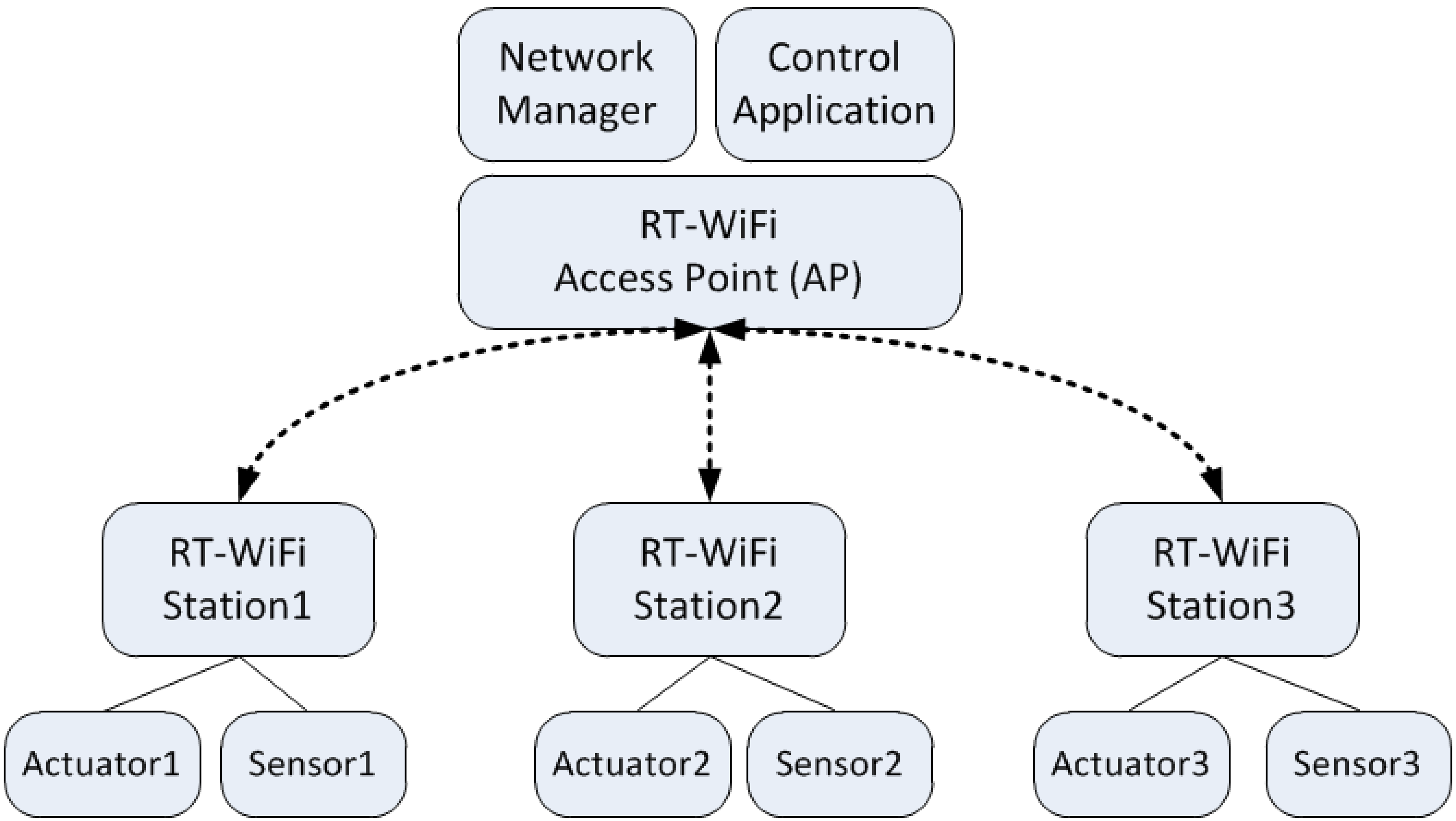
RT-WiFi: High-Speed Real-Time Communication Platform for Cyber-Physical Systems

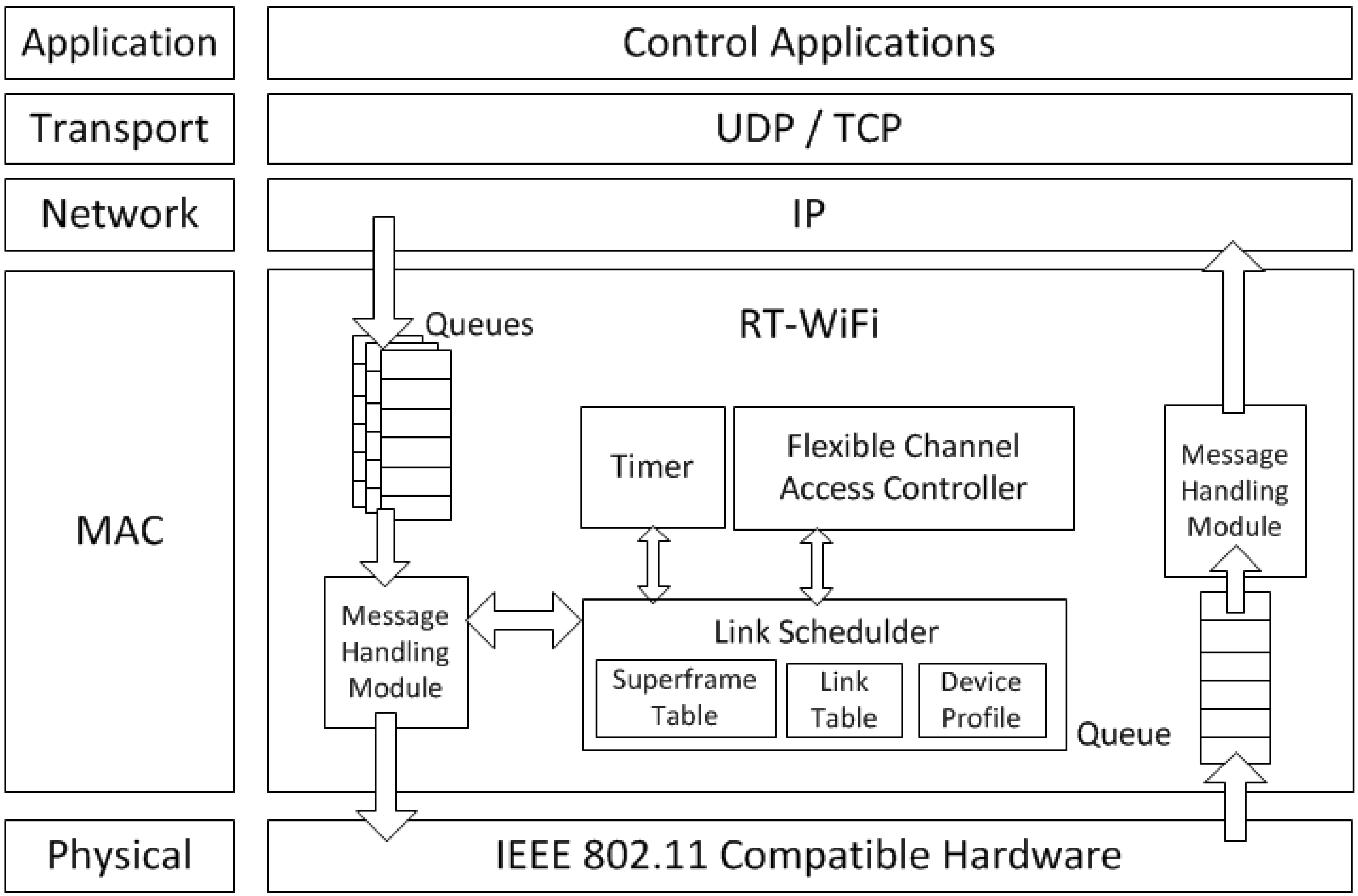
RT-WiFi Design Goals

1. Real-Time Data Delivery and High Sampling Rate
   – Aim to provide at least 1 kHz sampling rate
     • minimum requirement for many mechanical control systems
2. Flexible Configuration
   – Configurable parameters: sampling rate, predictability of real-time data delivery, reliability, co-existence with regular WiFi networks
3. Transparent System Design
   – Use commercial-off-the-shelf network card
   – Transparent to upper layer protocols

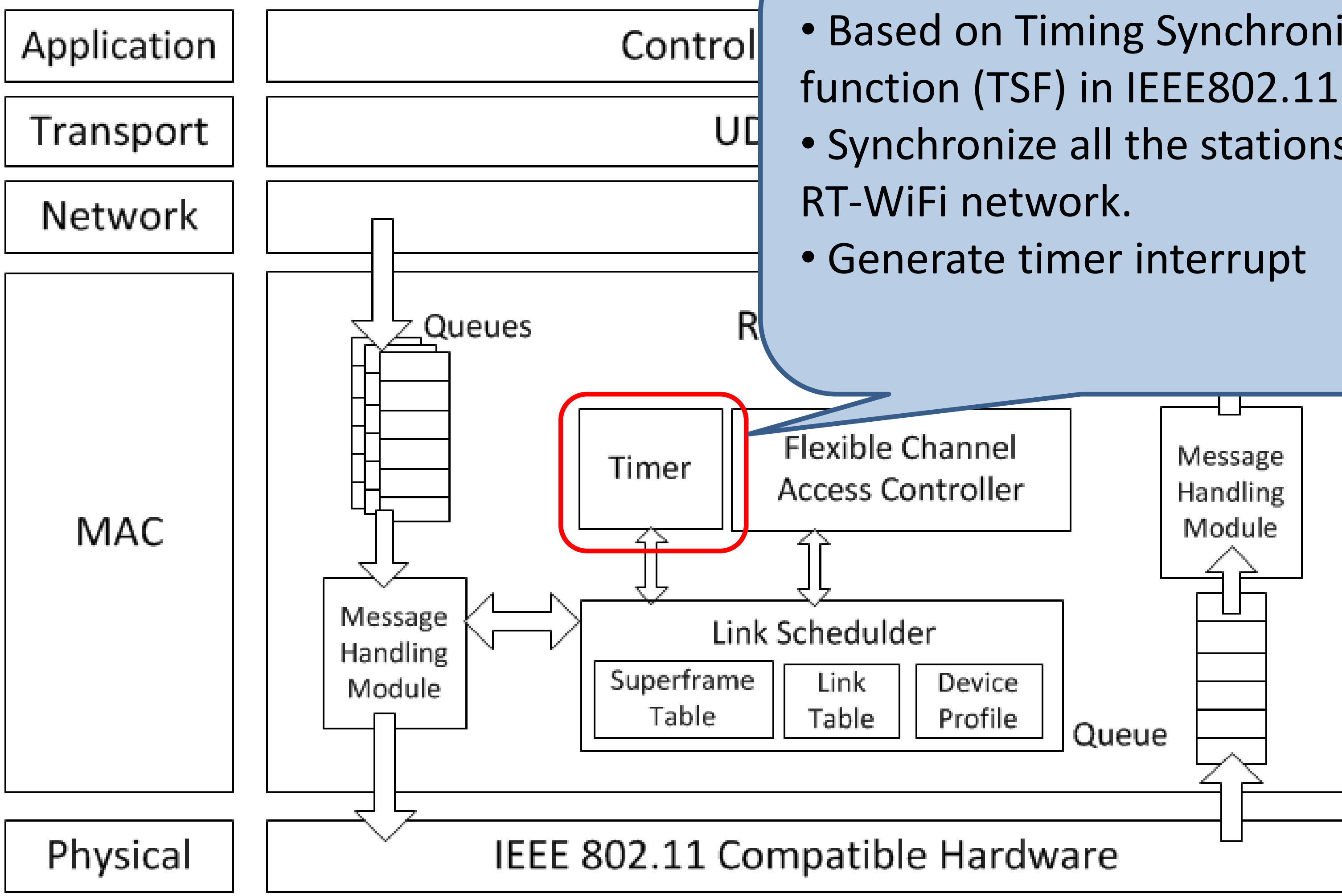
Overview of a Control System using RT-WiFi Network



Architecture of RT-WiFi Protocol



Architecture of RT-WiFi Protocol

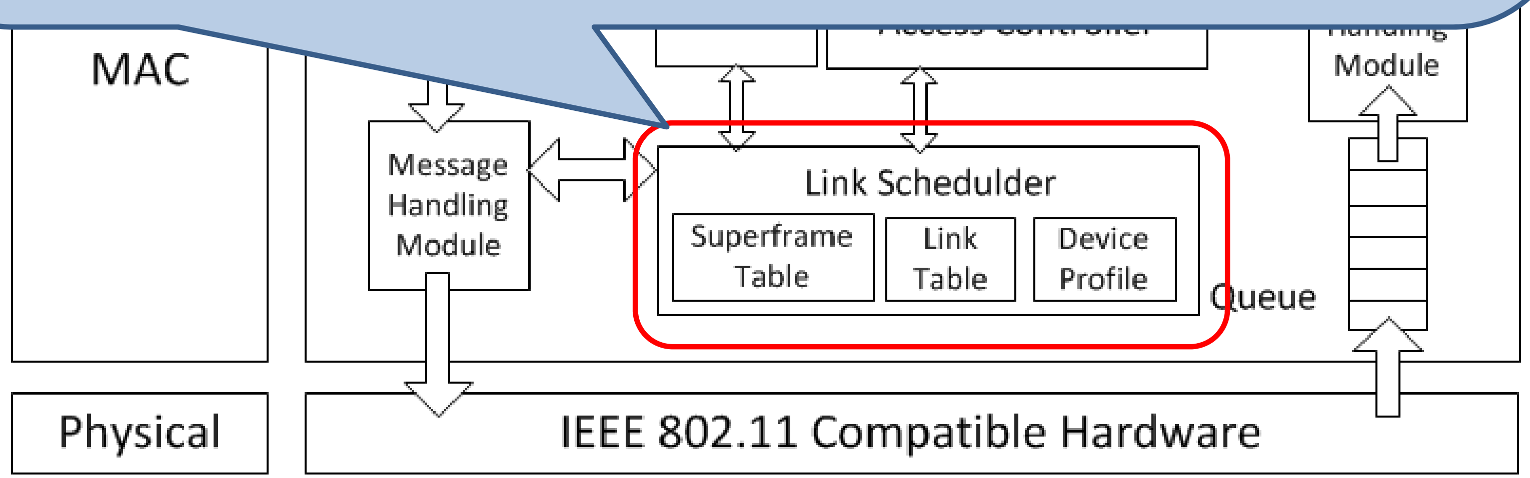


- Based on Timing Synchronization function (TSF) in IEEE802.11
- Synchronize all the stations in the RT-WiFi network.
- Generate timer interrupt

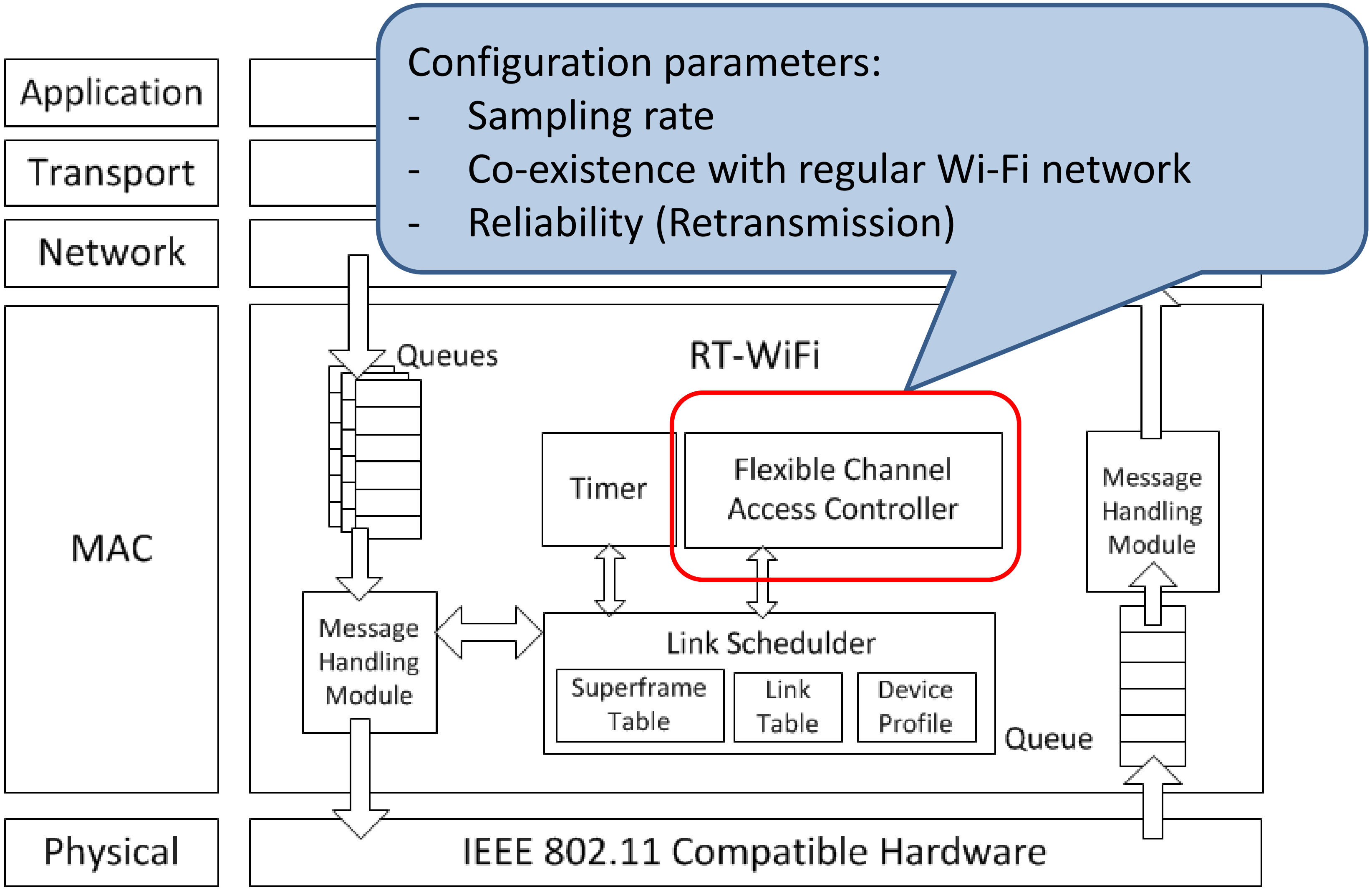
Architecture of RT-WiFi Protocol

- Coordinate channel access among the stations
- Link: Broadcast link, transmit link, receive link, shared link
- Superframe:

Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
AP Broadcast	Shared	STA1 ⇩ AP	AP ⇩ STA1	STA2 ⇩ AP	AP ⇩ STA2	STA3 ⇩ AP	AP ⇩ STA3



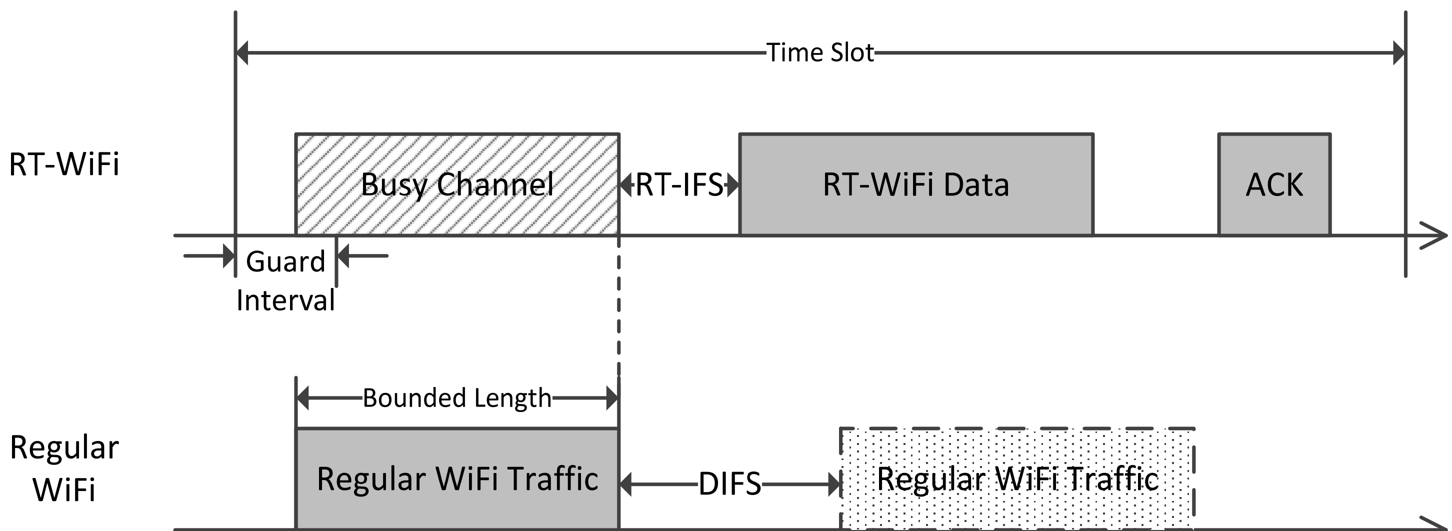
Architecture of RT-WiFi Protocol



Enabling Co-existence with Regular WiFi

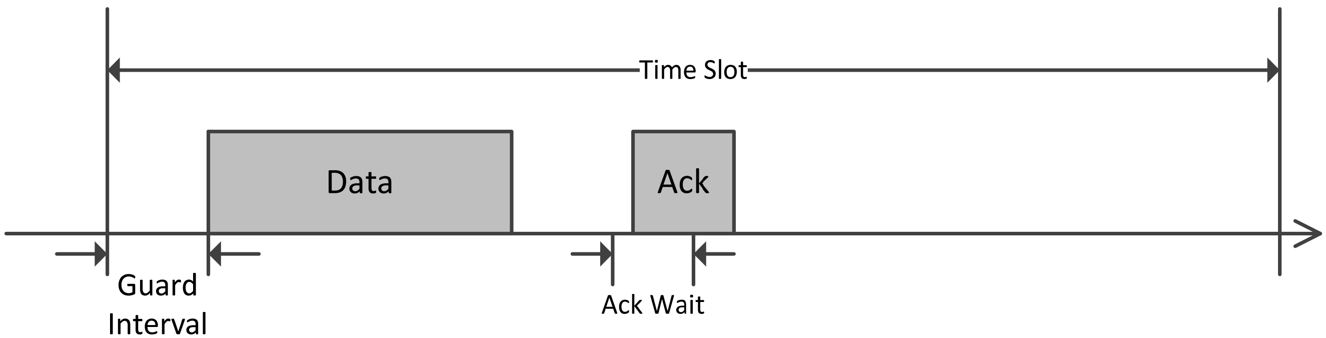
1. Assume bounded length for regular WiFi data frames to ensure bounded latency
   – Limit maximum transmission unit to at most some upper bound
   – Limit lowest data rate to at least some lower bound

2. For RT-WiFi:
   – Enable carrier sense
   – Use shorter interframe space (IFS)

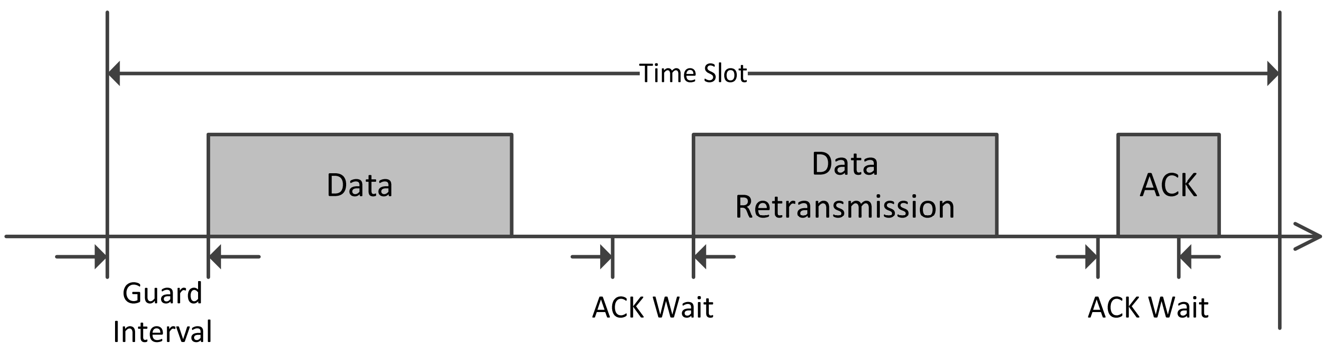


Increasing Reliability in RT-WiFi transmission

- Retransmission
  - In-slot retransmission
  - Out-of-slot retransmission



(a) Regular time slot



(b) Time slot with in-slot retransmission

RT-WiFi Testbed setup

Setup:

RT-WiFi baseline vs. regular WiFi

UDP socket program to emulate control applications

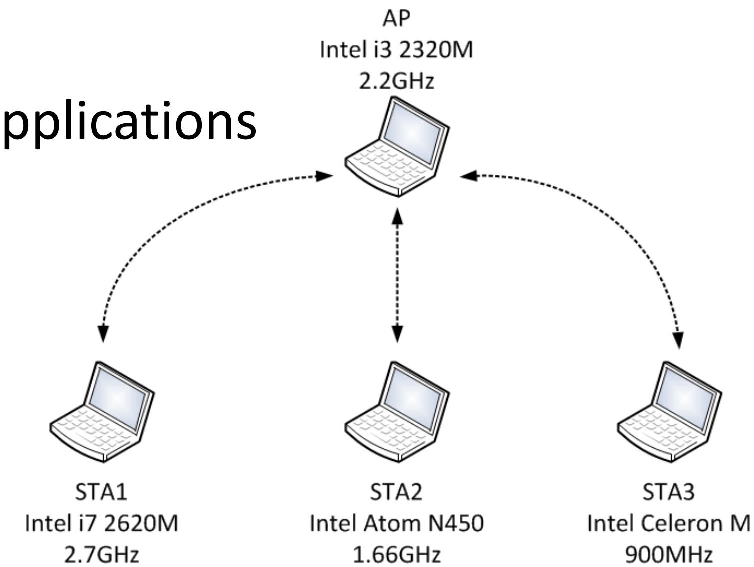
Metrics:

MAC layer to MAC layer latency

Packet loss ratio

Environment:

Interference free environment vs. office environment



Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
AP Broadcast	Shared	STA1 ↓ AP	AP ↓ STA1	STA2 ↓ AP	AP ↓ STA2	STA3 ↓ AP	AP ↓ STA3

Timeslot size: 500 µs

Results of RT-WiFi baseline (TDMA only) in Interference Free Environment

Link	Max Latency(µs)		Mean Latency(µs)		Latency stdev. (µs)		Loss Ratio	
	RT-WiFi	Wi-Fi	RT-WiFi	Wi-Fi	RT-WiFi	Wi-Fi	RT-WiFi	Wi-Fi
STA1->AP	535	16448	173	191	4.88	231.60	0.19%	0%
STA2->AP	529	13387	172	181	2.52	214.15	0.16%	0%
STA3->AP	525	13589	174	202	3.01	236.75	0.21%	0%
AP->STA1	827	16472	184	250	5.29	342.24	0.06%	0%
AP->STA2	544	17465	187	298	3.98	360.66	0.04%	0%
AP->STA3	1055	17049	188	248	4.87	325.50	0.01%	0%

16-32x

47-90x

Results of RT-WiFi baseline (TDMA only) in Office Environment

Link	Max Latency(μs)		Mean Latency(μs)		Latency stdev. (μs)		Loss Ratio	
	RT-WiFi	Wi-Fi	RT-WiFi	Wi-Fi	RT-WiFi	Wi-Fi	RT-WiFi	Wi-Fi
STA1->AP	3865	100078	176	401	25.86	1491.69	8.64%	0%
STA2->AP	4193	81499	171	348	27.62	1000.60	9.97%	0%
STA3->AP	3861	75298	174	429	25.16	1221.72	7.55%	0%
AP->STA1	1197	78089	184	788	16.86	2861.42	9.52%	0%
AP->STA2	1342	78923	189	790	15.19	2806.56	8.09%	0%
AP->STA3	2186	77860	189	799	19.03	2855.89	9.31%	0%

2.0-4.3x

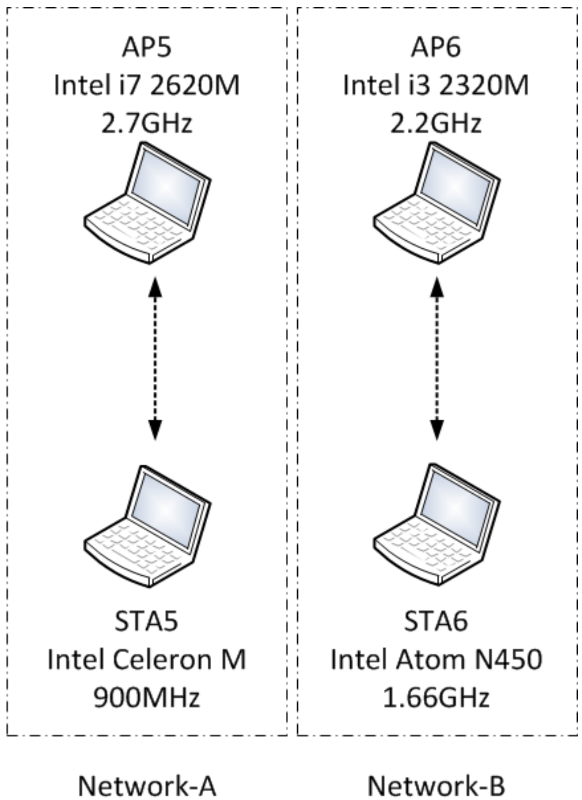
36-184x

Flexible Channel Access Controller Experiment Setup

- **Network A:**
  - Regular WiFi network
  - 10 Mbps UDP traffic generated by iperf
- **Network B:**
  - UDP program to emulate control applications
  - Compare:
    - Regular WiFi
    - RT-WiFi baseline
    - RT-WiFi with co-existence enabled
    - RT-WiFi with co-existence enabled and one in-slot retransmission enabled
- **Metrics:**

  MAC to MAC layer latency

  Packet loss ratio

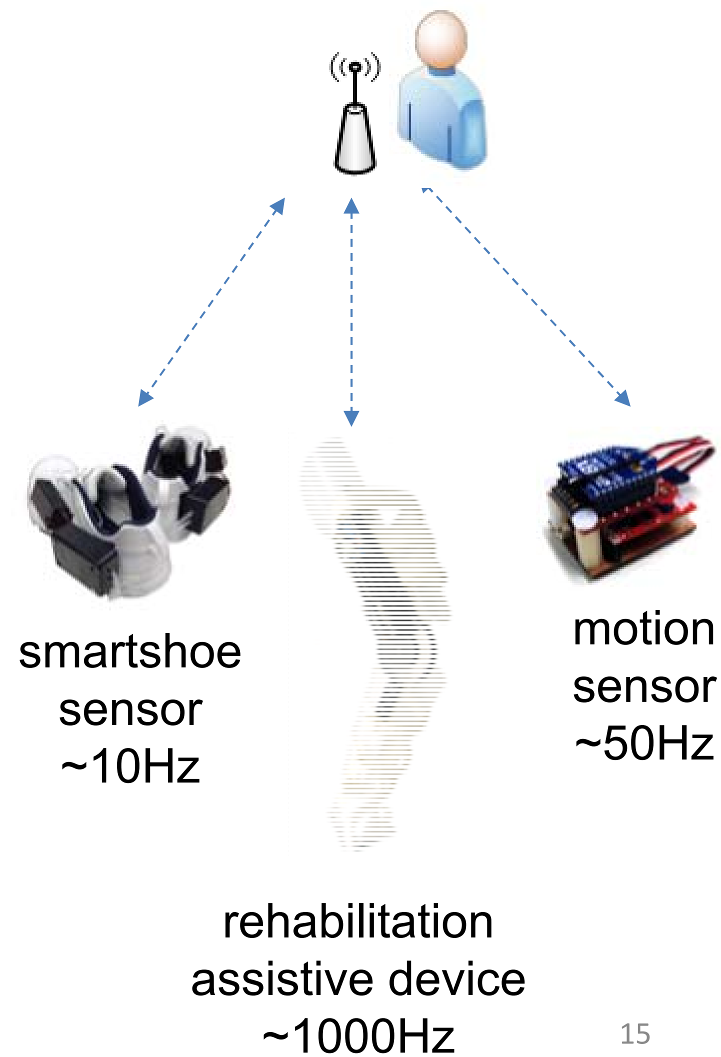


Flexible Channel Access Controller Experiment Results

	Max Latency (µs)	Mean Latency (µs)	Latency Stdev. (µs)	Loss Ratio
Regular WiFi	62629	580	1679.04	0%
RT-WiFi baseline	953	183	27.75	50.21%
RT-WiFi co-ex	2507	220	149.27	10.92%
RT-WiFi co-ex + retry	2831	254	166.37	4.96%

Data Link Layer Schedule Assignment for Minimizing Communication Jitter

- TDMA communication scheduler design for wireless CPS applications.
  - Meet diverse application and system requirements
    - Application performance requirements
    - Hardware capability
  - Minimize communication jitters
    - Easier controller design
    - Better application performance



RT-WiFi Communication Task Model

- Model communication tasks in an RT-WiFi network as a periodic task set $T = \{T_i\}_{i=1}^{n}$
- Each task T_i is defined as $T_i = (P_i^{min}, P_i^{max}, C_i)$
  - P_i^{min}: minimum sampling period supported by the task : determined by hardware and software capability
  - P_i^{max}: maximum sampling period required by the task: determined by controller requirement
  - C_i: number of time slots for data transmission
  - P_i^{min}, P_i^{max} and C_i are integers, representing the number of time slots
  - Scheduler allows preemption only at time slot boundaries.

Static Link Schedule Assignment

- Each task T_i consists of an infinite sequence of jobs. Each job is a fixed number (C_i) of data fragments to be transmitted, with each fragment taking one time slot to transmit.
- **Jitter** is defined as the variation of jobs' inter-completion times.
- Given a communication task set $T = \{T_i\}_{i=1}^{n}$ with $T_i = \left(P_i^{min}, P_i^{max}, C_i\right)$, the **Jitter-Free Scheduling (JFS) Problem** is to determine the period P_i and phasing $\{\phi_{i,1}, \dots, \phi_{i,C_i}\}$ for each task T_i, so that:
  1) $P_i^{min} \leq P_i \leq P_i^{max}$
  2) The *j*-th fragment of task T_i is scheduled at ($\phi_{i,j} + P_i \cdot k$)-th time slot, where $k = 0, 1, 2, \dots$
  3) only one fragment is scheduled at a time slot
  4) The network utilization $U = \sum_{i=1}^{n} \frac{C_i}{P_i}$ is minimized
- JFS problem is NP-hard

Harmonic Chain Based Jitter-Free (HCJF) Scheduler

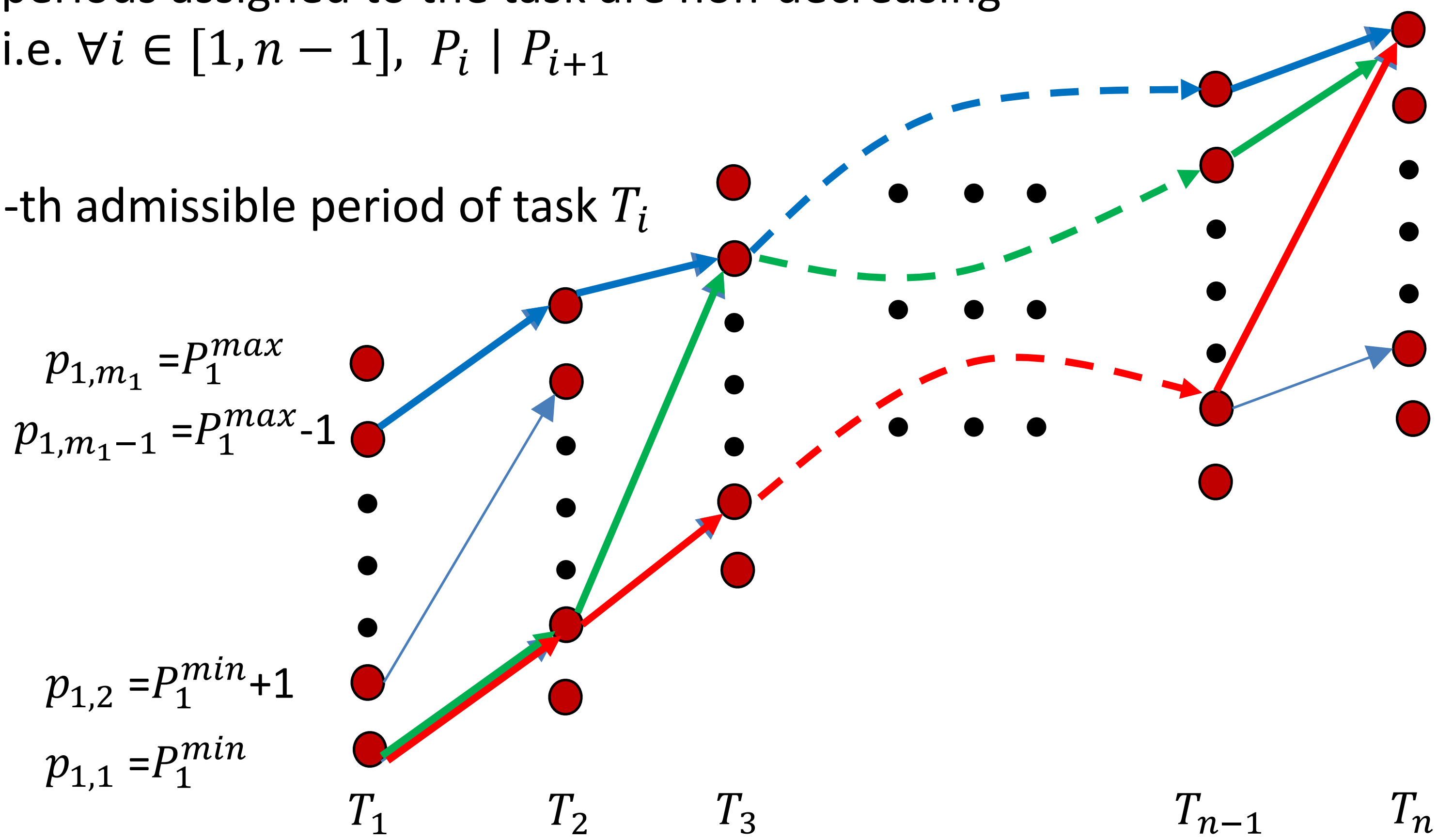
- Find a sufficient condition that yields a pseudo-polynomial time solution to the JFS problem.
- **Stage 1: Period Selection**
  - Select period $P_i \in [P_i^{min}, P_i^{max}]$ for each task, so that all the $\{P_i\}_{i=1}^{n}$ form a harmonic chain and the network utilization $U = \sum_{i=1}^{n} \frac{C_i}{P_i}$ is minimized.

  [A set S of positive integers forms a **harmonic chain** if and only if $\forall x, y \in S, (x|y) \vee (y|x)$.]
- **Stage 2: Phasing Assignment**
  - Assign phasing using a greedy algorithm

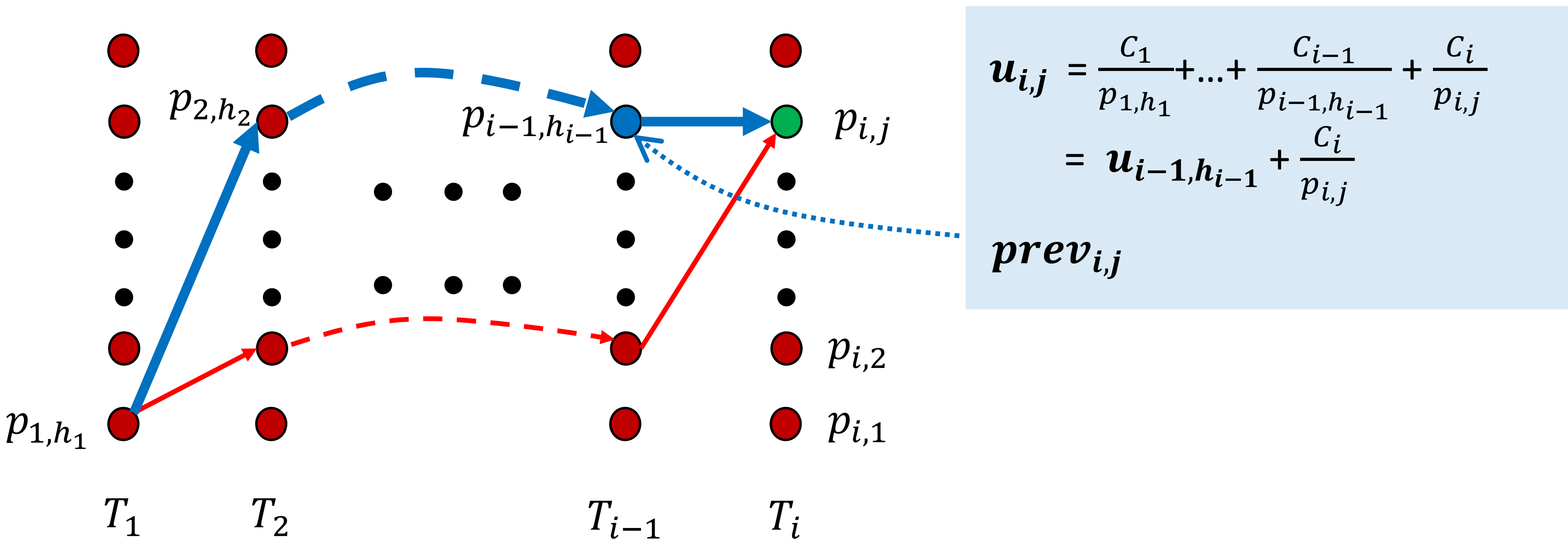
Period Selection Algorithm

- Without loss of generality, assume: for $i > j$:
  $P_i^{max} > P_j^{max}$, OR $P_i^{max} = P_j^{max}$ and $P_i^{min} \geq P_j^{min}$
- Under this assumption, we have proved that the periods assigned to the task are non-decreasing
  i.e. $\forall i \in [1, n-1],\ P_i \mid P_{i+1}$

$p_{i,j}$: j-th admissible period of task T_i



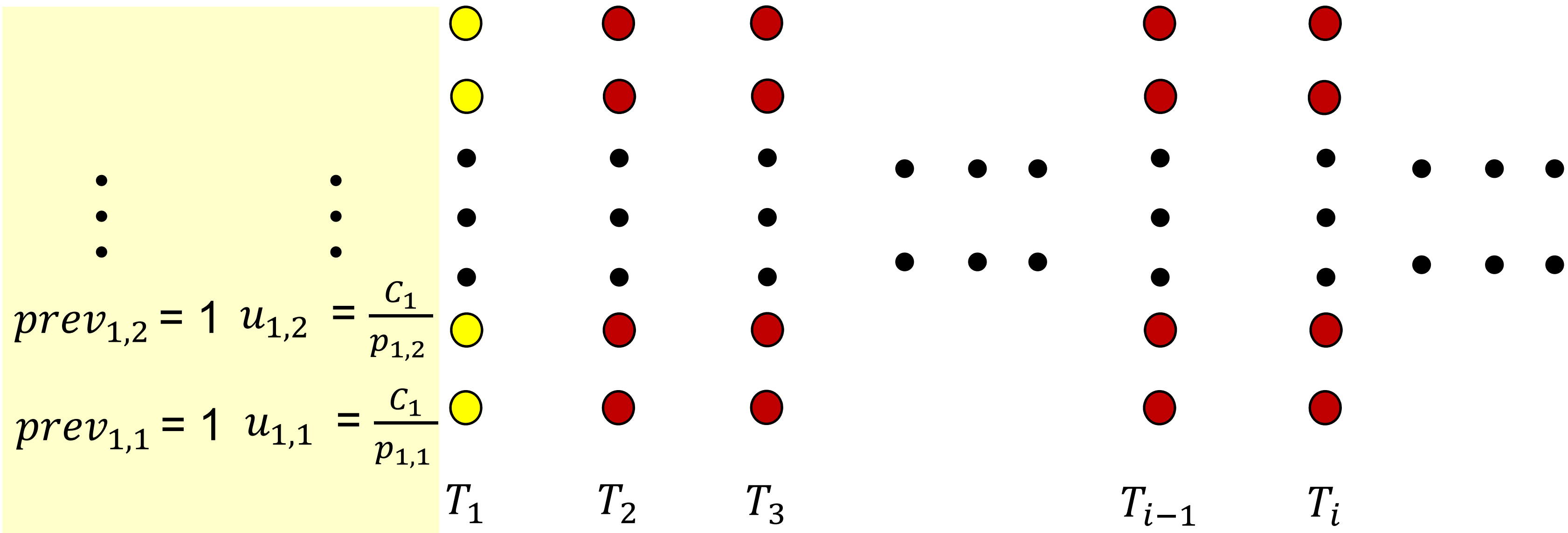
Period Selection Algorithm



If $p_{i,j}$ can form a harmonic chain with some admissible periods from T_1 to T_{i-1}, then:

- $\boldsymbol{u_{i,j}}$ stores the minimum utilization of these harmonic chain
- $\boldsymbol{prev_{i,j}}$ is the backpointer to the previous node in the harmonic chain with minimum utilization

Period Selection Algorithm

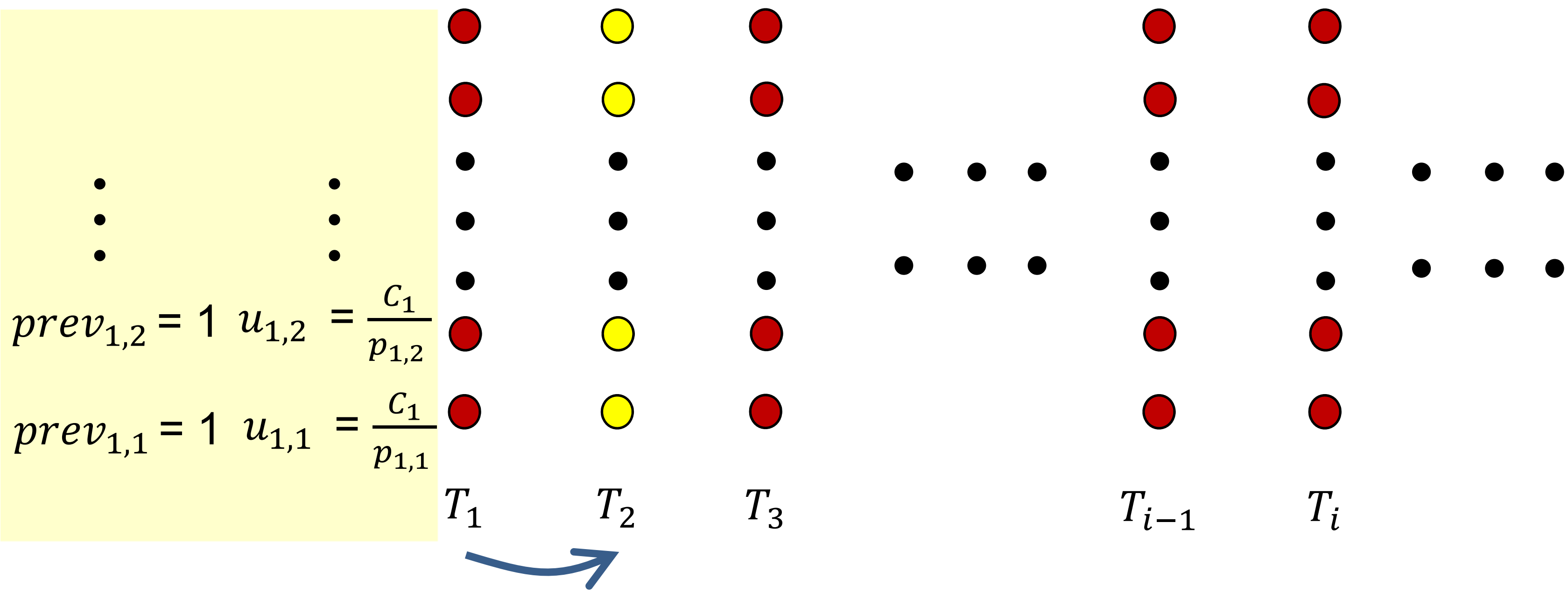


Initialization:

$$u_{1,j} = \frac{C_i}{p_{1,j}} \quad 1 \leq j \leq P_1^{max} - P_1^{min} + 1$$

$$prev_{1,j} = 1$$

Period Selection Algorithm



<u>Recursion:</u> Let set S = $\{x \mid (p_{i-1,x} \mid p_{i,j}) \wedge (prev_{i-1,x} \neq \text{null})\}$

If S $\neq \emptyset$:

$$u_{i,j} = \frac{C_i}{P_{i,j}} + \min_{y \in S} u_{i-1,y}$$

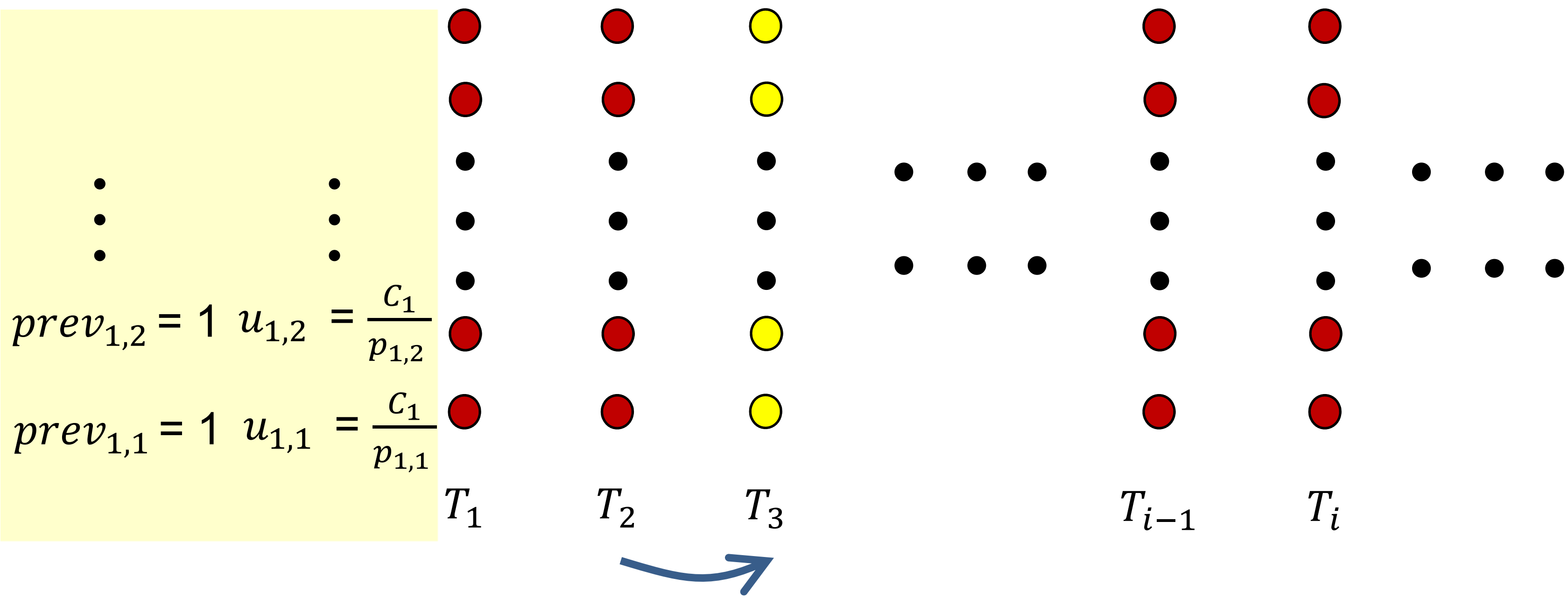
$$prev_{i,j} = \text{argmin}_{y \in S}\, u_{i-1,y}$$

If S = $\emptyset$:

$$u_{i,j} = +\infty$$

$$prev_{i,j} = \text{null}$$

Period Selection Algorithm



<u>Recursion:</u> Let set S = $\{x \mid (p_{i-1,x} \mid p_{i,j}) \wedge (prev_{i-1,x} \neq \text{null})\}$

If S $\neq \emptyset$:

$$u_{i,j} = \frac{C_i}{P_{i,j}} + \min_{y \in S} u_{i-1,y}$$

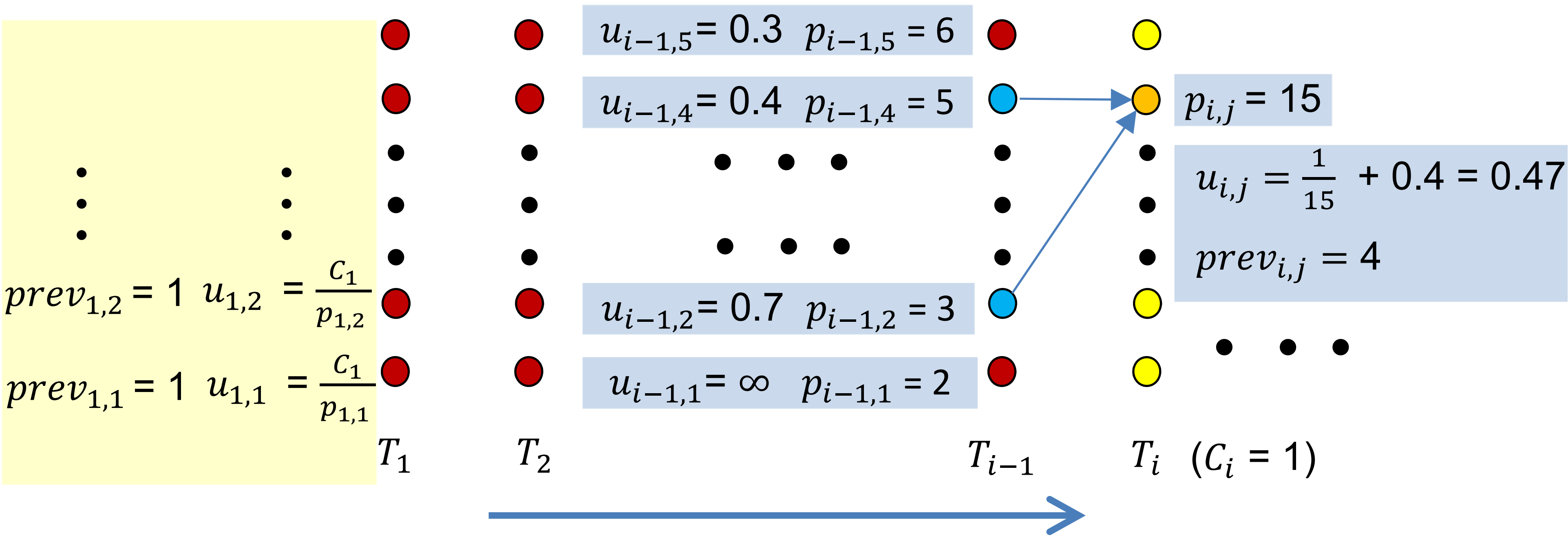
$$prev_{i,j} = \text{argmin}_{y \in S}\, u_{i-1,y}$$

If S = $\emptyset$:

$$u_{i,j} = +\infty$$

$$prev_{i,j} = \text{null}$$

Period Selection Algorithm



<u>Recursion:</u> Let set S = $\{x \mid (p_{i-1,x} \mid p_{i,j}) \wedge (prev_{i-1,x} \neq \text{null})\}$

If S $\neq \emptyset$:

$$u_{i,j} = \frac{C_i}{P_{i,j}} + \min_{y \in S} u_{i-1,y}$$

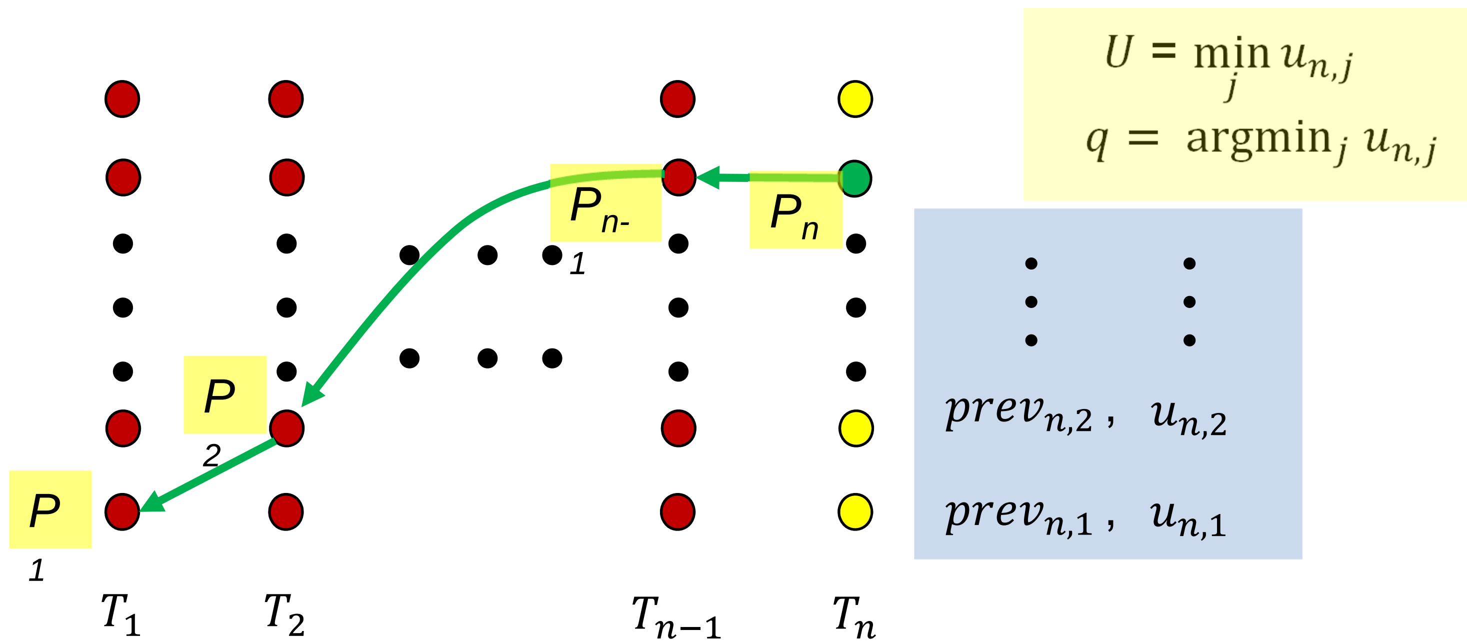
$$prev_{i,j} = \text{argmin}_{y \in S}\, u_{i-1,y}$$

If S = $\emptyset$:

$$u_{i,j} = +\infty$$

$$prev_{i,j} = \text{null}$$

Period Selection Algorithm



Termination:

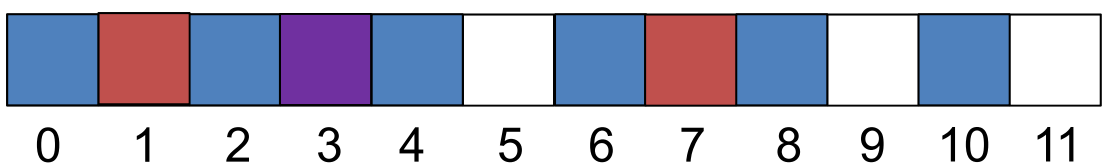
$U = \min_j u_{n,j} \quad 1 \le j \le m_n \quad q = \text{argmin}_j\, u_{n,j}$

If $U = +\infty$: Harmonic chain not found

If $U \neq +\infty$: The optimal network utilization is U, $P_n = p_{n,q}$, then follow the backpointers $prev$ to construct the full period sequence $P_{n-1}, P_{n-2}, \ldots P_1$

Phasing Assignment in Static Network

- Create a superframe with size P_n
- Assign phasing from T_1 to T_n in sequence
- Assign the first unallocated phasing in the schedule to a fragment of a task
- The task set is schedulable if the network utilization U is less than or equal to 1.



T_1: P_1= 2, C_1= 1 φ_1= 0
T_2: P_2= 6, C_2= 1 φ_2= 1
T_3: P_3= 12, C_3= 1 φ_3= 3

Dynamic Network Scheduling

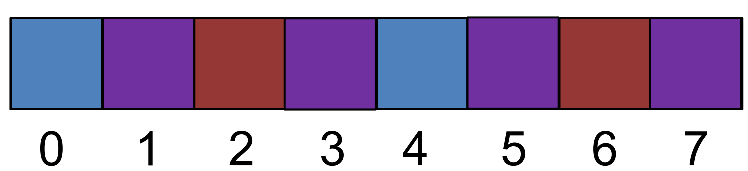
- Devices may join and leave the network frequently
- No prior knowledge on future requests
- Existing schedule may need to be adjusted to admit new tasks

Example:

At t_1, T_1: P_1= 4, C_1= 1 joins. φ_1= 0

At t_2, T_2: P_2= 4, C_2= 1 joins. ~~φ_2= 1~~ φ_2= 2

At t_3, T_3: P_3= 2, C_3= 1 joins. φ_3= 1



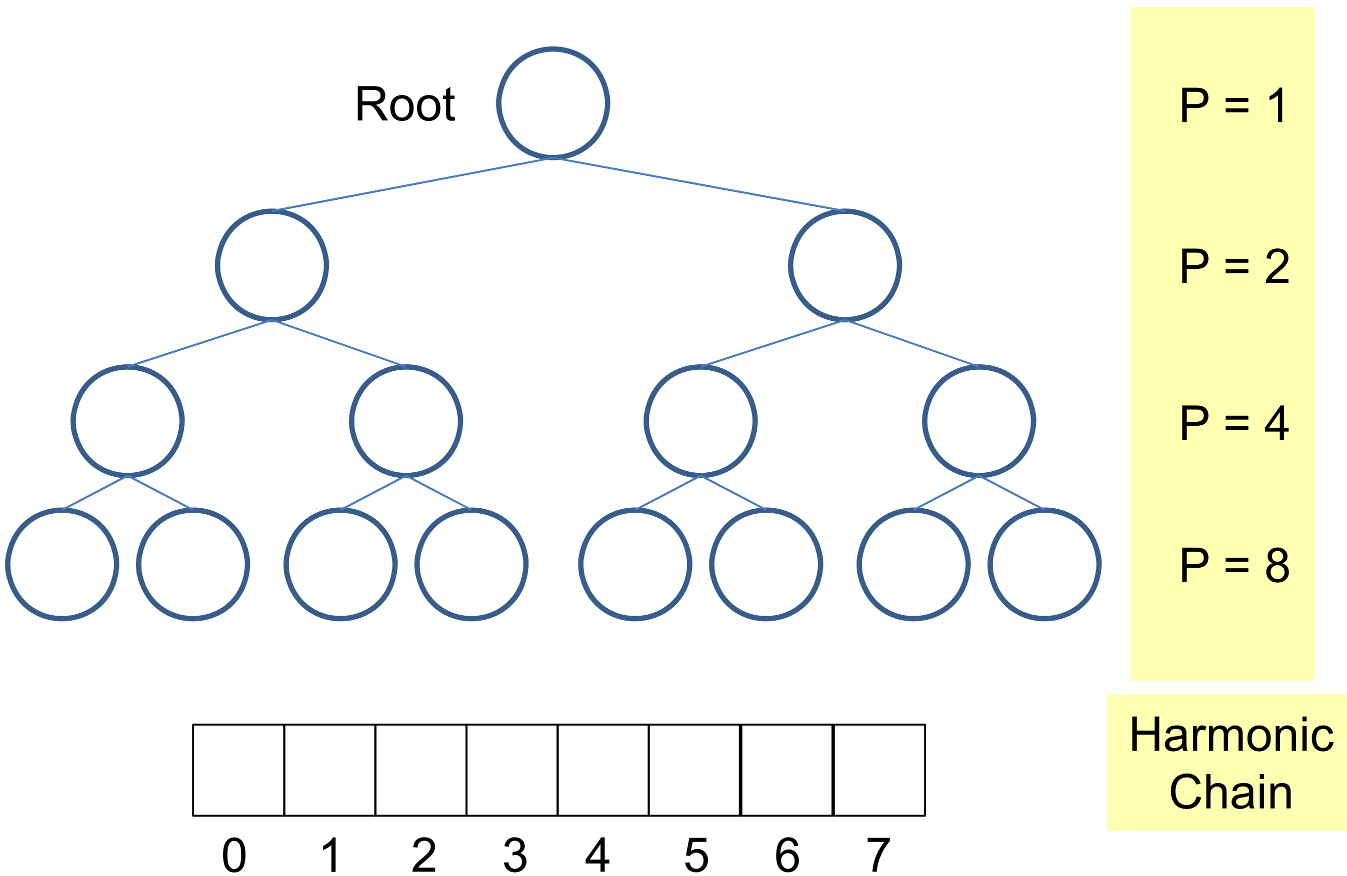
Dynamic Network Scheduling

- Devices may join and leave the network frequently
- No prior knowledge on future requests
- Existing schedule may need to be adjusted to admit new tasks

- Adjusting an existing schedule will cause:
  - network configuration overhead
  - control overhead to adjust sensor/actuator
- An efficient on-line scheduling algorithm to minimize schedule adjustment overhead is needed.

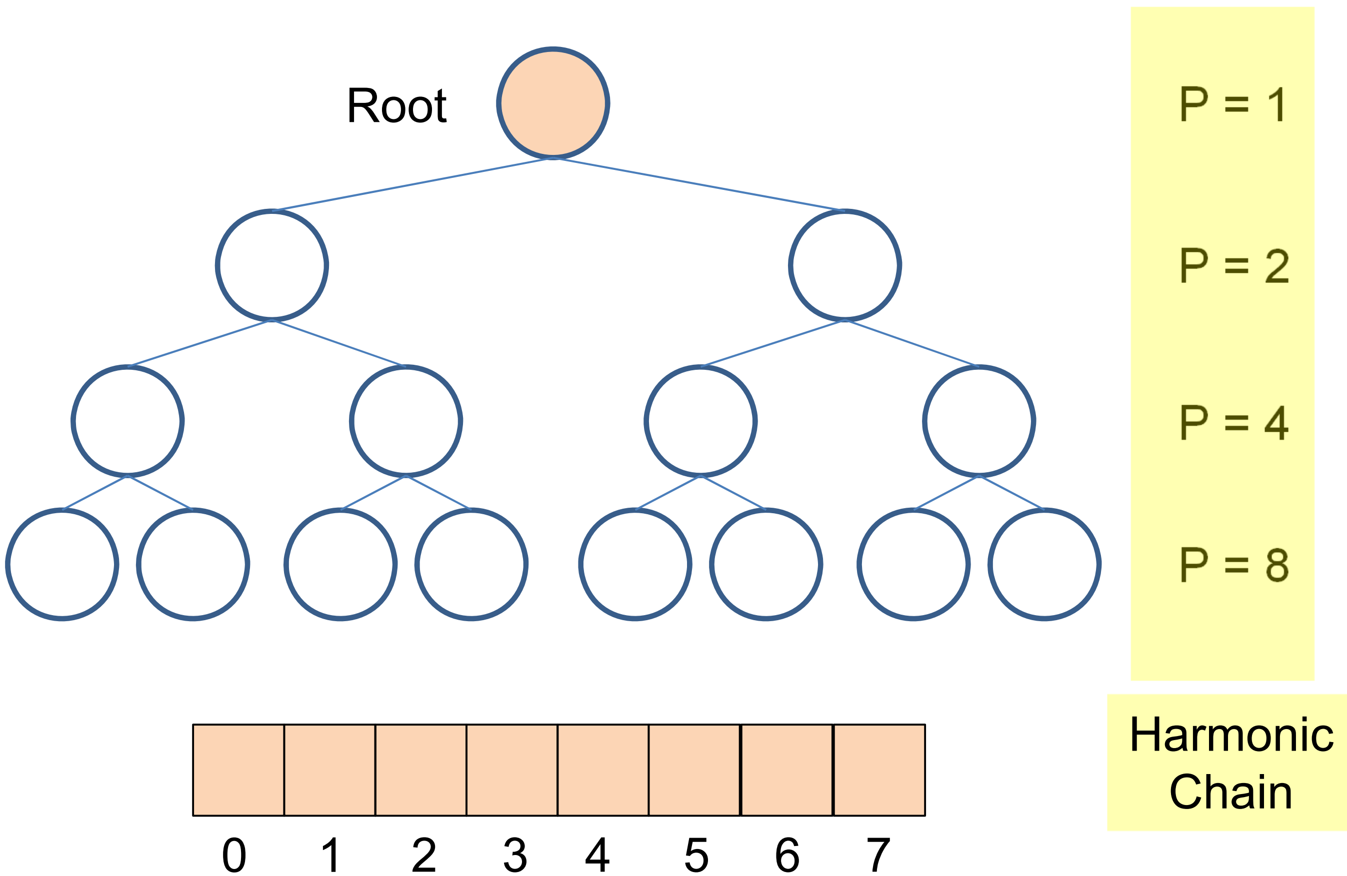
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule



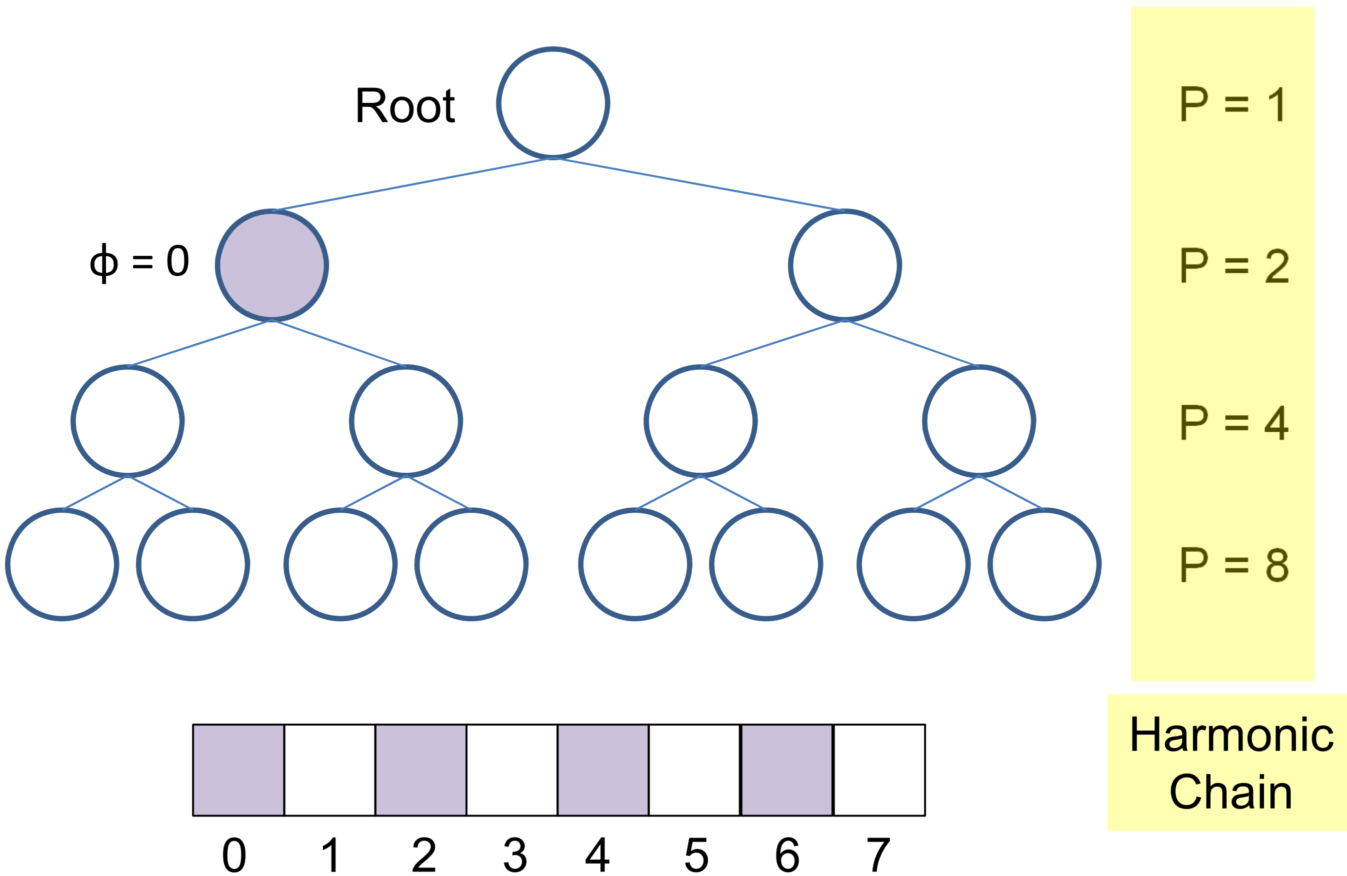
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule



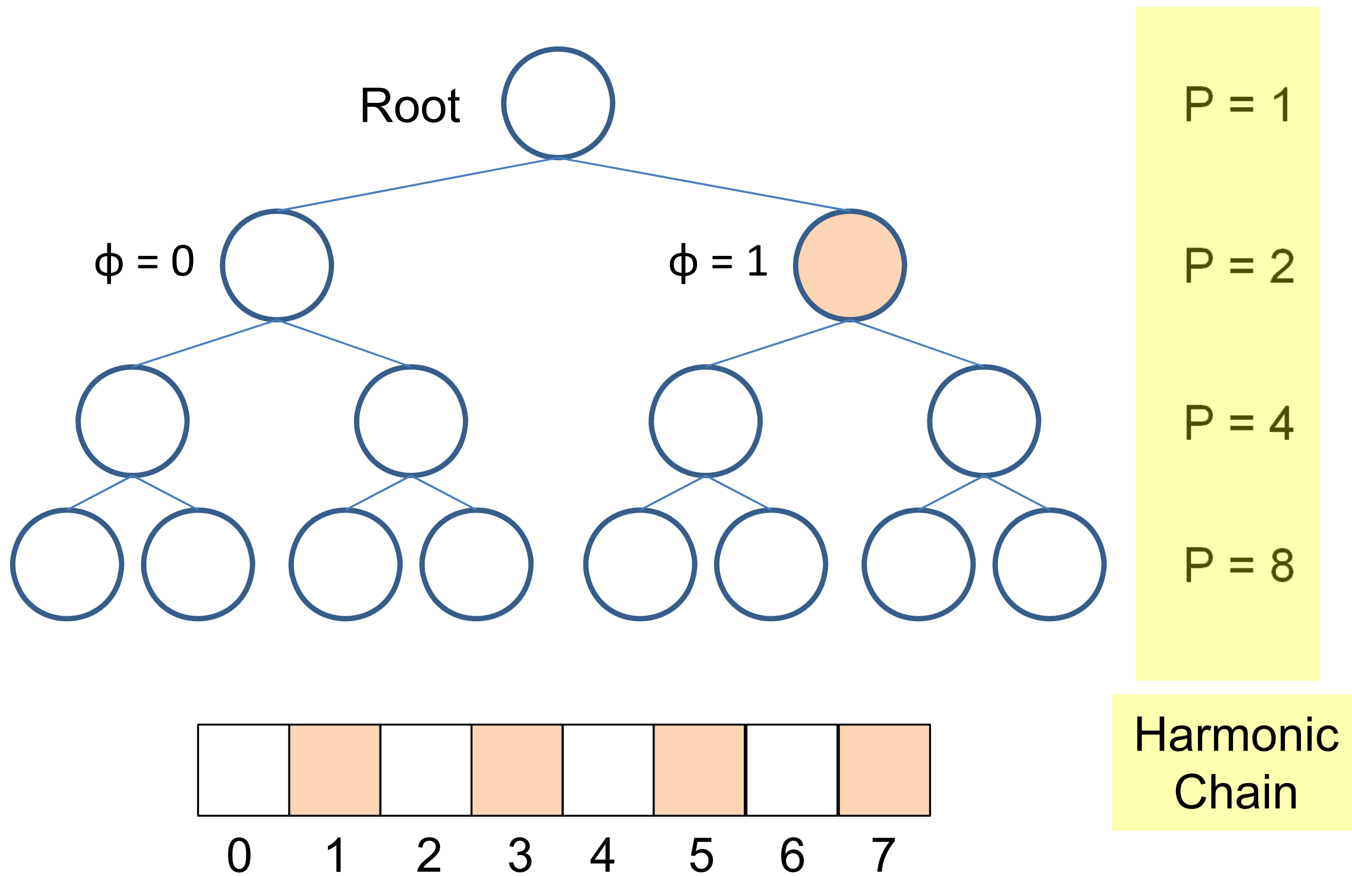
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule



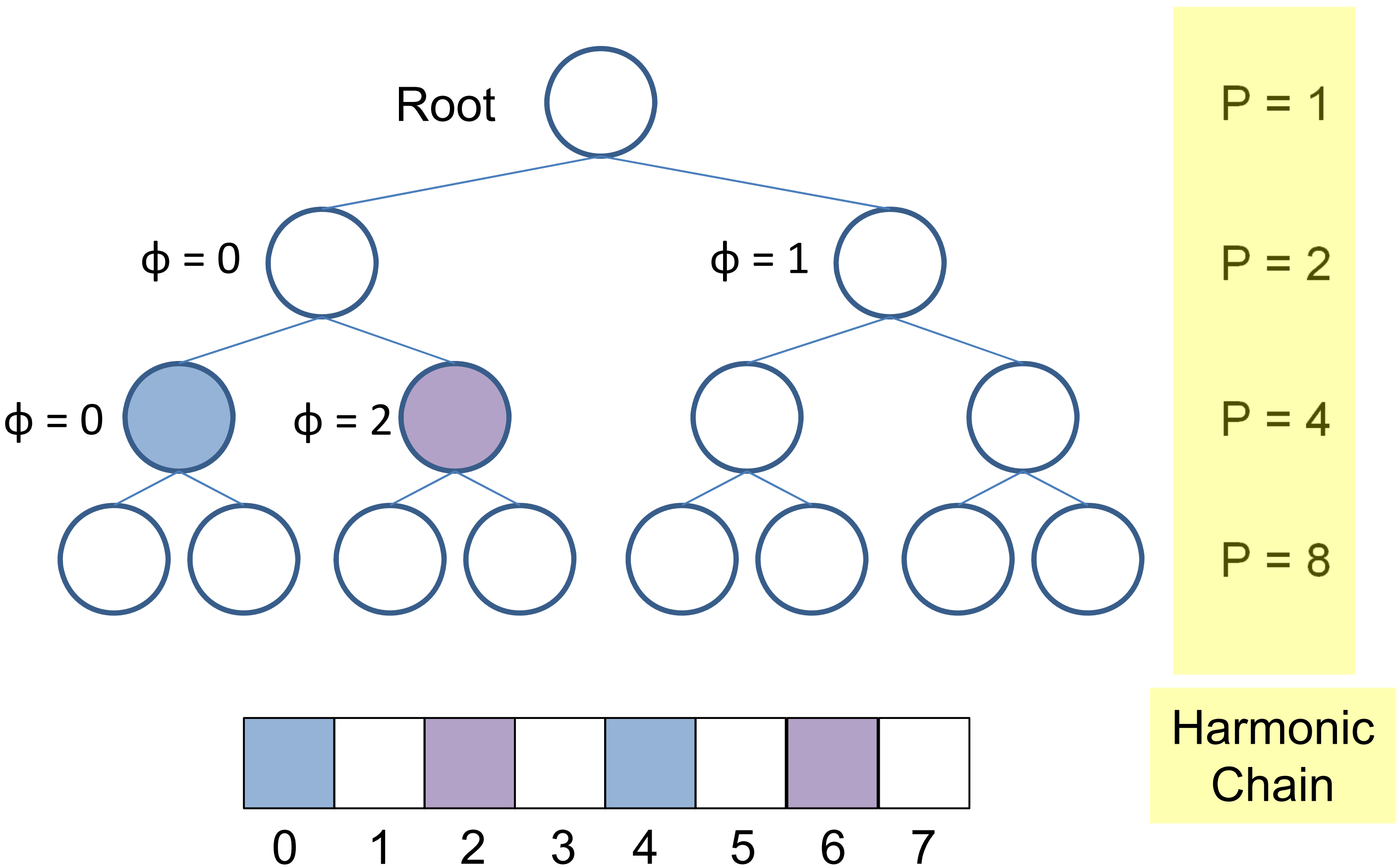
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule



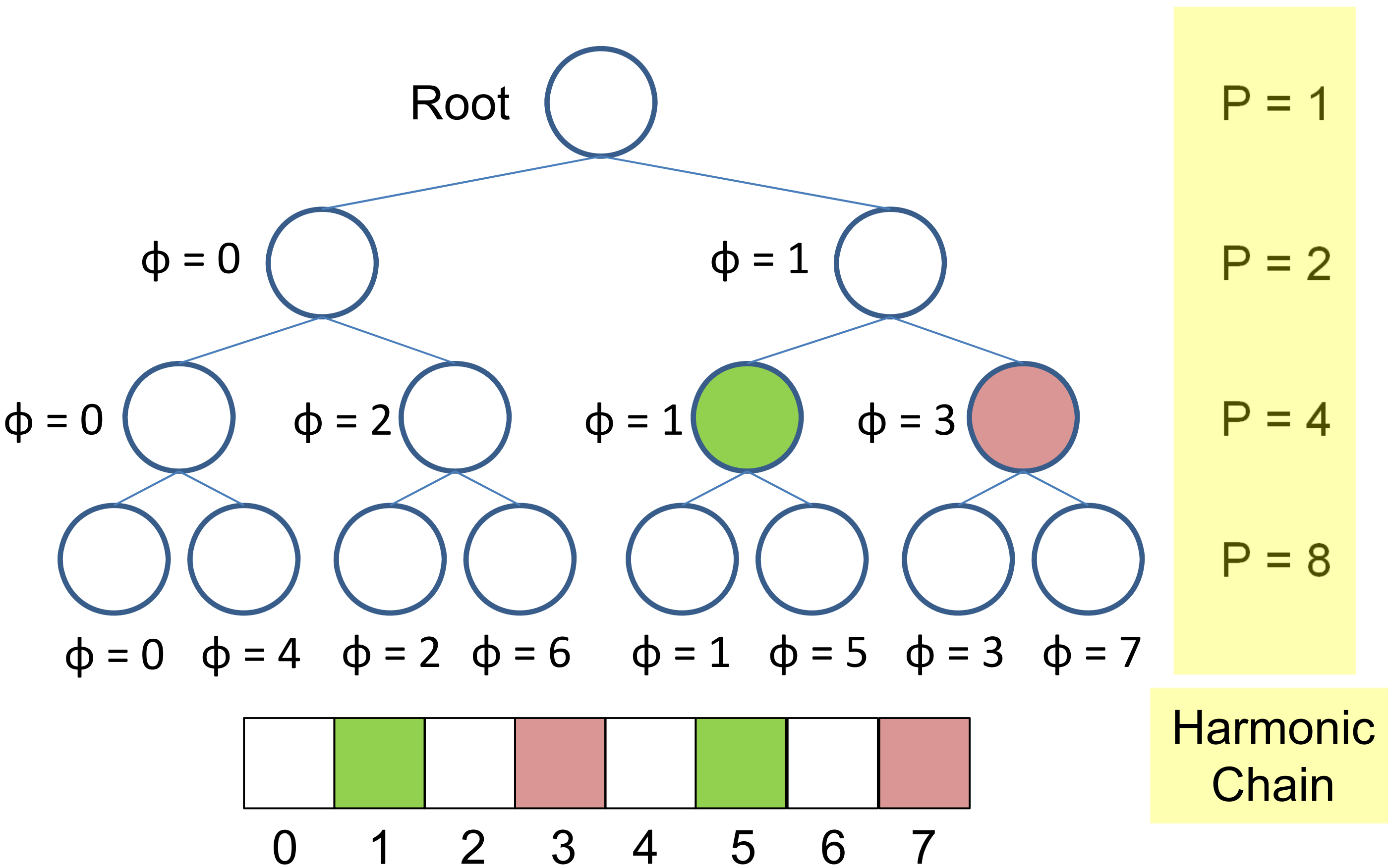
Dynamic Phasing Assignment

- S-tree: a data structure to represent the schedule

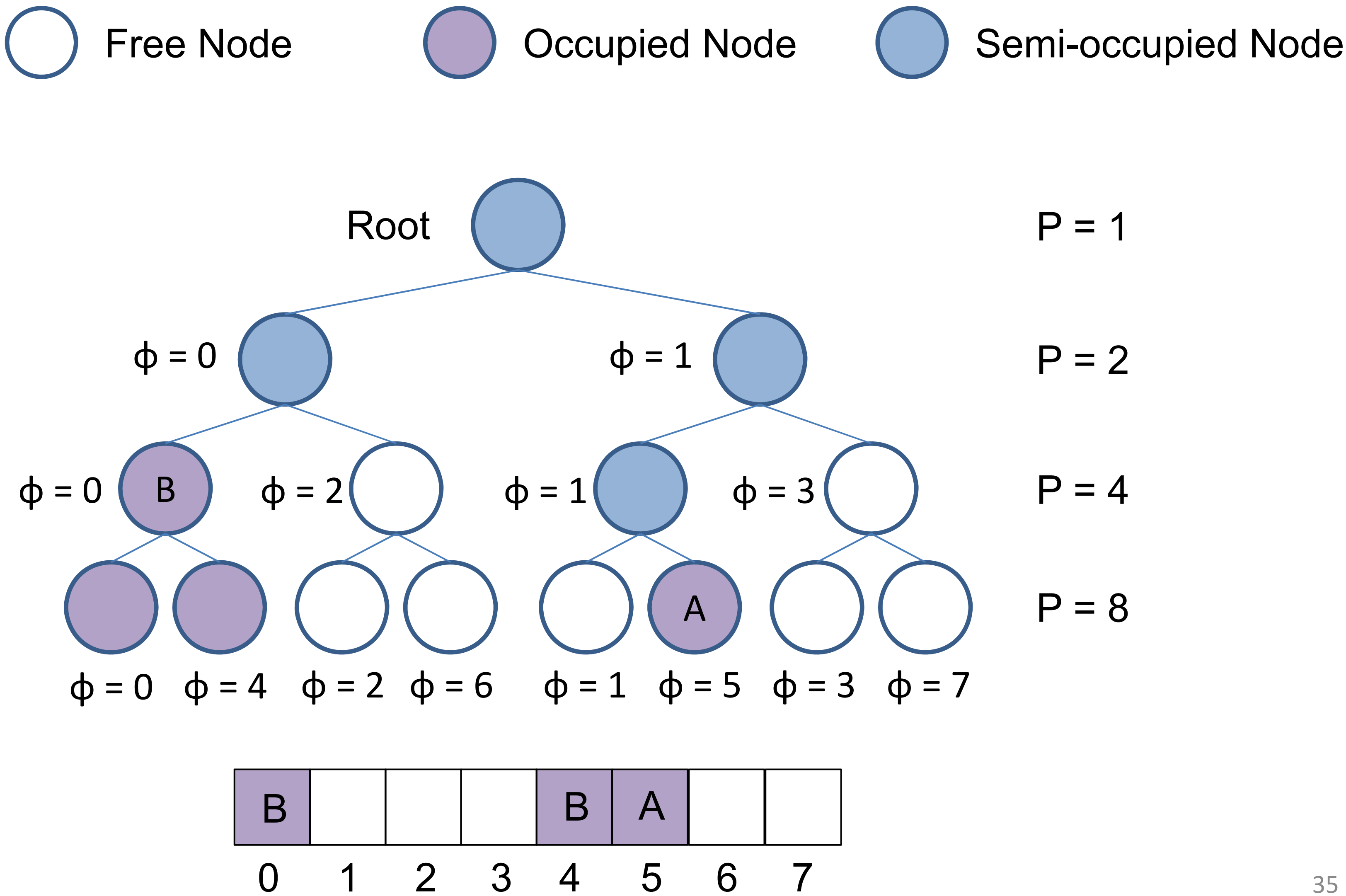


Dynamic Phasing Assignment

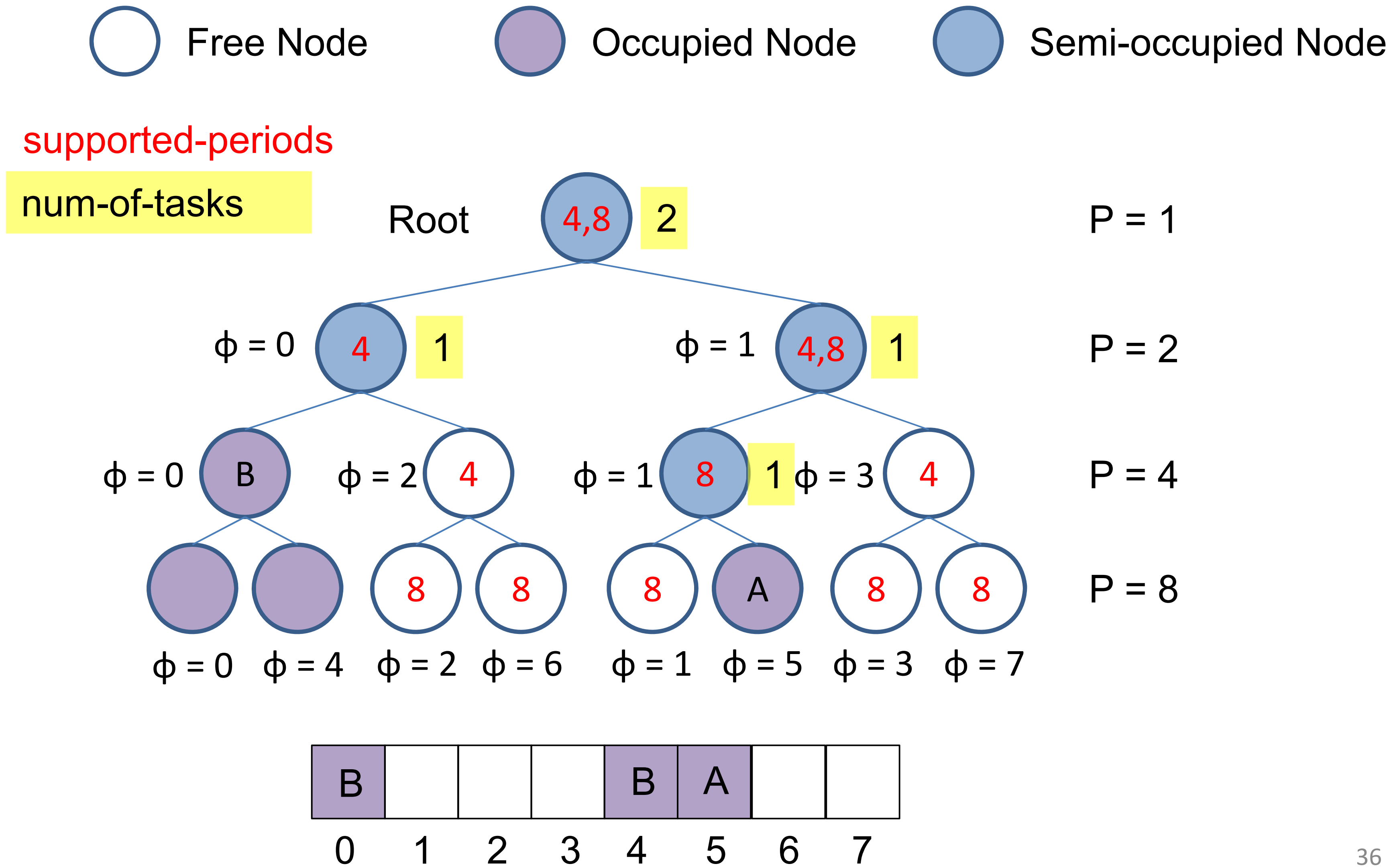
- S-tree: a data structure to represent the schedule



S-tree Node Status



S-tree Node Attribute:
supported-periods & num-of-tasks

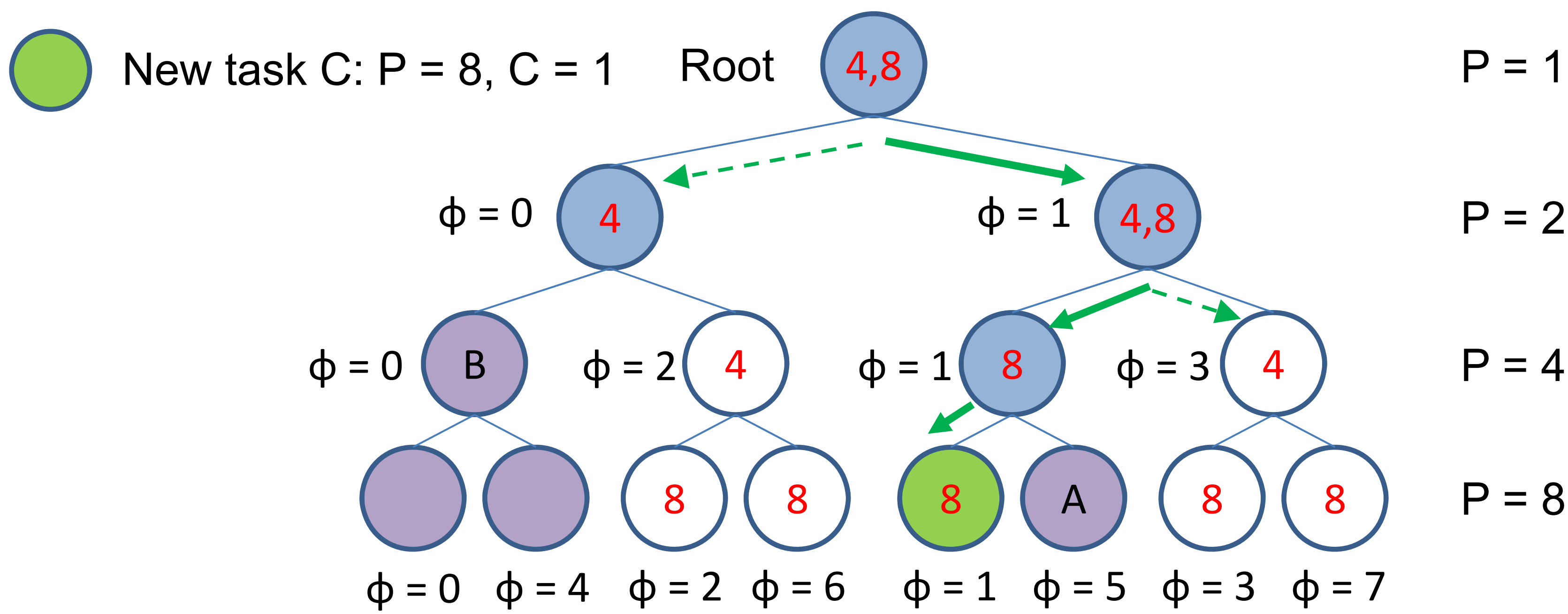


Add-task in S-tree

<u>Assignment Policy:</u>

To add a task with period P_i: start from the root, follow a path with supported-period ≤ P_i until we find a free node with period P_i

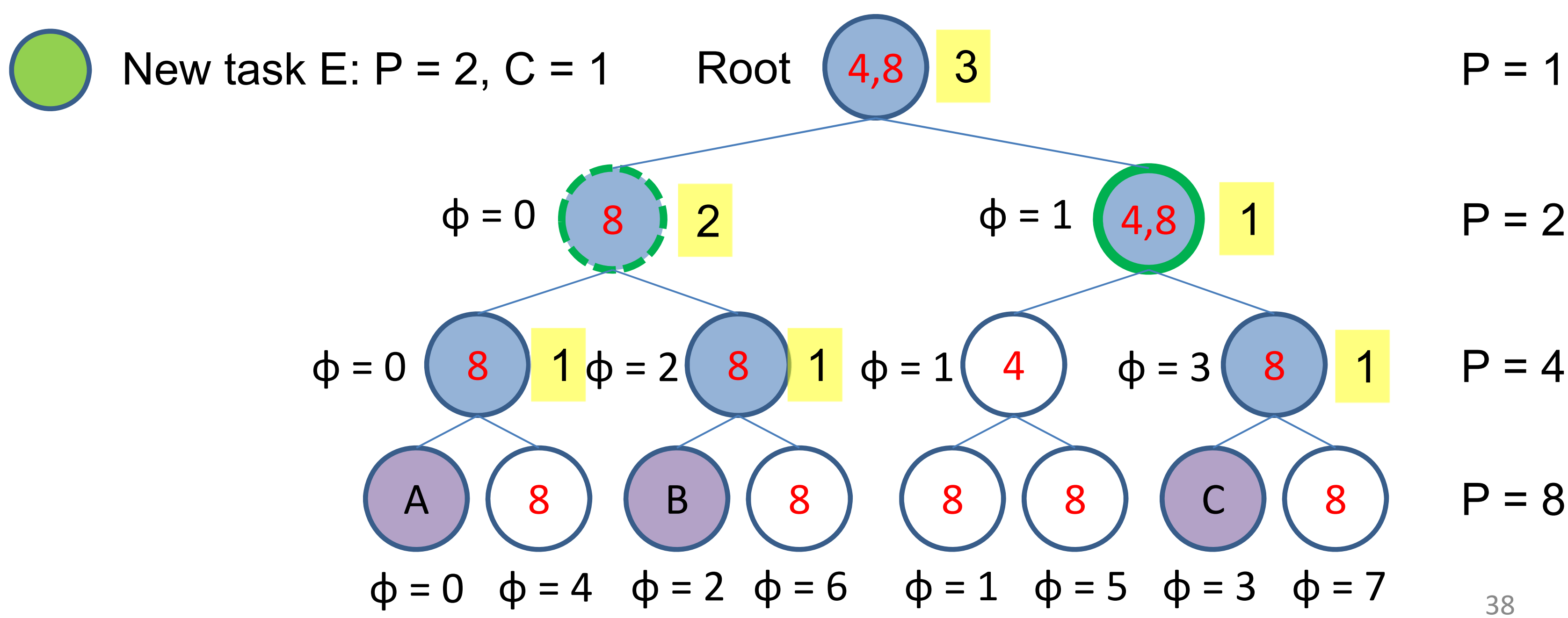
When more than one node can support the new task, select the node whose supported-period is closest to P_i.



Add-task in S-tree

Replacement Policy:

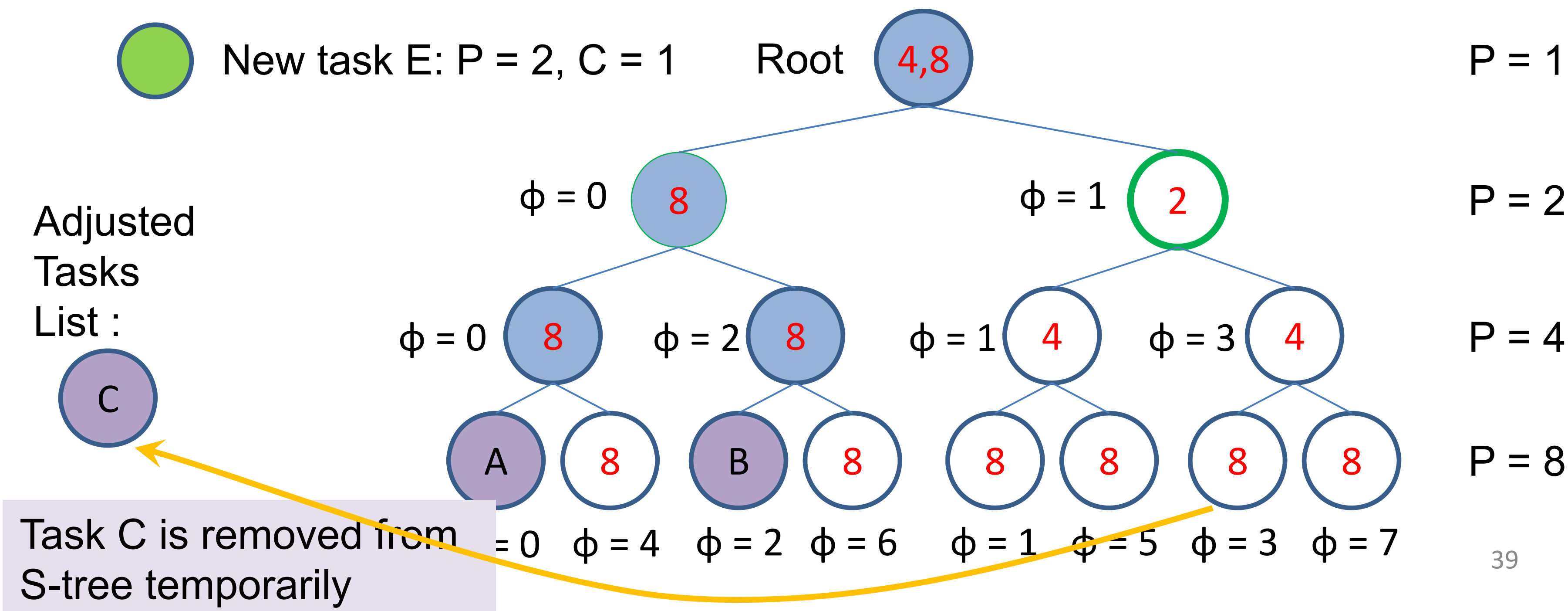
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.



Add-task in S-tree

Replacement Policy:

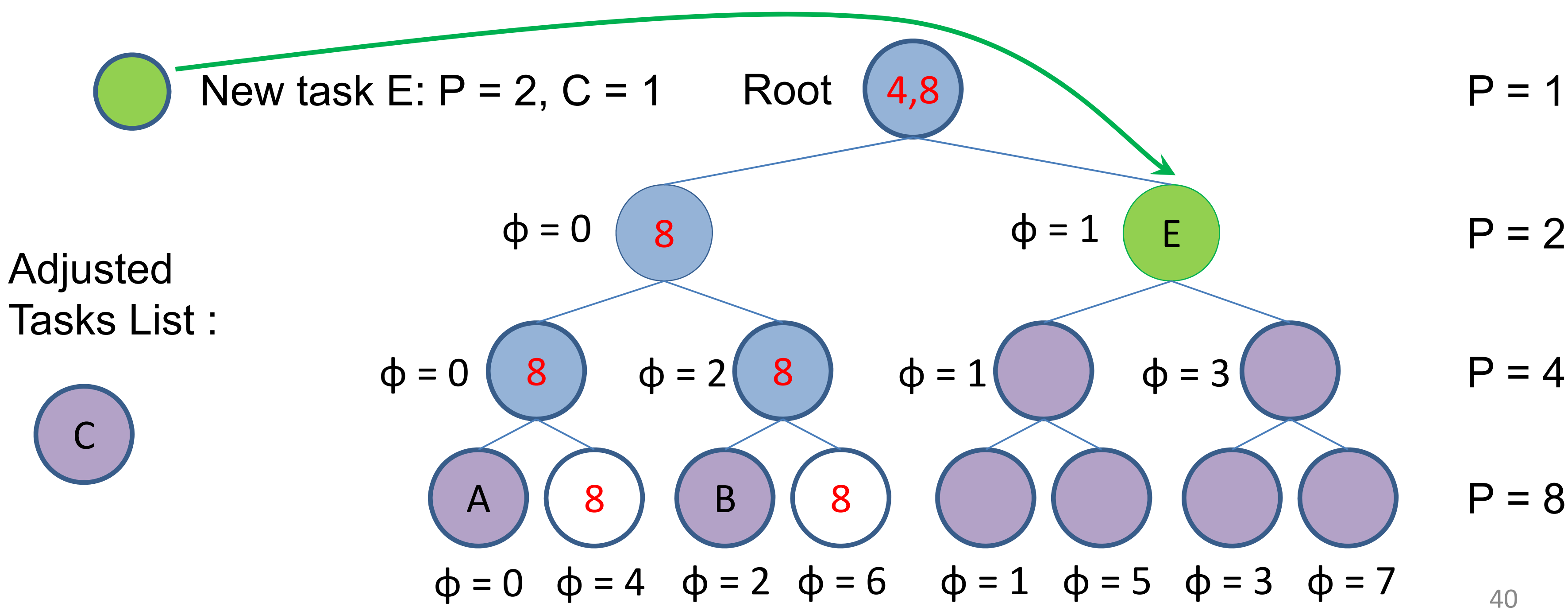
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.



Add-task in S-tree

Replacement Policy:

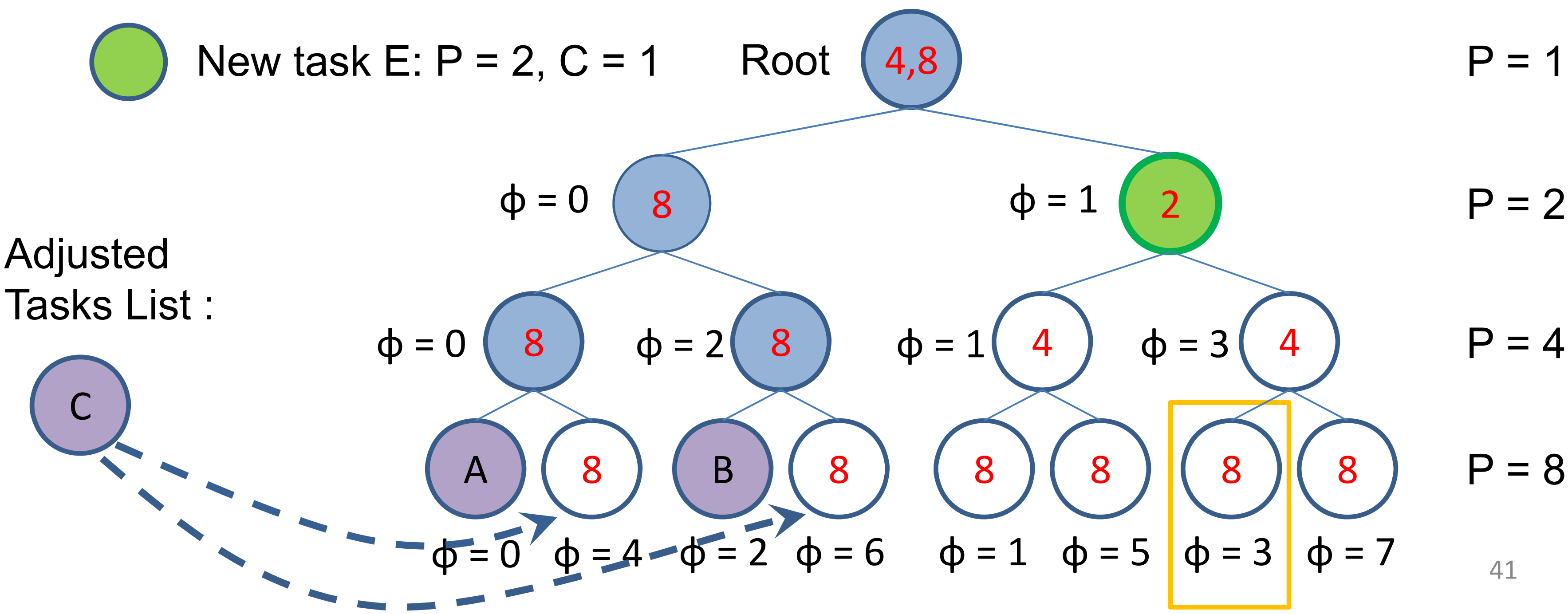
If only semi-occupied nodes are available for new task, the semi-occupied node with least number of tasks will be replaced.



Add-task in S-tree

Adjustment Policy:

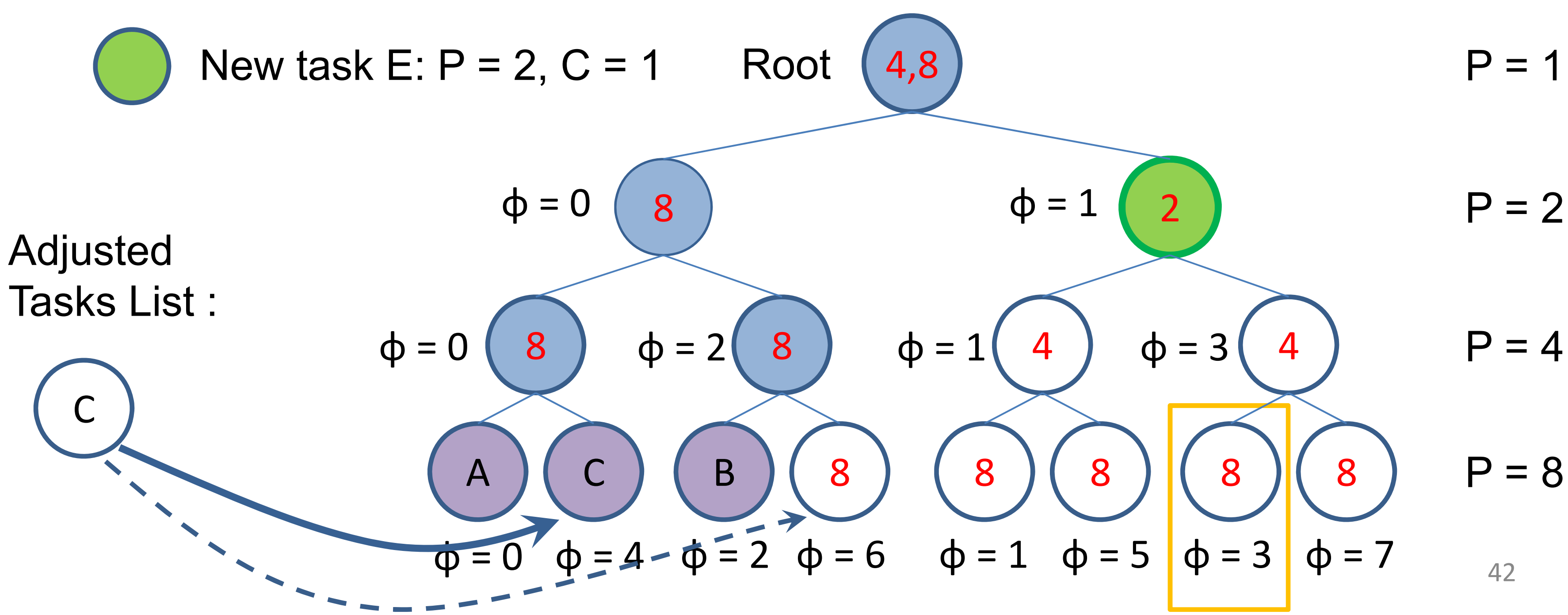
When adjusting the schedule of a task, we assign a node that has the closest phasing to its original phasing.



Add-task in S-tree

Adjustment Policy:

When adjusting the schedule of a task, we assign a node that has the closest phasing to its original phasing.



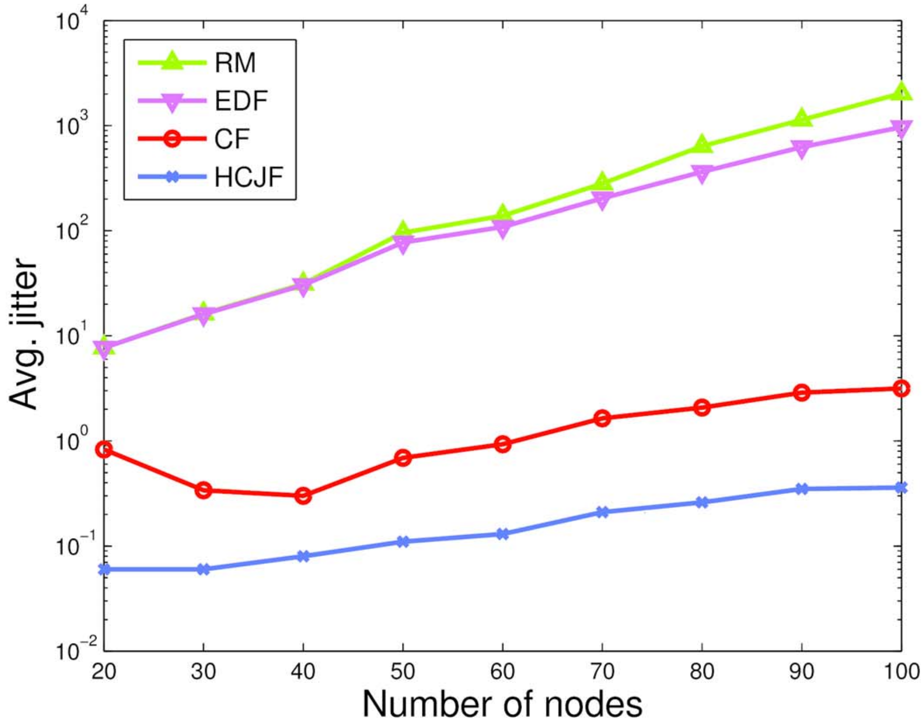
HCJF Scheduler Performance Evaluation - Network Performance

- Compare with:
  - Earliest Deadline First scheduler (EDF)
  - Rate Monotonic scheduler (RM)
  - Contention-Free periodic message scheduler (CF)

- Simulation Setup
  - Time slot size: 200 μs
  - Randomly generate 500 join/leave requests
  - P_{min}: [2, 20] uniformly distributed (simulating 250Hz to 2500Hz max. supported sampling rate)
  - P_{max}: [10, 600] uniformly distributed (simulating 8.33Hz to 500Hz min. required sampling rate)
  - Number of fragments: [1,3] Poisson distribution with $\lambda = 1$
  - 100 runs for each simulation

HCJF Scheduler Performance Evaluation - Network Performance (Cont.)

Overall Jitter



Number of Adjustments