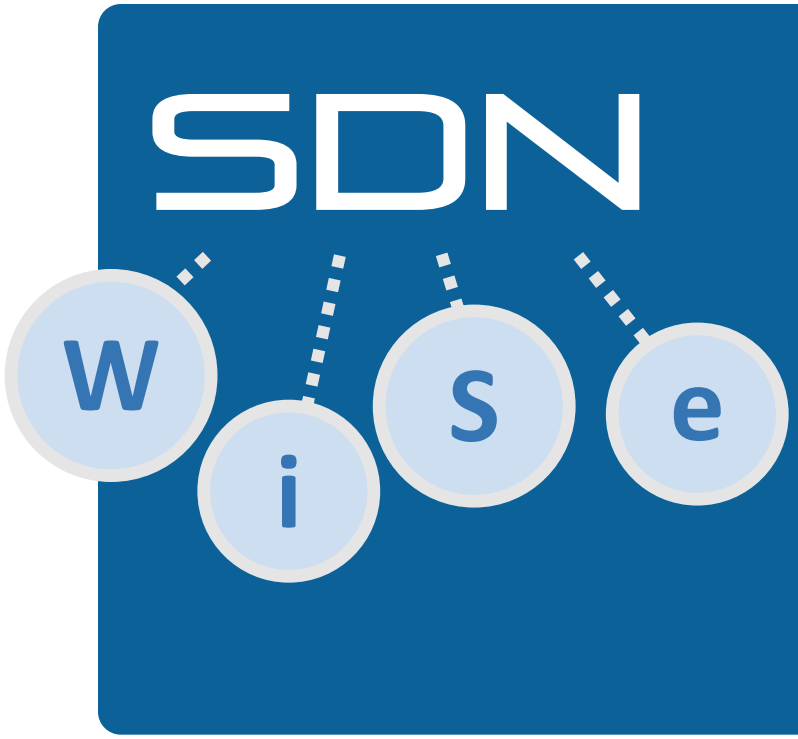


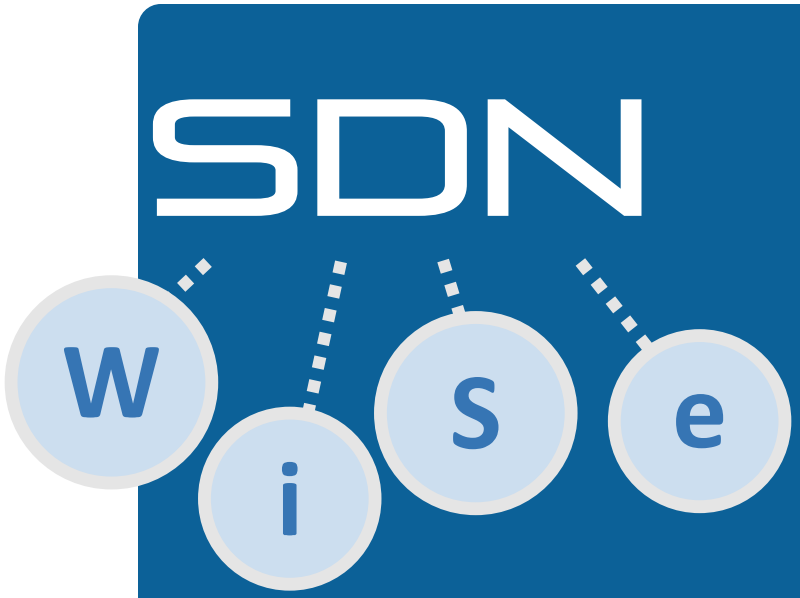
Exploiting *State* Information to Support QoS in Software-Defined WSNs

Paolo Di Dio*, Salvatore Faraci*, Laura Galluccio‡, **Sebastiano Milardo**†, Giacomo Morabito‡, Sergio Palazzo‡, and Patrizia Livreri†

*CNIT Research Unit Catania, Italy
‡University of Catania, Italy
†University of Palermo, Italy

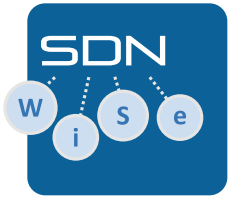
Outline

- Motivation
- Related Work
- Proposed Solution
- Simulations and Results
- Conclusions



Motivations

Motivations



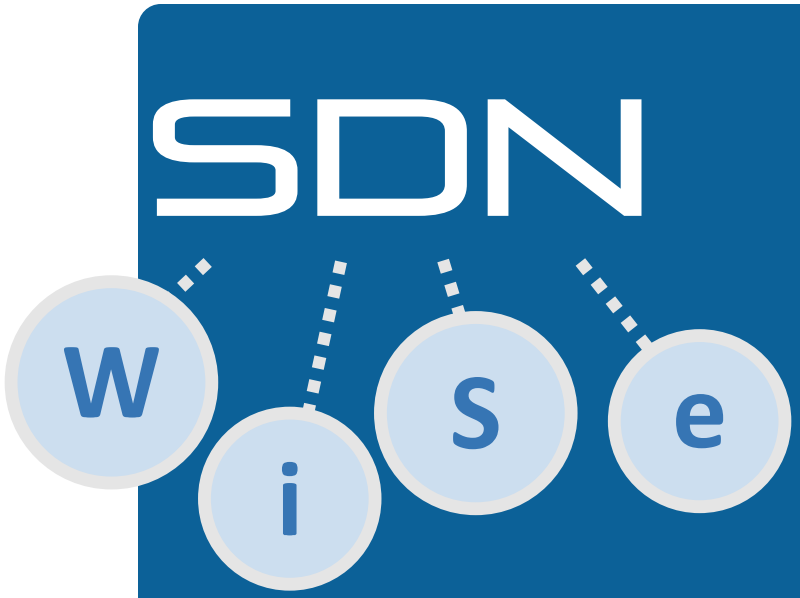
- Many WSNs deployed around the world
- The deployment is easier compared to wired networks…
- …but the management is harder!
- Different kind of data should be managed in different ways

Motivations

- IEEE Xplore results:
  - QoS in wired SDN networks: 173
  - QoS in wireless infrastructured SDN networks: 43
  - QoS in wireless infrastructureless SDN networks: none (up to now)

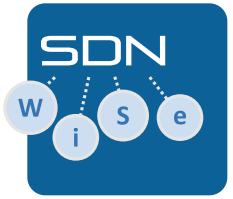
Proposed contribution

- To this purpose, we exploit the *state* information envisioned by SDN-WISE. In fact, state can represent the level of congestion of the node and can be used in a twofold manner:
  - Assign different packet drop probabilities to different traffic flows depending on the current level of congestion of a node;
  - Inform the Controller about the current level of congestion of a node so that it can calculate alternative rules for traffic flows in order to mitigate congestion.



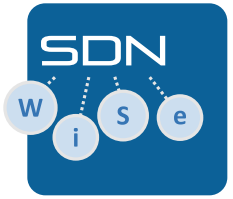
Related Works

SDN & OpenFlow



- Software Defined Networking (SDN) clearly separates:
  - **Data plane:** run by network Switches
  - **Control plane:** implemented by a software program running on a server (the Controller)
- Modifying the behavior of the network as easy as it is installing a new piece of software on a PC
- **OpenFlow** is the most popular implementation of the SDN paradigm
- **Flow Rules:** matching window, actions, stats

QoS in SDN & SDWN

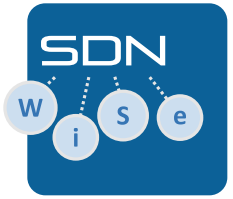


- Few papers targeting QoS support in wired SDN scenarios:
  - OpenQoS,
  - QoSFlow,
  - PolicyCop.
- QoS in Software Defined Wireless Networks (SDWN):
  - *Ethanol*, for 802.11 Wireless Networks

QoS in WSN

- The QoS support mechanisms developed for wired networks and traditional wireless networks cannot be applied in WSNs because usually they are too complex.
- Thus many of the works on this topic focus on the integration between the Application and the Network layer, while others focus on the MAC layer only.

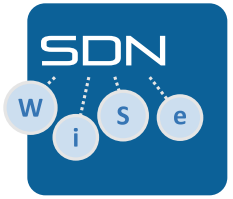
SDN in WSNs



- Few attempts to extend SDN to WSNs:
  - Software Defined Wireless Networks (SDWN), 2012
  - Sensor OpenFlow, 2012
- Different requirements:

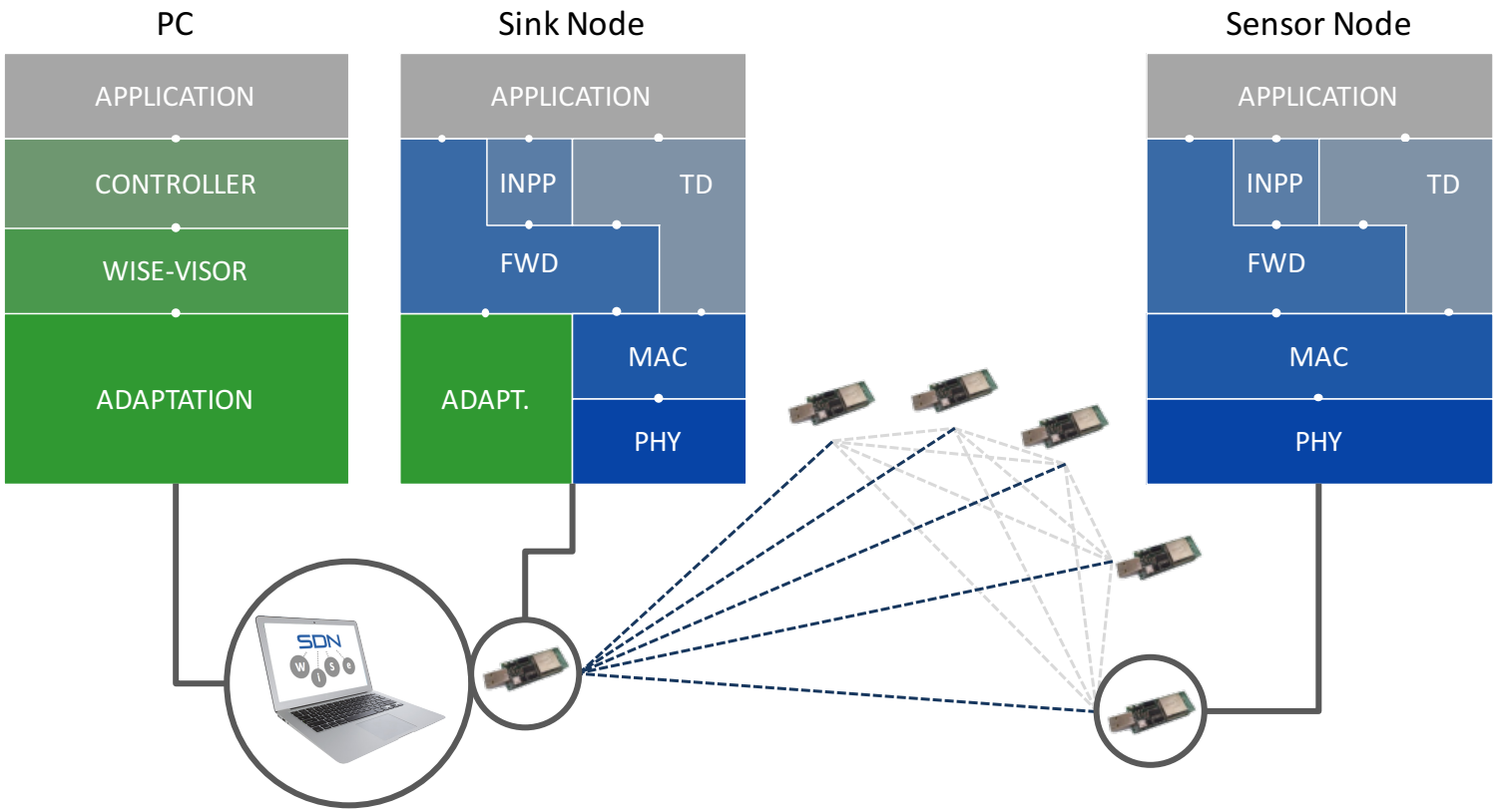
Traditional wired networks	WSNs
Velocity	Efficiency
	Flexibility
	Memory occupancy

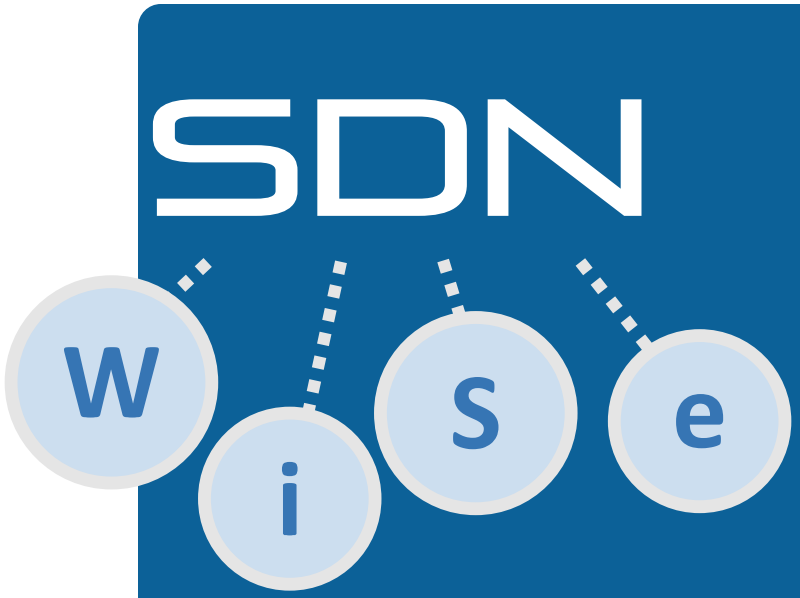
SDN-WISE: Basic concepts



- Directly derived by OpenFlow
- Separation (even physical) between
  - **data plane** (executed by sensor nodes)
  - **control plane** (executed by the Controller)
- When an event (e.g., the arrival of a packet) occurs sensor nodes behave as specified in the WISE Table
- If there is no relevant information in the WISE Table → Ask the Controller
- The Controller replies sending a new entry for the WISE Table
- A simple protocol defined to allow nodes to:
  - Learn the shortest path towards the (closest) sink(s)
  - Discover the neighboring nodes
  - Periodically report local information to the Controller (through the sink)
- **SDN-WISE is Stateful**

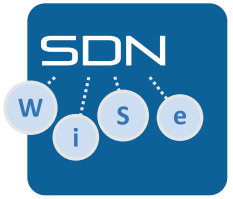
SDN-WISE: Architecture





Proposed Solution

Basic Concepts



- A state variable is used to represents the congestion of a node
- Diversify the handling depending on the congestion of the node and the priority level of the packet
- The Controller will provide all the rules needed
- QoS using Drop

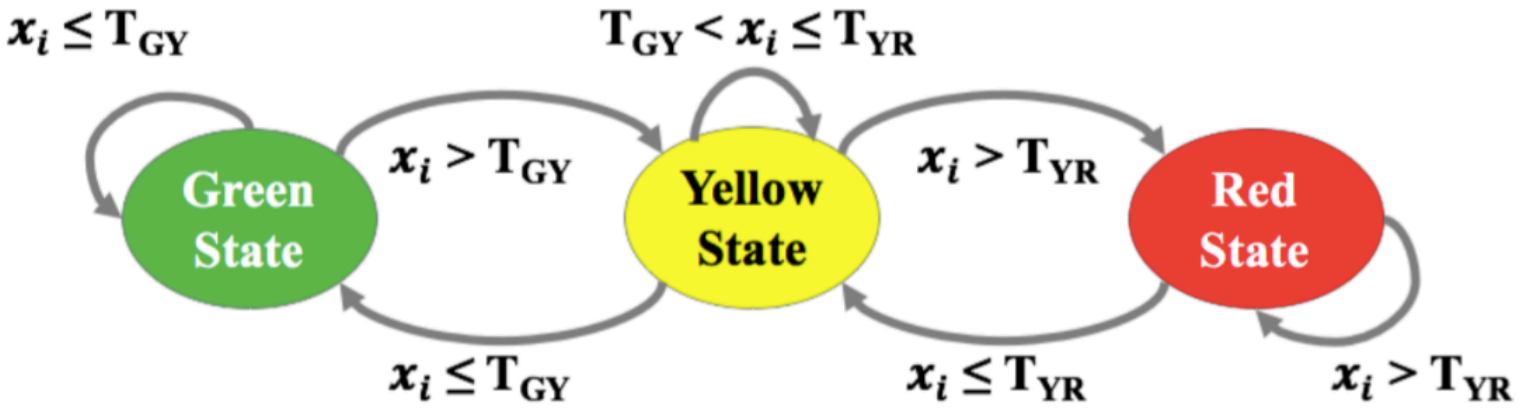
Load Balancing

- New Report Message

	Bit 0-7	Bit 0-7
Byte 0-9	SDN-WISE Header	
10	No. Hop	Battery Level
12	Congestion Level	N
14	$Address_1$	
16	$RSSI_1$	...
18	...	
...	$Address_n$	
...	$RSSI_n$	

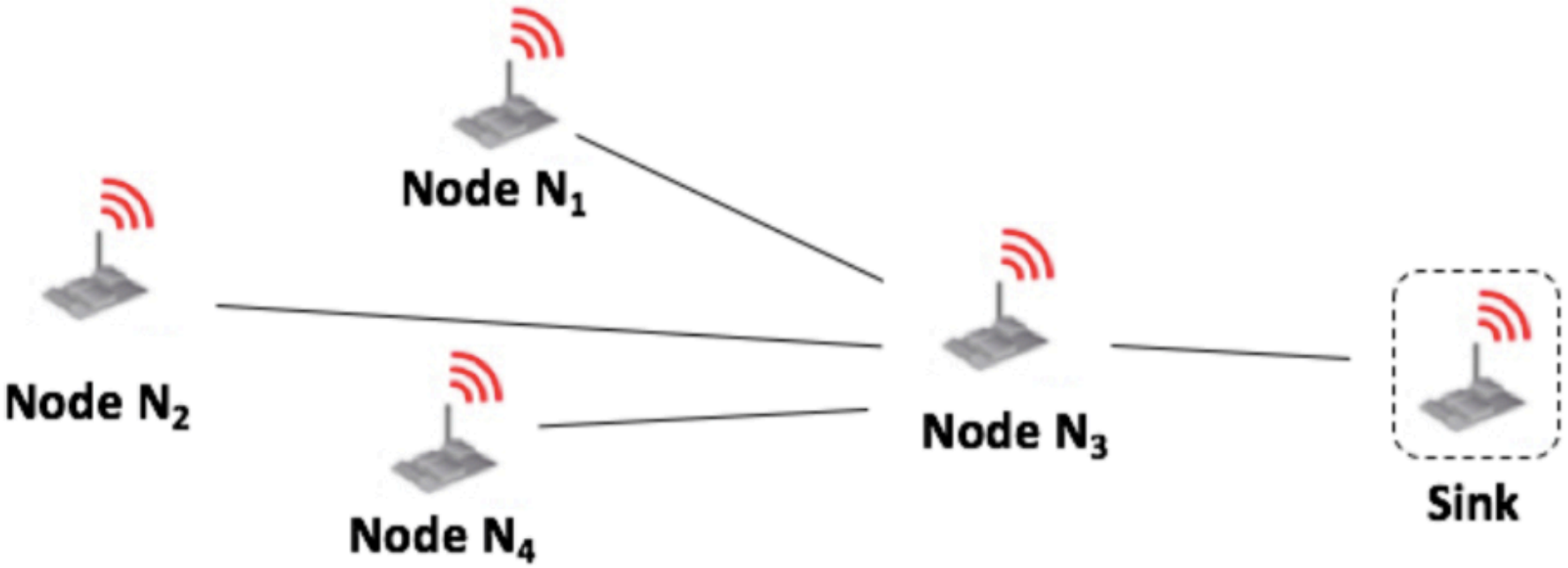
State of a node

- Thresholds on **TX buffer size**
- Dropping policies
  - Green: No Drop
  - Yellow/Red: drop probability is inversely propotional to the priority of the traffic flow



Simple example

- Network of 5 nodes



Details

- An example of a SDN-WISE flow table

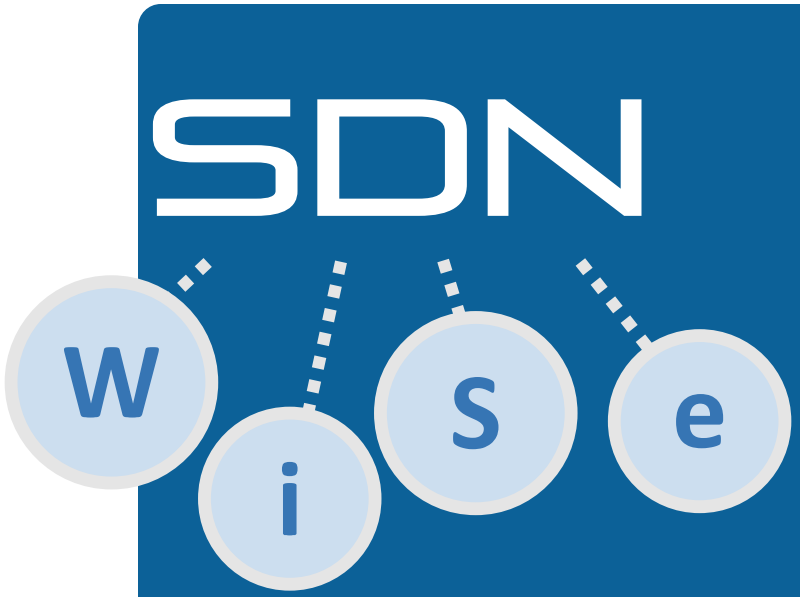
Matching Rules	Action
PACKET [SRC_ADDRESS] == N_1 and STATE_ARRAY [0] == RED and PACKET [PRIORITY_LEVEL] == C_1	DROP (10%, $Sink$)
PACKET [SRC_ADDRESS] == N_2 and STATE_ARRAY [0] == RED and PACKET [PRIORITY_LEVEL] == C_3	DROP (80%, $Sink$)

Estimation issues

- Holt Exponential smoother

$$x_i = \alpha x_{i-1} + (1 - \alpha) b_i$$

- b_i = instantaneous value of the buffer occupancy
- α = is a coefficient, in the range between 0 and 1, that characterizes the degree of filtering fluctuation. if α is low, fluctuations are not filtered and viceversa



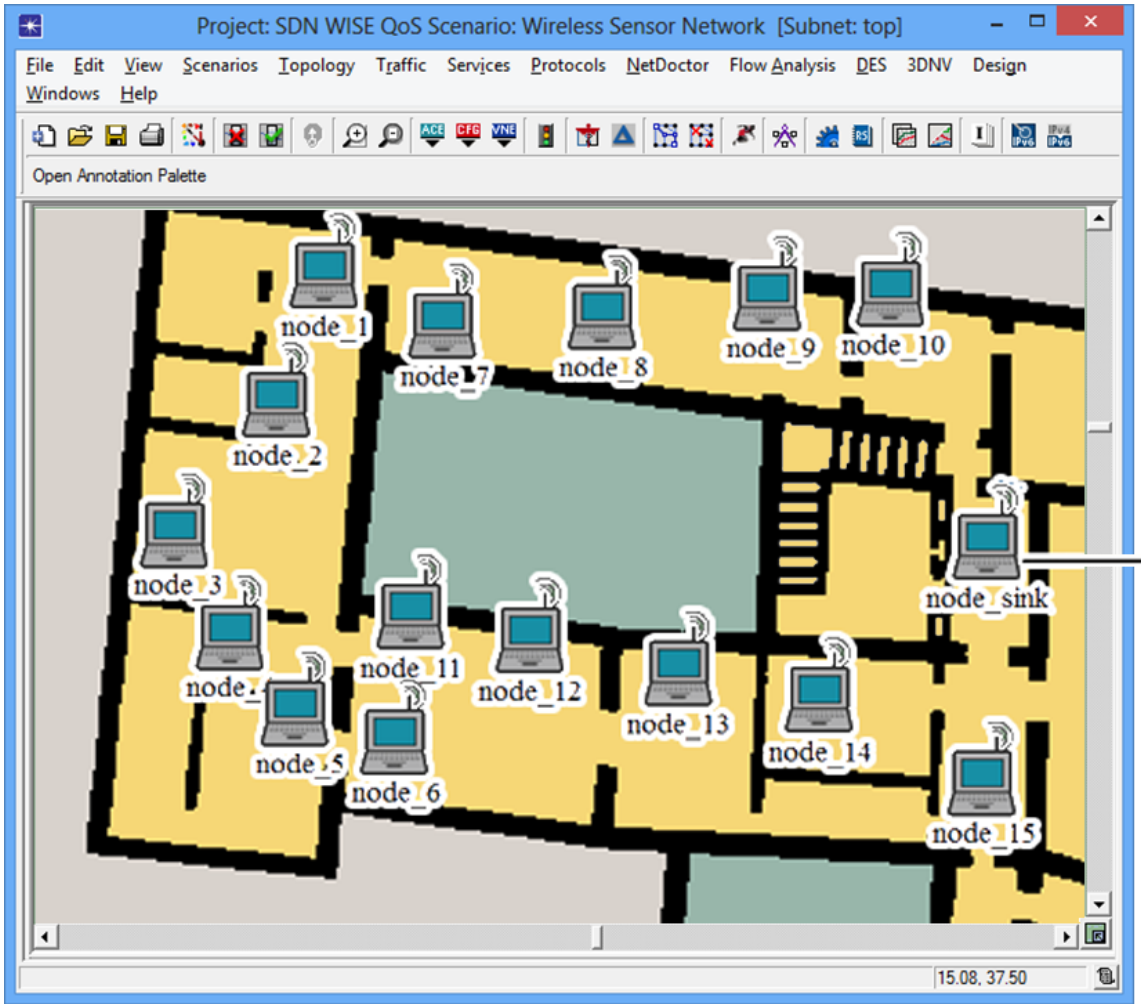
Simulations and Results

Simulation Campaign

- OPNET (16 node) + Controller + HLA
- Store max 120 packets
- Transitions
  - T_{GY} = 65 and T_{YR} = 85
  - T_{GY} = 75 and T_{YR} = 95
  - T_{GY} = 85 and T_{YR} = 105

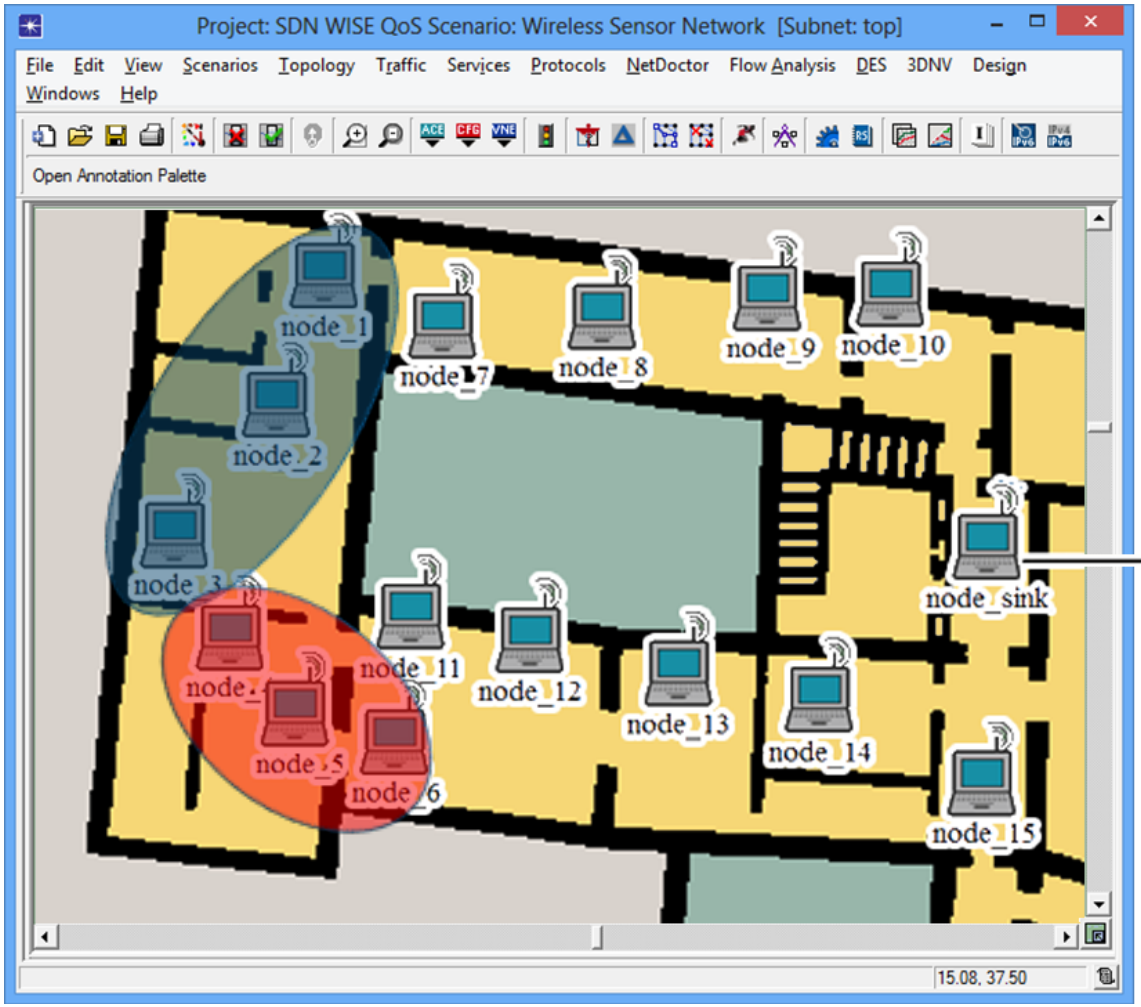
Drop probabilities

			Option 1	Option 2
	Green State	Yellow State	Red State	Red State
C_1	NO DROP	1%	5%	10%
C_2	NO DROP	3%	20%	45%
C_3	NO DROP	5%	40%	80%



Controller

HLA
(High Level Architecture)



Controller

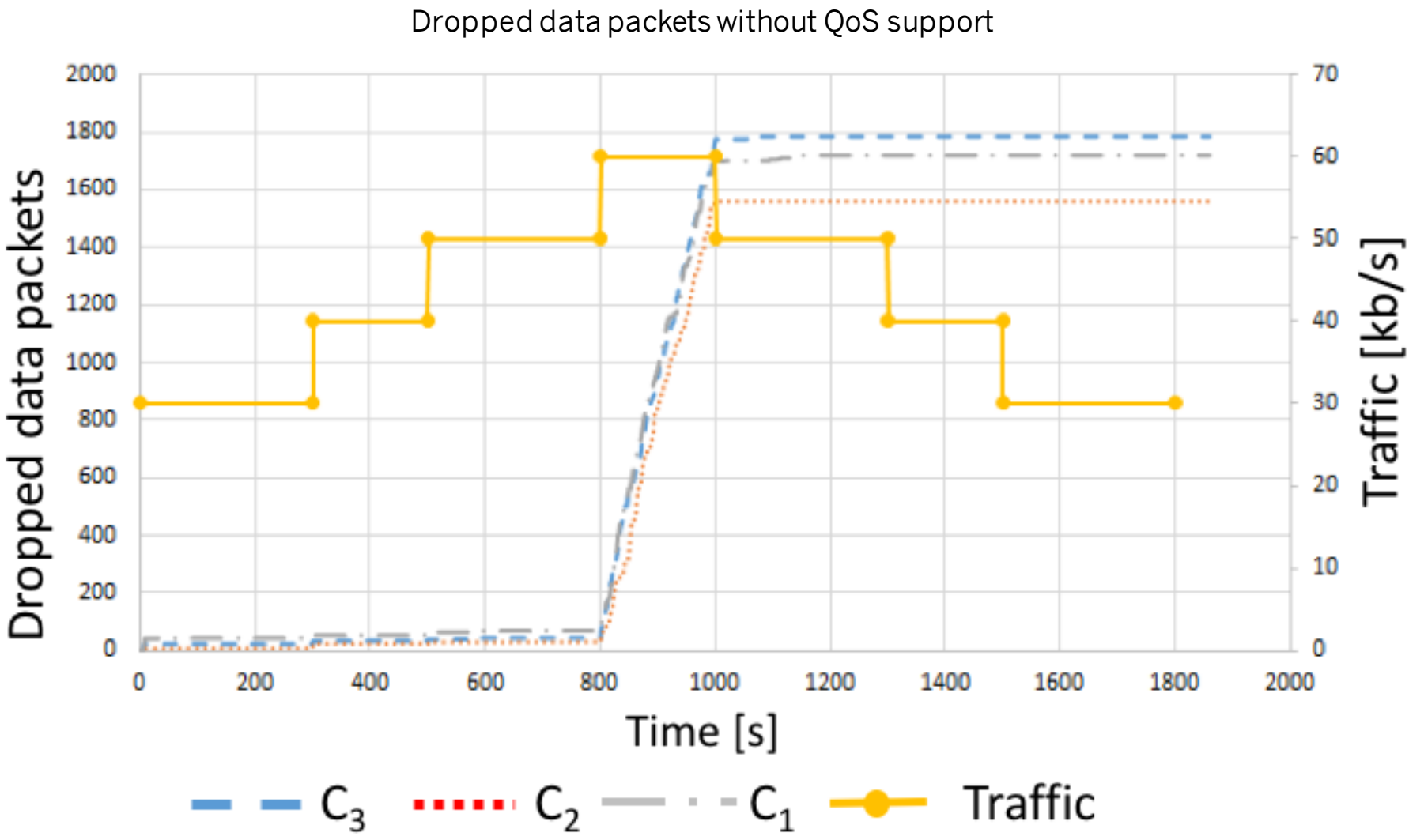
HLA
(High Level Architecture)

Simulations

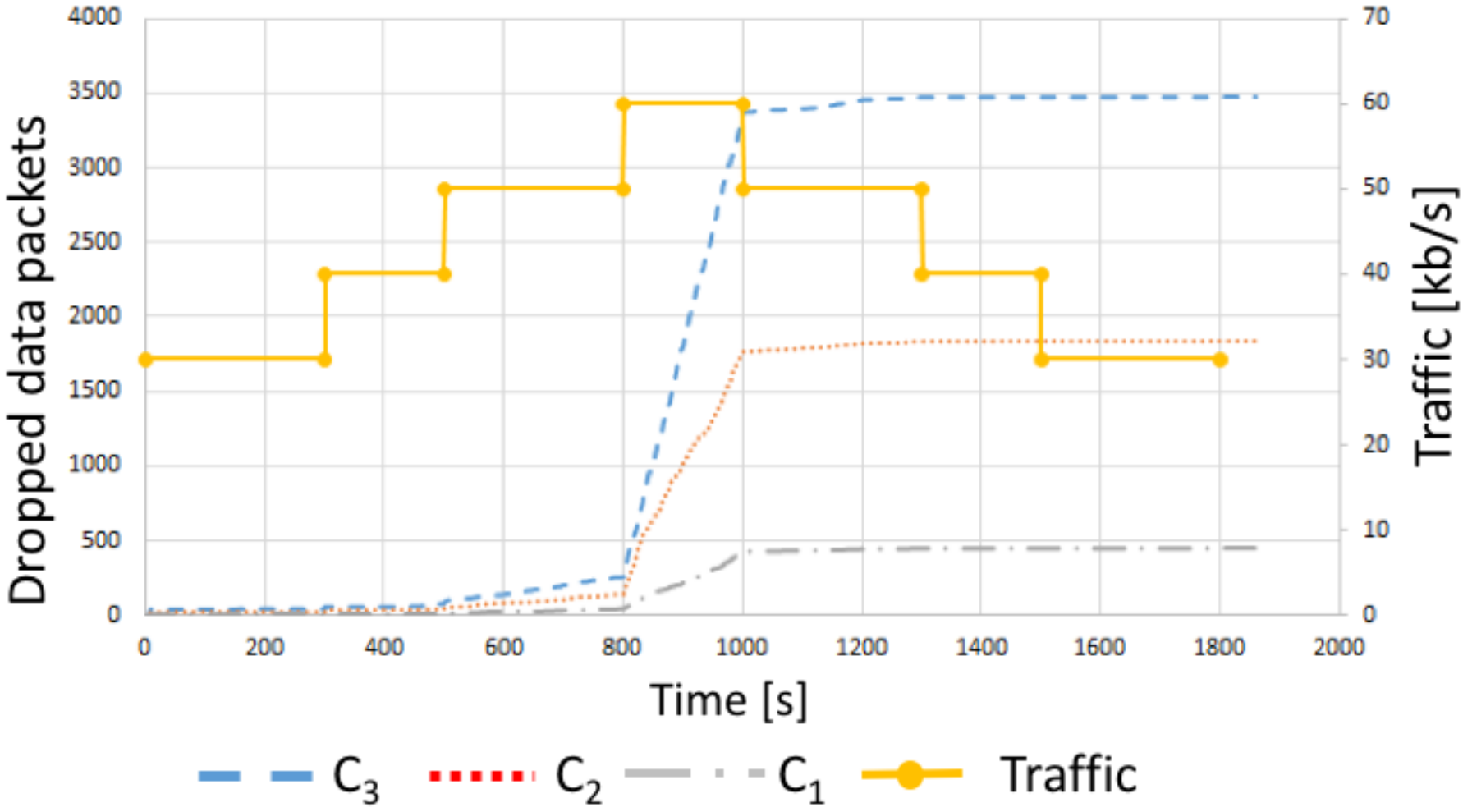
- node1, node2, node3 generate a traffic of 10 kb/s with priority level C1 , C2 , C3 , respectively, from the beginning to the end of the simulation time,
- node4 generates a traffic of 10 kb/s with priority level C1 from time 300 s to time 1500 s,
- node5 generates a traffic of 10 kb/s with priority level C2 from time 500 s to time 1300 s,
- node6 generates a traffic of 10 kb/s with priority level C3 from time 800 s to time 1000 s.

Simulations

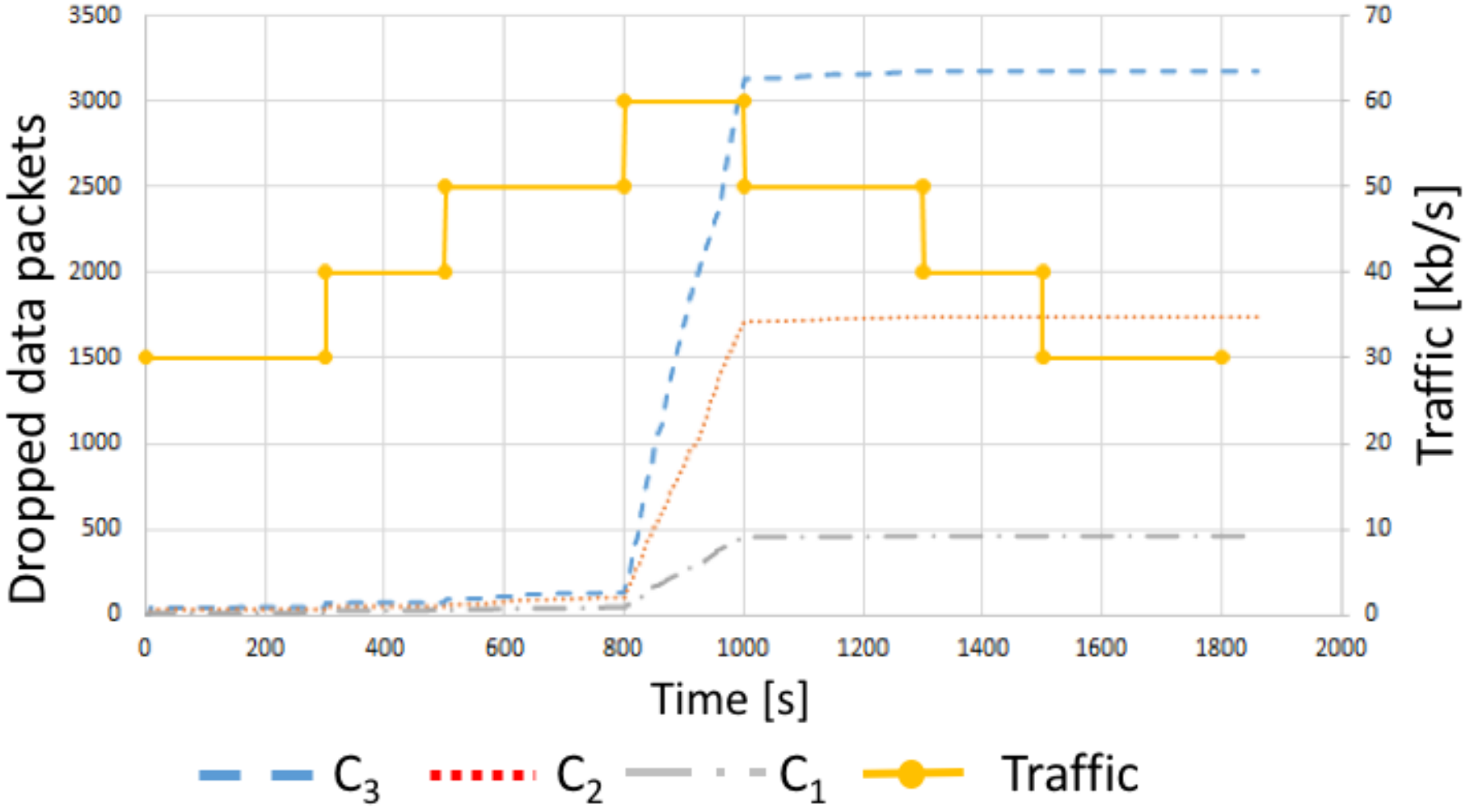
- No QoS
- QoS + No Dynamic Update
- QoS + Dynamic Update


Dropped data packets without QoS support
Dropped data packets
Traffic [kb/s]
Time [s]
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
10
20
30
40
50
60
70
C3
C2
C1
Traffic

Dropped data packets 65 – 85 (No Dynamic Update)

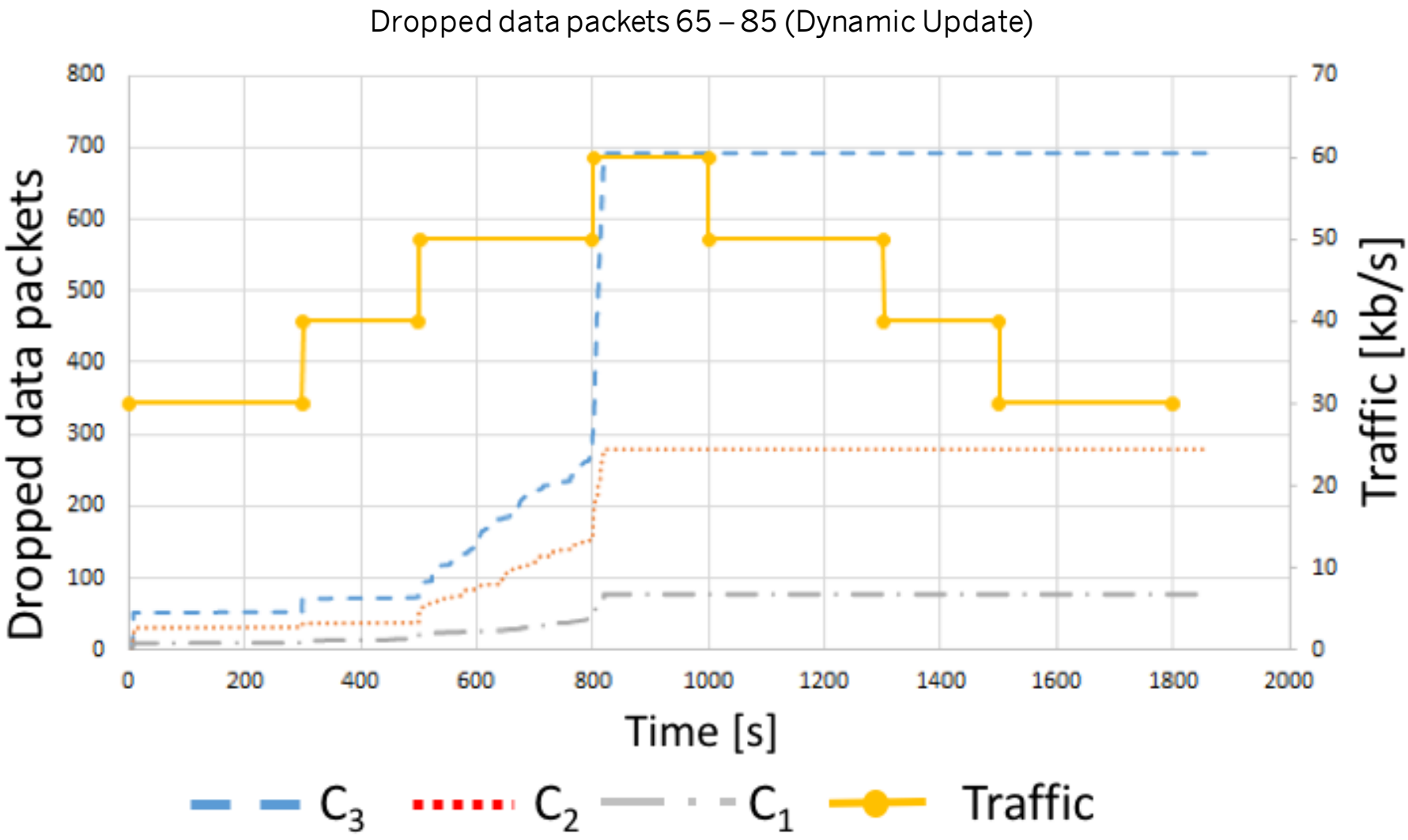


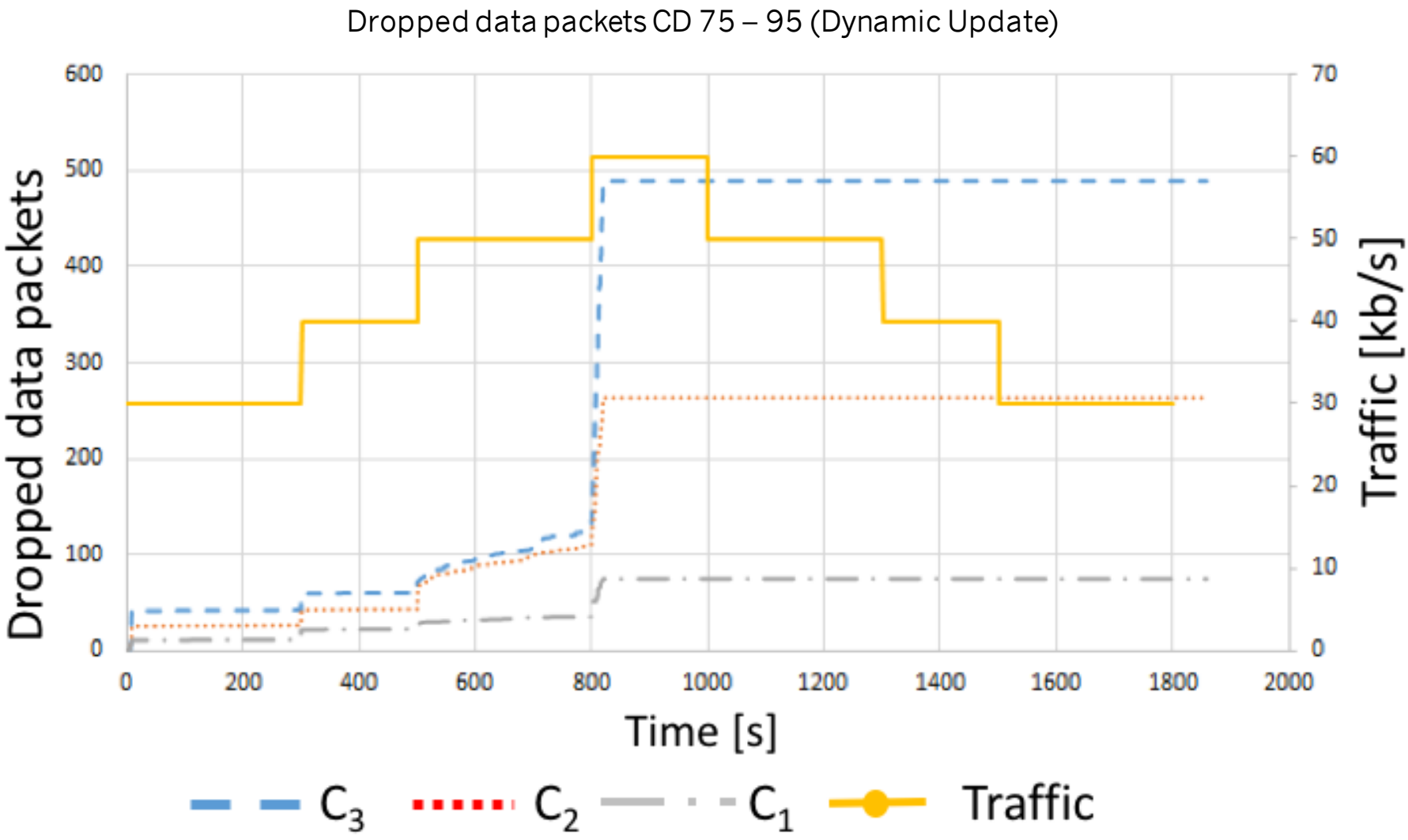
Dropped data packets 75 – 95 (No Dynamic Update)

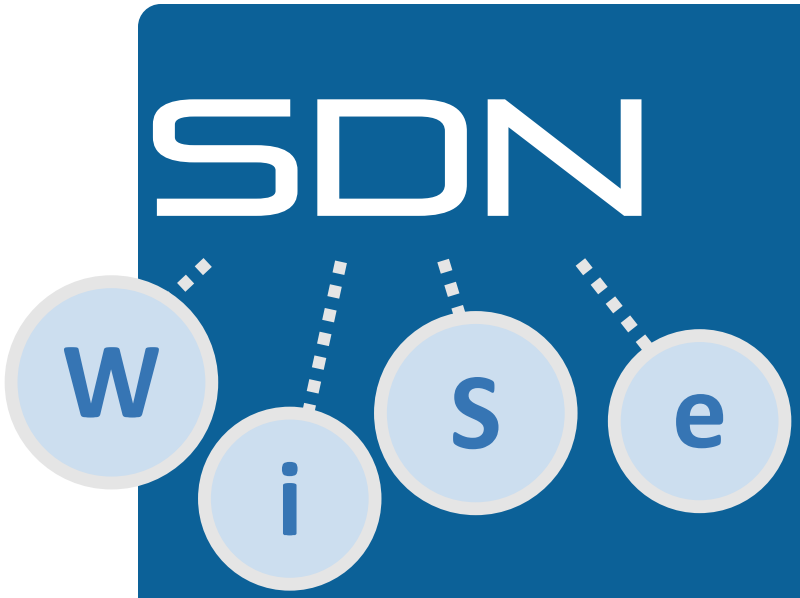


Results

	No QoS support	WCD low drop probability	WCD high drop probability		
Thres.	-	75-95	65-85	75-95	85-105
C_1	16.23 %	9.39 %	3.81%	4.17%	6.46%
C_2	16.96 %	15.65 %	16.16%	16.01%	16.44%
C_3	15.97 %	25.33%	30.93%	29.77%	26.26%


Dropped data packets 65 – 85 (Dynamic Update)
Dropped data packets
Traffic [kb/s]
Time [s]
0
100
200
300
400
500
600
700
800
0
10
20
30
40
50
60
70
0
200
400
600
800
1000
1200
1400
1600
1800
2000
C3
C2
C1
Traffic


Dropped data packets CD 75 – 95 (Dynamic Update)
Dropped data packets
Traffic [kb/s]
Time [s]
0
100
200
300
400
500
600
0
10
20
30
40
50
60
70
0
200
400
600
800
1000
1200
1400
1600
1800
2000
C3
C2
C1
Traffic



Conclusions

Conclusions

- We have introduced a mechanism that exploits the *stateful* nature of SDN-WISE to support differentiated levels of QoS in WSNs.
- The mechanism is based on the usage of state to give information about the congestion condition at the nodes.
- Each node, as shown by simulations, is able to handle traffic flows with different levels of QoS in different ways.
- Simulation results assess the effectiveness of the proposed solution to handle QoS.

THANK YOU