Reliable Dynamic Packet Scheduling over Lossy Real-Time Wireless Networks

Tao Gong[1], Tianyu Zhang[2,3], Xiaobo Sharon Hu[3], Qingxu Deng[4], Michael Lemmon[3], Song Han[1]

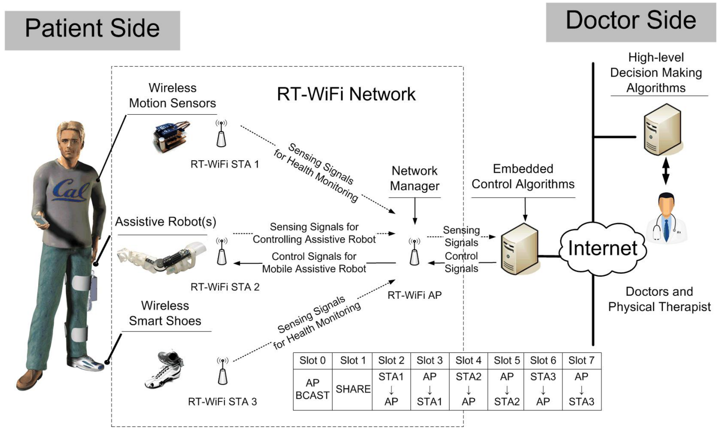
[1] University of Connecticut, Storrs, US

[2] Qingdao University, China

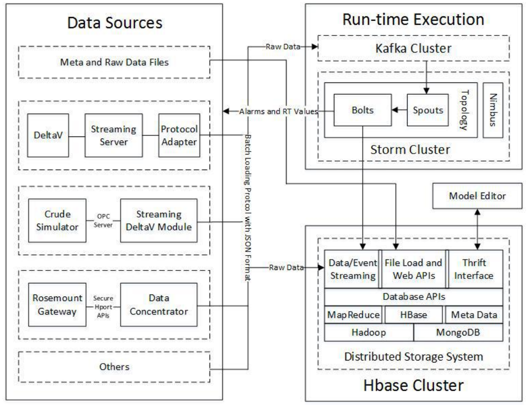
[3] University of Notre Dame, US

[4]Northeastern University, China

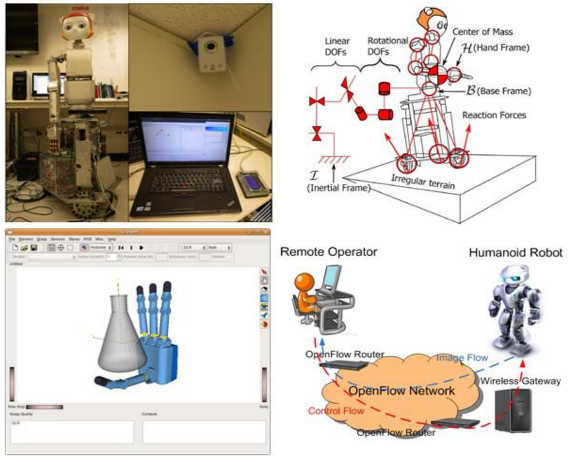
Real-Time Wireless Networks (RTWNs)



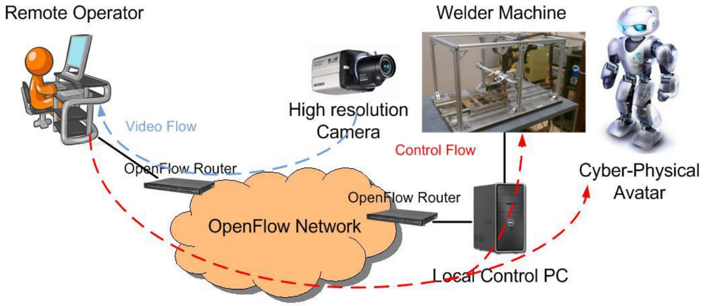
Network-based Rehabilitation System



Real-Time Analytics Platform for Process Control

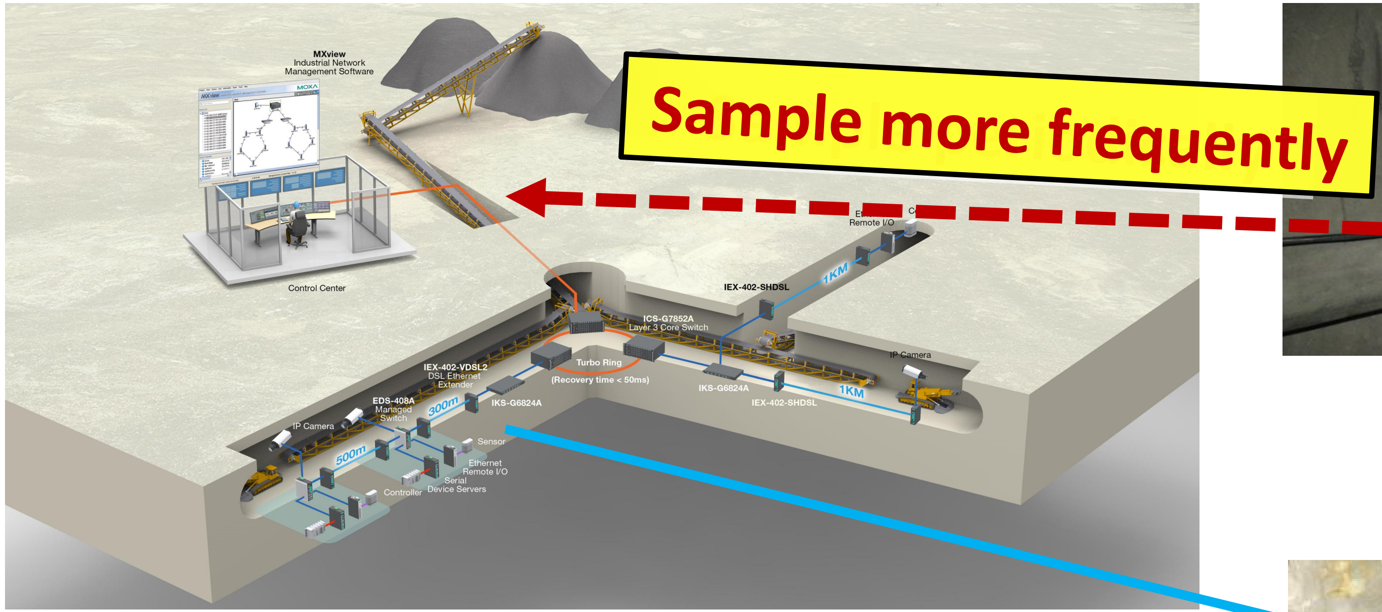


Cyber-physical Avatar



Remote and Real-time Welding System

An Example: Mining Monitoring System



- Packet delivery needs to be guaranteed
- External disturbance: unexpected changes in temperature/pressure, etc.
- Require more frequent monitoring/response

Problem and Challenges

- Requirements of a RTWN:
  - Environment data must be collected timely
  - Guaranteed data freshness and delivery
- QoS: how well it satisfies real-time deadlines and delivers packet
  - Packet scheduling is critical
- Challenges:
  - Lossy wireless link
  - External disturbances occur unexpectedly

What We Want to Achieve?

- Design a reliable on-line dynamic scheduling framework

- During nominal operation:
  - Determine a reliable static schedule
  - Each packet can be reliably delivered before its deadline

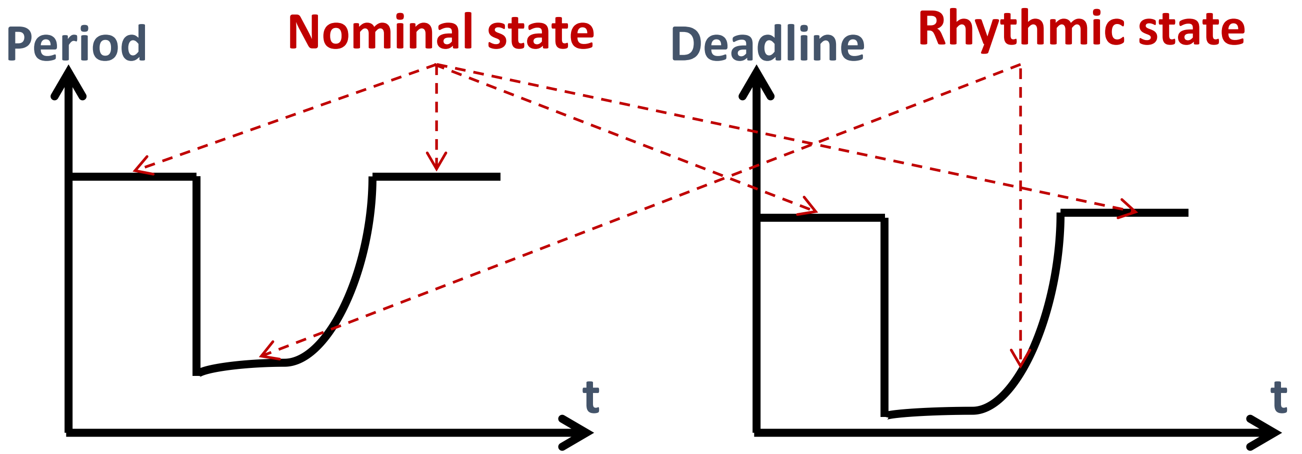
- Upon detecting a disturbance
  - Generate and distribute a temporary dynamic schedule
  - Deadlines and delivery of critical tasks are guaranteed
  - Minimize the impact to the rest of the tasks

Outline

- Motivation
- System Model
- Reliable Real-Time Dynamic Packet Scheduling
- Experimental Evaluation
- Conclusion

Rhythmic Task Model

- When nothing happens
  - All tasks follow regular periods
- When disturbance occurs
  - The corresponding task follows a specific release pattern

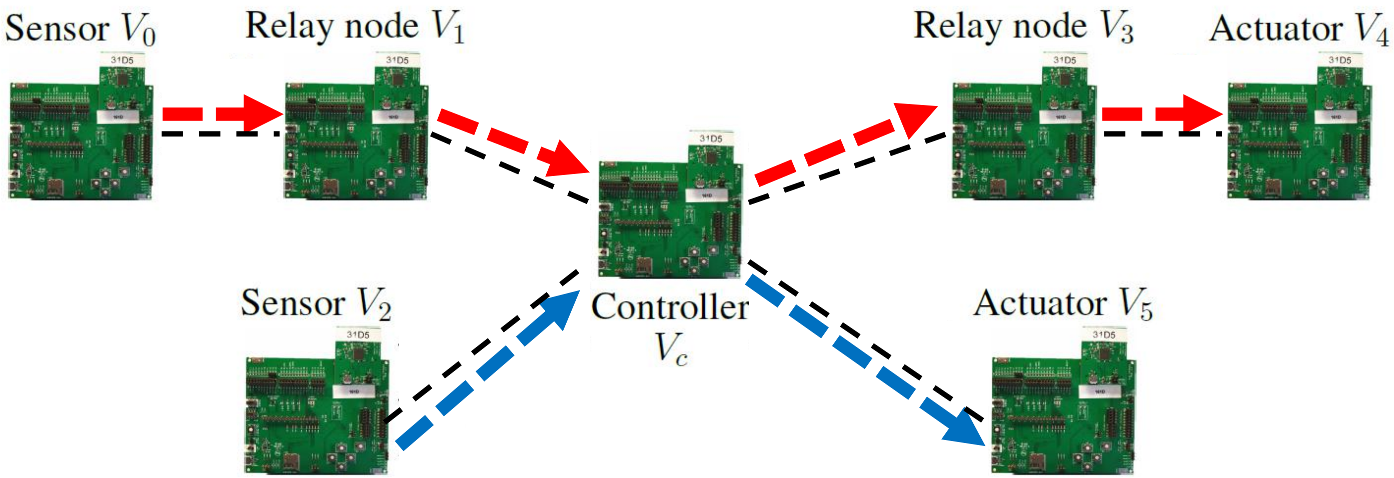


In general, any given pattern can be used in the rhythmic state

J. Kim, K. Lakshmanan and R. Rajkumar, *ICCPS*, 2012

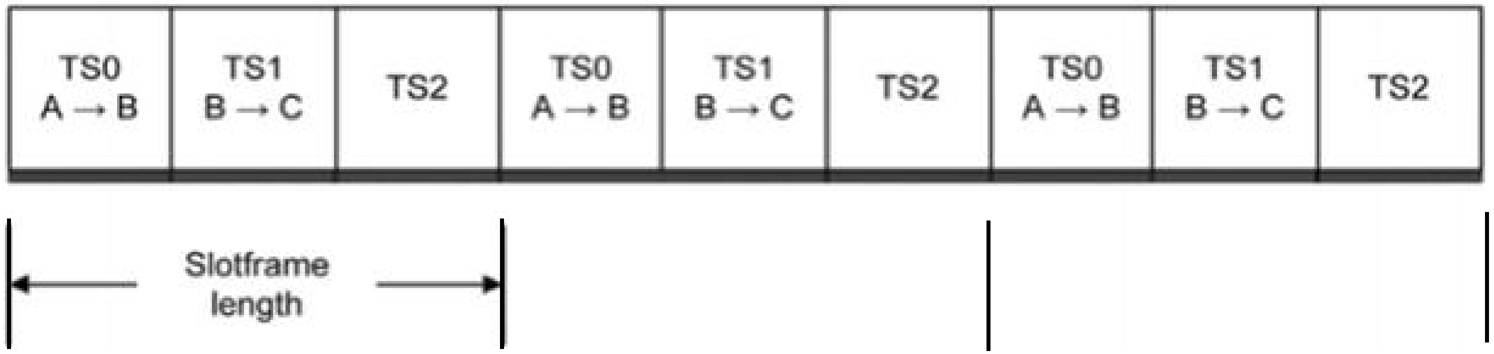
System Model

- Real-Time Wireless Network (RTWN) infrastructure
  - A controller, sensors, relay nodes and actuators sharing a channel
  - Nodes have computing capability
  - Link reliability is measured as link packet delivery ratio (PDR)
- Task model
  - Broadcast task and unicast tasks (periodic and rhythmic) release infinite packets
  - Each packet involves multiple transmissions, from a sensor to an actuator

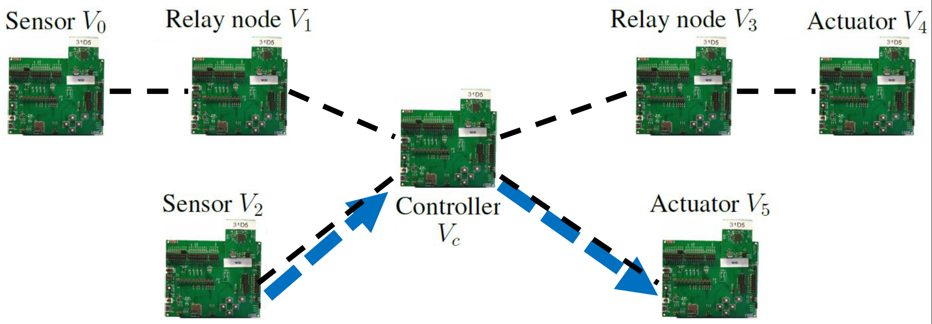


System Model: Communication Model

- Time-Division Multiple Access (TDMA) network
- Time slot (TS), slot
- Slotframe



Scheduling Model



- Transmission-based Scheduling (TBS)
  - A time slot is assigned to a transmission of a packet from a task
  - Sender and receiver(s) of the transmission will operate their RF modules

Slot 1	Slot 2	Slot 3	Slot 4
(1, 1)	(1, 1)	(1, 2)	(1, 2)

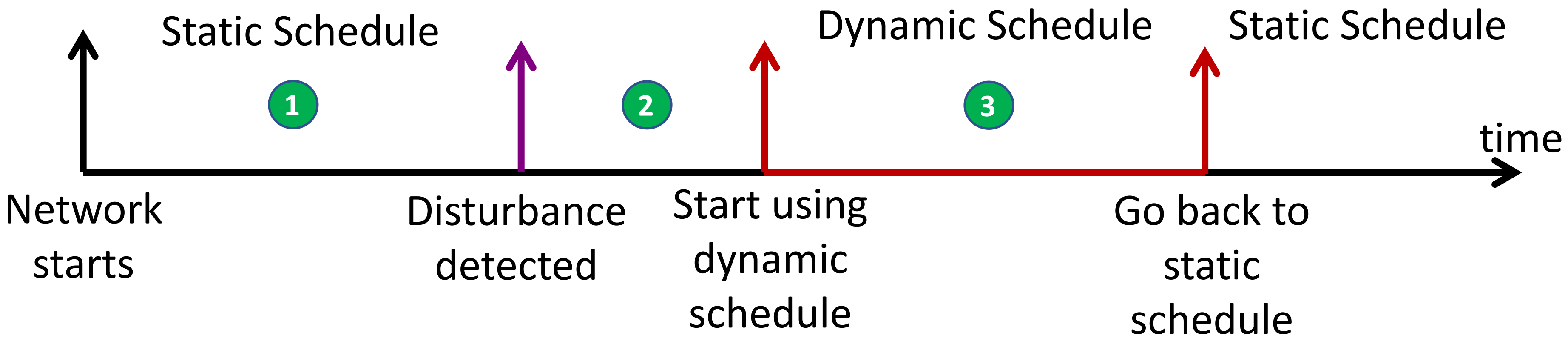
Assign 2 retries for each transmission (x, y) mean the y-th transmission of the packet from task x.

- Packet-based Scheduling (PBS)
  - A time slot is assigned to a packet of a task
  - Involved nodes decide radio operation (TX, RX, idle) at runtime

Slot 1	Slot 2	Slot 3	Slot 4
(1)	(1)	(1)	(1)

Assign 4 time slots for this packet

Problem Overview

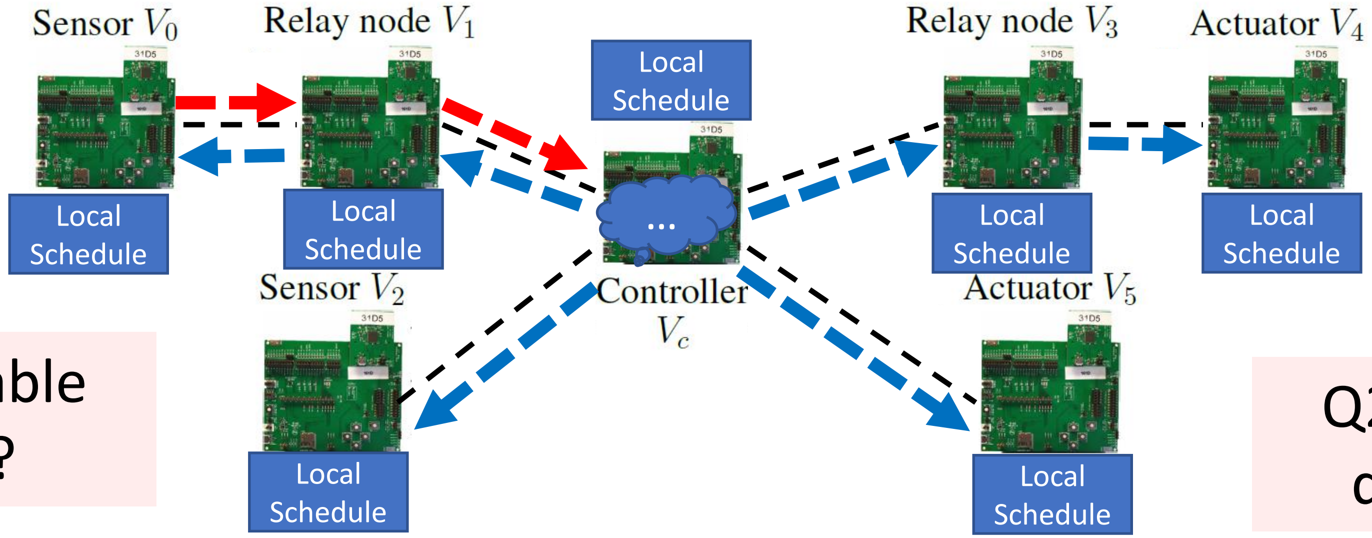


- No disturbance
  1. Determine a reliable static schedule.
- Upon detection of a disturbance, determine a dynamic schedule
  2. Collect disturbance information and distribute dynamic schedule.
  3. All rhythmic packets meet their deadlines and are reliable.
     Delivery of the periodic packets may be degraded

Outline

- Motivation
- System Model
  - Transmission-Based Scheduling (TBS)
  - Packet-Based Scheduling (PBS)
- Reliable Real-Time Dynamic Packet Scheduling (RD-PaS)
- Experimental Evaluation
- Conclusion

RD-PaS Overview



Q1. What is a reliable static schedule?

Q2. What is a desired dynamic schedule?

Network starts: Each node generates consistent schedule locally

Disturbance detected:
1. Rhythmic request
2. Determine degraded packets
3. Broadcast the degradation information

Start using dynamic schedule: Each node generates consistent dynamic schedule using degradation info

Go back to static schedule: Each node generates consistent schedule

time

Reliable Static Schedule

- Problem: Given link PDR (λ^L), task set T, determine the static schedule such that:
  - Reliability Constraint: All packets are scheduled such that end-to-end PDR (λ) meets the required reliability λ^R.
  - Timing Constraint: All packets are scheduled before their deadlines.

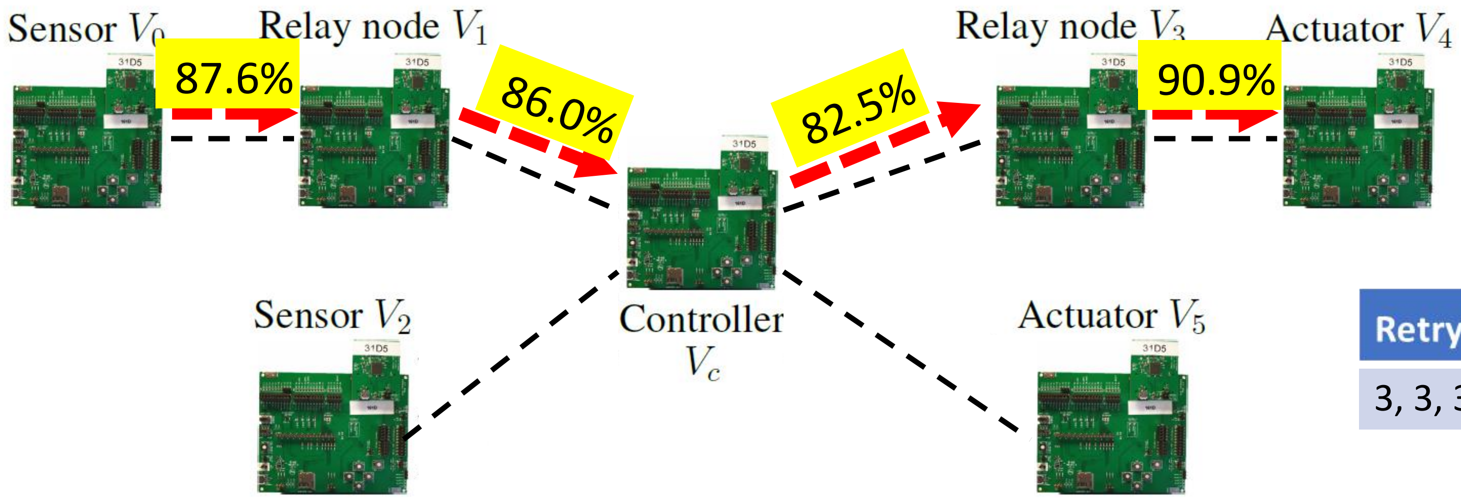
Reliable Static Schedule: Methodology

- Solve reliability constraint for each packet
  - How many retransmission slots are needed?
- Solve timing constraint
  - How to schedule these slots?

Reliable Static Schedule: reliability (TBS)

- Problem statement: Given the transmissions of a packet, and their link PDR (λ^L), find out a retry vector ($\vec{R}$) such that:
  - Total number of time slots (w) is minimized
  - Packet PDR (λ) achieves required reliability (λ^R)

$$\lambda = \prod_l 1 - \left(1 - \lambda_l^L\right)^{R[l]}$$



Retry vector ($\vec{R}$)	End-to-end PDR (λ)	Number of slots (w)
3, 3, 3, 3	0.989	12

Proposed solution: PDR table

- Start from $\vec{R} = [1,1,1,1]$
- Each time add 1 retry with maximum gain
- Stops when required PDR is reached (99%)

- Proved optimal
- Only last result is needed in static schedule

Number of slots (w)	End-to-end PDR (λ)	Retry vector ($\vec{R}$)
4	0.565	1, 1, 1, 1
5	0.664	1, 1, 2, 1
6	0.757	1, 2, 2, 1
7	0.851	2, 2, 2, 1
8	0.928	2, 2, 2, 2
9	0.952	2, 2, 3, 2
10	0.969	2, 3, 3, 2
11	0.982	3, 3, 3, 2
12	0.989	3, 3, 3, 3
13	0.994	3, 3, 4, 3

w^+

$\vec{R}^*$

Reliable Static Schedule: timing

- Map to single CPU scheduling problem
- Each packet has release time and deadline
- w^+ -> Worst Case Execution Time (WCET), for each packet
- Earliest Deadline First (EDF) can give optimal schedule
- Packet transmission (execution) follows retry vector

Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8
Task 1	Task 1	Task 1	Task 2	Task 2	Task 1	Task 1	

Rhythmic event: dynamic scheduling

- Check all packets are still schedulable
- If not schedulable, then packet degradation is needed

Number of slots	End-to-end PDR	Retry vector
4	0.565	1, 1, 1, 1
5	0.664	1, 1, 2, 1
6	0.757	1, 2, 2, 1
7	0.851	2, 2, 2, 1
8	0.928	2, 2, 2, 2
9	0.952	2, 2, 3, 2
10	0.969	2, 3, 3, 2
11	0.982	3, 3, 3, 2
12	0.989	3, 3, 3, 3
13	0.994	3, 3, 4, 3

Degradation ↑

Reliable → (13 slots)

Reliable Dynamic Schedule

- Problem statement: Given the packet set, Γ, determine w for each packet such that:
  - All rhythmic packets satisfy required reliability λ^R
  - Degradation information does not exceed broadcast payload
  - Total end-to-end PDR degradation is minimized

$$\forall \chi \in \Gamma, \min \sum \max\{0, \lambda^R - \lambda_\chi\}$$

NP-Hard

Proposed heuristic:

- Assign basic number of slots to each periodic packet.
- If not schedulable, drop smallest number of packets (a solved problem in literature).
- If schedulable, add slots to the packet that leads to best degradation improvement.

Number of slots	End-to-end PDR	Retry vector
4	0.565	1, 1, 1, 1
5	0.664	1, 1, 2, 1
6	0.757	1, 2, 2, 1
7	0.851	2, 2, 2, 1
8	0.928	2, 2, 2, 2
9	0.952	2, 2, 3, 2
10	0.969	2, 3, 3, 2
11	0.982	3, 3, 3, 2
12	0.989	3, 3, 3, 3
13	0.994	3, 3, 4, 3

Degradation

Reliable

Outline

- Motivation
- System Model
  - Transmission-Based Scheduling (TBS)
  - Packet-Based Scheduling (PBS)
- Reliable Real-Time Dynamic Packet Scheduling
  - Reliable Static Schedule
  - Reliable Dynamic Schedule
- Experimental Evaluation
  - Simulation
  - Testbed
- Conclusion

Simulation Evaluation

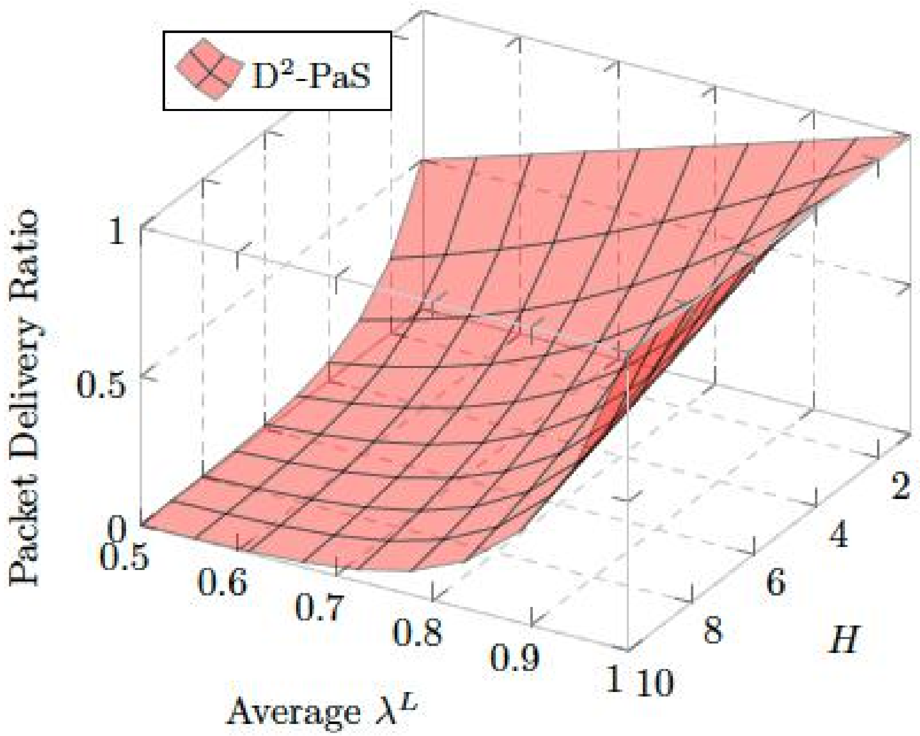


Figure 6 PDR in traditional RTWN scheduling

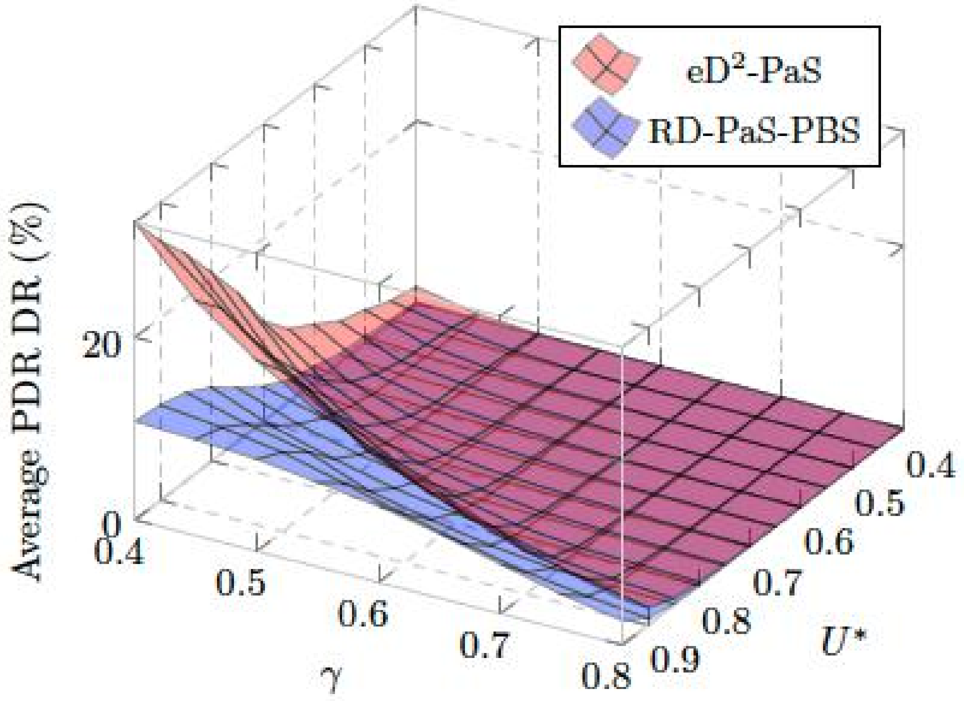
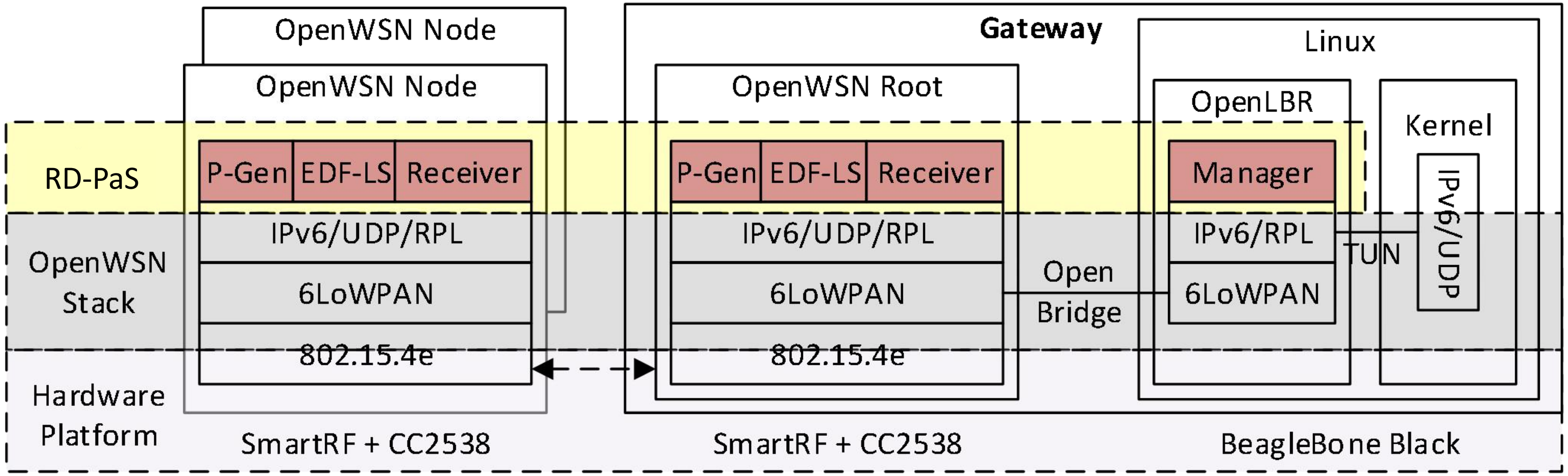


Figure 9 Comparison of the PDR degradation rate.

Testbed

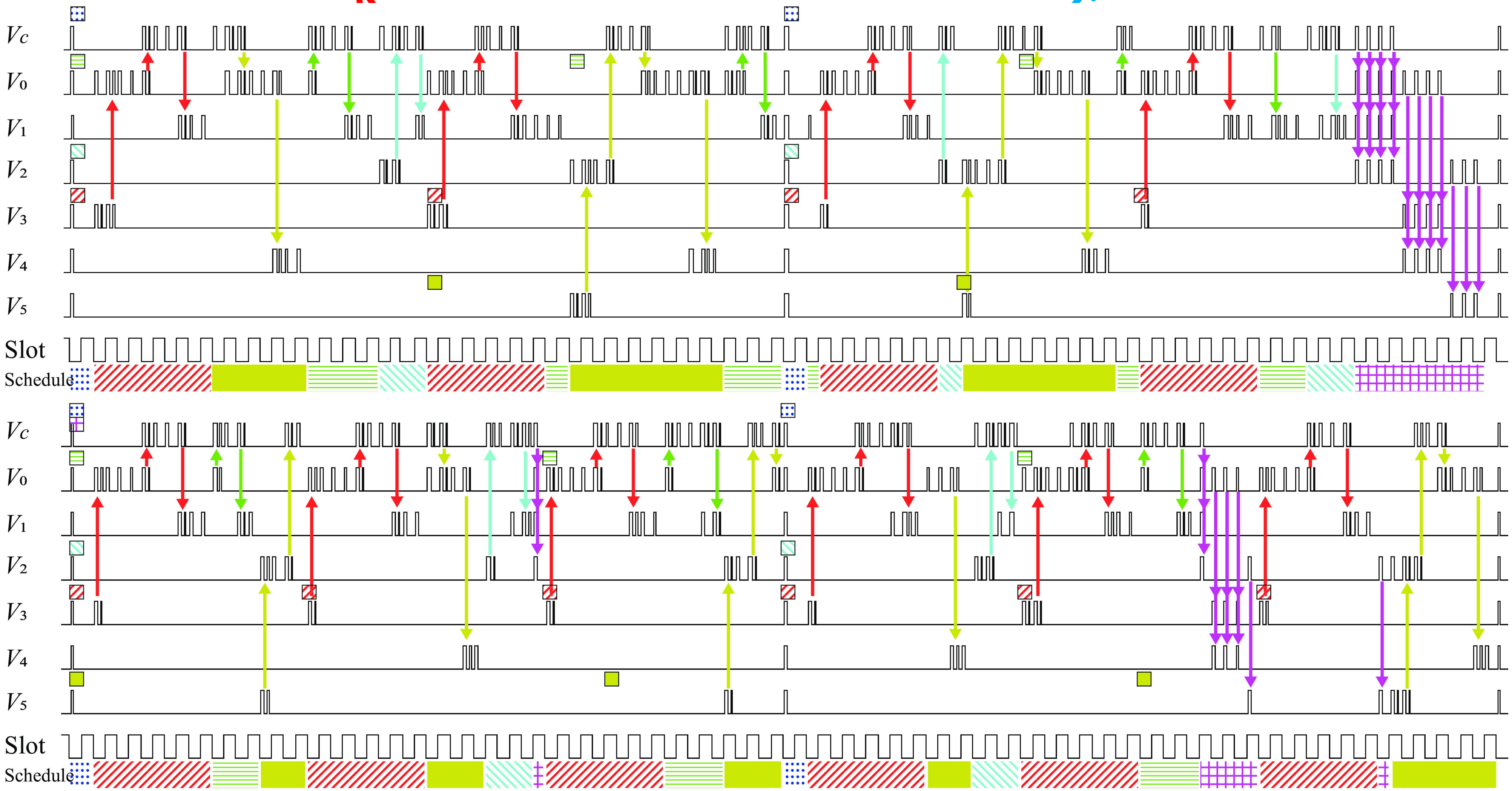
- Use TI CC2538 EMKs to form a mesh network
- Implement RD-PaS on an OpenWSN testbed
- Dynamic schedule generation in the application layer



Testbed-Based Experiment

7 CC2538 wireless devices

Logic analyzer



Conclusion:

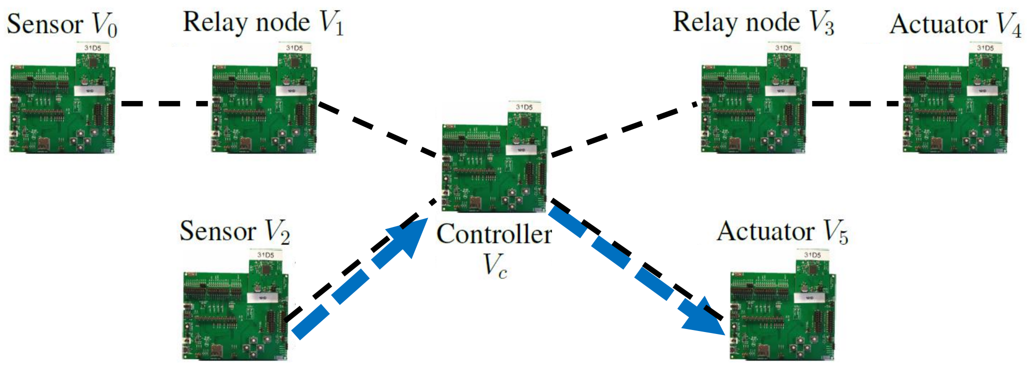
- RD-PaS solves reliable online static and dynamic scheduling problem considering each individual sensing packet.
  - It computes a reliable static schedule in nominal mode
  - It computes a dynamic schedule to compensate the new traffic and minimize the impact in rhythmic mode
- We studied the performance of RD-PaS
- We also implement RD-PaS in an OpenWSN testbed to validate the correctness

Thanks

Questions?

Packet-based Scheduling (PBS)

- A time slot is assigned to a packet of a task
- Involved nodes decide radio operation (TX, RX, idle) at runtime



Slot 1	Slot 2	Slot 3	Slot 4
(1)	(1)	(1)	(1)

← Assign 4 time slots for this packet

Performance comparison

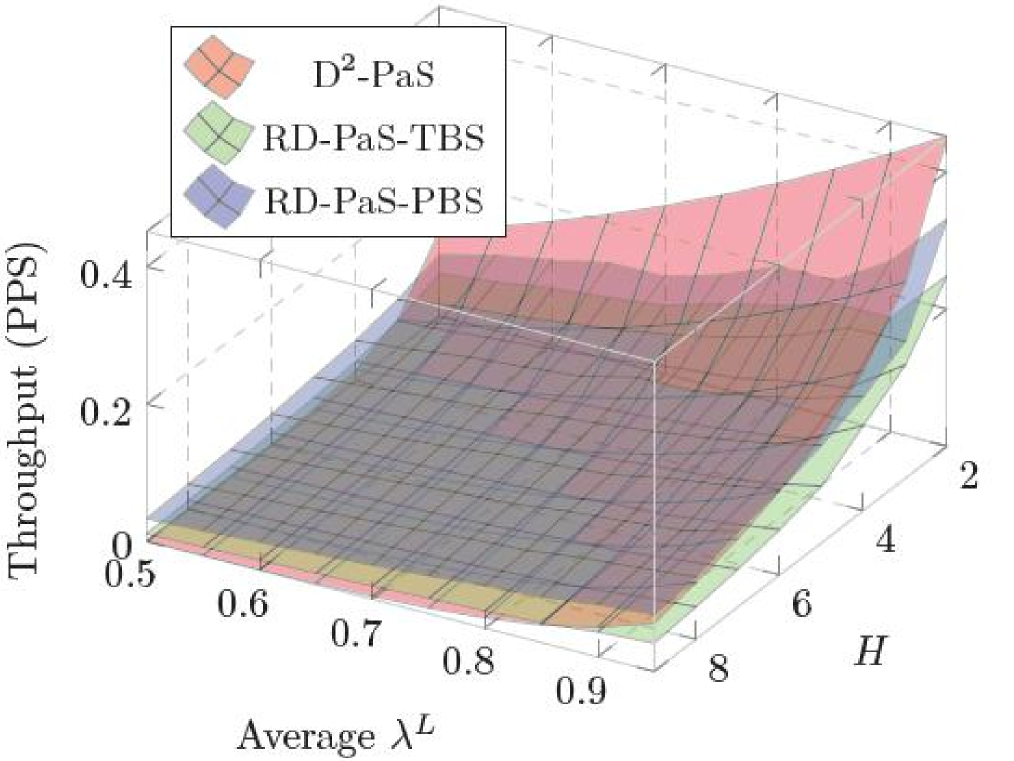


Figure 7 Throughput comparison among different scheduling frameworks.

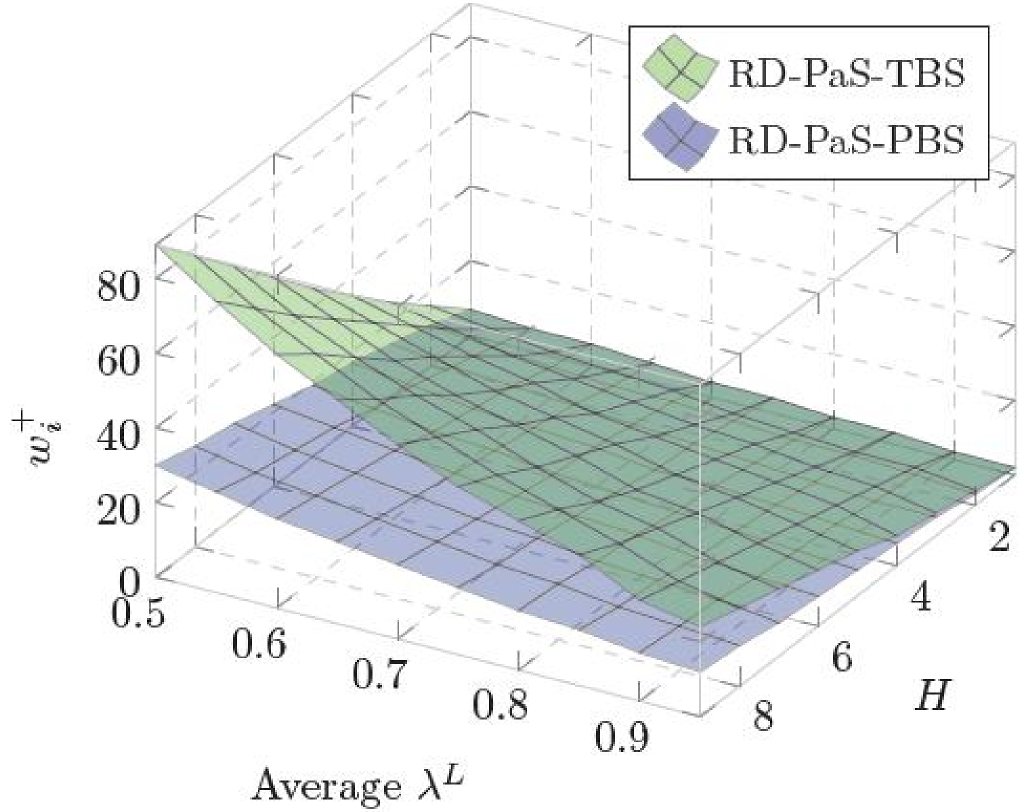


Figure 8 Comparison of w_i^+ in TBS and PBS.