Deterministic scheduling of periodic datagrams for low latency in 5G and beyond

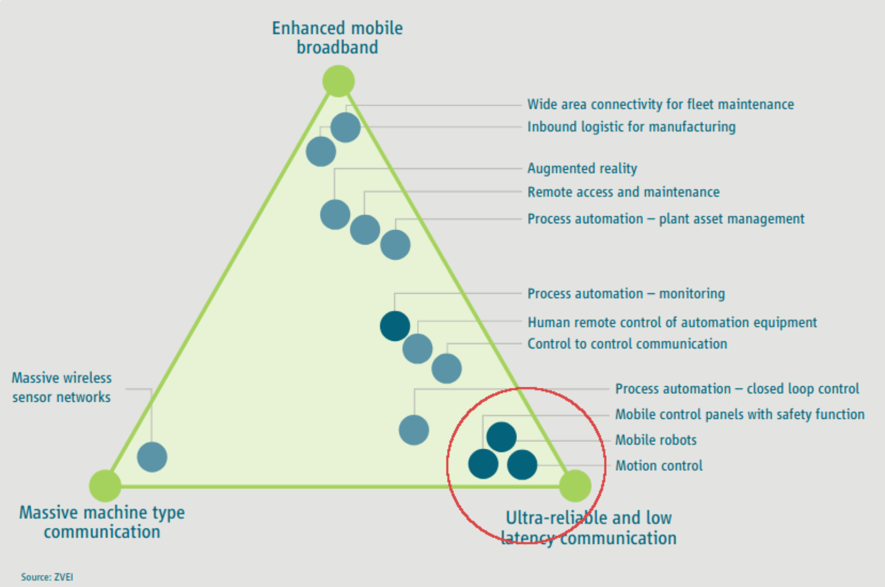
Maël Guiraud

Nokia Bell Labs France - DAVID, Université de Versailles Saint Quentin

February 24, 2022

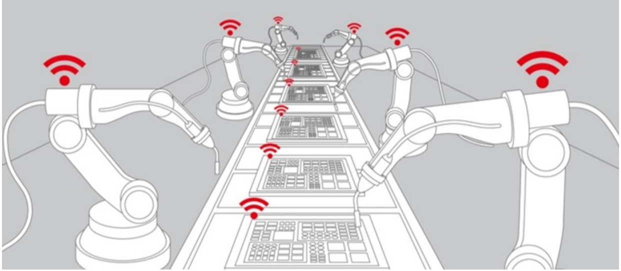
Introduction

5G Context


Enhanced mobile broadband
Wide area connectivity for fleet maintenance
Inbound logistic for manufacturing
Augmented reality
Remote access and maintenance
Process automation – plant asset management
Process automation – monitoring
Human remote control of automation equipment
Control to control communication
Massive wireless sensor networks
Process automation – closed loop control
Mobile control panels with safety function
Mobile robots
Motion control
Massive machine type communication
Ultra-reliable and low latency communication

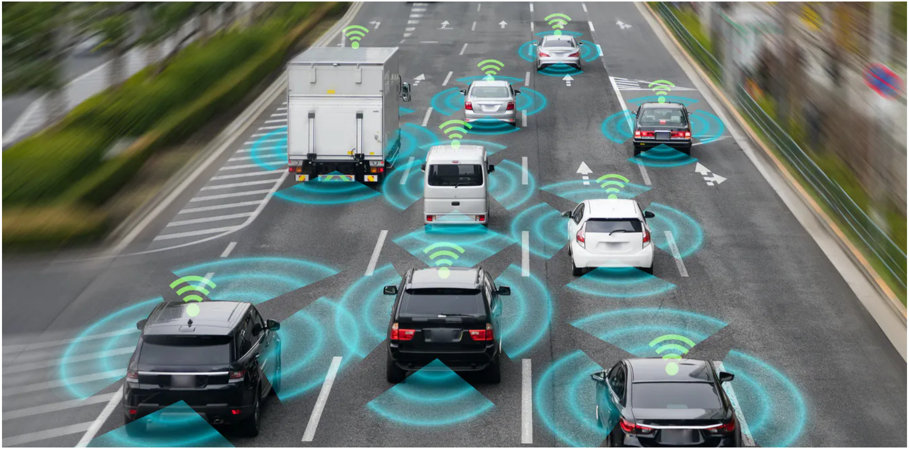
Source: ZVEI

Use cases



Industry 4.0

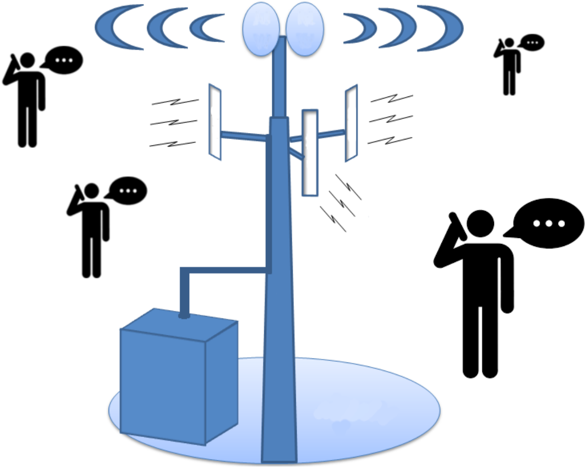
Use cases



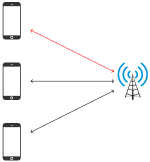
Autonomous Vehicle

A Radio Antenna

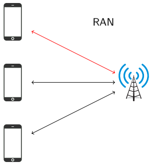
A Radio Antenna



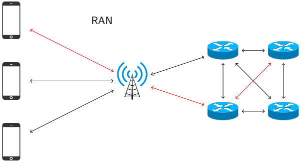
Radio Access Network



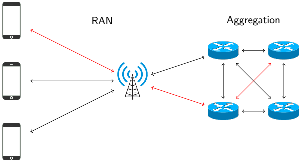
Radio Access Network



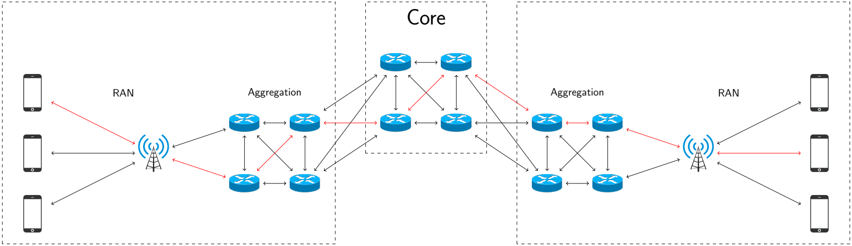
Aggregation network



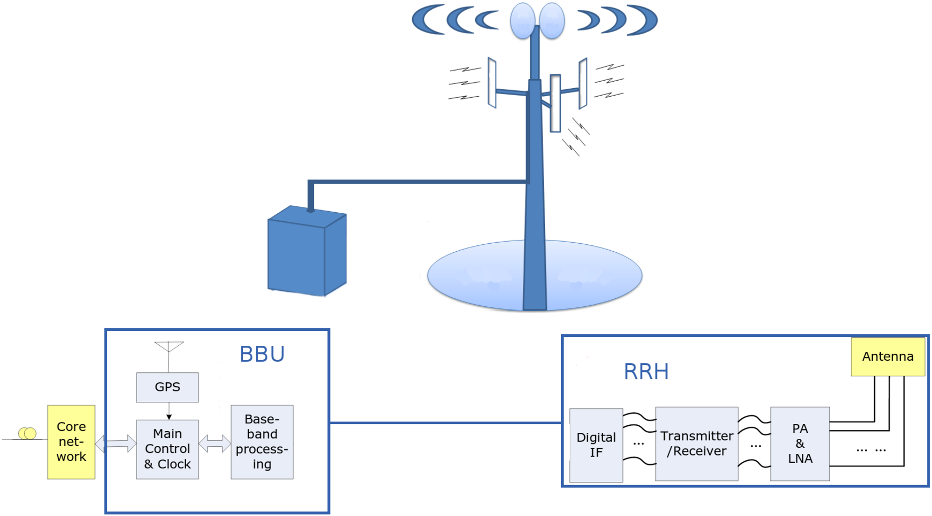
Aggregation network



An end-to-end communication between two terminals

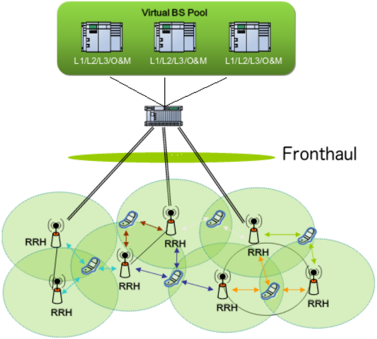


What does Cloud-RAN means?

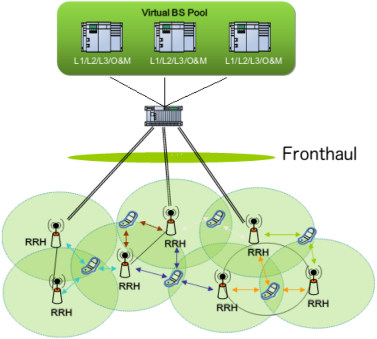


RU=RRH, Distributed/Centralized Unit=BBU

Cloud-RAN

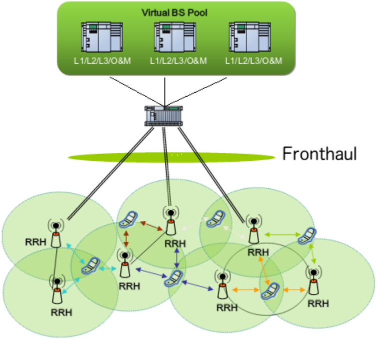

Virtual BS Pool
L1/L2/L3/O&M
L1/L2/L3/O&M
L1/L2/L3/O&M
Fronthaul
RRH
RRH
RRH
RRH
RRH
RRH
RRH
RRH

Cloud-RAN



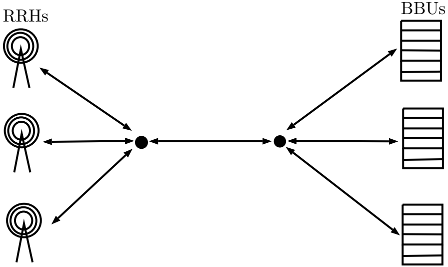
C-RAN aims to mutualize the computation ressources.

Cloud-RAN



C-RAN aims to mutualize the computation ressources.
The latency is constrained by protocol.

Problematic



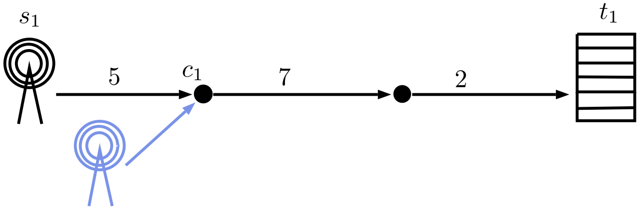
Statistical multiplexing:

- Buffering inducing additional latency
- Average guarantee on Latency

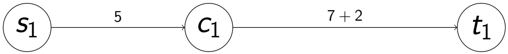
C-RAN specifications:

- 100% of the packets under a given latency.
- Minimize latency means an higher area cover.

Model: The routed network

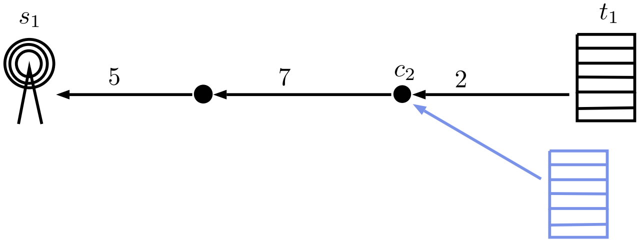


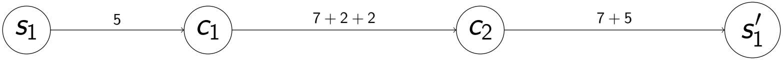
- Network → Weighted Directed Acyclic Multigraph
- Discrete time model: Physical Delay of a link → Weight of the arcs (tics).
- Only the contention points are represented in the graph



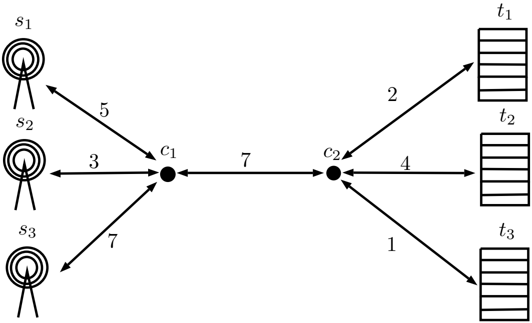
Model: The routed network

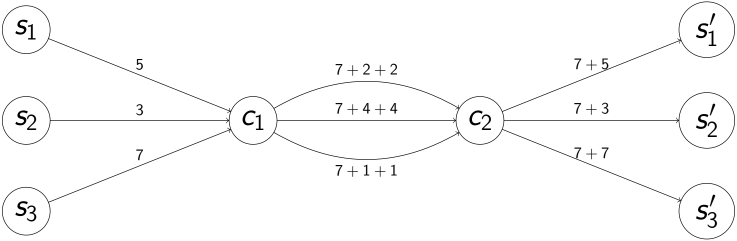
Both ways: from RRH to BBU (forward) then from BBU to RRH (backward)



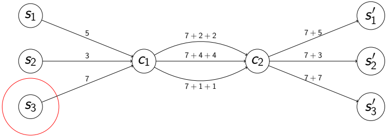


Model: The routed network

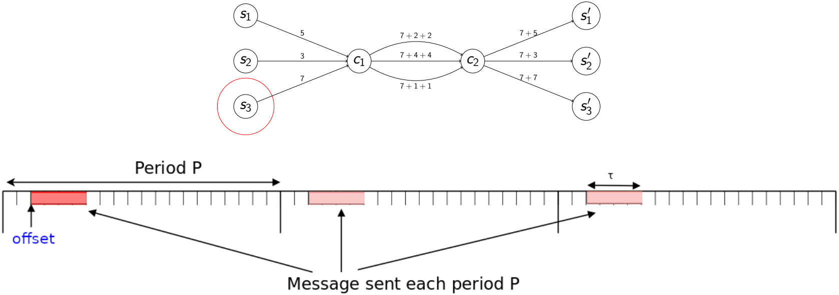




Model: The communication process



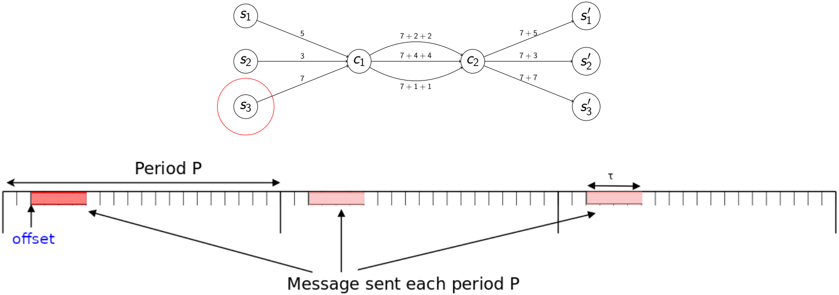
Model: The communication process



Every P units of time, a message of size τ is emitted from each RRH. P and τ are fixed by the context.

The process is periodic : each message is emitted in each period at the same time, called offset.

Model: The communication process



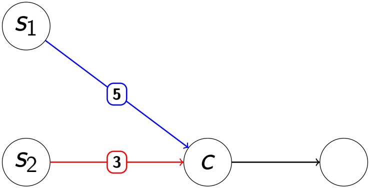
Message sent each period P

Definition

$$\text{Load} = \frac{\text{Size of the messages}}{\text{Period}} = \frac{3\tau}{P}$$

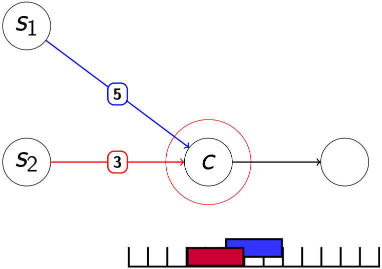
Model: The communication process

$P = 13, \tau = 3$



Model: The communication process

Example for $P = 13$ and $\tau = 3$.

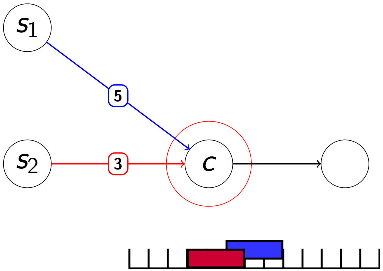


Definition

There is a collision between two routes when their messages go through the first vertex of a common arc at the same time.

Model: The communication process

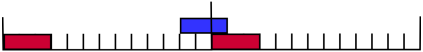
Example for $P = 13$ and $\tau = 3$.



Definition

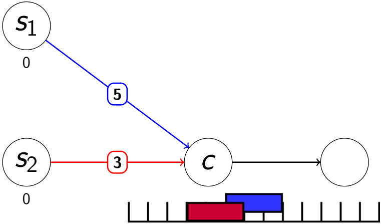
There is a collision between two routes when their messages go through the first vertex of a common arc at the same time.

Periodicity must be taken into consideration.



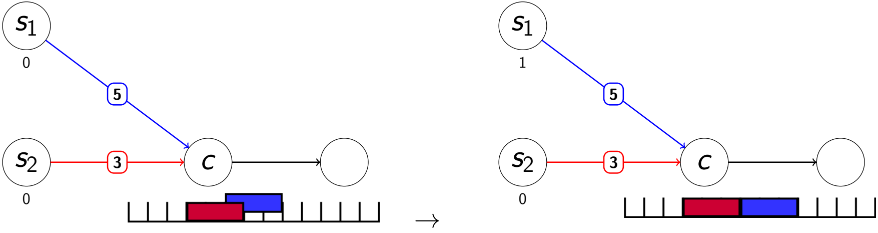
Assignment

Example for $P = 13$ and $\tau = 3$.



Assignment

Example for $P = 13$ and $\tau = 3$.

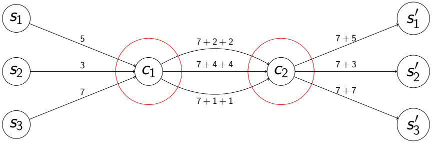


Choosing the offset such that there is no collisions.

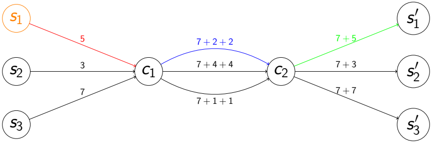
Definition

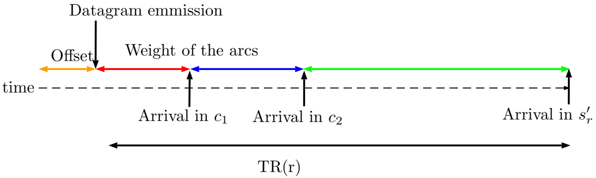
An Assignment is a choice of offset for each message.

A first problem

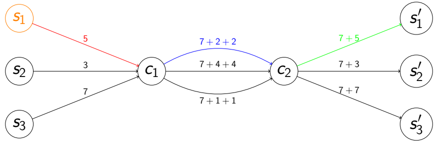


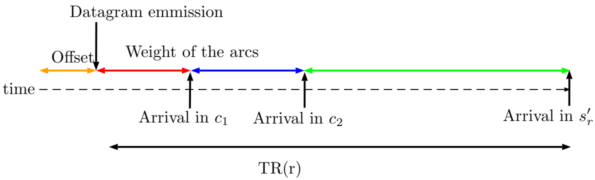
A first problem





A first problem




Datagram emmission

Offset

Weight of the arcs

time

Arrival in c_1

Arrival in c_2

Arrival in s'_r

TR(r)

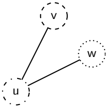
Periodic Assignment for Zero Latency (PAZL)

Given a routed network, find an assignment such that there is no collisions in c_1 and c_2.

Complexity

Theorem

PAZL is NP*-complete.*



→

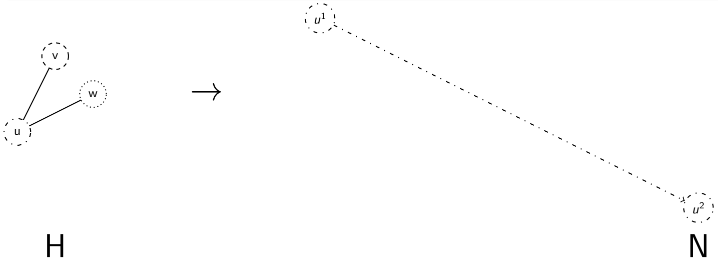
H

Reduction of an instance H of vertex-coloring to an instance of PAZL.

Complexity

Theorem

PAZL is NP*-complete.*

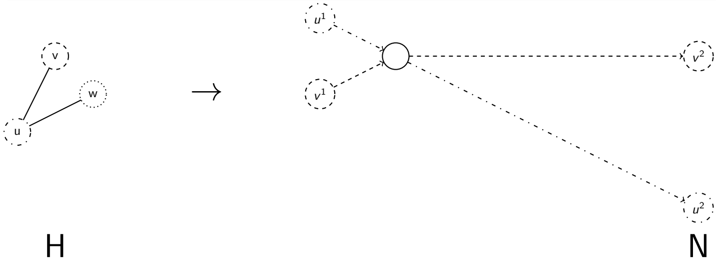


Reduction of an instance H of vertex-coloring to an instance of PAZL.

Complexity

Theorem

PAZL is NP*-complete.*

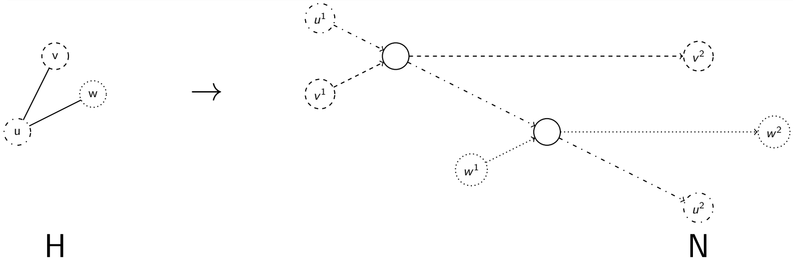


Reduction of an instance H of vertex-coloring to an instance of PAZL.

Complexity

Theorem

PAZL is NP*-complete.*

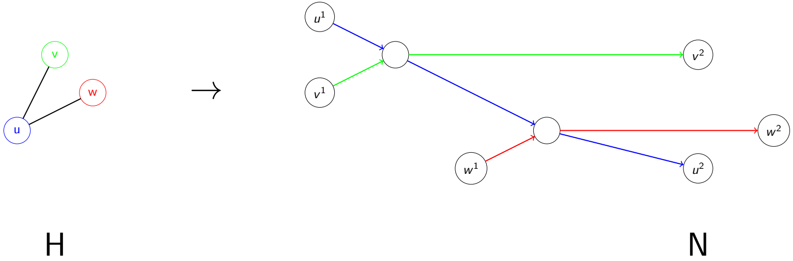


Reduction of an instance H of vertex-coloring to an instance of PAZL.

Complexity

Theorem

PAZL is NP*-complete.*

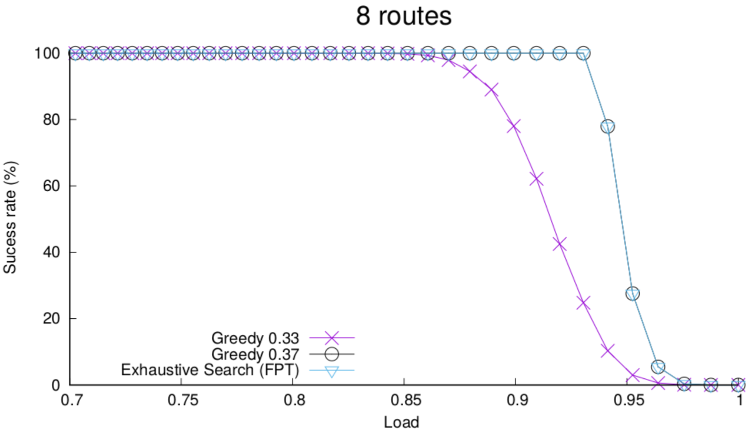


Reduction of an instance H of vertex-coloring to an instance of PAZL. A P-coloration of H is equivalent to a P-periodic Assignment of N.

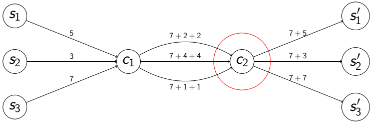
Solving PAZL

Algorithms to solve PAZL:

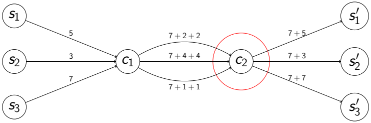
- Exsitence of a solution for moderate loads using polynomial time algorithms.
- FPT Algorithm with the number of routes as parameter.



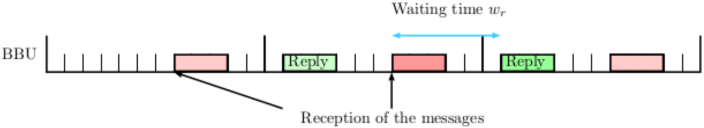
Assignment



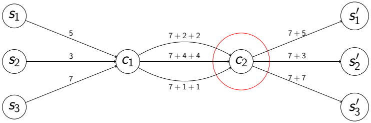
Assignment



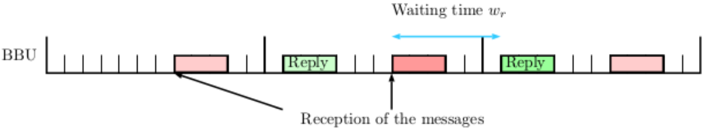
In c_2, one can choose the waiting time before sending back the answer.



Assignment



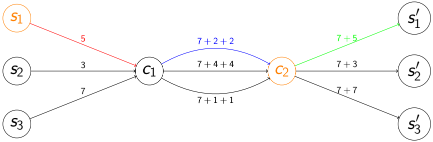
In c_2, one can choose the waiting time before sending back the answer.

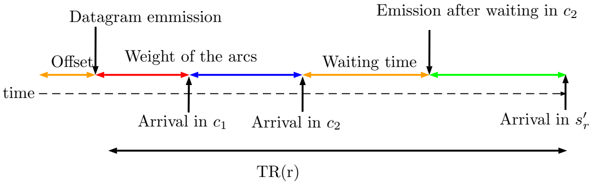


Definition

Redefinition of the notion of assignment: a choice of offsets and waiting times for each route without collisions.

Transmission Time




Datagram emmission

Emission after waiting in c_2

Offset

Weight of the arcs

Waiting time

time

Arrival in c_1

Arrival in c_2

Arrival in s'_r

TR(r)

Deadline and Problem

Each route must have a transmission time lower or equal to a given **deadline**.

Deadline and Problem

Each route must have a transmission time lower or equal to a given **deadline**.

Periodic Assignment for Low Latency (PALL)

Given a routed network, find an assignment such that the deadline constraint is satisfied for each route.

Deadline and Problem

Each route must have a transmission time lower or equal to a given **deadline**.

Periodic Assignment for Low Latency (PALL)

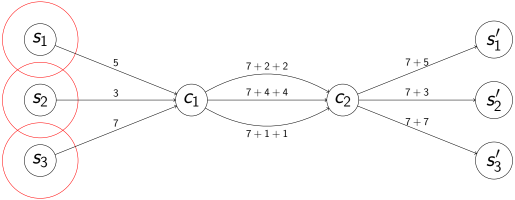
Given a routed network, find an assignment such that the deadline constraint is satisfied for each route.

PALL is NP-complete.

Solving PALL on simple topologies

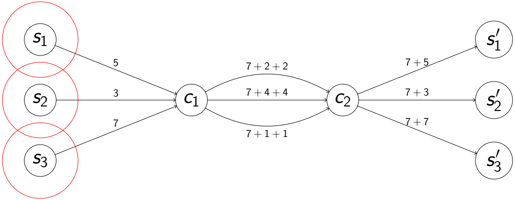
A two stage approach for PALL

First step: We fix the offset of the route such that there is no collisions in c_1.



A two stage approach for PALL

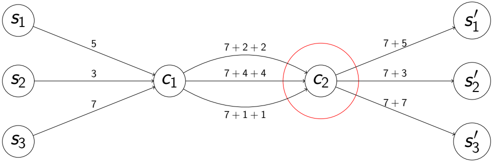
First step: We fix the offset of the route such that there is no collisions in c_1.



Second Step:

Waiting Time Assignment (WTA)

Given the routed network and the offsets for all routes, find an assignment satisfying the deadlines constraints.

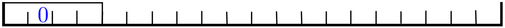


Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time					

A run of GreedyDeadline with $P = 20, \tau = 4$.

Step 1:

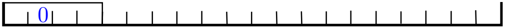


Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0				

A run of GreedyDeadline with $P = 20, \tau = 4$.

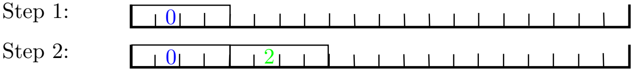
Step 1:



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	**5**	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0				

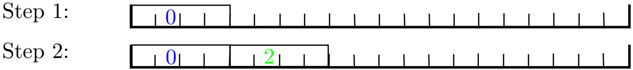
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	**5**	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0		1		

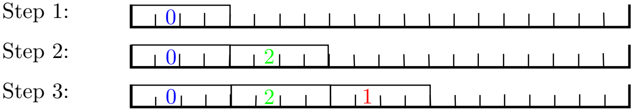
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0		1		

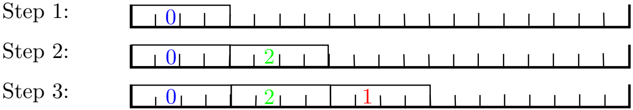
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0	6	1		

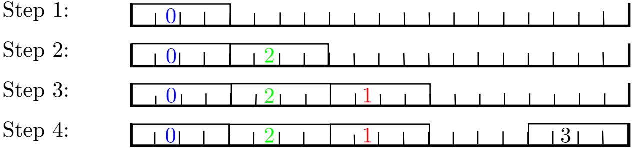
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0	6	1		

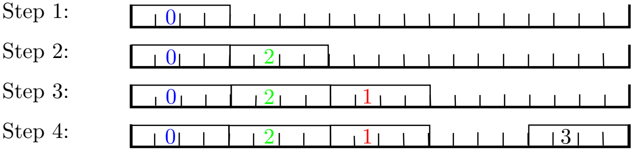
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0	6	1	0	

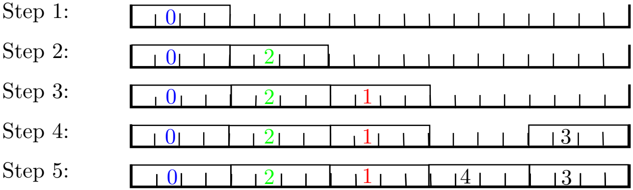
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0	6	1	0	

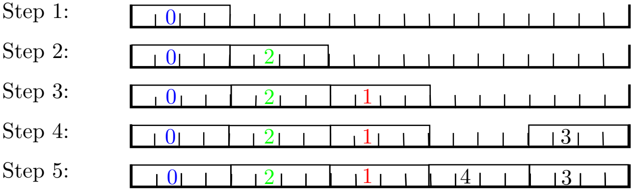
A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: A greedy algorithm

Route	0	1	2	3	4
Deadline	10	15	5	7	32
Arrival time in c_2	0	2	3	16	17
Waiting time	0	6	1	0	15

A run of GreedyDeadline with $P = 20, \tau = 4$.



Solving WTA: Polynomial time algorithms

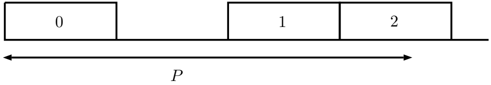
MLS: Polynomial time algorithm. Finds a solution minimizing the date at which all messages are scheduled.

Solving WTA: Polynomial time algorithms

MLS: Polynomial time algorithm. Finds a solution minimizing the date at which all messages are scheduled. Periodicity not managed.

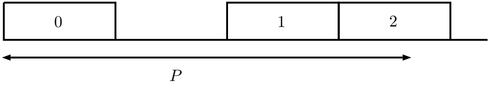
Solving WTA: Polynomial time algorithms

MLS: Polynomial time algorithm. Finds a solution minimizing the date at which all messages are scheduled. Periodicity not managed.

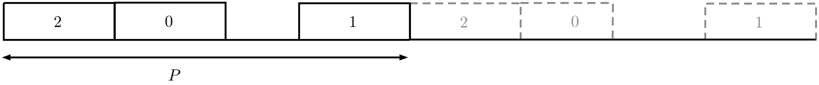


Solving WTA: Polynomial time algorithms

MLS: Polynomial time algorithm. Finds a solution minimizing the date at which all messages are scheduled. Periodicity not managed.

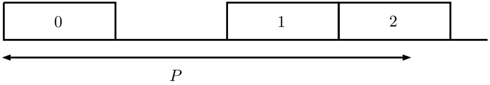


PMLS:

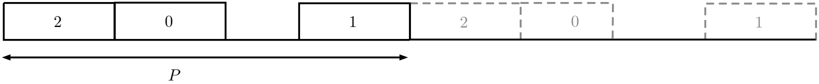


Solving WTA: Polynomial time algorithms

MLS: Polynomial time algorithm. Finds a solution minimizing the date at which all messages are scheduled. Periodicity not managed.

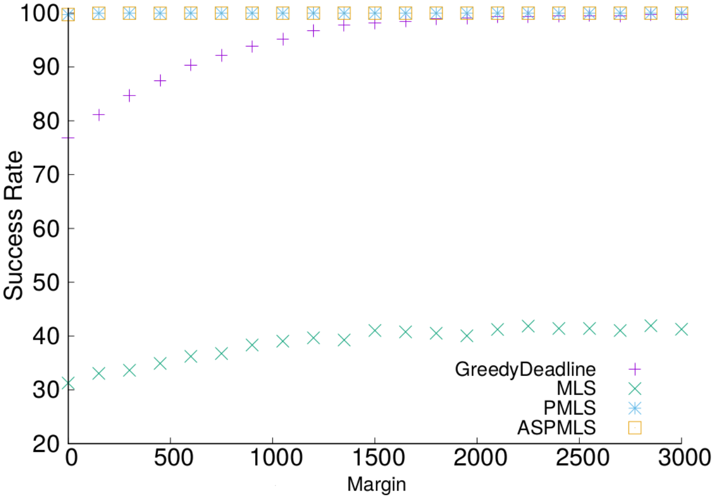


PMLS:

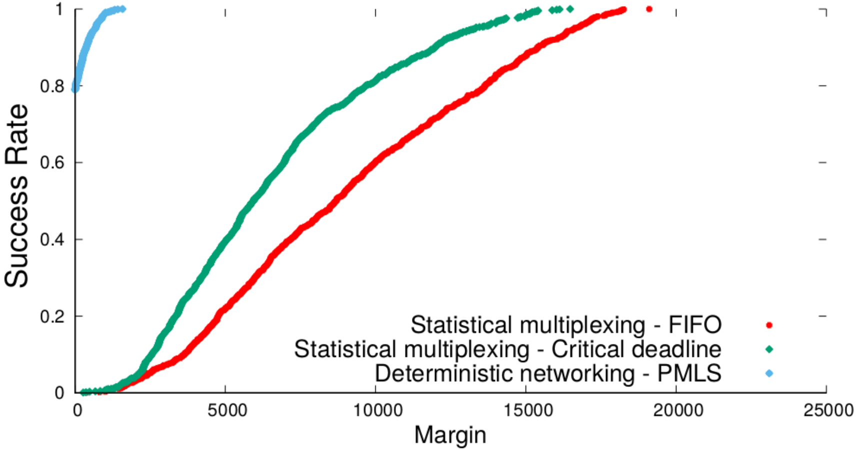


ASPMLS: FPT-Algorithm based on PMLS → Always find a solution if existing.

Results: Algorithms for WTA



Results: PMLS against statistical multiplexing



Mixing C-RAN traffic with Best-Effort Traffic

C-RAN traffic: High priority, deterministic traffic, scheduled with minimal latency.

Mixing C-RAN traffic with Best-Effort Traffic

C-RAN traffic: High priority, deterministic traffic, scheduled with minimal latency.

Best Effort traffic: Low priority, stochastic law on arrivals, no specific rules on forwarding.

Mixing C-RAN traffic with Best-Effort Traffic

C-RAN traffic: High priority, deterministic traffic, scheduled with minimal latency.

Best Effort traffic: Low priority, stochastic law on arrivals, no specific rules on forwarding.

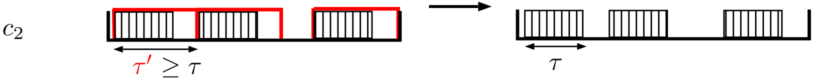
PMLS creates long sequences of C-RAN traffic in which there is no free spaces for BE traffic.

Mixing C-RAN traffic with Best-Effort Traffic

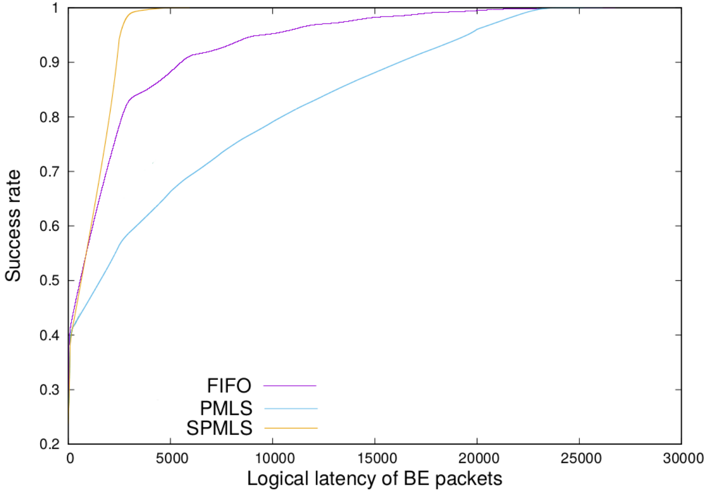
C-RAN traffic: High priority, deterministic traffic, scheduled with minimal latency.

Best Effort traffic: Low priority, stochastic law on arrivals, no specific rules on forwarding.

PMLS creates long sequences of C-RAN traffic in which there is no free spaces for BE traffic.

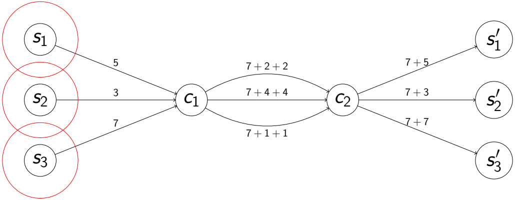


Mixing C-RAN traffic with Best-Effort Traffic



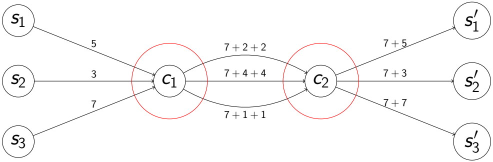
Synchronized PALL

Synchronized Version of PALL



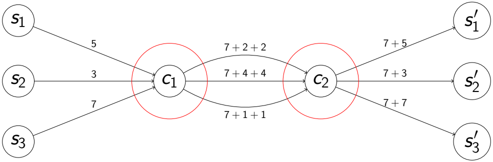
The antennas are **synchronized**: All messages are emited at the same date.

Synchronized Version of PALL



The antennas are **synchronized**: All messages are emited at the same date. There is no more offsets, but a buffer in each contention point.

Synchronized Version of PALL

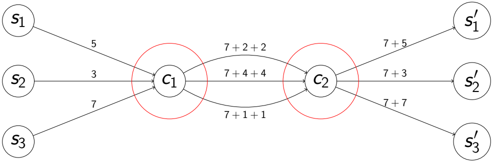


The antennas are **synchronized**: All messages are emited at the same date. There is no more offsets, but a buffer in each contention point.

Definition

An Assignment is a choice of buffering time for each message in every contention point.

Synchronized Version of PALL



The antennas are **synchronized**: All messages are emited at the same date. There is no more offsets, but a buffer in each contention point.

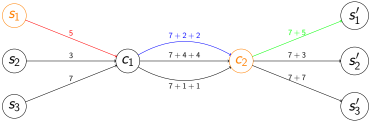
Definition

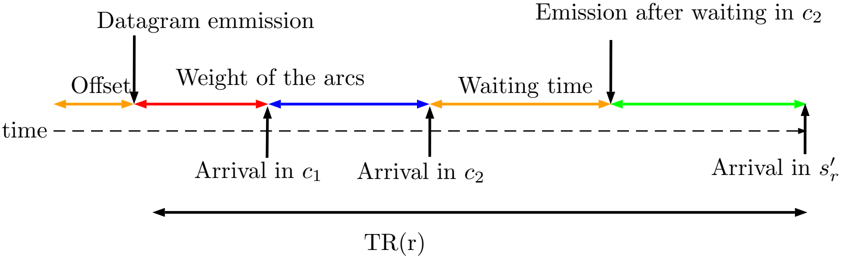
An Assignment is a choice of buffering time for each message in every contention point.

Synchronized Periodic Assignment for Low Latency (SPALL)

Given the routed network, find an assignment satisfying the deadlines constraints.

Transmission Time in SPALL





Datagram emmission

Emission after waiting in c_2

Offset

Weight of the arcs

Waiting time

time

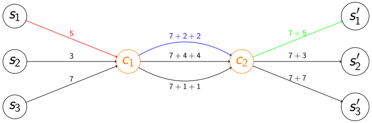
Arrival in c_1

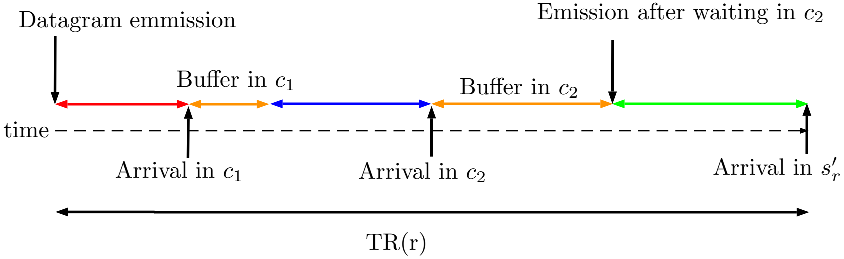
Arrival in c_2

Arrival in s'_r

TR(r)

Transmission Time in SPALL





Datagram emmission

Emission after waiting in c_2

Buffer in c_1

Buffer in c_2

time

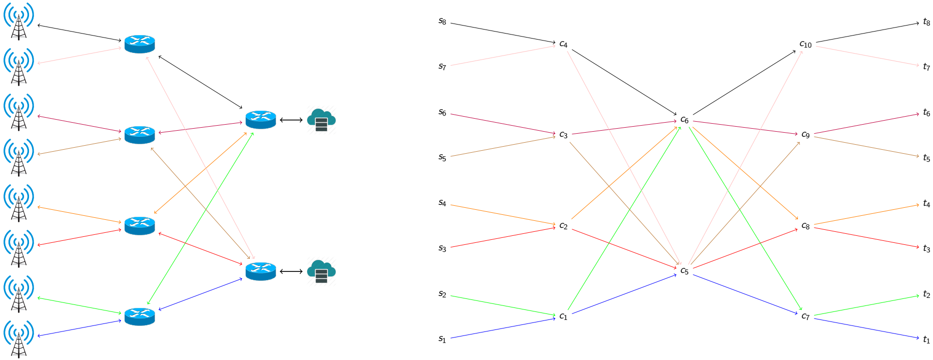
Arrival in c_1

Arrival in c_2

Arrival in s'_r

TR(r)

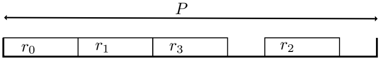
Deeper networks



Each contention point at the same level can be solved independantly.
Once all the contention points of the same level have been solved, we deal with the contention points of the next level.

Compact representation

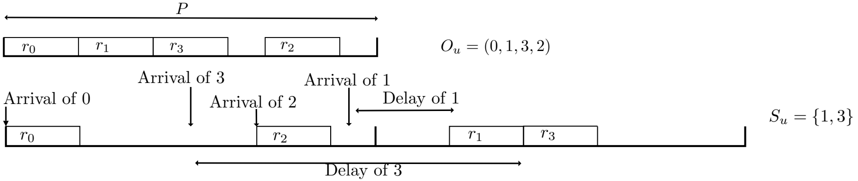
A compact representation of an assignment on a contention point *u* is a pair (O_u,...).



$O_u = (0, 1, 3, 2)$

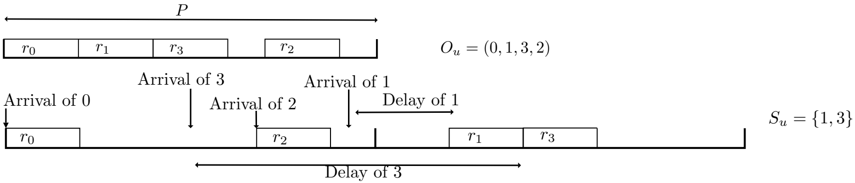
Compact representation

A compact representation of an assignment on a contention point u is a pair (O_u,S_u).



Compact representation

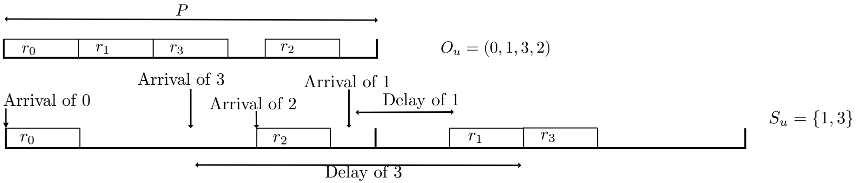
A compact representation of an assignment on a contention point u is a pair (O_u,S_u).



- Not all assignments have a compact representation.

Compact representation

A compact representation of an assignment on a contention point u is a pair (O_u, S_u).

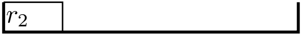


- Not all assignments have a compact representation.
- Several assignments can have the same compact representation.

Solving SPALL: Assignment from a compact representation

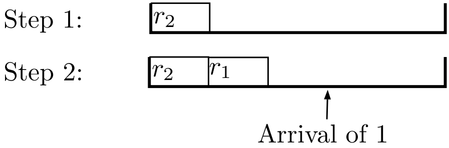
Inductive construction of an assignment from the compact representation.
Example for $((2, 1, 0, 3), \{1\})$ on a single contention point .

Step 1:



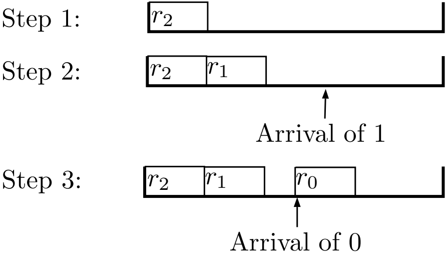
Solving SPALL: Assignment from a compact representation

Inductive construction of an assignment from the compact representation.
Example for $((2, 1, 0, 3), \{1\})$ on a single contention point .



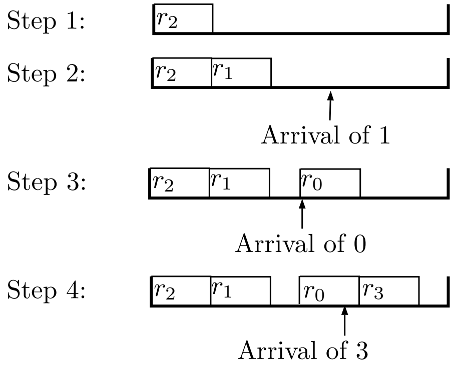
Solving SPALL: Assignment from a compact representation

Inductive construction of an assignment from the compact representation.
Example for $((2, 1, 0, 3), \{1\})$ on a single contention point .



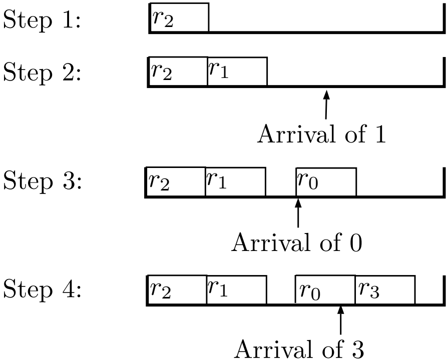
Solving SPALL: Assignment from a compact representation

Inductive construction of an assignment from the compact representation.
Example for $((2,1,0,3),\{1\})$ on a single contention point .



Solving SPALL: Assignment from a compact representation

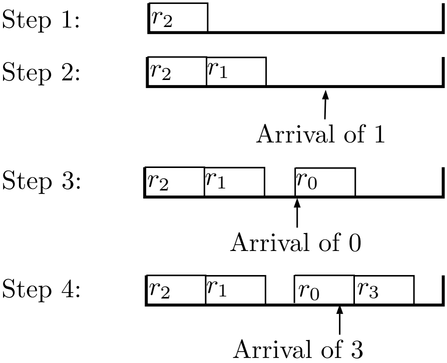
Inductive construction of an assignment from the compact representation.
Example for $((2, 1, 0, 3), \{1\})$ on a single contention point .



- The assignment built from a compact representation is minimal.

Solving SPALL: Assignment from a compact representation

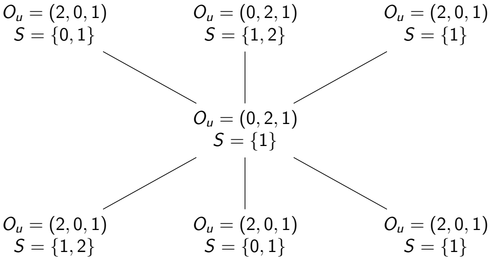
Inductive construction of an assignment from the compact representation.
Example for $((2, 1, 0, 3), \{1\})$ on a single contention point .



- The assignment built from a compact representation is minimal.
- Help to reduce the number of solutions to explore. Exponential in the number of routes an depth of the network.

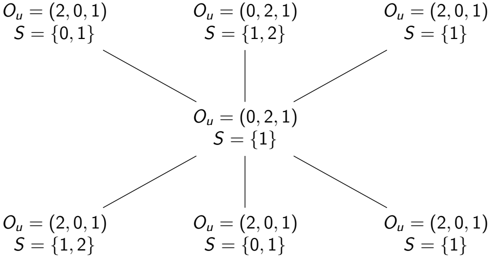
Neighborhood and Local Search Algorithms

Neighborood of a compact representation defined by a permutation:


$O_u = (2,0,1)$ $S = \{0,1\}$
$O_u = (0,2,1)$ $S = \{1,2\}$
$O_u = (2,0,1)$ $S = \{1\}$
$O_u = (0,2,1)$ $S = \{1\}$
$O_u = (2,0,1)$ $S = \{1,2\}$
$O_u = (2,0,1)$ $S = \{0,1\}$
$O_u = (2,0,1)$ $S = \{1\}$

Neighborhood and Local Search Algorithms

Neighborood of a compact representation defined by a permutation:



$O_u = (2,0,1)$
$S = \{0,1\}$

$O_u = (0,2,1)$
$S = \{1,2\}$

$O_u = (2,0,1)$
$S = \{1\}$

$O_u = (0,2,1)$
$S = \{1\}$

$O_u = (2,0,1)$
$S = \{1,2\}$

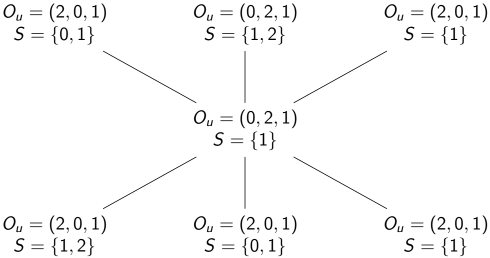
$O_u = (2,0,1)$
$S = \{0,1\}$

$O_u = (2,0,1)$
$S = \{1\}$

Initial Solution : Greedy similar to PALL.

Neighborhood and Local Search Algorithms

Neighborood of a compact representation defined by a permutation:



$O_u = (2,0,1)$
$S = \{0,1\}$

$O_u = (0,2,1)$
$S = \{1,2\}$

$O_u = (2,0,1)$
$S = \{1\}$

$O_u = (0,2,1)$
$S = \{1\}$

$O_u = (2,0,1)$
$S = \{1,2\}$

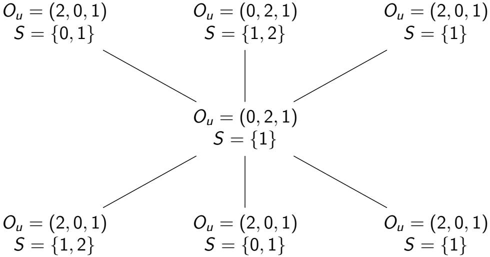
$O_u = (2,0,1)$
$S = \{0,1\}$

$O_u = (2,0,1)$
$S = \{1\}$

Initial Solution : Greedy similar to PALL.
Algorithms : Hill Climbing ,

Neighborhood and Local Search Algorithms

Neighborood of a compact representation defined by a permutation:



$O_u = (2, 0, 1)$
$S = \{0, 1\}$

$O_u = (0, 2, 1)$
$S = \{1, 2\}$

$O_u = (2, 0, 1)$
$S = \{1\}$

$O_u = (0, 2, 1)$
$S = \{1\}$

$O_u = (2, 0, 1)$
$S = \{1, 2\}$

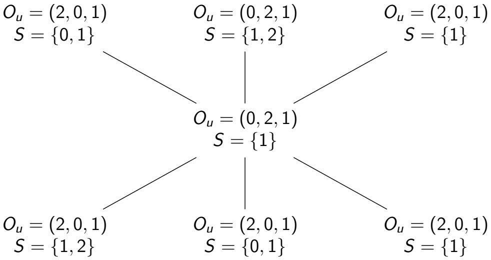
$O_u = (2, 0, 1)$
$S = \{0, 1\}$

$O_u = (2, 0, 1)$
$S = \{1\}$

Initial Solution : Greedy similar to PALL.
Algorithms : Hill Climbing , Tabu Search,

Neighborhood and Local Search Algorithms

Neighborood of a compact representation defined by a permutation:



$O_u = (2,0,1)$ $S = \{0,1\}$

$O_u = (0,2,1)$ $S = \{1,2\}$

$O_u = (2,0,1)$ $S = \{1\}$

$O_u = (0,2,1)$ $S = \{1\}$

$O_u = (2,0,1)$ $S = \{1,2\}$

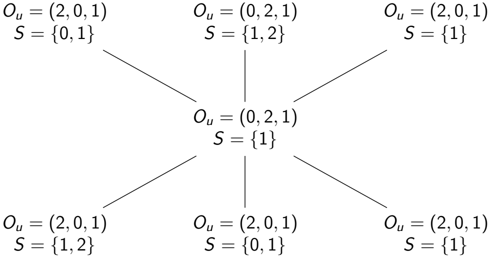
$O_u = (2,0,1)$ $S = \{0,1\}$

$O_u = (2,0,1)$ $S = \{1\}$

Initial Solution : Greedy similar to PALL.
Algorithms : Hill Climbing , Tabu Search, Simulated Annealing,

Neighborhood and Local Search Algorithms

Neighborood of a compact representation defined by a permutation:



$O_u = (2, 0, 1)$ $S = \{0, 1\}$

$O_u = (0, 2, 1)$ $S = \{1, 2\}$

$O_u = (2, 0, 1)$ $S = \{1\}$

$O_u = (0, 2, 1)$ $S = \{1\}$

$O_u = (2, 0, 1)$ $S = \{1, 2\}$

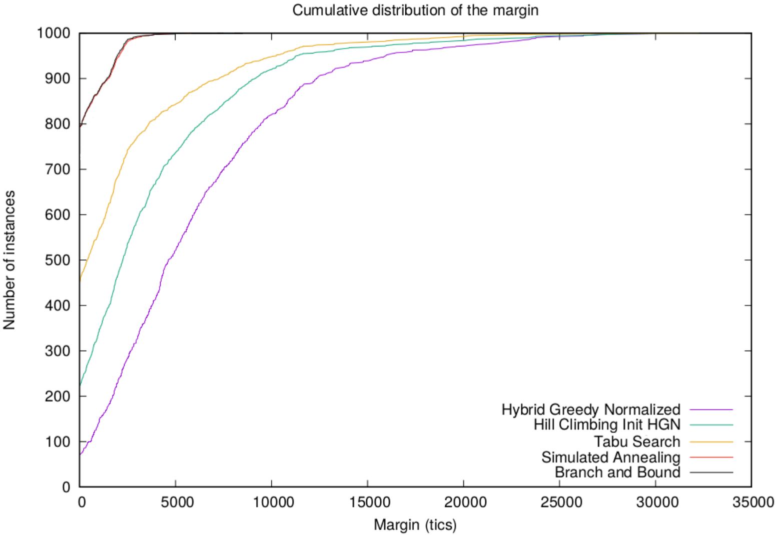
$O_u = (2, 0, 1)$ $S = \{0, 1\}$

$O_u = (2, 0, 1)$ $S = \{1\}$

Initial Solution : Greedy similar to PALL.

Algorithms : Hill Climbing , Tabu Search, Simulated Annealing, Branch and Bound.

Results: Algorithms to solve SPALL



Conclusion

Conclusion

Key result.

Deterministic Networking is the best way to manage deterministic flows.

Industrial prototype in development.

- Hot topic in telecommunications.
- 2 registered patents.

Complexification of the model.

- Different bandwidth.
- Not the same period for all messages.
- Different messages size.

Open Questions.

- Complexity of PALL on star networks.
- Performances of PALL algorithms on deeper networks.

Thank You
For your time and attention !

Published Papers:

- Dominique Barth, Maël Guiraud, Brice Leclerc, Olivier Marcé, Yann Strozecki
  Deterministic Scheduling of Periodic Messages for Cloud RAN. ICT 2018: 405-410
- Dominique Barth, Maël Guiraud, Yann Strozecki
  Deterministic Contention Management for Low Latency Cloud RAN over an Optical Ring. ONDM 2019: 479-491

Pre-print:

- Dominique Barth, Maël Guiraud, Brice Leclerc, Olivier Marcé, Yann Strozecki
  Deterministic Scheduling of Periodic Messages for Cloud RAN (2021, long version). Submitted to Networks.
- Maël Guiraud, Yann Strozecki
  Scheduling periodic messages on a shared link (2021). Submitted to MFCS.