SFCTMN

SFCTMN: a framework based on Continuous Time Markov Networks for throughput calculation in CSMA/CA-based Wireless Networks

A presentation by Francesc Wilhelmi

- 2018 -

Introduction to the framework

- Created by Sergio Barrachina-Muñoz and Francesc Wilhelmi
- Open source. Available at https://github.com/sergiobarra/SFCTMN
- Framework presented in Barrachina-Munoz, S., Wilhelmi, F., & Bellalta, B. (2018). Performance Analysis of Dynamic Channel Bonding in Spatially Distributed High Density WLANs. *arXiv preprint arXiv:1801.00594*.
  - Based on Bellalta, B., Zocca, A., Cano, C., Checco, A., Barcelo, J., & Vinel, A. (2014). Throughput analysis in CSMA/CA networks using continuous time Markov networks: a tutorial. In *Wireless Networking for Moving Objects* (pp. 115-133). Springer, Cham.
- Functionalities
  - Builds the CTMN of a given scenario (nodes location, transmit power, etc.)
    - ✓ Models the behavior of dynamic channel bonding (DCB)
    - ✓ Captures spatially distributed deployments (no need of fully overlapping WLANs)
  - Computes the long-term throughput obtained by each WLAN

Basics on CTMNs (I)

- Models CSMA/CA to capture interactions between WLANs
  - A node senses the channel idle if the detected energy level is lower than the Carrier Sense Threshold (CST)
  - A node is able to recover transmitted data if the energy level is equal or higher than the Data Communication Threshold (DCT)
- Assumptions:
  - The backoff countdown is continuous in time
  - Traffic saturation*
  - In case of collision, we use the Capture Effect (CE) threshold

* Recent research by Barrachina-Muñoz contemplates non-saturation in CTMNs. S. Barrachina-Munoz, F. Wilhelmi, and B. Bellalta. (2018). To overlap or not to overlap: Enabling Channel Bonding in High Density WLANs

Basics on CTMNs (II)

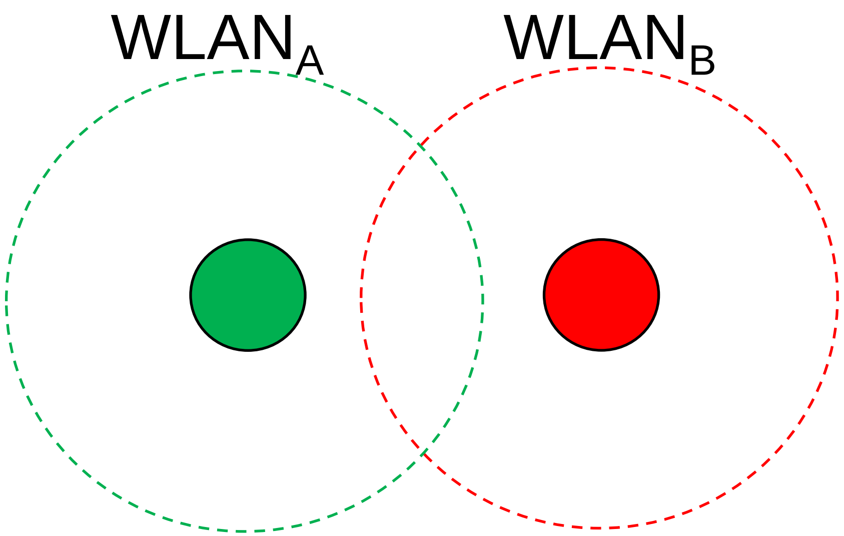


Fig. 1: Scenario 1, containing two non-overlapping WLANs

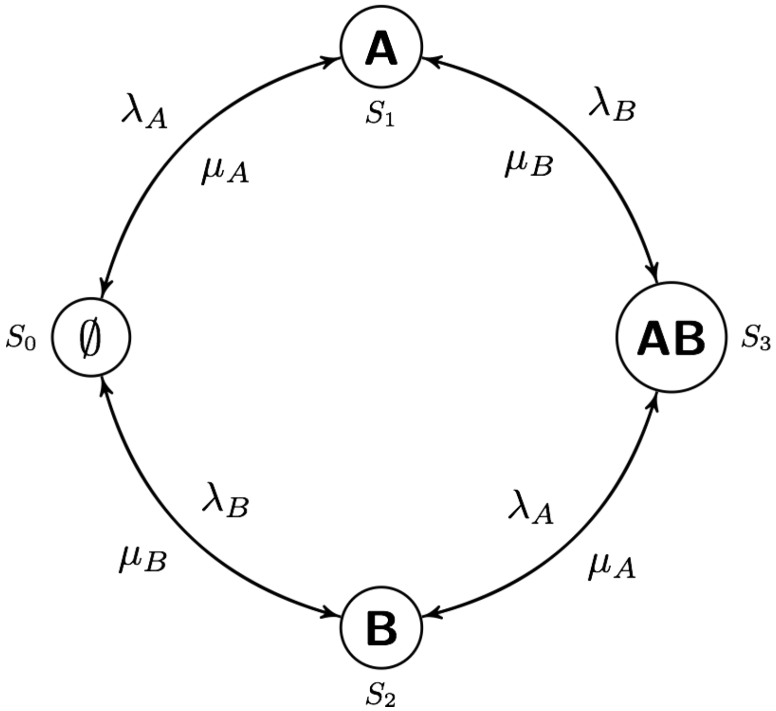


Fig. 2: CTMN of Scenario 1

- λ is the attempt rate: $\lambda = E[B]^{-1} = (CW - 1) / 2$
- μ is the transmission rate: $\mu = 1/T = R/L$

SF-CTMN Preliminaries

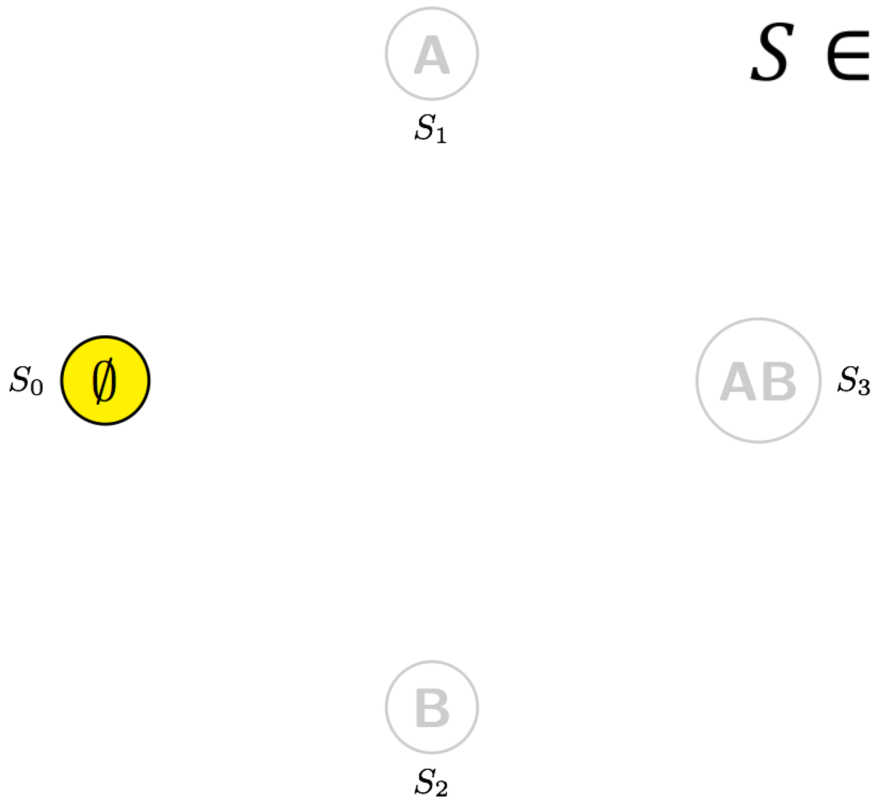
- Assumptions:
  - Considers Downlink (DL) traffic only
  - Saturation & Full buffer
  - The MCS depends on the SINR received at the Station (STA)
  - Data transmissions are successful if **i)** $P_{rx} \geq$ CST, **ii)** SINR $\geq$ CE
- Parameters:
  - IEEE 802.11ax MAC & PHY parameters
  - Propagation models: FSPL, IEEE 802.11ax – residential scenario, etc.
  - Channelization: IEEE 802.11, logarithmic, adjacent channels, etc.
  - Spectrum access protocol: several Dynamic Channel Bonding (DCB) techniques

Algorithm for constructing CTMNs

- Roughly,
  - Generate a space of feasible states (S) from the global space (ψ)
  - For a given state *s*, discover a **backward** transition if exists an state in which the **active** WLANs change their status while the others remain the same
  - For a given state *s*, discover **forward** transitions if exists an state in which the **inactive** WLANs change their status while the others remain the same
- Define transitions as functions of the expected backoff, E[B], and the average transmission time E[T], respectively.

Building a CTMN (I)

State 0: the space of feasible states contains only the **empty** state



$$S \in \{0\}$$

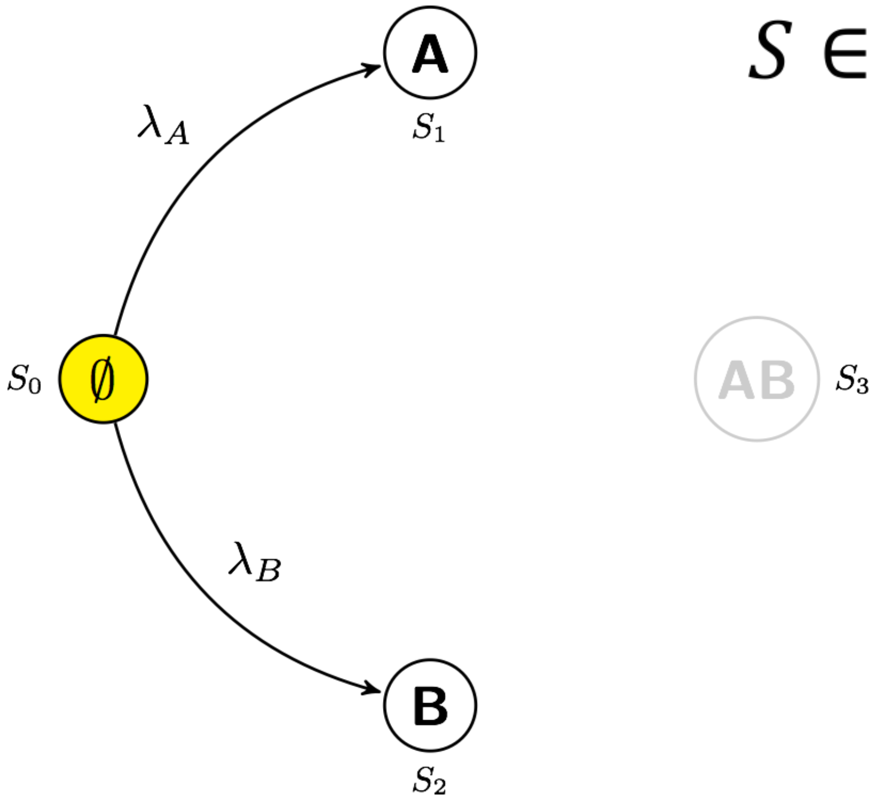
Fig. 3: Building a CTMN

Discover new states and transitions

- Is $WLAN_A$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_A$ is active and $WLAN_B$ remains the same
- Is $WLAN_B$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_B$ is active and $WLAN_A$ remains the same

Building a CTMN (I)

State 0: the space of feasible states contains only the **empty** state



$$S \in \{0, A, B\}$$

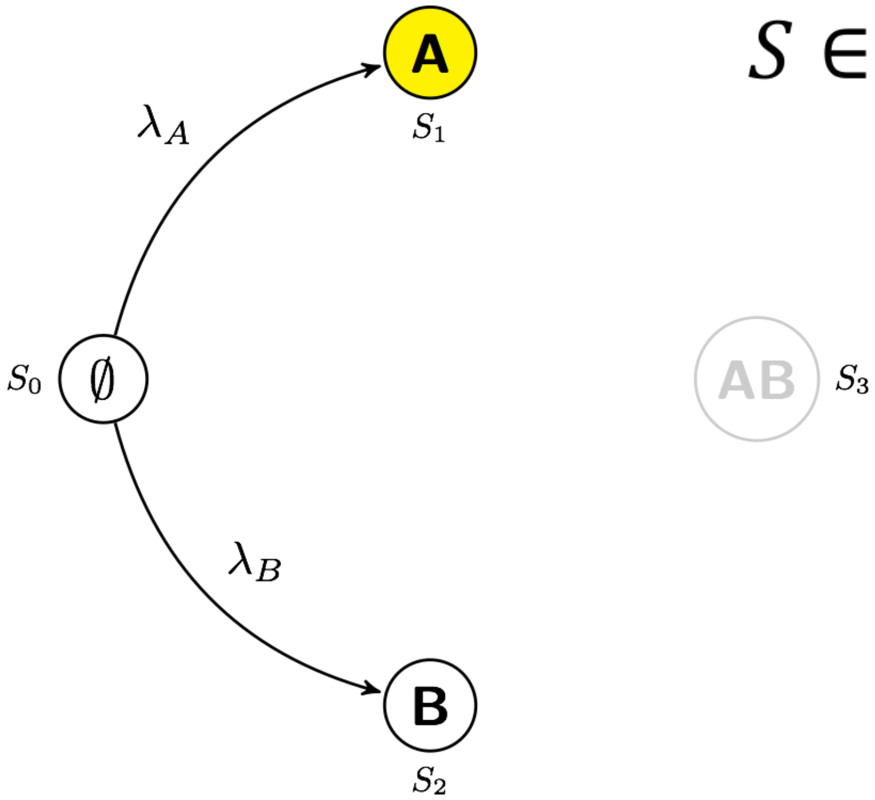
Fig. 3: Building a CTMN

Discover new states and transitions

- Is $WLAN_A$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_A$ is active and $WLAN_B$ remains the same
- Is $WLAN_B$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_B$ is active and $WLAN_A$ remains the same

Building a CTMN (II)

State 1: $WLAN_A$ is active



$$S \in \{0, A, B\}$$

Fig. 3: Building a CTMN

Discover new states and transitions

- Is $WLAN_A$ active? **Yes**
  - Backward transition: $WLAN_A$ is not active and $WLAN_B$ remains the same
  - Forward transition: conditions not accomplished
- Is $WLAN_B$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_B$ is active and $WLAN_A$ remains the same

Building a CTMN (II)

State 1: $WLAN_A$ is active

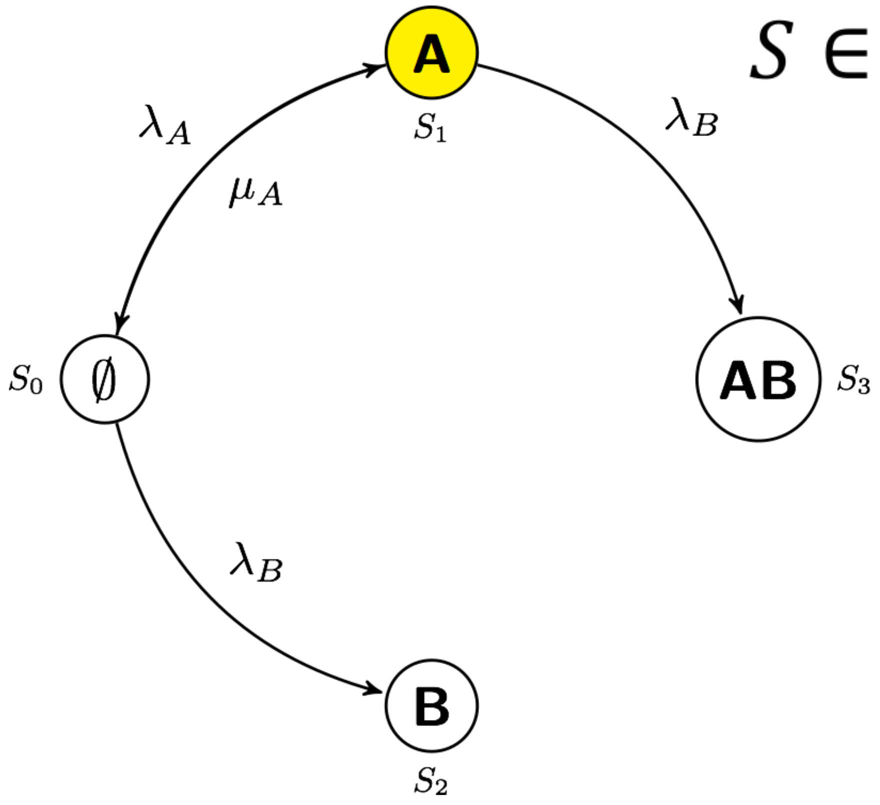


Fig. 3: Building a CTMN

$$S \in \{0, A, B, AB\}$$

Discover new states and transitions

- Is $WLAN_A$ active? **Yes**
  - Backward transition: $WLAN_A$ is not active and $WLAN_B$ remains the same
  - Forward transition: conditions not accomplished
- Is $WLAN_B$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_B$ is active and $WLAN_A$ remains the same

Building a CTMN (III)

State 2: $WLAN_B$ is active

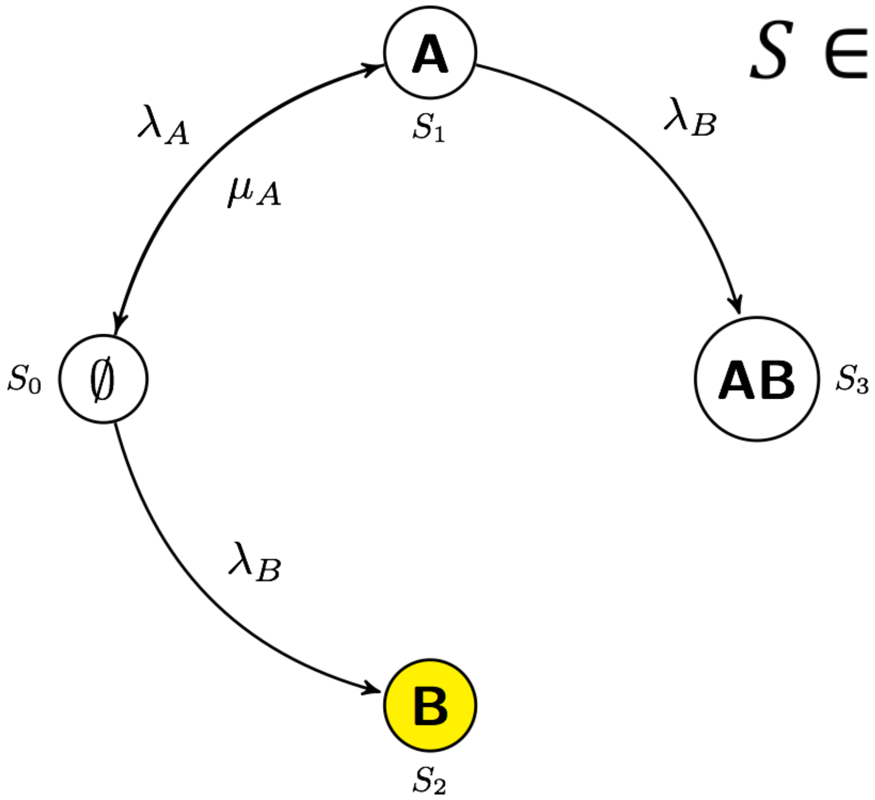


Fig. 3: Building a CTMN

$$S \in \{0, A, B, AB\}$$

Discover new states and transitions

- Is $WLAN_A$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_A$ is active and $WLAN_B$ remains the same
- Is $WLAN_B$ active? **Yes**
  - Backward transition: $WLAN_B$ is not active and $WLAN_A$ remains the same
  - Forward transition: conditions not accomplished

Building a CTMN (III)

State 2: $WLAN_B$ is active

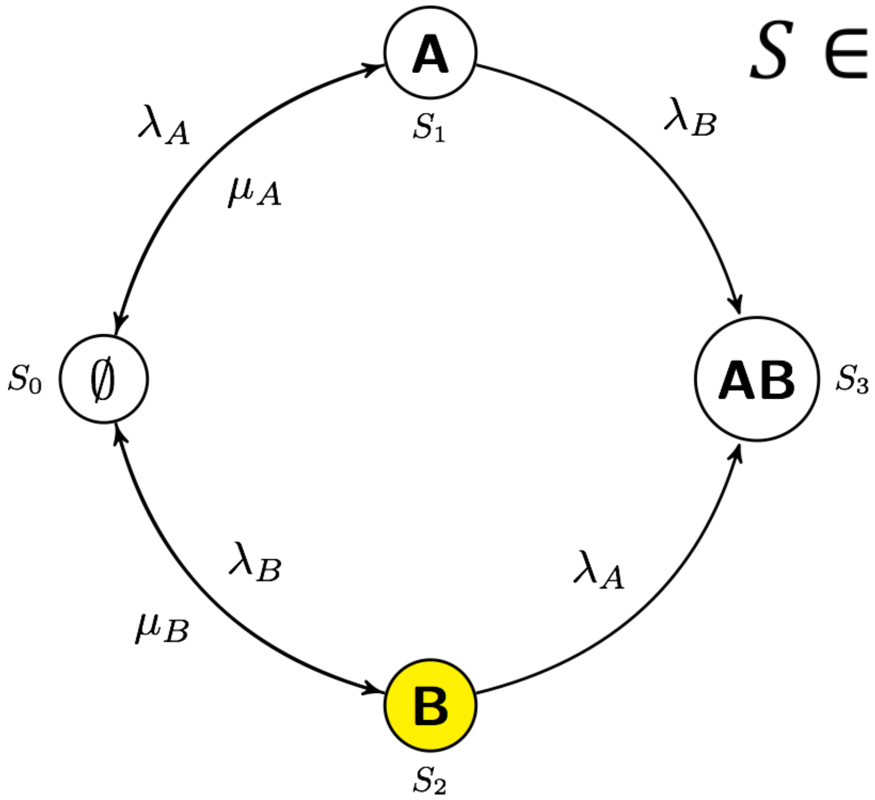


Fig. 3: Building a CTMN

$$S \in \{0, A, B, AB\}$$

Discover new states and transitions

- Is $WLAN_A$ active? **No**
  - Backward transition: conditions not accomplished
  - Forward transition: $WLAN_A$ is active and $WLAN_B$ remains the same
- Is $WLAN_B$ active? **Yes**
  - Backward transition: $WLAN_B$ is not active and $WLAN_A$ remains the same
  - Forward transition: conditions not accomplished

Building a CTMN (IV)

State 3: Both $WLAN_A$ and $WLAN_B$ are active

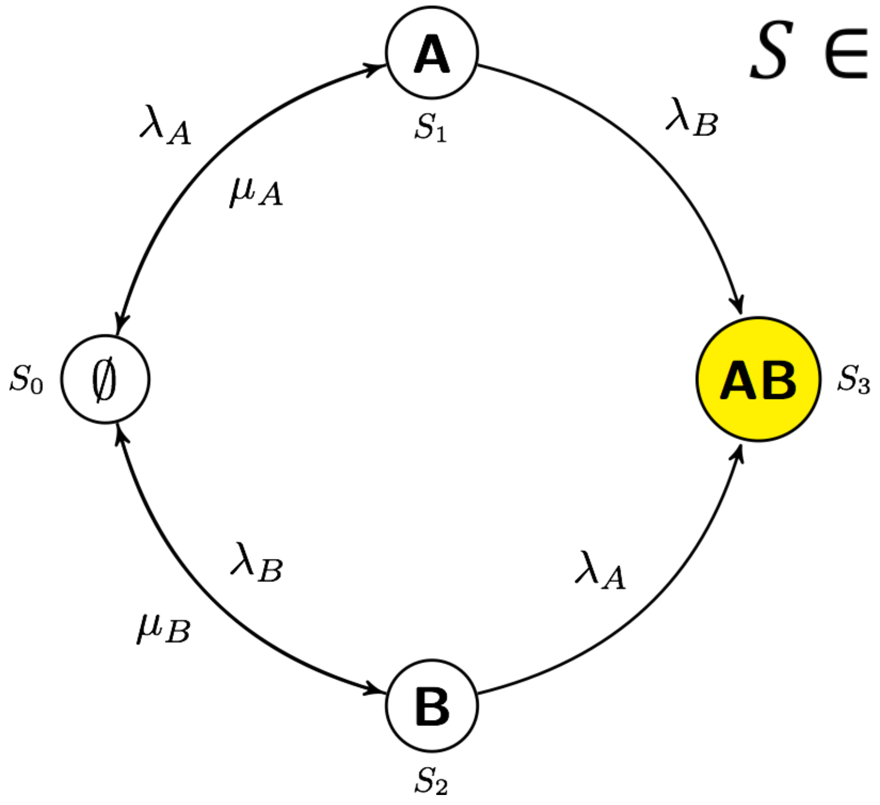


Fig. 3: Building a CTMN

$$S \in \{0, A, B, AB\}$$

Discover new states and transitions

- Is $WLAN_A$ active? **Yes**
  - Backward transition: $WLAN_A$ is not active and $WLAN_B$ remains the same
  - Forward transition: conditions not accomplished
- Is $WLAN_B$ active? **Yes**
  - Backward transition: $WLAN_B$ is not active and $WLAN_A$ remains the same
  - Forward transition: conditions not accomplished

Building a CTMN (IV)

State 3: Both $WLAN_A$ and $WLAN_B$ are active

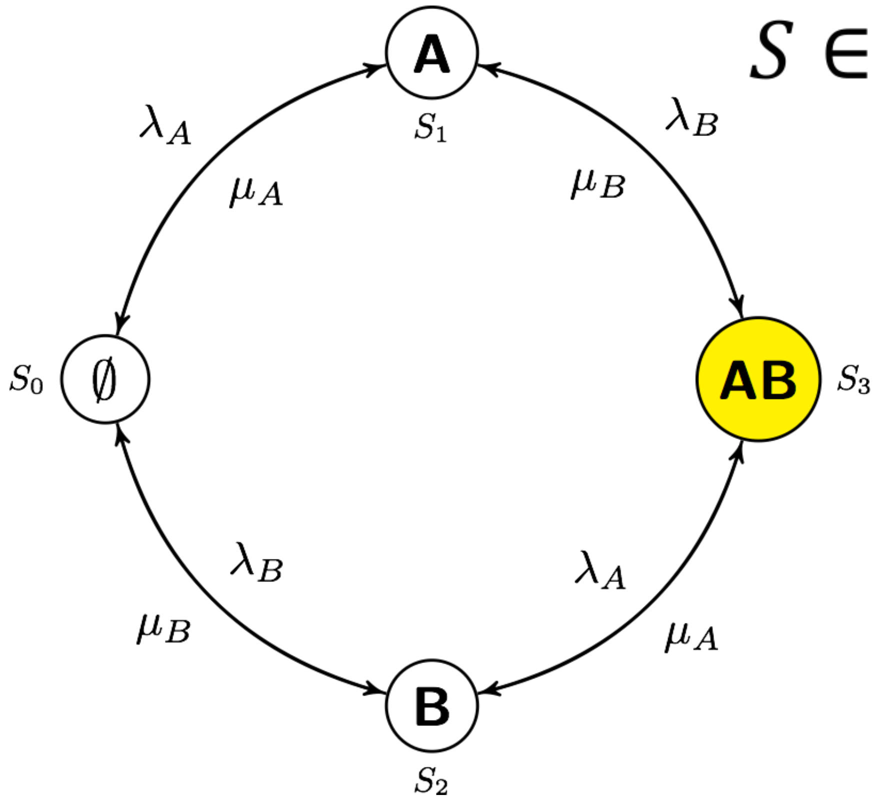


Fig. 3: Building a CTMN

$$S \in \{0, A, B, AB\}$$

Discover new states and transitions

- Is $WLAN_A$ active? **Yes**
  - Backward transition: $WLAN_A$ is not active and $WLAN_B$ remains the same
  - Forward transition: conditions not accomplished
- Is $WLAN_B$ active? **Yes**
  - Backward transition: $WLAN_B$ is not active and $WLAN_A$ remains the same
  - Forward transition: conditions not accomplished

Throughput Calculation (I)

Once the CTMN is built

1. Compute the probability of being in each state:

$$\pi_s = \frac{\prod_{i \in s} \theta_i}{\sum_{s \in \Omega} \prod_{i \in s} \theta_i}$$, given that $\theta_i := \lambda_i / \mu_i$ and $$\pi_\emptyset = \frac{1}{\sum_{s \in \Omega} \prod_{i \in s} \theta_i}$$

2. Compute the throughput as follows:

$$x_i = \frac{E[L_i]}{E[T_i]} \left(\sum_{s \in \Omega : i \in s} \pi_s \right)$$

Throughput Calculation (II)

The CTMN may not be reversible!

- $WLAN_B$ cannot transmit if WLANs A and C occupy the channel
- $WLAN_B$ can transmit if either $WLAN_A$ or $WLAN_C$ occupy the channel
- $WLAN_B$ experiences collisions in state *ABC* due to the additive interference
- State *ABC* cannot be reached from *AC*
- There is a backward transition from *ABC* to *AC*

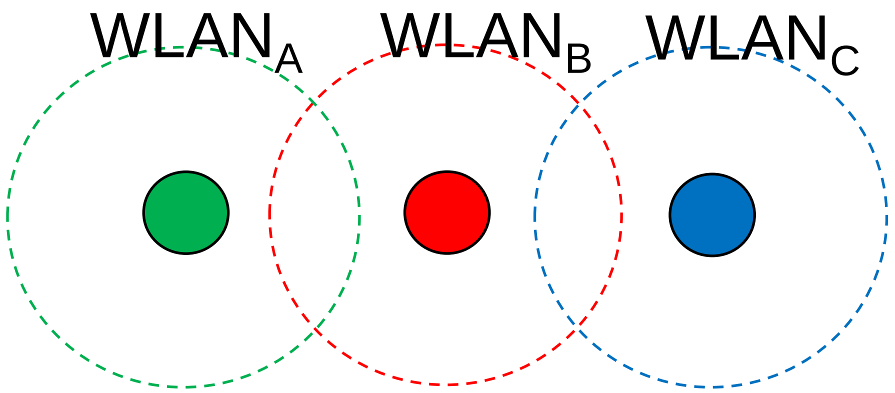


Fig. 4: Scenario 2 framing additive interference at $WLAN_B$

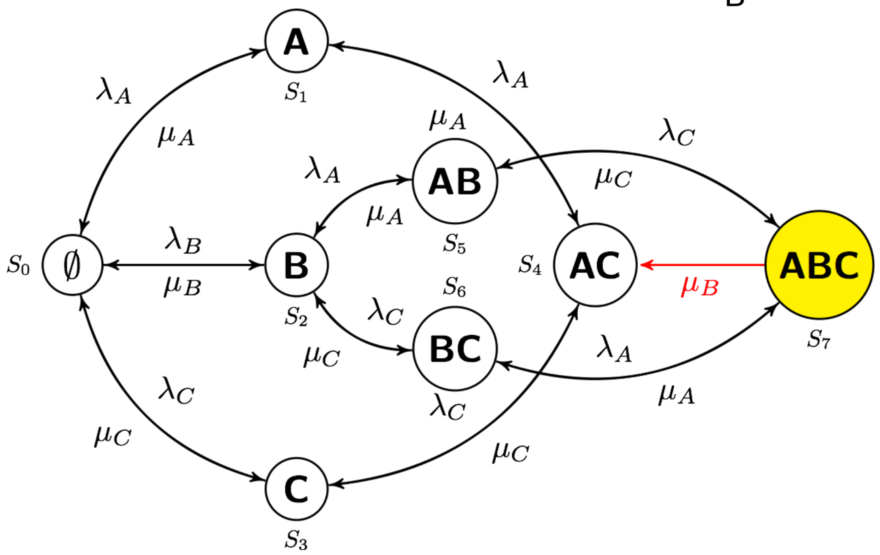


Fig. 5: CTMN from Scenario 2

Throughput Calculation (III)

Then, compute the throughput by:

1. Build the transitions matrix Q

$$Q = \begin{bmatrix} -(\lambda_A + \lambda_B + \lambda_C) & \lambda_A & \lambda_B & \lambda_C & 0 \\ \mu_A & -(\mu_A + \lambda_A) & 0 & \lambda_C & \lambda_A \\ \mu_B & 0 & -(\mu_B + \lambda_B) & 0 & 0 \\ \mu_C & \lambda_A & 0 & -(\mu_C + \lambda_C) & \lambda_C \\ 0 & \mu_A & 0 & \mu_C & -(\mu_A + \mu_C) \end{bmatrix}$$

2. Solve πQ = 0 to obtain the probability π of being in each state
3. Compute the throughput according to π

Input

- The user introduces an .csv file to indicate the WLANs' configuration
  - Primary channel, left and right channels, transmit power, CCA, λ, position AP, position STA, etc.
- In addition, another file specifies the system configuration
  - Path-loss model, ambient noise, access protocol, bandwidth per channel, etc.

1	% code	primary	left ch	right ch	tx_power	cca	lambda	x_ap	y_ap	z_ap	x_sta	y_sta	z_sta	legacy_n...	cw
2	1	1	1	1	20	-82	14815	0	0	0	0	4	0	0	512
3	2	1	1	1	20	-82	14815	20	0	0	20	4	0	0	512

Fig. 6: Input sample for the SF-CTMN framework

Output (I)

- Display information can be shown if the corresponding flag are activated
  - Channel allocation
  - WLANs' map
  - Chains of PSI (global states) and S (feasible states)
  - Throughput per WLAN

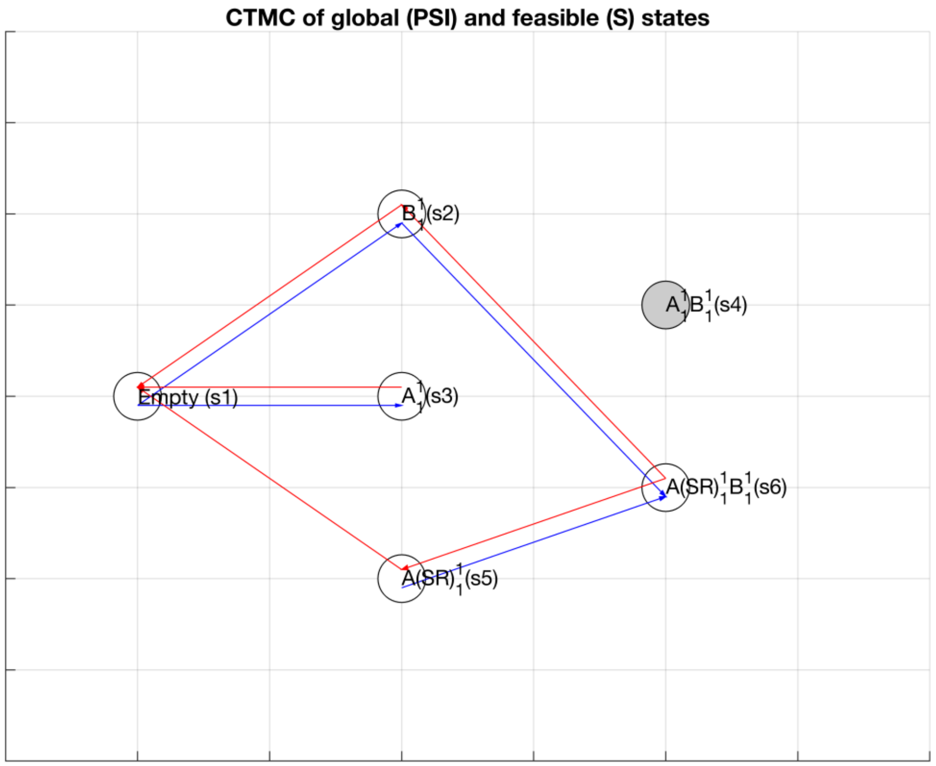


Fig. 7: Output generated by the SF-CTMN framework

Output (II): example with DCB

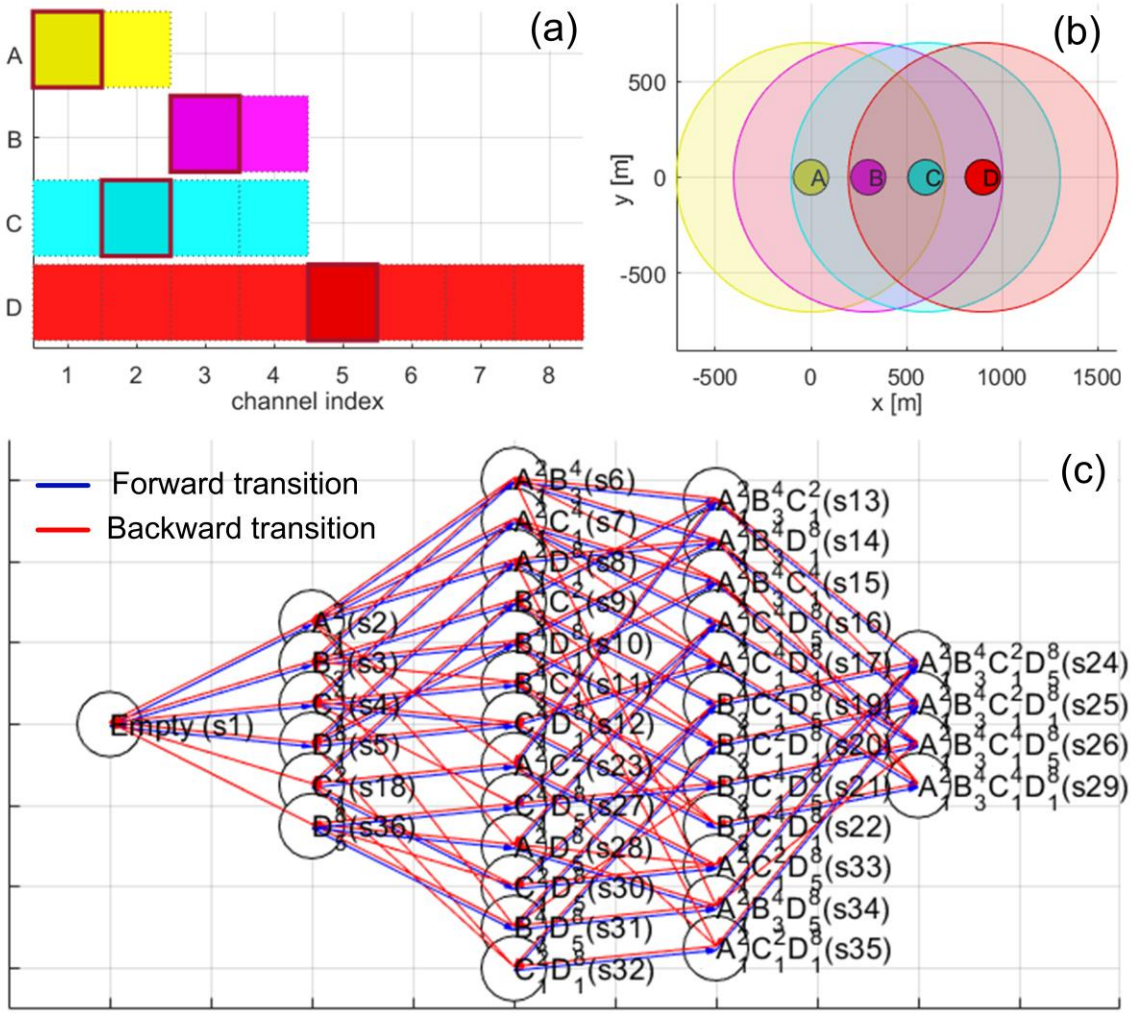


Fig. 8: scenario with DCB and spatial distribution.
a) Channel allocation of the WLANs.
b) Deployment of the network and carrier sense ranges.
c) CTMN with states, backward and forward transitions.