

The 2nd International Workshop on Communications and Sensor Networks (ComSense-2014) Towards a Dual-Mode Adaptive MAC Protocol (DMA-MAC) for Feedback-based Networked Control Systems

Admar Ajith Kumar Somappa^{a,b}, Knut Øvsthus^a, Lars Michael Kristensen^a

^aFaculty of Engineering and Business Administration, Bergen University College, Norway

^bFaculty of Engineering and Science, University of Agder, Norway

Abstract

Automated control systems play an important part in many industrial domains and the medium used for communication between devices in these systems is in transition from wired to wireless for cost reasons. Control systems have strict requirements on delay, throughput, and reliability, that vary with time during operation. Addressing these requirements requires predictable and robust protocols to be employed, and they must be adaptive to the varying states of the controlled process. In this article, we propose a dual-mode adaptive medium access control protocol that caters for two main operation modes in control systems: the steady mode operation, and the transient mode operation. We present initial performance analysis results focusing on energy consumption, including a comparison with a related single-mode industrial monitoring and control protocol.

© 2014 The Authors. Published by Elsevier B.V.

Selection and peer-review under responsibility of Elhadi M. Shakshuki.

Keywords: Wireless Sensor Actuator Networks, Process Control, Automation, Industrial Application.

1. Introduction

A Wireless Sensor Actuator Network (WSAN)¹ consists of sensors and actuators that use radios to send, relay, and receive information. WSANs are applied across several domains including process and factory automation. The use of WSANs in these application domains has had a significant effect on the process industry, by reducing operating costs and increasing automation. With actuators being part of the system, feedback-based control-loop automation is one of the main applications. Feedback-based control systems that use wired or wireless solutions for data transfer are known as Networked Control Systems (NCSs)². A general control system consists of a reference input, plant output, control input, sensors, actuators, and a controller and a typical feedback-based control loop is shown in figure 1. Traditionally, wired communication has been used in NCSs to achieve high reliability, low delay, and high bandwidth. With the advancement in WSAN technology, industry has started utilizing wireless solutions for NCSs. There are several challenges that are to be addressed when implementing wireless solutions for process control applications³. The primary issues that arise as a result of switching from a wired to wireless medium are packet loss, reduced bandwidth, and increased latency. Considering the fact that nodes in WSAN are generally battery powered, energy efficiency is also an important factor in addition to satisfying real-time requirements.

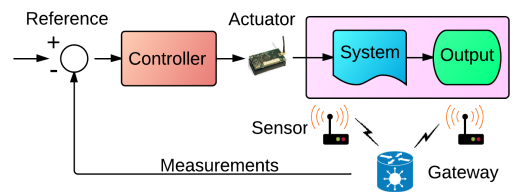


Fig. 1: Feedback based control loop

E-mail address: aaks,kovs,lmkr@hib.no

1877-0509 © 2014 The Authors. Published by Elsevier B.V.

Selection and peer-review under responsibility of Elhadi M. Shakshuki.

In this article, we focus on process control systems based on wireless communication. Process control generally has two states: steady and transient. In the transient state, data values vary rapidly with time, generating large amounts of traffic. Once the data values start stabilizing, the process is said to be in steady state, and the amount of data traffic required is then low. Thus, we propose a medium access control (MAC) protocol designed to have two operational modes to cater for varying process control states. The protocol is called the Dual-Mode Adaptive MAC protocol (DMA-MAC) which defines operation modes based on the two process states. A transient mode that supports the transient state and a steady mode to support steady state. The sensors communicate less frequently during the steady mode thus saving energy and reducing interference to other operations in the vicinity. In a wireless sensor node, the transceiver consumes far more energy than the micro-controller⁴. Thus, by reducing the communication of information during steady state, energy consumption can be reduced. The network architecture considered, consists of a sink node, sensor nodes, and actuator nodes in a tree topology. With sensor/actuator nodes at the lowest level known as leaf nodes, and the parent/child nodes at organized higher levels. The sink node is assumed to be wire powered and is powerful relative to the sensors and actuators. In DMA-MAC operation, the switch between the operational modes is important. In case the measurement crosses a certain threshold (steady state error interval), sensors will notify the sink of this change, and the sink will take the necessary steps to initiate transient mode operation. The switch from transient to steady mode is decided by the sink, and the sensors are notified. The switch decision is made based on the underlying process model, which captures the characteristic features of the process. The main advantage of having a dual-mode adaptive protocol over single-mode protocols is for the case with process models that spends less time in transient state than in the steady state. For such a case the DMA-MAC protocol consumes far less energy than a single-mode protocol. Key design considerations and assumptions used for the design of DMA-MAC are as follows: The setting of thresholds for the sensors to detect the switch is assumed to be based on the process model. For multi-variable input models, we design the transient mode operation to continue until the slowest of the input reaches steady state.

1.1. Important features of DMA-MAC

The aim of our protocol in comparison to existing protocols is to preserve the key performance metrics such as reliability, throughput, low delay, and at the same time improve energy efficiency. The main features that the proposed protocol focuses on, and the overhead caused due to the design are as follows: *Predictability*: process control system procedures have predictable performance since they are pre-calculated to ensure smooth operation. Simulations are performed based on the given process model, to ensure proper design, and to test for predictable performance. From the communication part, with the usage of TDMA based superframes, DMA-MAC has highly predictable performance for the wireless communication. *Reliability*: with the use of reliability mechanisms similar to GinMAC⁵ (a single-mode protocol), reliability is maintained. *Energy conservation*: in the steady mode DMA-MAC protocol has lower data transmission. We propose this protocol for process control scenario where the steady state is dominant compared to the transient state, and hence we have lower energy consumption compared to protocols having single-mode of operation. This is due to the fact that protocols with single-mode of operation should be adhering to transient mode operation at all times to have the required response in transient state. *Overhead*: DMA-MAC protocol has a complex protocol structure due to the dual-mode operation, which results in larger code space, and hence affects hardware performance. Also, the need to ensure efficient state switch detection, and relay within a given duration, is especially critical for state switch from steady to transient.

2. Related Work

GinMAC⁵ is a protocol proposed in the GINSENG⁶ project for process monitoring systems. The main features of this protocol are *Offline dimensioning*, *Exclusive TDMA* (Time Division Multiple Access) and *Delay Conform Reliability Control*. The design of GinMAC was aimed at satisfying real-time constraints for sensors and actuators. The TDMA structure and offline dimensioning makes the protocol predictable. We use GinMAC protocol as a basis for our protocol design and then also for comparison. Breath⁷ is a protocol aimed at reducing energy consumption and providing delay guarantees. Breath is based on Carrier Sense Multiple Access (CSMA) mechanisms and uses clustering techniques. The Breath protocol was designed for control applications. But with the use of CSMA and in the attempt to be adaptive to random conditions, the protocol is aimed at applications that has random possibilities or has continuously varying sampling intervals (time duration between each sensor reading). For process control applications which are known to have precise process models that define transient and steady states along with steady state error,

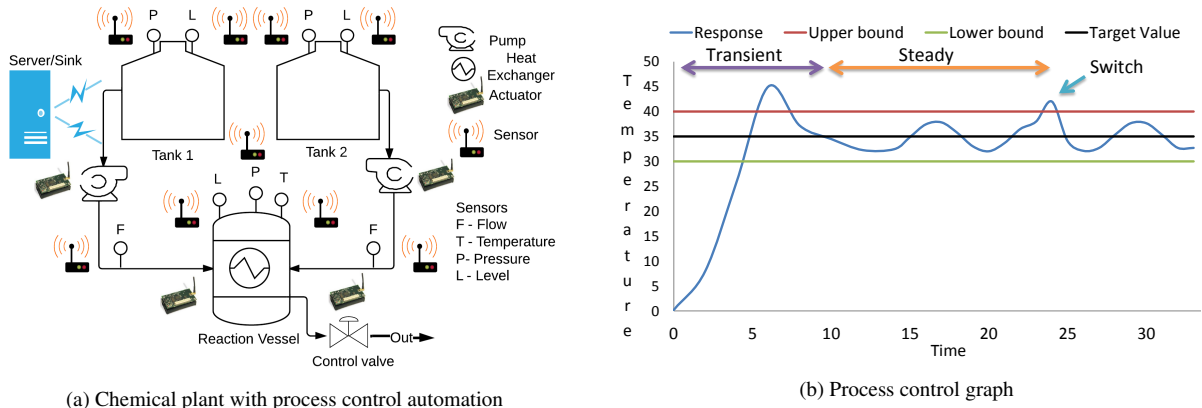


Fig. 2: Scenario description and chart representation of process control variables

could benefit from the use of schedule based protocols, increasing predictability, and also energy efficiency. In⁷, the authors also assume that requirements for TDMA such as having a network manager, are complex requirements. This is in contrast to our protocol, where we assume that these are a common part of the network solution. Thus the two protocols aim at solving different problems. Generic wireless solutions proposed as industrial standards generally have an architecture that consists of a network manager and other powerful middleware. Two popular wireless standards are wirelessHART⁸ and ISA100.11a⁹. Our proposed DMA-MAC protocol can be used in combination with both wirelessHART and ISA100.11a, which are based on the 802.15.4 standard.

3. Networked Control System Example

We define an example scenario of a chemical processing plant, employing NCSs for automating a chemical process using sensors and actuators. The example deals with different kinds of sensors including temperature, pressure, and level. The scenario is similar to the one used in one of the case studies for the GinMAC¹⁰ protocol. Figure 2a shows a schematic diagram of the process. We assume that two chemicals A and B are stored in two different tanks (Tank 1 and Tank 2) and then mixed in the reaction vessel. The reaction is continuously monitored and controlled for temperature, pressure, and level using a feedback-based control system. The goal of the system is to supervise the mixing operation. This is done by continuously measuring all parameters (temperature, pressure, and level) via sensors and making necessary control decisions. The mixing procedure starts with letting the chemicals flow to the reaction vessel. During this process, the flow measurement and management is very important. After the flow is completed and the chemical reactions take place, it is important to continuously manage pressure and temperature. The temperature is controlled via heat exchangers in the reaction vessel. Initially, when the chemicals flow and the reaction starts, there is possibly a high rate of change of temperature and pressure in the reaction vessel (assuming flow is constant). This indicates a transient state. After a while, when the flow is completed, and a given temperature is reached, the rate of change in temperature and pressure will be small. This indicates a steady state. For the example in figure 2a, a typical process control graph is illustrated in figure 2b. The graph is plotted for temperature varying with time, continuously monitored and controlled. Similar behaviour is exhibited by the remaining parameters (pressure and level). We have a target value for temperature (set point or reference), and two bounds (upper and lower) between which the temperature may vary, while the system is still considered to be in the steady state.

4. The DMA-MAC Protocol

The DMA-MAC is proposed based on the dual-mode operation of process control as introduced above. It has one mode for the transient state, and one mode for the steady state. This allows for service differentiation, thus facilitating support for the faster data rate in the transient mode, and saving energy by reducing the data rate in the steady mode. We define two separate superframes to represent slot distribution in the two different operation modes. DMA-MAC is entirely based on a TDMA slot mechanism in both superframes except for alert messages. Alert messages handling is explained below. One important part of DMA-MAC operation is the decision of switching the mode of operation. The transient to steady mode switch is done by the sink based on preset characteristics, i.e. when system is continuously in steady mode operation for the past 10s (sensor reading is within steady state intervals). The steady to transient switch is more critical than the former, and is detected and notified by sensors to the sink. We use thresholds to identify the

switch from steady to transient. As represented in the process control graph in figure 2b, the lower bound and upper bound define the threshold. When in steady mode operation, the sensors constantly monitor for measurements that go beyond the threshold, and when such an event occurs, the sink is notified. The sink then sends a signal indicating the switch of operational mode. Below, we discuss in detail the two operational modes of DMA-MAC.

Transient mode operation. The process changes rapidly during transient state, generating large volume of data sampled at high frequency, to trace the rapidly changing process variables (such as temperature). The sensors transmit this data to the sink. Actuators continuously act on this data. We have a short superframe with N_t (t for transient) slots and smaller sleep duration compared to steady mode operation, to increase data reliability. In general, transient mode operation is data-intensive and hence also energy intensive. The transient mode operation superframe is based on the GinMAC⁵ superframe with similar structure except for the placement of actuator slots. The superframe structure for transient mode operation is illustrated in figure 3a and consists of:

- 1 notification slot for information dissemination from the sink to all sensors and actuators. This includes time synchronization, switching of operational mode, change of thresholds, and other control data. With the switch notification, all nodes are notified to switch in the next superframe.
- S slots for sensor data transmission towards the sink.
- $S * R_T$ re-transmission slots for increased reliability. Here R_T is the number of re-transmissions per sensor data. By setting the number of re-transmissions for transient and steady mode operation, we can have further control the tradeoff between energy conservation and reliability.
- 1 slot for information processing at the sink.
- A_c slots for actuator data transmission from the sink.
- The remaining slots out of the N_t slots are used as sleep slots.

Steady mode operation. The steady mode operation in DMA-MAC is defined to operate during the steady state of the process. During the steady state, the data-rate requirement of the controller is low, and thus we also keep the communication of the sensed data low to save energy. The superframe structure for steady mode operation has N_s (S for steady) slots and is designed to be a multiple of N_t (number of slots in transient mode). This is to ensure that the maximum delay between the detection of a threshold violation and change of superframes is two transient superframe in length. The superframe structure is shown in figure 3b, and consists of:

- N_s/N_t notification slots for information dissemination, similar to transient mode operation. With the switch notification, all nodes are notified to switch in the next N_t part of the steady mode superframe.
- S slots for sensor data transmission towards the sink.
- $S * R_S$ re-transmission slots, here R_S is based on the link conditions, similar to re-transmission slots in GinMAC. Also, here R_S is the number of re-transmissions per sensor data set for steady mode operation.
- 1 slot for information processing at the sink.
- A_c slots for actuator data transmission from the sink.
- $Al * (N_s/N_t)$ slots for alerting the sink of a need for switch of operational modes from steady to transient. After every N_t slots in the steady mode superframe, we repeat the notification slots, and alert slots.
- The remaining slots out of the N_s slots are used as sleep slots.

Alert slots are used to indicate the switch of process state from steady to transient state. The switch is detected by the sensors based on thresholds determined from the process model. The sending procedure for alert messages does not adhere to TDMA, but is based on a special procedure. During the alert message sending slots, mainly the parent nodes are awake to relay any information coming from their children. All leaf nodes are at sleep, unless they have

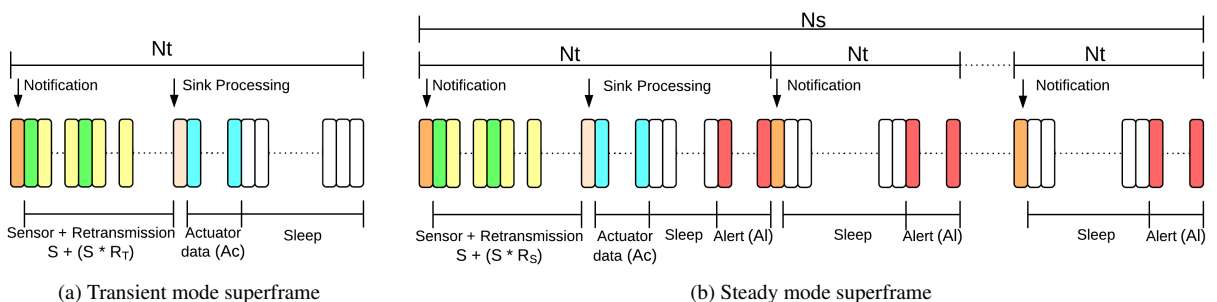


Fig. 3: Superframes for the two modes of the DMA-MAC protocol

alert messages to be sent. Alert messages are usually short messages and two (one duplicate) of these messages are sent in the same slot to increase the probability of receiving the alert message. Any of the child nodes could send an alert message to its parent in alert slots. Thus we could have a possibility where two child nodes of the same parent have alert messages created and need to send them in the same slot. To solve the possible collision at the relay node, we introduce a random delay in the interval $[0..slotDuration/n]$ before sending the first alert message, and then again a random delay within the same range $[0..slotDuration/n]$ as the previous one before sending the next alert message. We use $n \geq 3$, to ensure that the two alert messages can be accommodated in the same slot. Thus, we attempt to increase the probability of at least one of the alert messages reaching the sink, and the operational mode switch being triggered. An implicit ACK mechanism is used, where the node that generated alert message listens to the channel in the next slot to make sure that the alert message is relayed towards the sink by the parent node. We can also consider the addition of CSMA for alert message transmission, to further reduce collision.

The *Sink* sets the threshold depending on the process model given as input by the process model designers. The sink is also responsible for detecting the change in states from transient to steady, make the necessary switch in operational modes, and inform all sensors and actuators under its management. Timing synchronization is handled by the sink, which is a basic requirement for having TDMA operation. *Time Synchronization* information is sent in the information notification slot every λ frames. It is sent in the beginning of the frame for transient mode operation. For steady mode operation it is sent in the beginning of the superframe and then after every Nt slots. We assume a powerful sink and that the message sent by the sink is received by all nodes in the network. We use acknowledgements (ACK) for all packets sent by the sensors. For packets sent by sink, no ACK is used. The ACK packets are sent within the same time slot after the data packets. On failure of the data or ACK packets, re-transmission slots are used.

5. Initial Evaluation of Energy consumption

In this section, we describe the energy model for both GinMAC and DMA-MAC, and provide a first comparison between them. We assume that the transient mode superframe and the GinMAC superframe are of the same length to have similar performance for transient state. We use δ to denote the slot duration (e.g. 10ms). The main radio states being TX - Transmission, RX - Receiving, and SLEEP - Sleeping. We define $P_{RadioState}$ as the power consumed by the radio in the defined radio state. We denote energy consumed in a given radio state by $E_{radioState}$. $E_{Protocol}$ is the total energy spent in a single superframe by a given protocol (having one type of superframe structure). $E_{SLEEP}^{superFrameType}$ is used to describe the total sleep energy spent in one superframe of a given superframe type. Similarly, $E_{protocol}^{superFrameType}$ is used to describe the total energy spent in one superframe of a given superframe type of the respective protocol. We define α as the total number of transient superframes for a given time duration T . $E_{protocol}(\alpha)$ denotes the total energy spent for α superframes using the given protocol. p defines the probability of transient superframes appearing and $(1 - p)$ defines the probability of steady superframes appearing. We present this analysis for one hop networks, and we assume in the case of alerting, that the switch always happens in the last alert duration of the superframe. We assume worst case for re-transmissions, i.e., that all retransmissions are used. A more detailed calculation of energy consumption would also take into account the switching power and time. For simplicity, we neglect these, and the computation of energy spent by sink which is wire powered is omitted. Based on the assumptions, we have:

$$E_{TX} = P_{TX} * \delta, E_{RX} = P_{RX} * \delta, E_{SLEEP} = P_{SLEEP} * \delta, \text{ for worst-case analysis we assume that the entire slot is used.}$$

$$E_{SLEEP}^{Transient} = E_{SLEEP} * (Nt - 1 - S - (S * R_T) - Ac), \text{ subtraction of 1 is for notification slot.}$$

Alert is considered only once in a steady superframe, after which transient superframe starts and hence:

$$E_{SLEEP}^{Steady} = E_{SLEEP} * (Ns - (1 * (Nt/Ns)) - S - (S * R_S) - Ac - Al), \text{ (1 * (Nt/Ns)) is for notification slots} \quad (1)$$

$$E_{GinMAC}(\alpha) = \alpha * (E_{RX} * 1 + E_{TX} * S + E_{TX} * (S * R_T) + E_{RX} * Ac + E_{SLEEP}^{Transient}) \text{ for } \alpha \text{ superframes in time T.} \quad (2)$$

$$E_{DMA-MAC}^{Transient} = E_{GinMAC}$$

$$E_{DMA-MAC}^{Steady} = E_{RX} * (Nt/Ns) + E_{TX} * S + E_{TX} * (S * R_S) + E_{RX} * Ac + E_{TX} * Al + E_{SLEEP}^{Steady} \quad (3)$$

$$E_{DMA-MAC}(\alpha) = p * \alpha * E_{DMA-MAC}^{Transient} + (1 - p) * \frac{\alpha}{Ns/Nt} * E_{DMA-MAC}^{Steady} \quad (4)$$

Table 1 summarizes the relative energy consumption of DMA-MAC and GinMAC. For conditions where we have high number of steady state superframes with $p < 0.1$ for transient superframe to appear during operation, the DMA-MAC has far better energy performance than GinMAC. For other possibilities when the process state has $p < 0.5$ for the next frame to be transient frame, DMA-MAC has still lower consumption of energy. For process models where transient state dominates through the process duration it is advisable to use single-mode protocols like GinMAC especially to prevent the overhead, and this is represented by the $p = 1$ case.

On a simpler note, the main advantage of energy saving is in the steady state of DMA-MAC as shown for a small comparison in Figure 4. Here we show the superframe of normal steady mode (SF), and the steady mode superframe (SF^*) where a switch happens once for a time duration T . Assuming that energy spent on sleep and alert are negligible for a given time T (with α superframes), the energy consumed by DMA-MAC steady state is approximately equal to the energy consumed by GinMAC divided by the multiple (Ns/Nt), we have:

$$E_{DMA-MAC}^{Steady}(\alpha) \simeq \frac{\alpha}{Ns/Nt} * E_{GinMAC} \quad \text{or from eqn.(2):} \quad E_{DMA-MAC}^{Steady}(\alpha) \simeq \frac{E_{GinMAC}(\alpha)}{Ns/Nt} \quad (5)$$

6. Conclusion

In this article, we have proposed a WSN protocol DMA-MAC that is specifically designed for process control applications and requirements. Also, we consider the system dynamics for the process control applications. The system dynamics for process control include two prominent states: steady state and transient state. We have based the protocol design on reflecting the system dynamics states with dual-mode operation. With dual-mode operation, we aim to obtain a better energy efficiency preserving the required reliability and delay requirements. By construction and initial performance analysis, we see that DMA-MAC is energy efficient for certain process conditions that we discussed in the energy consumption evaluation section. An important part of future work is to conduct a more elaborate performance analysis based on other important metrics for process control applications including delay, packet error rate, throughput, and reliability. Furthermore, we will extend the energy evaluation to consider multi-hop communication. A practical evaluation of the proposed design is also planned both in terms of validating that the new protocol is better in practice, and also in terms of figuring out whether there are elements of the design which, in practice is difficult to implement.

References

1. I. F. Akyildiz, I. H. Kasimoglu, Wireless sensor and actor networks: research challenges, *Ad Hoc Networks* 2 (4) (2004) 351–367.
2. J. P. Hespanha, P. Naghshtabrizi, Y. Xu, A survey of recent results in networked control systems, Vol. 95, 2007, pp. 138–162.
3. G. Zhao, Wireless sensor networks for industrial process monitoring and control: A survey., *Network Protocols & Algorithms* 3 (1) (2011) 806–838.
4. A. Bachir, M. Dohler, T. Watteyne, K. Leung, MAC essentials for wireless sensor networks, *IEEE Communications Surveys Tutorials* 12 (2) (2010) 222–248.
5. P. Suriyachai, J. Brown, U. Roedig, Time-critical data delivery in wireless sensor networks, in: *Proceedings of DCOSS, 2010*, pp. 216–229.
6. T. O Donovan, J. Brown, U. Roedig, C. J. Sreenan, J. do O, A. Dunkels, A. Klein, J. Silva, V. Vassiliou, L. Wolf, GINSENG: Performance control in wireless sensor networks, in: *SECON, 2010*, pp. 1–3.
7. P. Park, C. Fischione, A. Bonivento, K. Johansson, A. Sangiovanni-Vincent, Breath: An adaptive protocol for industrial control applications using wireless sensor networks, *IEEE Transactions on Mobile Computing* 10 (6) (2011) 821–838.
8. J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, Wirelesshart: Applying wireless technology in real-time industrial process control, in: *IEEE Real-Time and Embedded Technology and Applications Symposium, 2008, 2008*, pp. 377–386.
9. Draft standard ISA100.11a, in: *Internal working draft, International Society of Automation, 2008*.
10. F. Busching, W. Pottner, D. Brokelmann, G. Von Zengen, R. Hartung, K. Hinz, L. Wolf, A demonstrator of the ginseng-approach to performance and closed loop control in wsns, in: *Ninth International Conference on Networked Sensing Systems (INSS), 2012*, pp. 1–2.

Condition	Energy consumption
$p < 0.1$	$E_{DMA-MAC}(\alpha) \ll E_{GinMAC}(\alpha)$
$p < 0.5$	$E_{DMA-MAC}(\alpha) < E_{GinMAC}(\alpha)$
$p = 1$	$E_{DMA-MAC}(\alpha) = E_{GinMAC}(\alpha)$

Table 1: Results summary

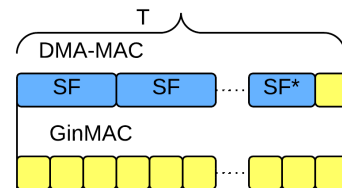


Fig. 4: Steady state