

PMSA: Power Mode Selection Algorithm in IEEE 802.11 WLANs operating TWT

**Loïc Desgeorges, Thomas Begin, Isabelle Guérin
Lassous**

loic.desgeorges@univ-lyon1.fr

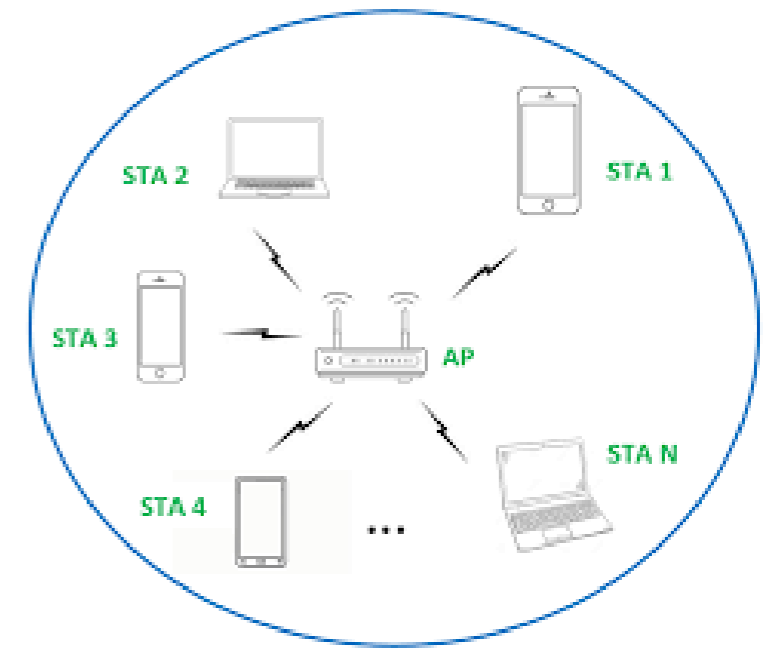
Univ. Lyon 1, ENS Lyon, Inria, CNRS

GDR RSD - 19 / 03 / 2026



Introduction

- There is a growing expectation for computer networks to implement energy-saving mechanism
- WLAN technology: Wi-Fi
- Set of stations (STAs) that can operate: in active mode or power save (PS) mode
 - Active: CSMA/CA
 - PS: use an energy saving mechanism



State of the art

- This paper focus on Target Wake Time
 - IEEE 802.11ah (further enhanced in ax)
 - Restricting communications to specific time window
- To define the time window: TWT scheduler
 - Optimizing the throughput (limit the collision)
 - Optimizing the energy saving (long sleep)
- **These works make the assumption that the network is underloaded**
 - **In such a case, we may have to switch between active and PS**

SCHEDULING WITH TWT

**PMSA: SELECTING THE POWER MODES
OF STAs**

RESULTS


Plan

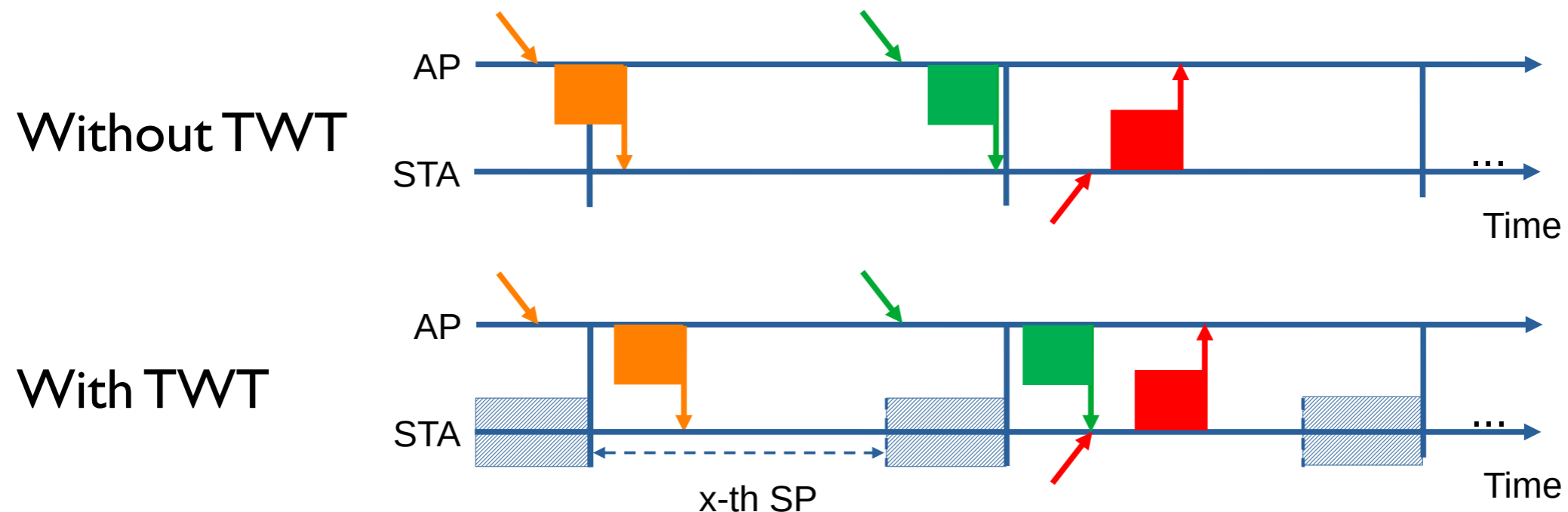
SCHEDULING WITH TWT

**PMSA: SELECTING THE POWER MODES
OF STAs**

RESULTS

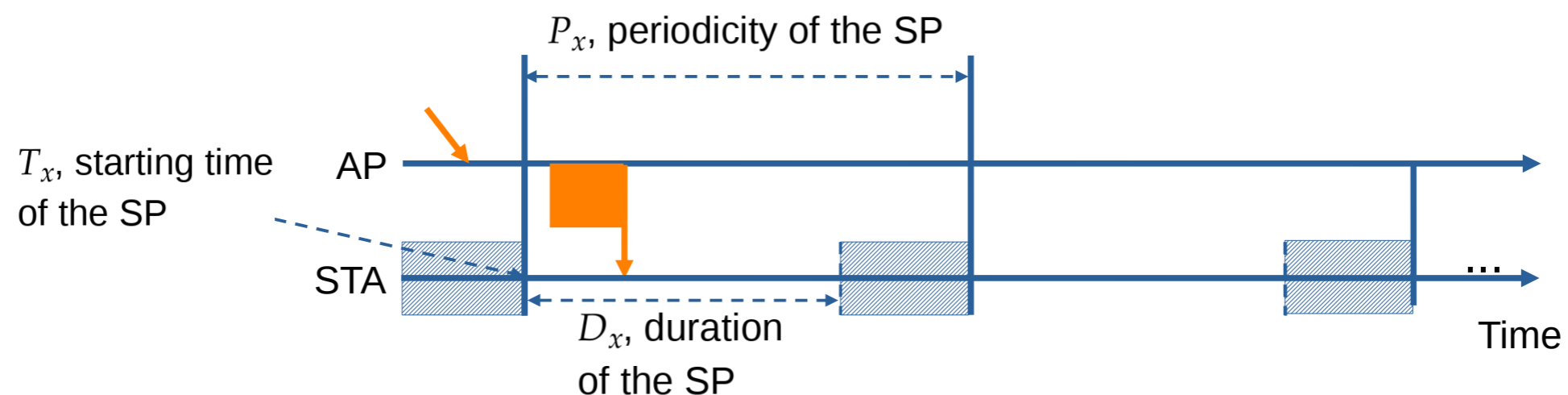
Presentation of TWT

- Dividing time into Service Periods (SP)
 - Turn off the Radio outside this period: 
- The SP can end by a TWT Teardown frame
- Negotiation between the AP and the STA



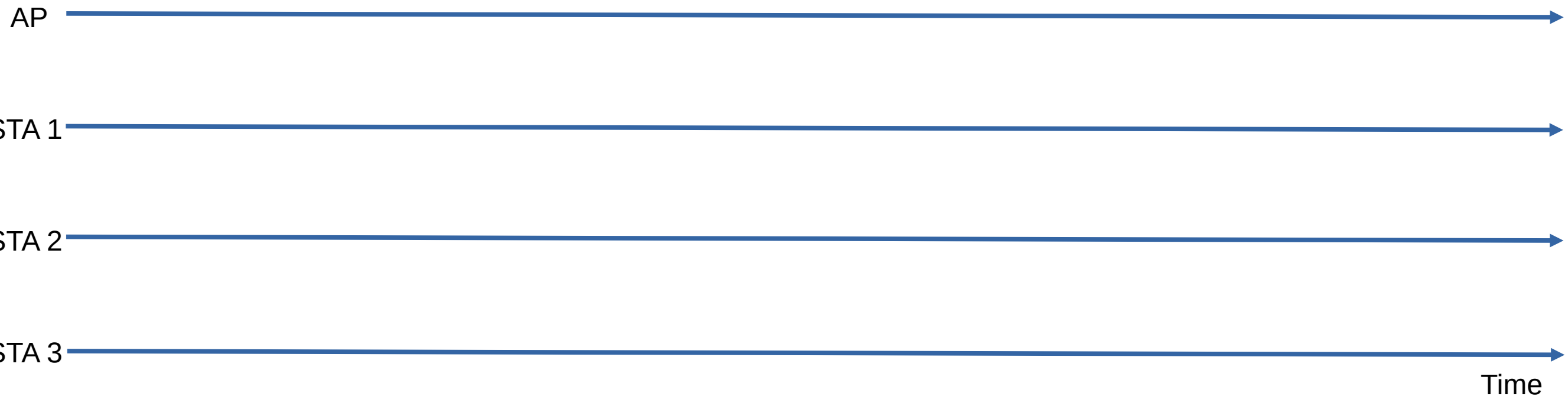
TWT agreements

- TWT agreements
 - Tx starting time of the SP
 - Px periodicity of the SP
 - Dx duration of the SP
- TWT Mode of operations
 - Unsolicited TWT
 - Individual SP
 - Implicit TWT
 - Unannounced TWT



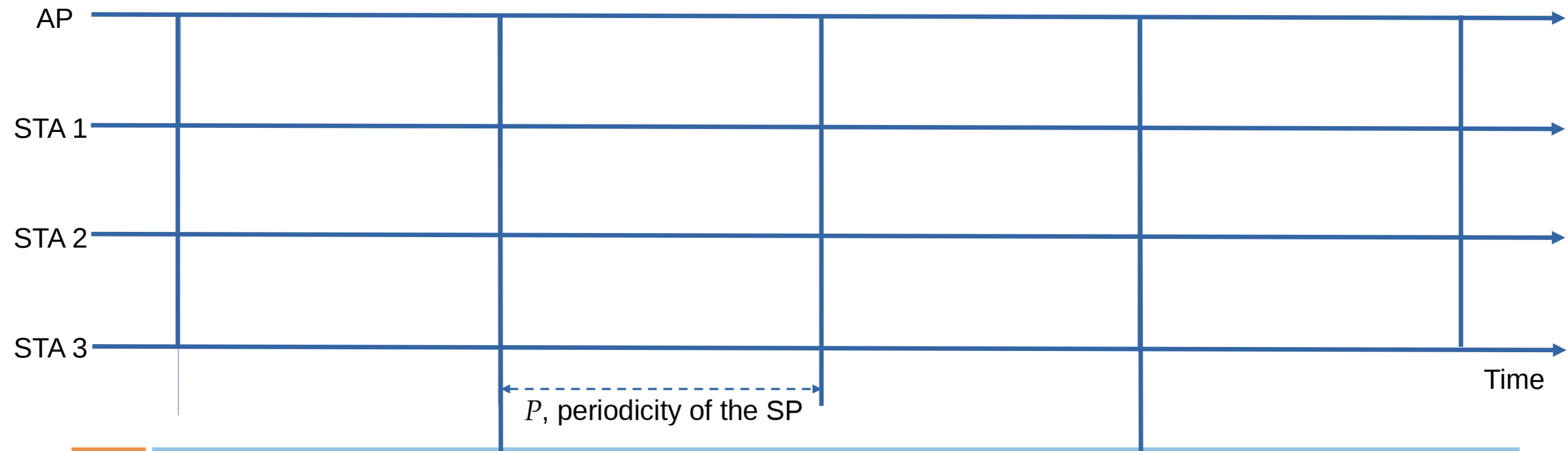
Scheduling with TWT

- Given a set of A STAs, how do we orchestrate the TWT agreements?
 - TWT scheduler
- Inspired from the literature
 - *Target Wake Time Scheduling Strategies for Uplink Transmission in IEEE 802.11ax Networks*, IEEE Wireless Communications and Networking Conference (WCNC)



Scheduling with TWT

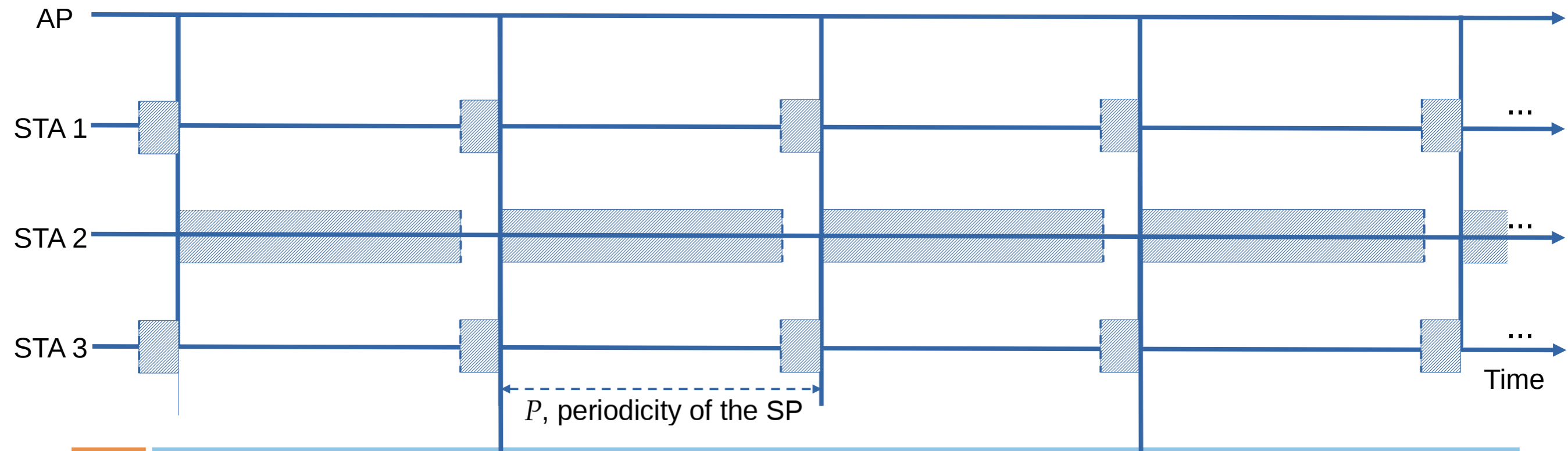
- Given a set of A STAs, how do we orchestrate the TWT agreements?
 - TWT scheduler
- Inspired from the literature
 - The same period P for each TWT agreements



Scheduling with TWT

- Inspired from the literature
 - The same period P for each TWT agreements
 - There are M non overlapping SPs
 - Rank the A STAs (according to the traffic rate) and group them in M group. Here, $A = 3$, $M = 2$ and $STA3$ traffic rate \gg $STA2$ tr $>$ $STA1$ tr
 - Computation of the duration the SP

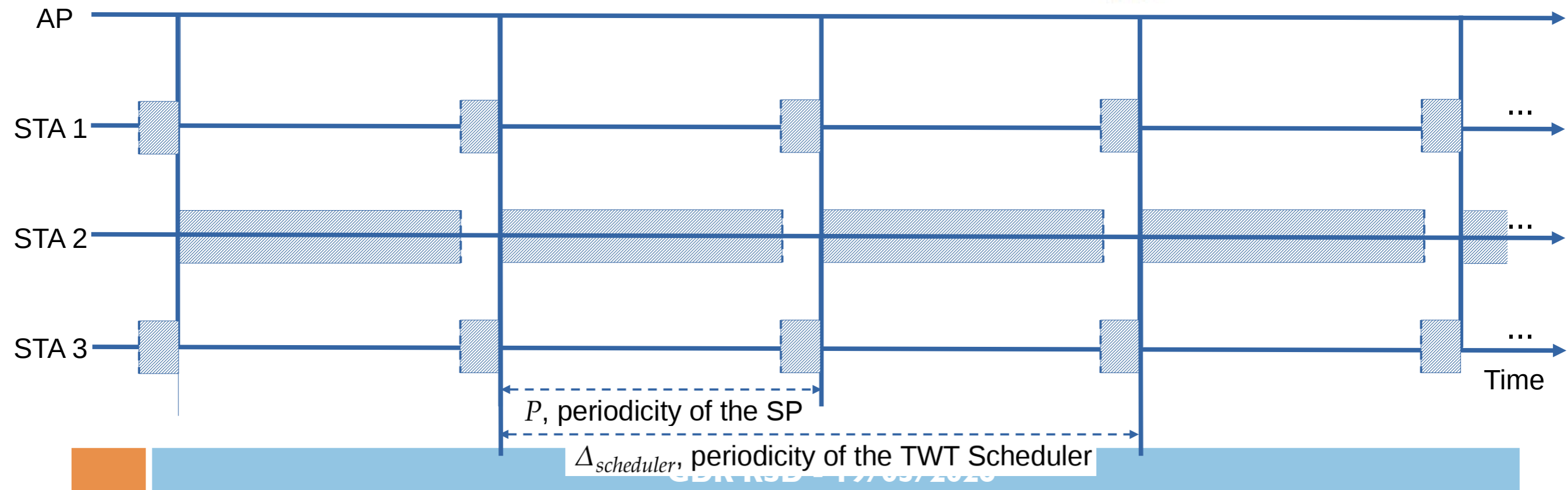
$$D_x = \frac{\sum_{i \in \Omega_x} \frac{\lambda_i}{\mu_i}}{\sum_{i=1}^A \frac{\lambda_i}{\mu_i}} \times P$$



Scheduling with TWT

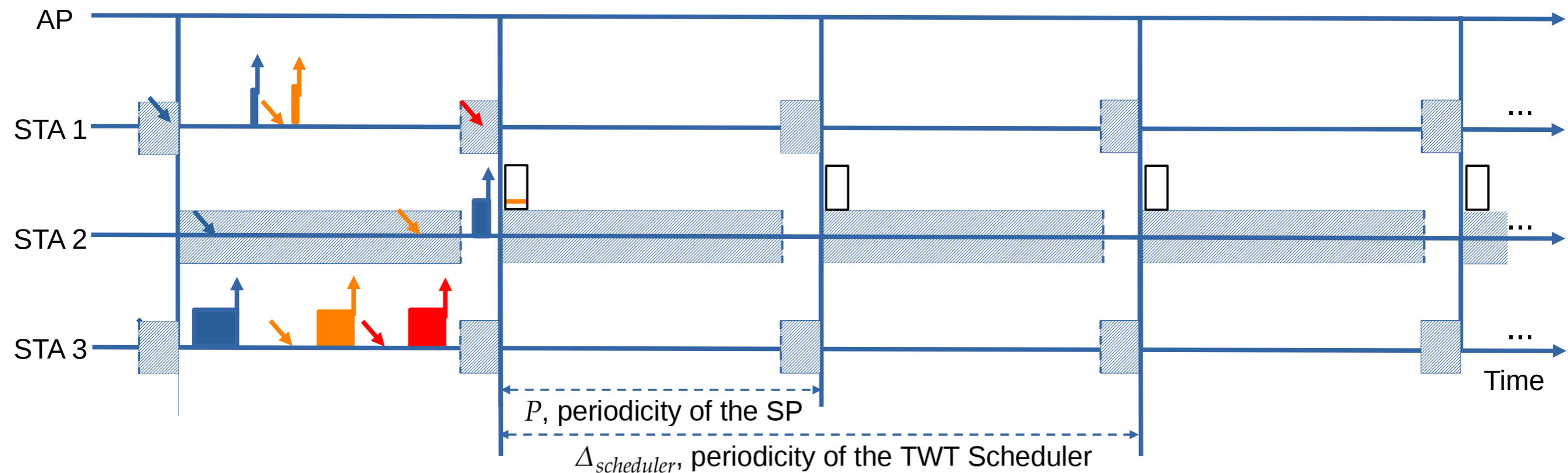
- Inspired from the literature
 - The same period P for each TWT agreements
 - There are M non overlapping SPs
 - Rank the A STAs and group them in M group
 - Computation of the duration the SP

$$D_x = \frac{\sum_{i \in \Omega_x} \frac{\lambda_i}{\mu_i}}{\sum_{i=1}^A \frac{\lambda_i}{\mu_i}} \times P$$



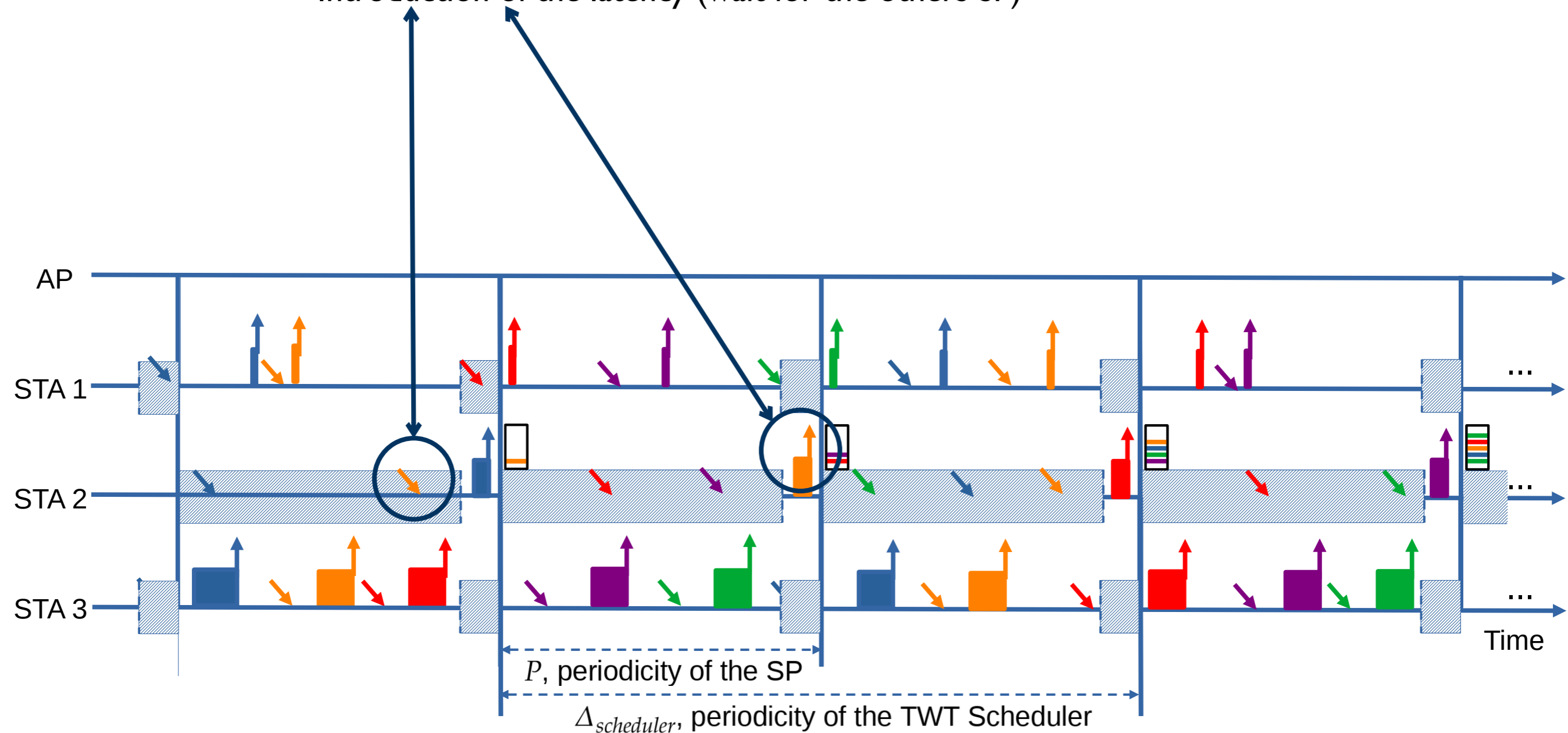
Scheduling with TWT

- With the TWT Scheduler:
- ↙ Arrival of a packet
- █ Transmission of the packet. The width corresponds to the transmission time
- State of the buffer (just represented for STA 2)



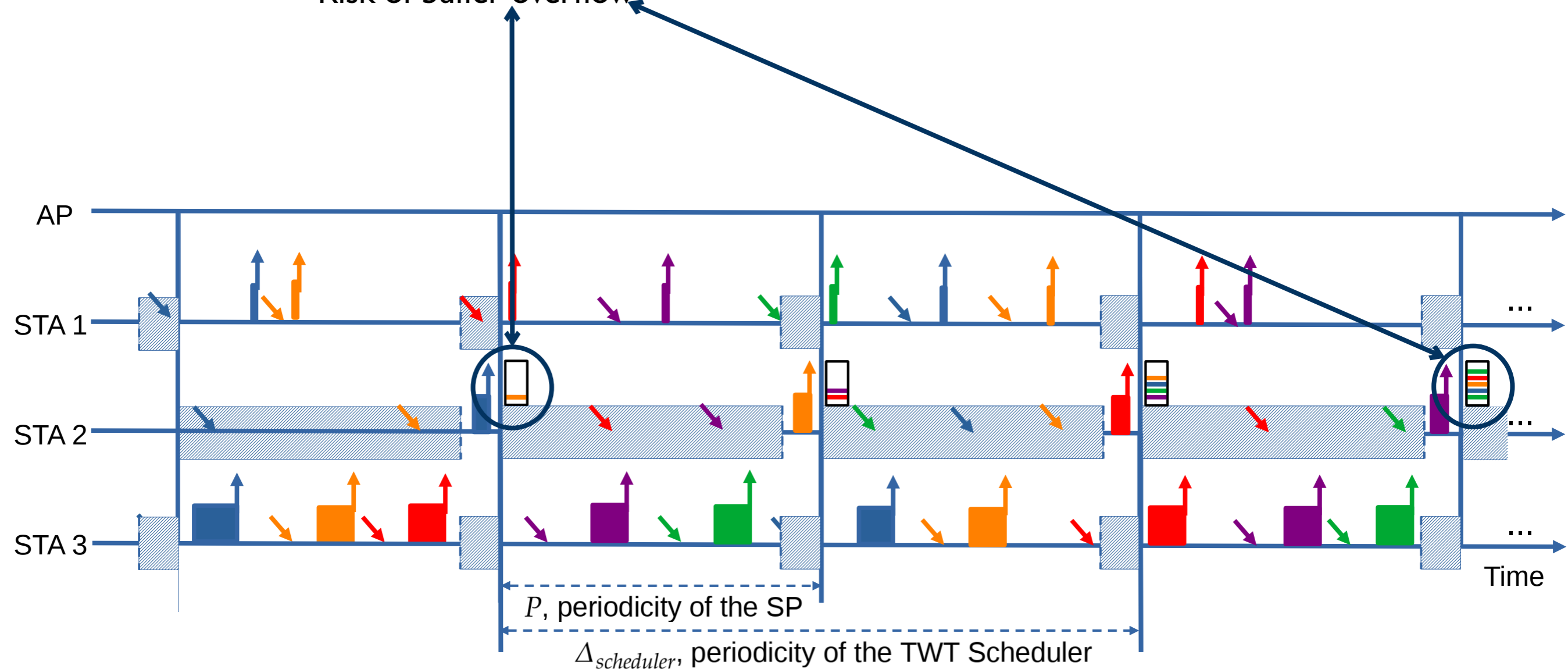
Scheduling with TWT

- With the TWT Scheduler:
 - At the end of the period: one packet in the buffer (has to wait another period)
 - Introduction of the latency (wait for the others SP)



Scheduling with TWT

- With the TWT Scheduler:
 - At the end of the period: one packet in the buffer
 - Introduction of the latency (wait for the others SP)
 - Risk of buffer overflow



Plan

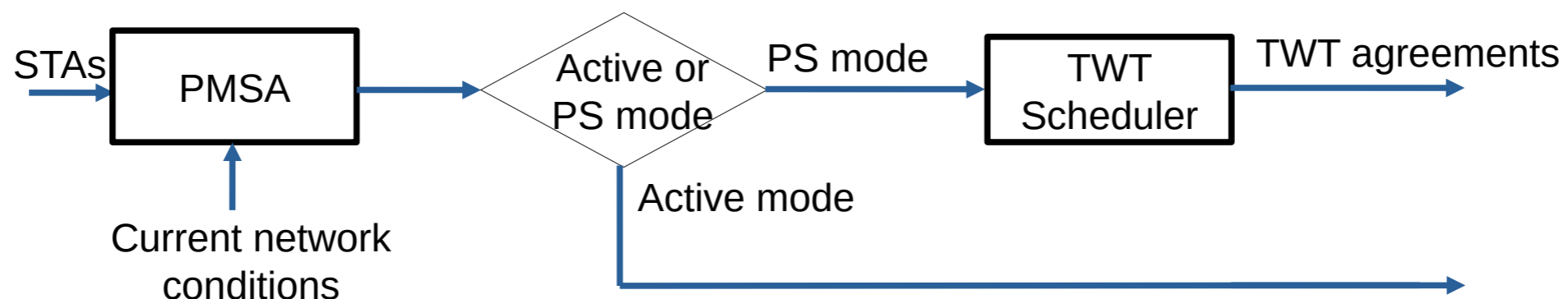
SCHEDULING WITH TWT

**PMSA: SELECTING THE POWER MODES
OF STAs**

RESULTS

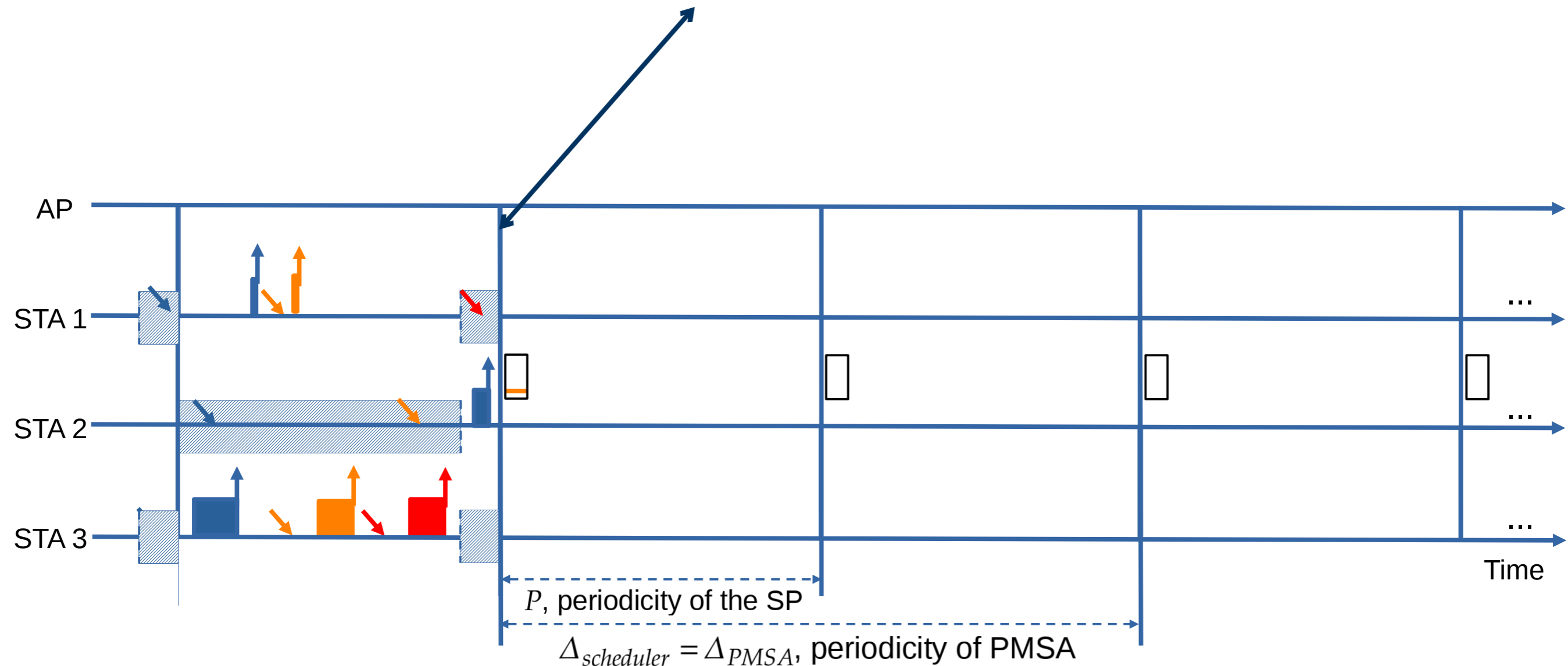
Power Mode Selection Algorithm

- STA can operate in two modes: active or PS mode
 - TWT scheduler only consider PS mode
 - This limit the flexibility of these schedulers, especially in case of congestion
- To overcome it: Power Mode Selection Algorithm
 - Not a Scheduler but a complement
 - Decide the appropriate power mode (active or PS)



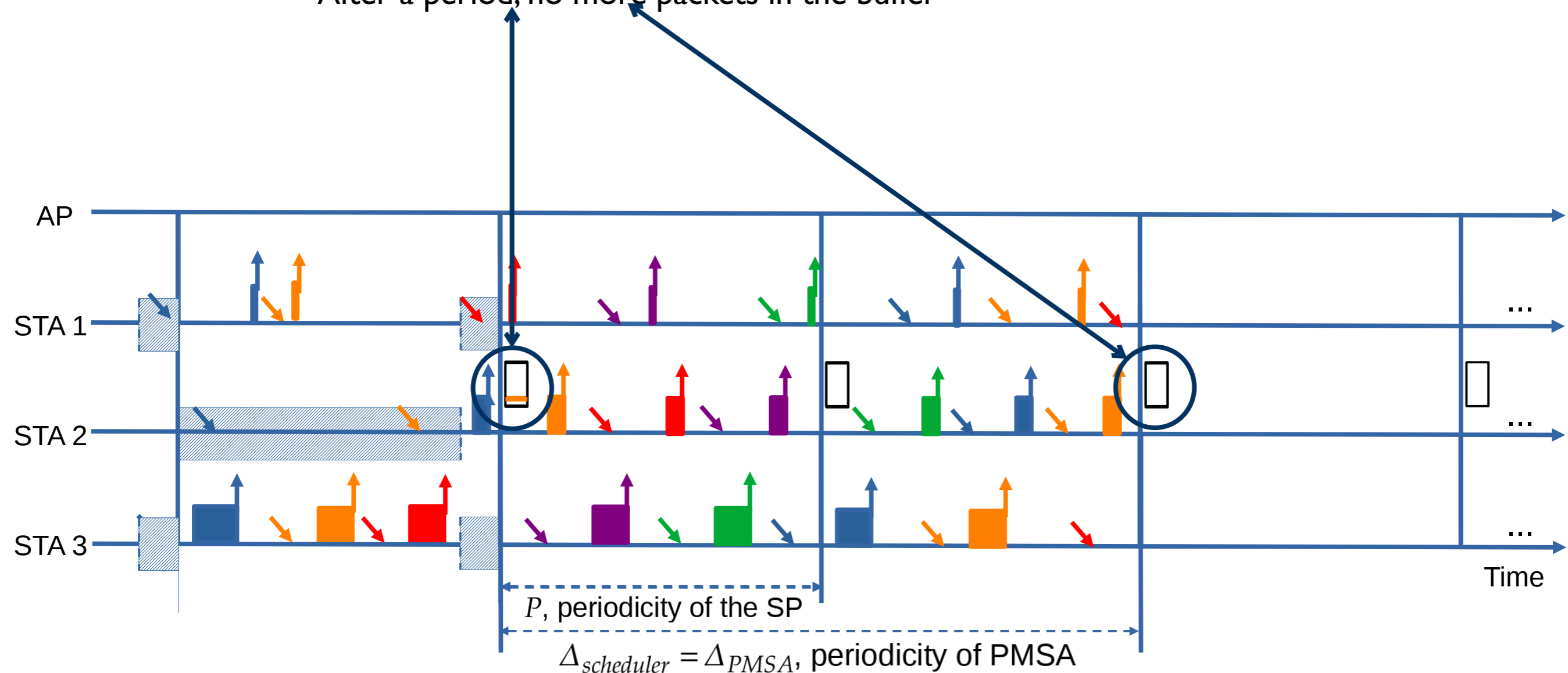
Power Mode Selection Algorithm

- With PMSA:
 - Detection of the congestion
 - Switch all the STAs on the active mode
 - We chose to switch all the STAs to avoid the risk of disturbance on one SP



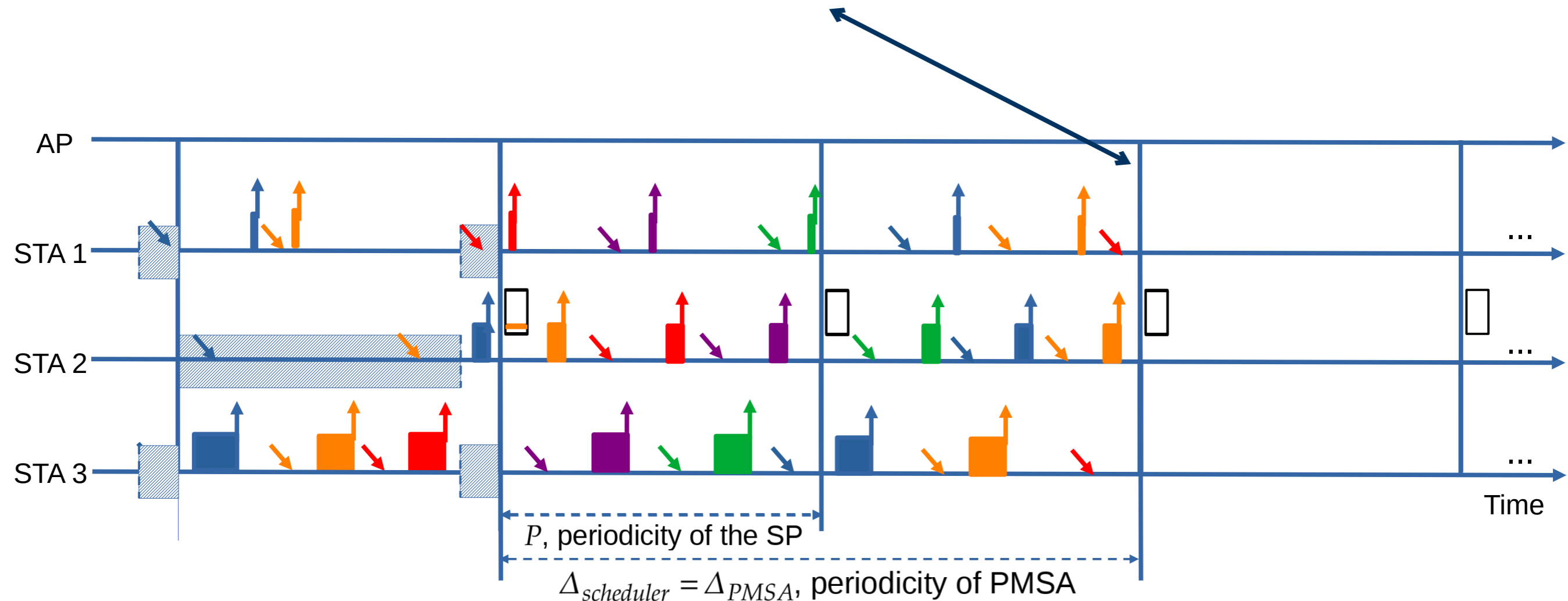
Power Mode Selection Algorithm

- With PMSA:
 - Detection of the congestion
 - Switch all the STAs on the active mode
 - After a period, no more packets in the buffer



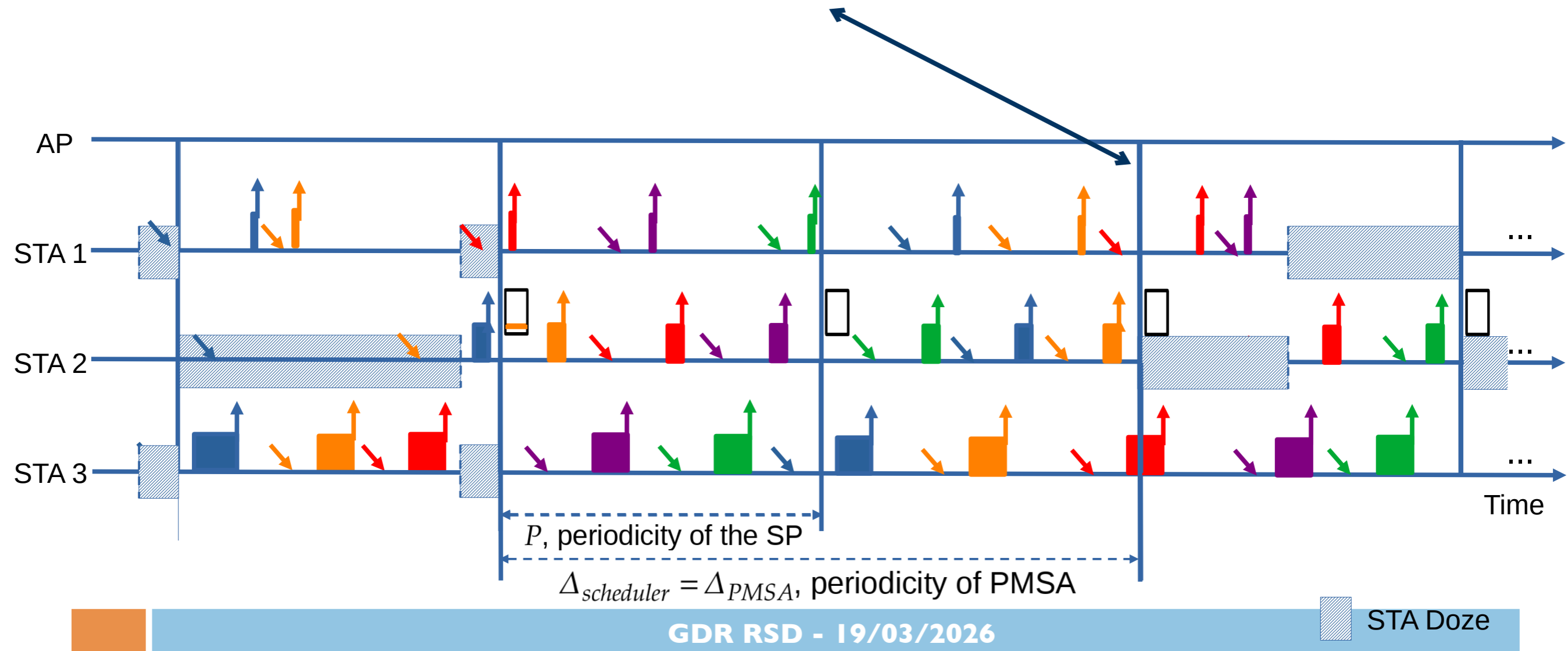
Power Mode Selection Algorithm

- With PMSA:
 - Detection of the congestion
 - Switch all the STAs on the active mode
 - After a period, no more packets in the buffer
 - STA 1 and STA 2 have moderate traffic rate



Power Mode Selection Algorithm

- With PMSA:
 - Detection of the congestion
 - Switch all the STAs on the active mode
 - After a period, no more packets in the buffer
 - STA 1 and STA 2 have moderate traffic rate: switch to PS mode



Power Mode Selection Algorithm

- The algorithm is based on the occupancy rates
 - One per SP

Algorithm 1: Pseudo-code for PMSA.

```
1: while True do
2:   Collect and compute  $(\rho_1, \dots, \rho_M)$  during a new  $\Delta_{PMSA}$  period
3:   if  $\exists x \in [1, M]$  such that  $\rho_x > \tau$  then
4:     for all STAs: switch to active mode and disable TWT
5:   else if  $\Gamma$  is not empty then
6:     Rank STAs in ascending order of their ratio  $\frac{\lambda_i}{\mu_i}$ 
7:     for  $i \in \Gamma$  do
8:        $x \leftarrow ((i - 1) \bmod M) + 1$ 
9:       if  $1 - \rho_x > \frac{\lambda_i}{\mu_i}$  then
10:        switch  $i$ -th STA to PS mode
11:         $\rho_x \leftarrow \rho_x + \frac{\lambda_i}{\mu_i}$ 
12:       end if
13:     end for
14:   end if
15:   Update  $\Gamma$ 
16:   Invoke TWT scheduler for the STAs in PS mode
17: end while
```

Power Mode Selection Algorithm

- The algorithm is based on the occupancy rates
 - One per SP
 - If there is one SP with a rate too high: switch ALL the STAs to active mode

Algorithm 1: Pseudo-code for PMSA.

```
1: while True do
2:   Collect and compute  $(\rho_1, \dots, \rho_M)$  during a new  $\Delta_{PMSA}$  period
3:   if  $\exists x \in [1, M]$  such that  $\rho_x > \tau$  then
4:     for all STAs: switch to active mode and disable TWT
5:   else if  $\Gamma$  is not empty then
6:     Rank STAs in ascending order of their ratio  $\frac{\lambda_i}{\mu_i}$ 
7:     for  $i \in \Gamma$  do
8:        $x \leftarrow ((i - 1) \bmod M) + 1$ 
9:       if  $1 - \rho_x > \frac{\lambda_i}{\mu_i}$  then
10:        switch  $i$ -th STA to PS mode
11:         $\rho_x \leftarrow \rho_x + \frac{\lambda_i}{\mu_i}$ 
12:       end if
13:     end for
14:   end if
15:   Update  $\Gamma$ 
16:   Invoke TWT scheduler for the STAs in PS mode
17: end while
```

Power Mode Selection Algorithm

- The algorithm is based on the occupancy rates
 - One per SP
 - If there is one SP with a rate too high: switch ALL the STAs to active mode
 - If there are STAs in active mode: determine the ones to switch to PS mode
- How:
 - Rank the STAs in ascending order of their ratio traffic rate / physical data rate
 - Check if there is enough space available on the next SP (according to their occupancy rate)

Algorithm 1: Pseudo-code for PMSA.

```
1: while True do
2:   Collect and compute  $(\rho_1, \dots, \rho_M)$  during a new  $\Delta_{PMSA}$  period
3:   if  $\exists x \in [1, M]$  such that  $\rho_x > \tau$  then
4:     for all STAs: switch to active mode and disable TWT
5:   else if  $\Gamma$  is not empty then
6:     Rank STAs in ascending order of their ratio  $\frac{\lambda_i}{\mu_i}$ 
7:     for  $i \in \Gamma$  do
8:        $x \leftarrow ((i - 1) \bmod M) + 1$ 
9:       if  $1 - \rho_x > \frac{\lambda_i}{\mu_i}$  then
10:        switch  $i$ -th STA to PS mode
11:         $\rho_x \leftarrow \rho_x + \frac{\lambda_i}{\mu_i}$ 
12:       end if
13:     end for
14:   end if
15:   Update  $\Gamma$ 
16:   Invoke TWT scheduler for the STAs in PS mode
17: end while
```

Power Mode Selection Algorithm

- The algorithm is based on the occupancy rates
 - One per SP
 - If there is one SP with a rate too high: switch ALL the STAs to active mode
 - If there are STAs in active mode: determine the ones to switch to PS mode
- How - Oracle:
 - Test all the combination of switch possible

Algorithm 1: Pseudo-code for PMSA.

```
1: while True do
2:   Collect and compute  $(\rho_1, \dots, \rho_M)$  during a new  $\Delta_{PMSA}$  period
3:   if  $\exists x \in [1, M]$  such that  $\rho_x > \tau$  then
4:     for all STAs: switch to active mode and disable TWT
5:   else if  $\Gamma$  is not empty then
6:     Rank STAs in ascending order of their ratio  $\frac{\lambda_i}{\mu_i}$ 
7:     for  $i \in \Gamma$  do
8:        $x \leftarrow ((i - 1) \bmod M) + 1$ 
9:       if  $1 - \rho_x > \frac{\lambda_i}{\mu_i}$  then
10:        switch  $i$ -th STA to PS mode
11:         $\rho_x \leftarrow \rho_x + \frac{\lambda_i}{\mu_i}$ 
12:       end if
13:     end for
14:   end if
15:   Update  $\Gamma$ 
16:   Invoke TWT scheduler for the STAs in PS mode
17: end while
```

Plan

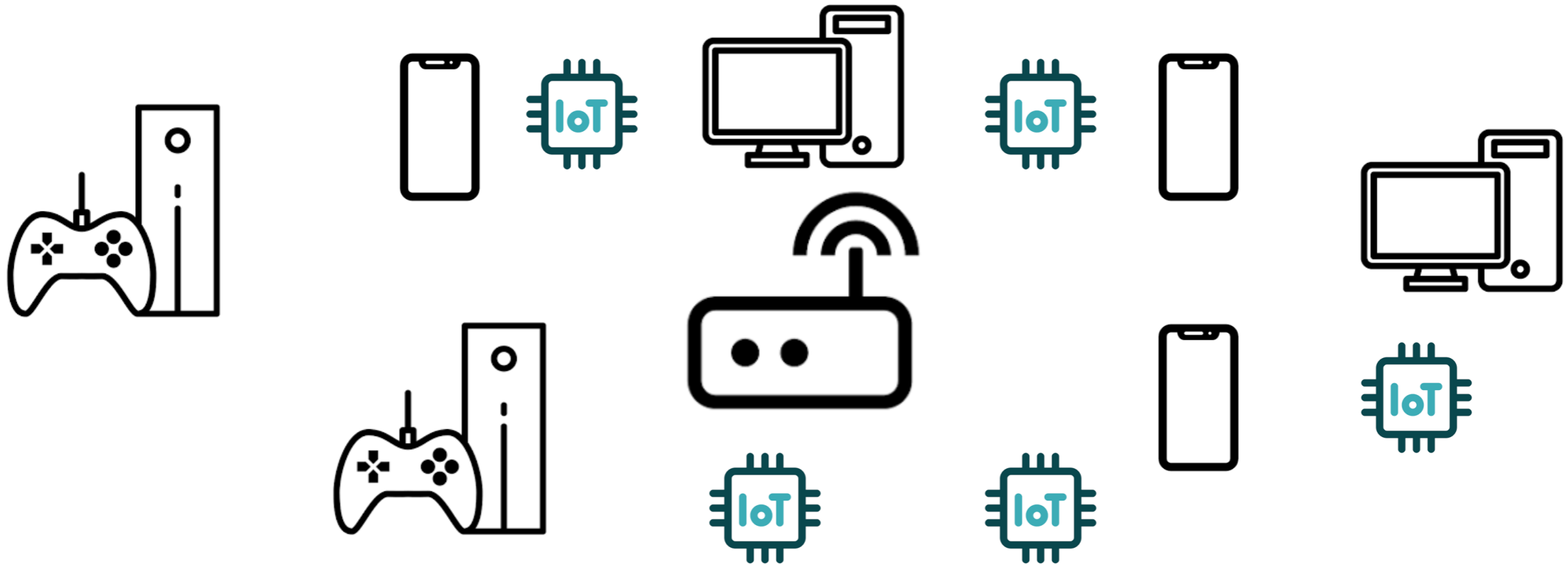


SCHEDULING WITH TWT

**PMSA: SELECTING THE POWER MODES
OF STAs**

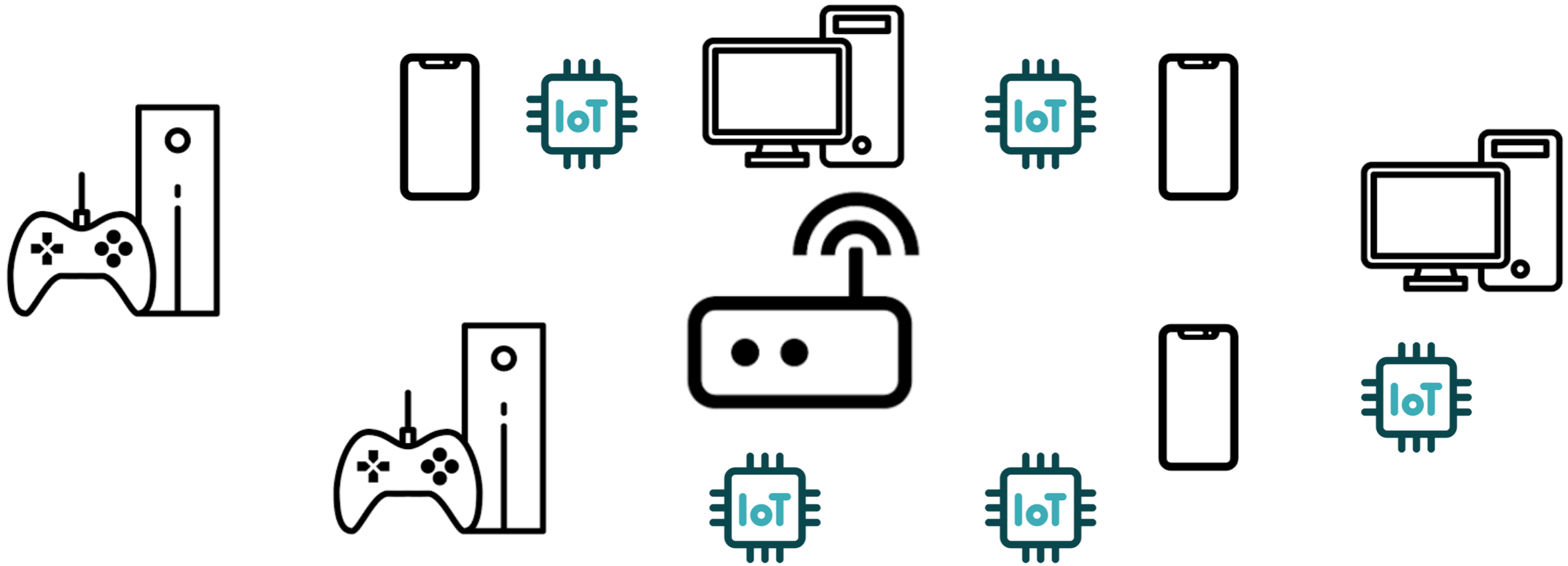
RESULTS

Results: Preliminaries



- 6 types of traffic: ToIP, IoT, Video conferencing, video gaming, streaming, streaming 4K
- 2 scenarios: PMSA under loaded conditions and PMSA under various load conditions

Results: Preliminaries



- Simulator: WE3S

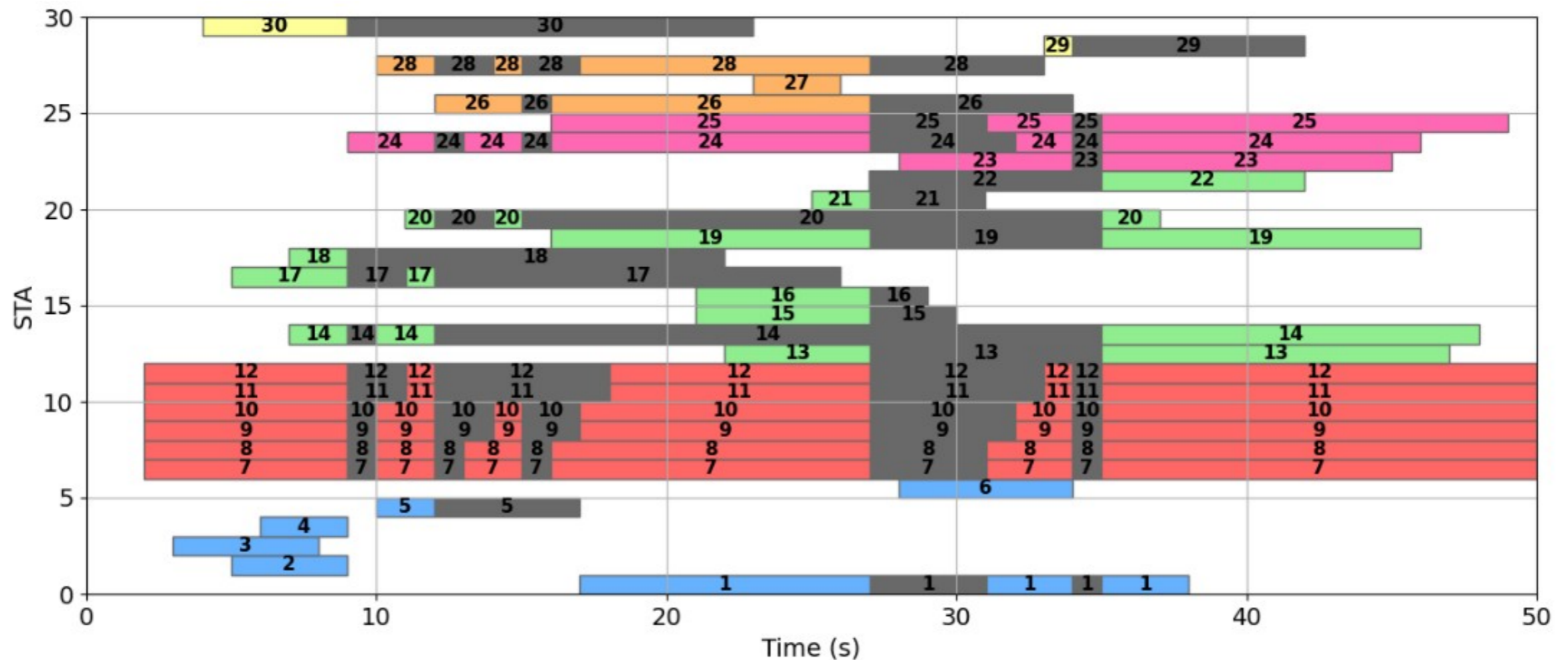
- Specifically designed to study energy-saving mechanisms in IEEE 802.11-based WLANs.

- 4 strategies:

- Classical
- w/o PMSA
- PMSA
- Oracle

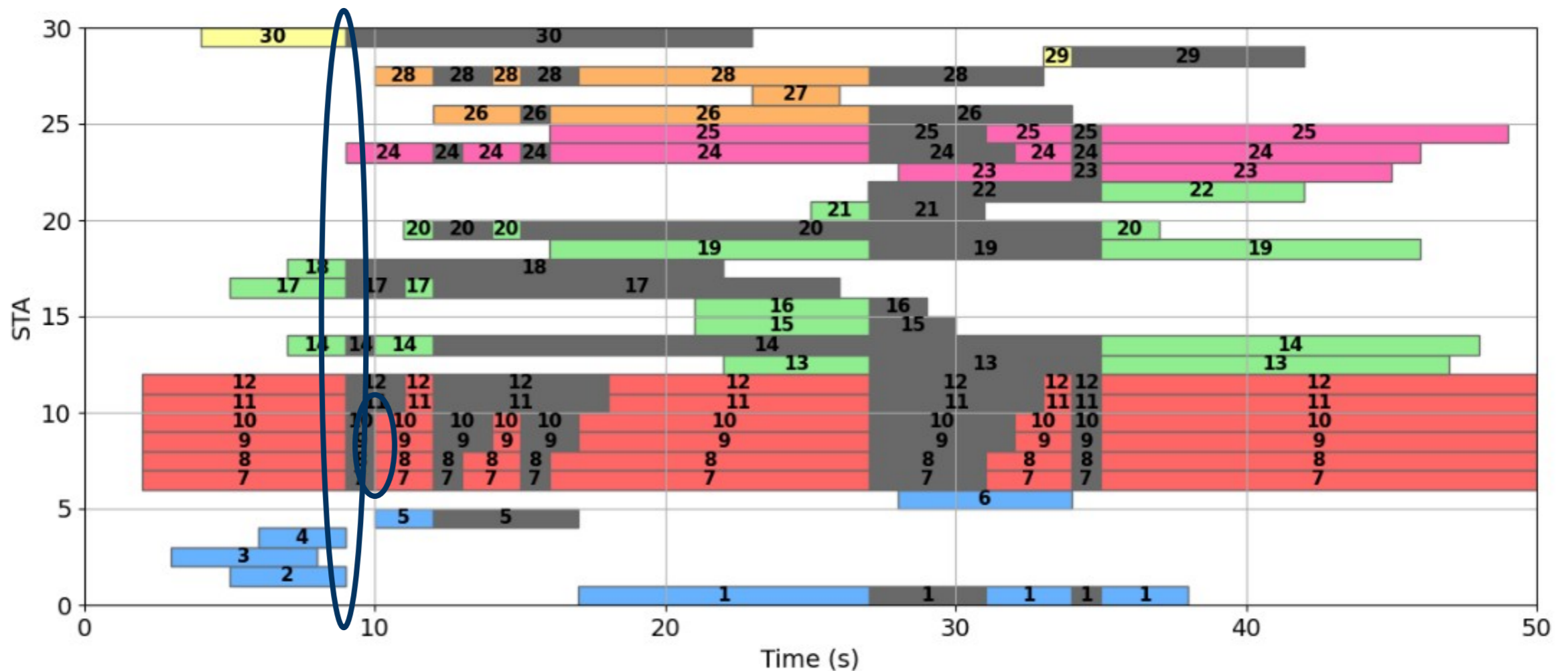
Results: PMSA under loaded conditions

- ToIP: blue, IoT: red, video conferencing: green, video gaming: pink, streaming: orange, 4K streaming: yellow
- $M = 5,$



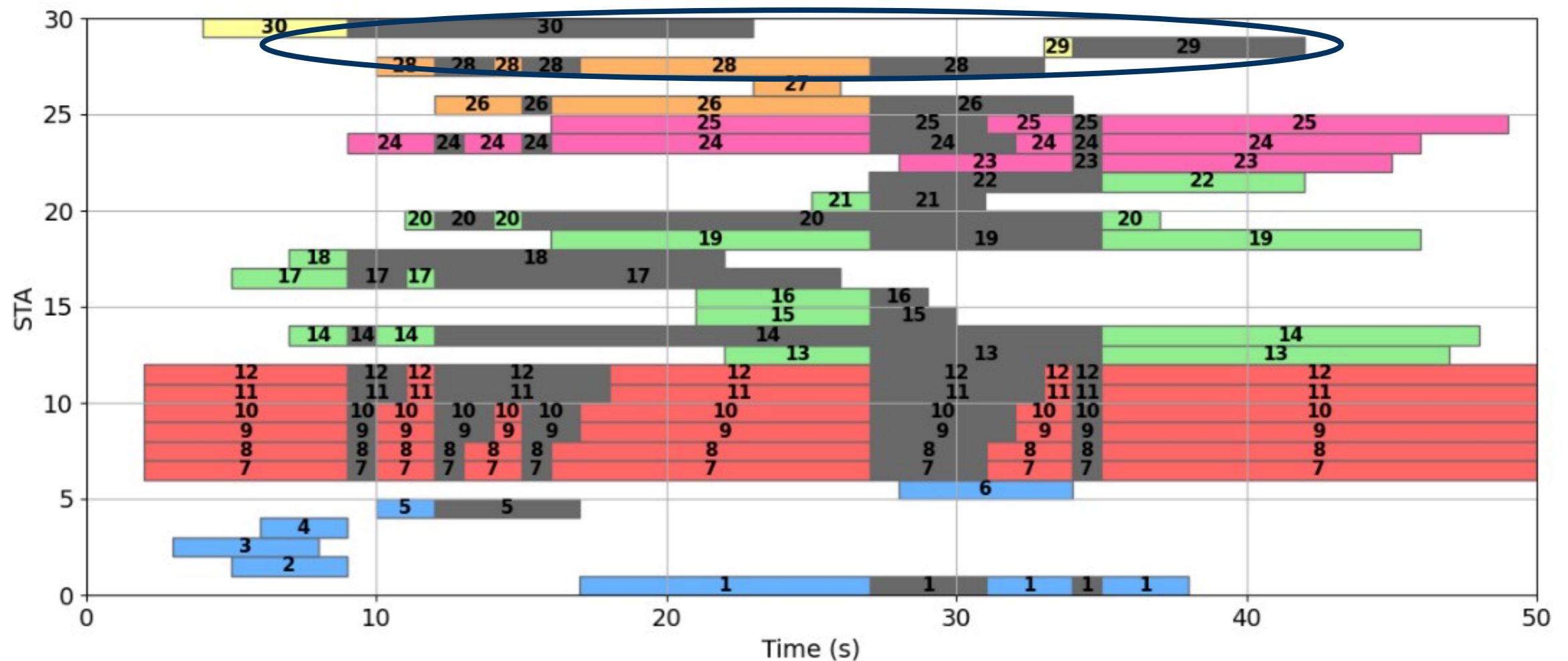
Results: PMSA under loaded conditions

- Evolution of the modes:
 - $t=9$: switch to active mode
 - $t=10$ and 11 : switch to PS mode



Results: PMSA under loaded conditions

- Behavior of the streaming 4K
 - Switch to active mode and stay in this mode



Results: PMSA under loaded conditions

- No packet loss with PMSA while 9% for video gaming or 25% for streaming 4K
- Delay reasonable and $< 120\text{ms}$ for delay sensitive application with PMSA (while \gg for w/o PMSA)
- Oracle not always the best: global optimisation

Type of STAs	classical	w/o PMSA	PMSA	oracle
ToIP	1.2	286.4	23.6	18.2
IoT	0.5	112.2	23.7	24.2
Video conferencing	1.9	801.9	32.5	22.2
Video gaming	1.5	286.9	15.2	13
Streaming	2.7	592.8	2.8	4.2
4K Streaming	2.2	517.6	2.4	2.3

TABLE V: Scenario 1. Mean packet delay (in ms) per STA according to their type and the relative increase as compared to the classical strategy.

Results: PMSA under loaded conditions

- No packet loss with PMSA while 9% for video gaming or 25% for streaming 4K
- Delay reasonable and $< 120\text{ms}$ for delay sensitive application with PMSA (while \gg for w/o PMSA)
- Oracle not always the best: global optimisation

Type of STAs	classical	w/o PMSA	PMSA	oracle
ToIP	1.2	286.4	23.6	18.2
IoT	0.5	112.2	23.7	24.2
Video conferencing	1.9	801.9	32.5	22.2
Video gaming	1.5	286.9	15.2	13
Streaming	2.7	592.8	2.8	4.2
4K Streaming	2.2	517.6	2.4	2.3

TABLE V: Scenario 1. Mean packet delay (in ms) per STA according to their type and the relative increase as compared to the classical strategy.

Results: PMSA under loaded conditions

- No packet loss with PMSA while 9% for video gaming or 25% for streaming 4K
- Delay reasonable and $< 120\text{ms}$ for delay sensitive application with PMSA (while \gg for w/o PMSA)
- Oracle not always the best: global optimisation

Type of STAs	classical	w/o PMSA	PMSA	oracle
ToIP	1.2	286.4	23.6	18.2
IoT	0.5	112.2	23.7	24.2
Video conferencing	1.9	801.9	32.5	22.2
Video gaming	1.5	286.9	15.2	13
Streaming	2.7	592.8	2.8	4.2
4K Streaming	2.2	517.6	2.4	2.3

TABLE V: Scenario 1. Mean packet delay (in ms) per STA according to their type and the relative increase as compared to the classical strategy.

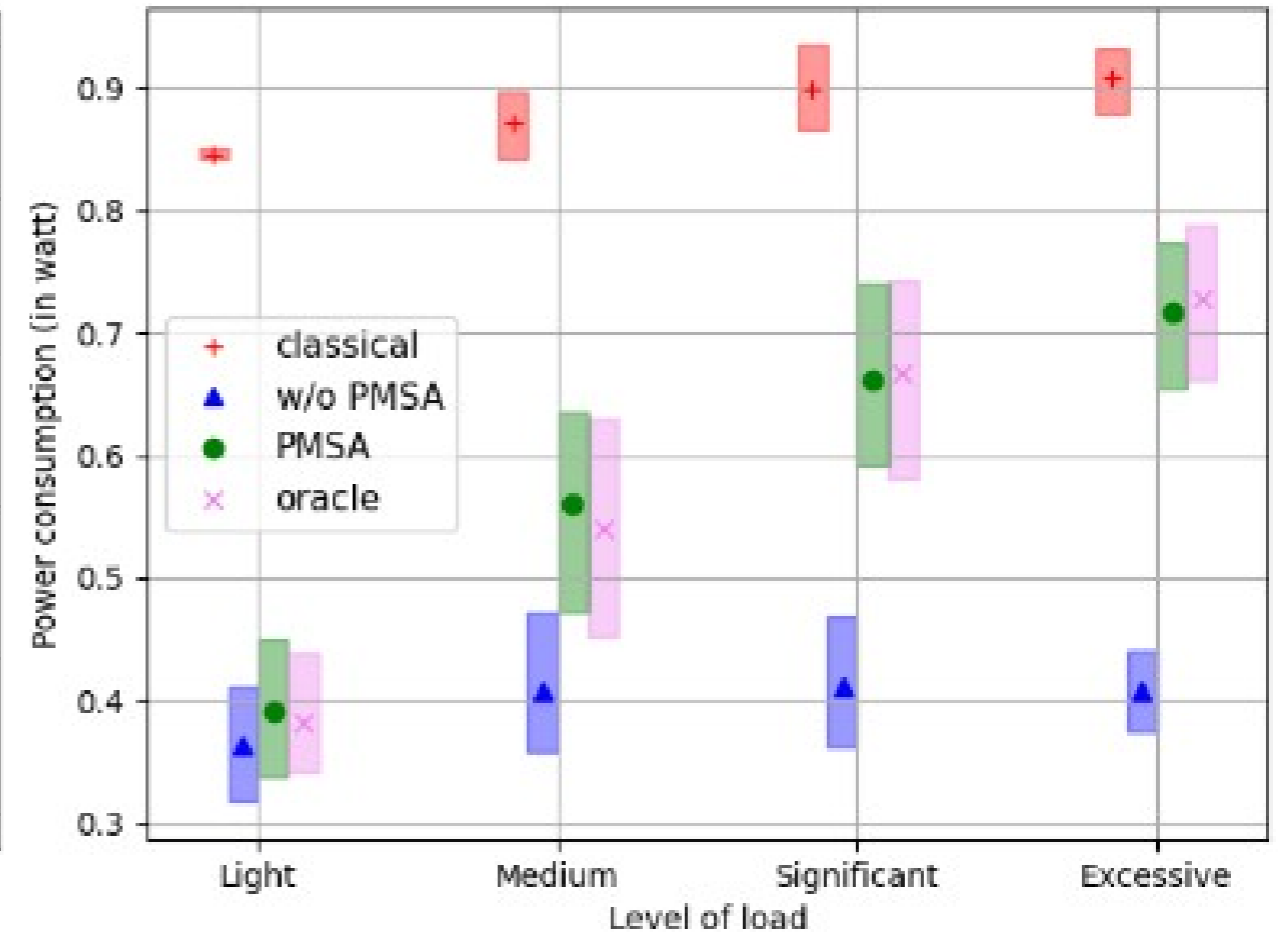
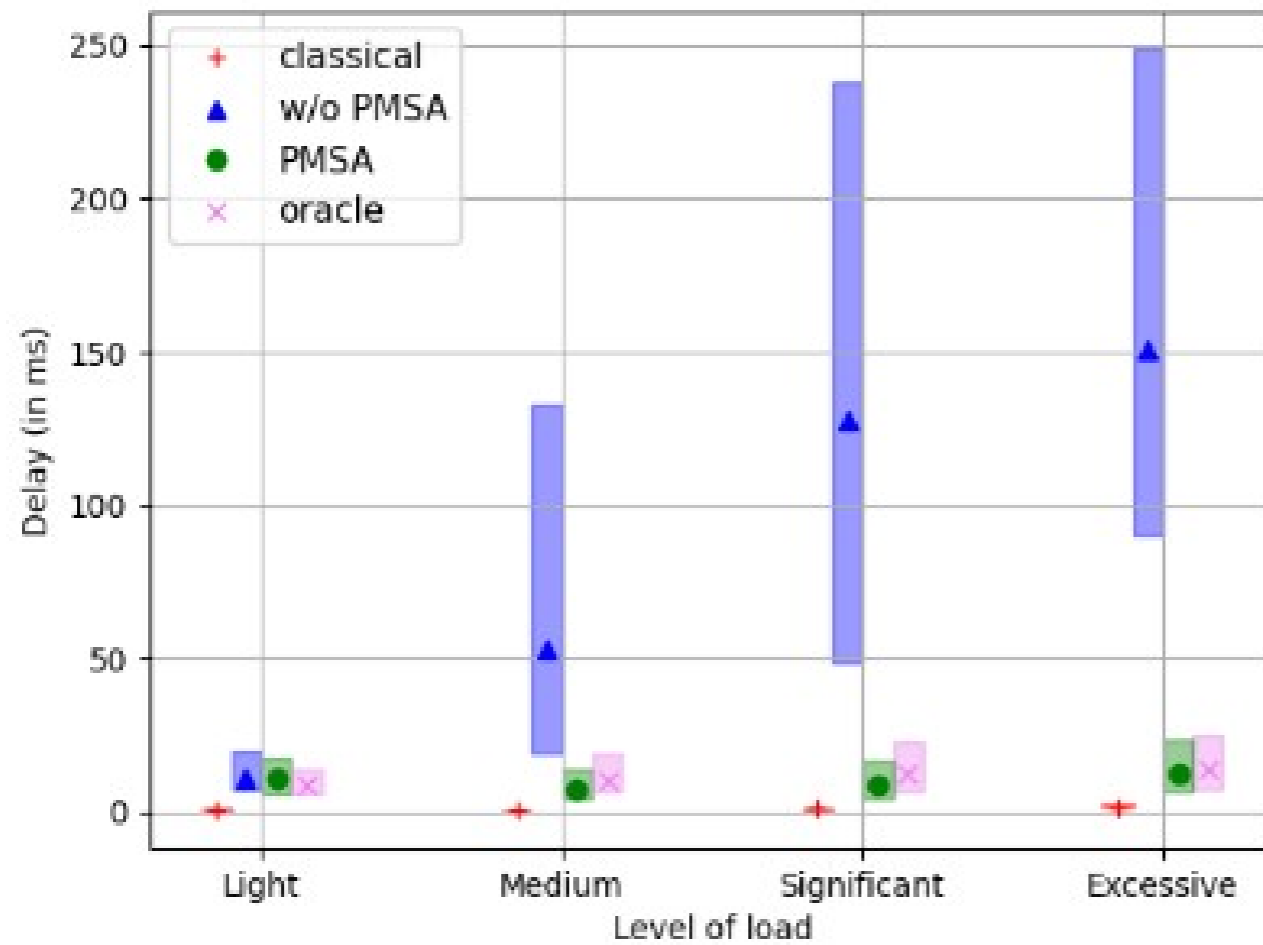
Results: PMSA under loaded conditions

- Without PMSA there are more than 60% of economy whatever the traffic rate
- With PMSA, the consumption is also divided with guarantee of performance
 - Not for the 4K streaming (same consumption)
 - The STAs with moderate traffic rate can save energy (37% of economy for IoT and 28% for video gaming)
- The coexistence between active and PS mode is going well

Results: PMSA under various loaded conditions

- We examined 600 different WLANs:
 - Light (less than 1 Mbps)
 - Medium (between 1 and 7 Mbps)
 - Significant (between 7 and 15 Mbps)
 - Excessive (more than 15 Mbps)
- $M = 5, ,$

Results: PMSA under various loaded conditions



Conclusion

- PMSA: switch mode to deal with heavy load conditions.
- With TWT scheduler, packet delays and losses can become significant under high load
- PMSA maintains limited delays and avoids packet losses, regardless of the network load level
- Future Work: Develop a method to enhance PMSA face to handle bursty traffic

Thank you ! Questions ?

**Loïc Desgeorges, Thomas Begin, Isabelle Guérin
Lassous**

loic.desgeorges@univ-lyon1.fr

Univ. Lyon I, ENS Lyon, Inria, CNRS

GDR RSD - 19 / 03 / 2026

