UNIVERSITÀ CA' FOSCARI DI VENEZIA
Dipartimento di Informatica
Technical Report Series in Computer Science

Rapporto di Ricerca CS-2009-9

Ottobre 2009

L. Gallina, S. Rossi

A Process Calculus for Mobile Ad-Hoc Networks Analysis

Dipartimento di Informatica, Università Ca' Foscari di Venezia
Via Torino 155, 30172 Mestre–Venezia, Italy

A Process Calculus for Mobile Ad-Hoc Networks Analysis

Lucia Gallina and Sabina Rossi

Dipartimento di Informatica, Università Ca' Foscari, Venice
{lgallina,srossi}@dsi.unive.it

Abstract. We propose an extension of CMN (Calculus of Mobile Ad-Hoc Networks), a process calculus to study behavioural theory of Mobile Ad-Hoc Networks. The operational semantics of this calculus is given both in terms of a *Reduction Semantics* and in terms of a *Labelled Transition Semantics*. LTS semantics is used to introduce notions of simulation and bisimulation, and we prove that bisimulation is a complete characterisation of *Reduction Barbed Congruence*. Using bisimulation we prove some new properties of our CMN, to demonstrate how our extension have enriched the calculus. Finally, as a practical example of the usefullness of CMN, we use our model to realize one of the most used routing protocols for Mobile Ad-Hoc Networks: the AODV (Ad-Hoc On-Demand Distance Vector Routing).

1 Introduction

1.1 Mobile Ad-Hoc Networks

A mobile ad-hoc network (MANET) is a self-configuring network composed of devices connected by wireless links. The network is composed of both mobile and stationary nodes, then its topology may change rapidly and unpredictably, Mobile devices are free to move randomly and organize themselves arbitrarily. Mobile Ad-Hoc Networks are built using wireless technology, the devices communicate with each other via radio transceivers, using the protocol IEEE 802.11 (WiFi) [24]. This type of communication has a physical scope, because a radio transmission spans over a limited area. Therefore it necessarily must be applicated a routing protocol, proper to wireless dynamic systems.

As Mobile Ad-Hoc Networks communicate in a self organized way without depending on any fixed infrastructure, they are the best solution to the problem of providing a communication system in emergency environments, as a natural catastrophe or a military conflict. However they are vulnerable to many attacks, because the nodes only communicate using radiofrequency channels, which cannot be private. They are also power limited, because constituted of various devices as mobile phones and notebooks. The dynamic nature of this kind of networks makes the management of the transmissions and of the routing protocol much more complicated. The ad hoc networks are self organized, so the good

behavior of the system depends on the cooperation among the connected nodes: this characteristic can make the network more vulnerable to damages caused, not only by malicious nodes, but also by "lazy" nodes, that are devices which, for power saving, do not cooperate with the other nodes; this bad behavior can originate problems especially in the management of the packets routing. Since the Mobile Ad-Hoc networks are used for the management of critical situations, where the transmitted data are often important and confidential, we are now going to enumerate the main properties to be preserved in planning an ad hoc network [25], [4], [19].

Security issues in mobile Ad-Hoc Networks

Authentication
Enables node to ensure the identification of the nodes they are communicating with.

Integrity
Guarantees that the message does not be modified or corrupted during its transmission

Confidentiality
Ensures that confidential data are never disclosed to unauthorized entities. In some environments also the information about the physical locations of the nodes must be protected (*Location Confidentiality*).

Availability
Ensures the survivability of network services despite denial of service attacks. On the network layer, an adversary could disrupt the routing protocol to disconnect the network. On the higher layers an adversary could bring down high-level services.

Access control
Guarantees the control of network's data flow.

Non-repudiation
Ensures that a device transmitting a message cannot deny having sent it. This future is useful for the detection of compromised nodes.

No traffic diversion
Ensures a control of the information traffic, protecting the network from malicious nodes and from non-cooperative nodes, that do not forward the messages in order to save power.

Cooperation and fairness
Ensures the cooperations of all the nodes of the network.

Power control
Ensures that any action of a node respects the power capacity of the devices (the mobile Ad-hoc networks are constituted of many kinds of devices, as mobile phones and notebooks).

Since the Ad-Hoc Networks are often used in critical situations (especially in case of war), it is not important only to preserve the properties we have just

enumerated, but it is also necessary to preserve the network from security problems [25], [4], [19]. We are going now to list the possible attacks to a MANET. Any attack on ad hoc networks can be categorized as passive or active attack. In a passive attack the malicious node only listens to the traffic, without disturbing the network, while in an active attack the malicious node disturbs the normal operation of the network.

Attacks to mobile Ad-Hoc Networks

Message tampering attack
A node could intercept and alter the content of a packet, compromising data integrity

Message dropping attack
A malicious node could intercept packets and drop them, hampering the correct communication of the nodes in the network.

Message reply attack
Malicious nodes could eavesdrop on the packets transmitted, and reply those packets again later.

Identity falsification
A node could alterate the information about its identity.

Impersonation
A node could communicate with the network, using the identity of an other device.

Network obstructing attack
A malicious node could generate packets only to overload the network, obstructing the correct communications of the nodes.

Non-cooperation attack
A selfish node, that wants to save power, could refuse to cooperate to the correct packet routing within the network.

Not only malicious or selfish nodes can obstruct the correct behavior of a network, but also nodes that have been compromised by physical damages, which can cause non-cooperations, or the transmission of corrupted data within the network (Consider for example the case of a node whose route table has been compromised: this can produce a series of transmission failures).

1.2 Process calculi studying the behavioral theory of MANETS

Ad hoc networks are still a new branch of wireless communication; many researchers have yet tried to create a process calculus in order to model them correctly, for the analysis of their properties and problems. The first step of this work has been then a bibliographic research choosing one of the process calculi already created for modeling MANETS in order to study it and eventually optimize it with some modifications. We paid particular attention to four different calculi, which we have analyzed in parallel evaluating their qualities and defects. All the models have been studied using the *process algebra*, in particular there

have been created extensions of calculi as CCS [13] and π-calculus [14], described using LTS semantics (Labelled Transition System). In order to prove behavioral properties of ad hoc networks, The notions of simulation and bisimulation are always introduced [20], [8], [3]. Properties to be preserved in the realization of a mobile ad hoc network are various and sometimes contradictory (for example if we need to preserve data integrity we have to ignore the necessity of power saving, while an excessive power saving could compromise network availability and data integrity), so each model has paid attention to a different characteristic of MANET, then the calculi that have been analyzed result to be really different from each other.

Singh, Ramakrishnan e Smolka have designed the ω-calculus [22], a conservative extension of the π-calculus [14]. The key future of this calculus is the separation of a node's communication and computational behavior from the description of its physical transmission range. The latter is modelled annotating a process with the set of group names to which it belongs. Since the ω-calculus is a conservative extension of the π-calculus, *scope extrusion* is defined (nodes can create new names and privately share them with other devices). The peculiarity of this calculus is that not only broadcast transmissions are permitted, but also multicast and unicast communications. The communications in the ad hoc networks are realized using WiFi, so only broadcast transmissions are permitted; however it can result useful a representation of multicast and unicast communication, because some routing protocols use them (as the AODV, which is one of the best routing protocol for MANETS) and their realization is anyway possible using the standard IEEE 802.11.

Jens Chr. Godskesen has proposed CMAN (Calculus for Mobile Ad Hoc Networks) [9], where connections between the nodes of the network is expressed using bidirectional links. A tag is associated to each node, containing the logical locations of the nodes it is connected with. Rules of CMAN allow scope extrusion. Contrarily to the other calculi we found in literature, here the network topology has been represented with bidirectionals links between the logical locations of the devices (nodes are not associated to a transmission radius nor a physical location). The author had chosen this solution believing that dealing with the node's behavior in its intentions with the network and its neighbors separately from its physical position could simplify the model.

Nanz and Hankin have introduced the CBS$^{\#}$ [15], an extension of CBS (Calculus of Broadcasting Systems) [17]; the peculiarity of this calculus is the choice of representing a node as a couple composed of a process and a store associated to a location. This solution allows the representation in detail of the network, that is constituted of devices that have an own store. Nevertheless the actions executed on the store are internal to the node, so they are not observable actions: their representation make the calculus heavy without a real optimization. The other important peculiarity characterizing the CBS$^{\#}$ is that the transmission is not considered an atomic action, but, when a node executes an output action, the topology of the network may change arbitrarily before the reception of the message by the neighbors of the sender. On contrary in the other calculi (as CMN

or CMAN) transmissions are considered atomic actions: topology of the network cannot change during a transmission. Notice that, even though the transmission is an atomic action, the input action is not deemed a direct consequence of an output, but information broadcasted to the network can be lost, meaning that no nodes of the network received it.

Finally Merro has introduced CMN [11] (Calculus of Mobile Ad-Hoc Networks), an extension of CCS [13], where the topology of the network is represented by a set of nodes associated with a location and a transmission radius. Contrarily to the other calculi we paid attention to, the connectivity of a node is defined by a physical transmission area, and not by a group of nodes. This characteristic allows one to describe more in detail the observability of the network, and this is the main reason why we have chosen this calculus. By a deep analysis of CMN, we found some limitations of this calculus, as the absence of a rule for arbitrary disconnections and connections of stationary nodes, or the impossibility of representing multicast and unicast communication; even though mobile ad hoc networks use only radio frequencies, which do not allow one to make a channel private, in some cases it is necessary specifying the particular receivers of a message.

1.3 Contribution

We propose CMN$^{\#}$, an extension of CMN, defining a new sintax and semantics. In particular we made two modifications:

- The first modification is about the output actions, in particular we added a tag constituted of a set of locations. This tag is associated to the channel of the transmission, and it indicates the intended recipients of a packet transmitted. Ad hoc networks use the radio-frequencies to communicate, so each packet transmitted will be always receivable by the whole network, because no channels can be private. Nevertheless the specification of the recipients of the messages transmitted allows us to analyze better the behavior of the networks consequently to each transmission; for example we can decide if the transmission radius of the node sending the message is sufficient to reach all the nodes it is addressed to. Using this new tag we can moreover represent unicast and multicast communications; even if channels are not private in mobile ad hoc networks, it is anyway useful a tag specifying which are the nodes that are really interested in receiving a message.
- The second modification has been made by adding four new rules modelling connections and disconnections of the nodes in the network. Since mobile nodes can disconnect from a location, and then reconnect in a different site, while stationary nodes are always bound to a specific location, we introduce different rules for disconnections (and connections) of mobile and stationary nodes.

In order to prove that our extention has really optimized the CMN, we demonstrate that all the properties proved with the original calculus are still

valid, and we introduce also new properties that are really important for the study of mobile ad hoc networks, and that we could not prove with the older CMN.

Our work is organized as following:

- In the second and third sections we introduce sintax and semantics of CMN$^{\#}$, semantics is given both in terms of *Reduction semantics* and in terms of LTS (Labelled Transition System).
- In the fourth section, using LTS semantics, we introduce the notions of Simulation and Bisimulation, and we prove that Bisimulation completely characterizes Reduction Barbed Congruence.
- The fifth section is dedicated to the definition of some properties about CMN$^{\#}$: some of these properties cannot be proved using the older CMN. In this section we also introduce an example of the concrete employment of MANETS, through which we can study the properties we introduced, describing their usefulness in the real world.
- The last section is dedicated to the development (using CMN$^{\#}$) and to the analysis of one of the most used routing protocols in MANETS management: the AODV (Ad-Hoc On-Demand Distance Vector Routing). We choose exactly the AODV to be implemented, because this protocol has the peculiarity of using not only broadcast, but also unicast and multicast communication; this characteristic made CMN not suitable for a correct implementation of AODV because this calculus does not allow the representation of multicast and unicast communication, while with our modifications, we can realize all these kinds of transmissions. In this section, as in the fifth, we propose an example of a concrete employment of AODV, showing the usefulness of its development,that allows us to analyze the protocol and to study its behavior.

2 CMN$^{\#}$

CMN (*Calculus of Mobile Ad Hoc Networks*) is a process calculus proposed by Massimo Merro [11] to model mobile ad hoc networks. In CMN a MANET is modelled as a collection of nodes, running in parallel, and using channels to broadcast messages. In this section an extension of CMN is presented, that enables us to keep focus on some properties of this kind of network, in particular the new calculus proposed supports multicast and unicast communication, and allows one to model the arbitrary and unexpected disconnections and connections of the nodes in the network.

In Table 1 we define the sintax of CMN$^{\#}$. This is defined in a two-level structure: the lower one for processes, the upper one for networks. the channel names set is separated from the names set.

$\mathbf{C}$ = channels set, $d, c \in \mathbf{C}$;
$\mathbf{N}$ = names set. In particular letters m, n are used for nodes, l, k, h for locations and r for transmission radii;

Table 1: **Sintax**

Networks		
M,N	$::= \mathbf{0}$	Empty network
	$\mid M_1 \mid M_2$	Parallel composition
	$\mid (\nu c)M$	Channel restriction
	$\mid n[P]^{\mu}_{\lambda,r}$	Node (or device)
Processes		
P,Q,R	$::= \mathbf{0}$	Inactive process
	$\mid c(\tilde{x}).P$	Input
	$\mid \bar{c}_L\langle\tilde{w}\rangle.P$	Output
	$\mid [w_1 = w_2]P, Q$	Matching
	$\mid A\langle\tilde{w}\rangle$	Recursion
Mobility tags		
μ	$::= \mathtt{m}$	Mobile node
	$\mid \mathtt{s}$	Stationary node
Nodes' locations		
λ	$::= l \mid k$	Connected node
	$\mid nil$	Disconnected mobile node
	$\mid nil(l)$	Disconnected stationary node
Channels' description tags		
L	$::= \{l_1, l_2, l_3, ...\}$	Multicast/unicast channel
	$\mid \infty$	Broadcast channel
	$\mid \epsilon$	Empty channel

$\mathbf{X}$ = variables set (x, y, z);
$\mathbf{V}$ = values set $(\mathbf{X} \cup \mathbf{N} \in \mathbf{V})$;

Values set includes names, variables and in general, any basic value (integers, booleans, etc.). Letters u, v are used for closed values, and w for open values. A tuple $a_1, ..., a_k$ of names is represented by $\tilde{a}$. Networks are collections of nodes (which represent devices), running in parallel, using channels to communicate messages. Network $\mathbf{0}$ denotes empty network. $M_1|M_2$ represents the parallel composition of two networks. In $(\nu c)M$ channel c is private to the nodes of M. The restriction operator models channel restriction but not channel creation.

Processes are sequential and live within the nodes. Process $\mathbf{0}$ denotes inactive processes. The input process $c(\tilde{x}).P$ can receive a tuple of values via channel c and continues as P, with $\tilde{w}$ substituted for $\tilde{x}$. We write $\{\tilde{w}/\tilde{x}\}P$ for the substitution of $\tilde{x}$ with $\tilde{w}$ in P (where $|\tilde{x}| = |\tilde{w}|$). The output process $\bar{c}_L\langle\tilde{w}\rangle.P$ can send a term $\tilde{w}$ via channel c and continue as P. The tag L is used to list the locations of the recipients of the message transmitted; in particular $L = \infty$ means a broadcast transmission, otherwise the message is transmitted with multicast communication (unicast if the set L has only one member). Synctactically L may be a variable, but it must be a set of locations when the prefix is ready to fire. Process $[w_1 = w_2]P, Q$ is the standard if-then-else: it behaves as P if $w_1 = w_2$, as Q otherwise. We write $A\langle\tilde{w}\rangle$ to denote a process defined via a (possibly recursive) definition $A(\tilde{x}) \stackrel{\text{def}}{=} P$, with $|\tilde{x}| = |\tilde{w}|$.

Each node, if connected, has a location and a transmission radius. Nodes cannot be created or destroyed. We write $n[P]^{\mu}_{\lambda,r}$ for a node named n (that is the logic location of the device in the network), located at λ, with r transmission radius, μ mobility tag, and executing a process P. μ is $\mathtt{m}$ for mobile nodes, and $\mathtt{s}$ for stationary nodes; λ denotes the physical location of the node, and it is: a generic location l if the device is a mobile node connected to the network; a fixed location l_n if the device is stationary, connected, node named n; the tag nil if the device is a mobile node disconnected,; the tag $nil(l_n)$ if the device is a stationary, disconnected, node located at l_n.

The possibility of the nodes to communicate with each other is verified looking at the physical locations and the transmission radii, in other words if a node broadcasts a message, this information will be received only by nodes that lie in the area delimited by the transmission radius of the sender. In the definition of the operational semantics we then assume the possibility of comparing locations so to determine wether a node lies or not within the transmission cell of another node. We do so by means of a function $d(\cdot, \cdot)$ which takes two locations and returns their distance. If one of the arguments of this function is nil or $nil(l_n)$, the value returned will be ∞, because disconnected nodes cannot receive any message.

In the process $c(\tilde{x}).P$ variable x is bound in P, giving rise to the standard notions of α-conversion and free and bound variables, denoted by $fv(\cdot)$ and $fb(\cdot)$ respectively. Similarly, in a network of the form $(\nu c)M$, the channel name c is bound in M and the notions of α-conversion and free and bound channels, $fc(\cdot)$

and $bc(\cdot)$, are defined accordingly.

Processes and networks are then identified up to α-conversion. More formally terms are considered as representatives of their equivalence class with respect to $\equiv_\alpha$, and these representatives will always be chosen so that bound names are distinct from free names. A context $\mathcal{C}[\cdot]$ is defined as a network term with a hole, denoted by $[\cdot]$. Contexts are generated by the following grammar:

$$\mathcal{C}[\cdot] ::= [\cdot] \mid [\cdot]|M \mid M|[\cdot] \mid (\nu c)[\cdot] \tag{1}$$

A number of conventions are used to simplify the notation. Parallel composition of networks has lower precedence with respect to restriction. $\prod_{i\in I} M_i$ means the parallel composition of all the networks M_i, $\forall i \in I$. We write $(\nu \tilde{c})M$ as an abbreviation for $(\nu c_1)...(\nu c_k)M$. To denote unicast communication we write c_l for $c_{\{l\}}$. We write $\bar{c}_L\langle w\rangle$ for $\bar{c}_L\langle w\rangle.\mathbf{0}$, $\mathbf{0}$ for $n[\mathbf{0}]^\mu_r$ and $[w_1 = w_2]P$ as an abbreviation of $[w_1 = w_2]P, \mathbf{0}$. We assume that there are no free variables in a network (while there can be free channels). The absence of free variables is trivially maintained as the network evolves. Moreover, as node identifiers denote device network addresses we assume that in any network each node identifier is unique.

Table 2: **Structural Congruence**

$n[[v = v]P, Q]^\mu_{\lambda,r} \equiv n[P]^\mu_{\lambda,r}$	(Struct Then)
$n[[v_1 = v_2]P, Q]^\mu_{\lambda,r} \equiv n[Q]^\mu_{\lambda,r} \quad v_1 \neq v_2$	(Struct Else)
$n[A\langle\tilde{v}\rangle]^\mu_r \equiv n[\{\tilde{v}/\tilde{x}\}P]^\mu_{\lambda,r}$ if $A(\tilde{x}) \stackrel{\text{def}}{=} P \wedge |\tilde{x}| = |\tilde{v}|$	(Struct Rec)
$M\|N \equiv N\|M$	(Struct Par Comm)
$(M\|N)\|M' \equiv M\|(N\|M')$	(Struct Par Assoc)
$M\|\mathbf{0} \equiv M$	(Struct Zero Par)
$(\nu c)\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Res)
$(\nu c)(\nu d)M \equiv (\nu d)(\nu c)M$	(Struct Res Res)
$(\nu c)(M\|N) \equiv M\|(\nu c)N$ If $c \notin fc(M)$	(Struct Res Par)
$M \equiv M$	(Struct Refl)
$N \equiv M$ if $M \equiv N$	(Struct Symm)
$M \equiv M''$ if $M \equiv M' \wedge M' \equiv M''$	(Struct Trans)
$M\|M' \equiv N\|M' \quad \forall M'$ if $M \equiv N$	(Struct Cxt Par)
$(\nu c)M \equiv (\nu c)N \quad \forall c$ if $M \equiv N$	(Struct Cxt Res)

2.1 Reduction Semantics

The dynamics of the calculus are specified by the *Reduction Relation* over networks ($\rightarrow$), described in Table 3. As usual in process calculi, the reduction

semantics relies on an auxiliary relation, called structural congruence ($\equiv$), defined in Table 2. Structural congruence brings the participants of a potential interaction into contiguous positions.

Rule (R-Bcast) models the transmission of a tuple of messages $\tilde{v}$ using channel c_L. Mobile ad hoc networks are always implemented with different kinds of devices, and nodes communicate using radio frequencies, that enable only broadcast of message (monopolizing channels is not permitted). Otherwise a node may decide to communicate with a specific node (or group of nodes), this is the reason why we decided to associate each output action with the set of transmission recipients. The cardinality of this set indicates the kind of communication that has been used: if a node broadcasts a message (the recipient is the whole network), it will use a channel tagged with the infinite set (for example c_∞); if the message has only a recipient (unicast transmission), the sender will use a channel as c_l; finally, if the cardinality of L is a finite number more than one there will be a multicast transmission. The recipients set indicates which are the nodes really interested in receiving that particular information, but we know that every message sent from a node will be potentially received by all the devices lying within the transmission cell of the sender, because radio frequencies do not allow one to make a channel private. We have decided to specify anyway the recipients set, to better describe the behavior of the network during the transmissions. If two nodes want to share a secret, they must use cryptography to hide the message.

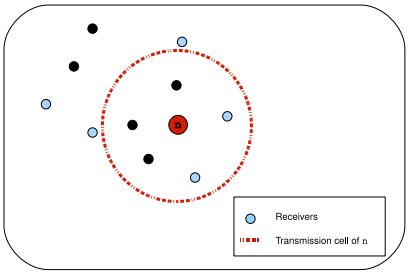
In our caulculus transmission is a *non-blocking action*: transmission proceeds even if there is no other process listening for messages.This is an instantaneous action and the message transmitted will be received only by those nodes which lie in that instant in the transmission area of the sender. Notice that when a transmission occurs, some receivers within the range of the transmitter might not receive the message. This may be due to several reasons that concern the instability and dynamism of the environments where ad hoc networks are usually installed. In terms of observation this corresponds to a local activity of the network which an external observer is not party to. In this calculus movement is considered an atomic action: while moving, a node cannot do anything else.

Rule (R-Move) models arbitrary and unpredictable movements of mobile nodes. δ denotes the maximum distance that a node can cover in a computational step. Moreover there are specific rules modelling arbitrary connections and disconnections of nodes, due to several reasons (as hardware or software problems of the device). Notice that stationary nodes can only disconnect or connect, but they cannot move. Remaining rules are standard in process calculi.

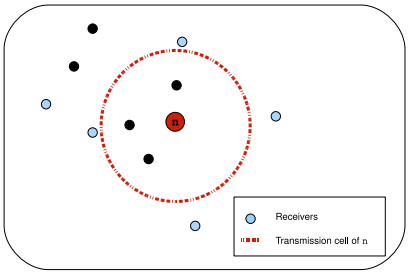
The symbol $\longrightarrow^*$ denotes the reflexive and transitive closure of $\longrightarrow$, Figure 1 shows an example of observable action. Suppose that a node $\mathtt{n}$ (the red node in figure) broadcasts a message to a set L of devices. Black nodes represent all the locations of the network not included in L, while light blue nodes represent locations in L, which are the real receivers of the message. Figure 1a models the case in which at least one of the locations in L lies in the transmission area of the sender, while Figure 1b models a *non-observable action*, where none of the locations in L is able to receive the message.

Fig. 1: Transmission observability

(a) Observable action



(b) Non-observable action



2.2 Behavioral Semantics

In operational semantics two terms are deemed equivalent if they have the same observational behaviour in all possible contexts. The central actions of the calculus here proposed are input and output of a message, but only the output action is observable. An observer cannot be sure whether an intended receiver actually receive a given value. Instead, if a node receive a message, then surely someone must have sent it (the network never invents messages!). Following Milner and Sangiorgi [20] we use the term *Barb* as a synonymous of observable. However our definition of barb, compared to the one proposed in [11] presents an important difference: a transmission is considered an observable action only if at least one location of the set of receivers is able to receive the message, in other words if a location in L associated to the channel c can receive the message transmitted.

Definition 1 (Barb). *Let* $M \equiv (\nu \tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r}|M')$, *with* $c \notin \tilde{d}$.
If $\exists k \in L \wedge d(l,k) \leqslant r$ *then* $M \downarrow_c$
If $M \longrightarrow^* M' \downarrow_c$ *then* $M \Downarrow_c$.

This definition ensures that, for a given process $M \equiv (\nu \tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r}|M')$, if $M \downarrow_c$, we can be sure that at least one of the recipients of the message $\tilde{v}$ is able to correctly listen to the transmission. This process will be in the form $N \equiv (\nu \tilde{d}_i)(n_i[c(\tilde{x}).Q]^{\mu_i}_{l_i,r_i}|N')$, with $c \notin \tilde{d}_i$ and $l_i \in \{k : k \in L \wedge d(k,l) \leqslant r\}$.

Definition 2. *A relation* $\mathcal{R}$ *is* barb preserving *if* $M\,\mathcal{R}\,N$ *and* $M \downarrow_c$ *implies* $N \Downarrow_c$

Definition 3. *A relation* $\mathcal{R}$ *is* reduction closed *if* $M\,\mathcal{R}\,N$ *and* $M \longrightarrow M'$ *implies the existence of some* N' *such that* $N \longrightarrow^* N'$ *and* $M'\,\mathcal{R}\,N'$

Definition 4. *A relation* $\mathcal{R}$ *is* contextual *if* $M\,\mathcal{R}\,N$ *implies* $\mathcal{C}[M]\,\mathcal{R}\,\mathcal{C}[N]$ *for all contexts* $\mathcal{C}[\cdot]$.

Table 3: **Reduction semantics**

$$\text{(R-Bcast)}\ \frac{\forall i \in I.d(l,l_i) \leq r \wedge |\tilde{x}| = |\tilde{v}|}{n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | \prod_{i\in I} n_i[c(\tilde{x_i}).P_i]^{\mu_i}_{l_i,r_i} \rightarrow n[P]^{\mu}_{l,r} | \prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i}}$$

$$\text{(R-sDisc)}\ \frac{-}{n[P]^{\mathtt{s}}_{l_n,r} \rightarrow n[P]^{\mathtt{s}}_{nil(l_n),r}} \qquad \text{(R-sConn)}\ \frac{-}{n[P]^{\mathtt{s}}_{nil(l_n),r} \rightarrow n[P]^{\mathtt{s}}_{l_n,r}}$$

$$\text{(R-mDisc)}\ \frac{-}{n[P]^{\mathtt{m}}_{l,r} \rightarrow n[P]^{\mathtt{m}}_{nil,r}} \qquad \text{(R-mConn)}\ \frac{-}{n[P]^{\mathtt{m}}_{nil,r} \rightarrow n[P]^{\mathtt{m}}_{l,r}}$$

$$\text{(R-Move)}\ \frac{d(l,k) \leq \delta}{n[P]^{\mathtt{m}}_{l,r} \rightarrow n[P]^{\mathtt{m}}_{k,r}} \qquad \text{(R-Par)}\ \frac{M \rightarrow M'}{M|N \rightarrow M'|N}$$

$$\text{(R-Struct)}\ \frac{M \equiv N\ N \rightarrow N'\ N' \equiv M'}{M \rightarrow M'} \qquad \text{(R-Res)}\ \frac{M \rightarrow M'}{(\nu c)M \rightarrow (\nu c)M'}$$

Definition 5 (Reduction barbed congruence). *reduction barbed congruence, written $\cong$, is the largest symmetric relation over networks, which is reduction closed, barb preserving, and contextual.*

3 LTS Semantics

In this section we describe the **LTS** semantics (*Label Transition System*) of CMN$^{\#}$. LTS has two sets of rules: one for processes and one for networks.

Table 4 presents the LTS for processes. Transitions are of the form $P \xrightarrow{\eta} P'$, where η ranges over input and output actions. More precisely $c\tilde{v}$ and $\bar{c}\tilde{v}$ denote, respectively, input and output of a tuple $\tilde{v}$ of values at channel c. The grammar for η is:

$$\eta ::= c\tilde{v} \mid \bar{c}_L\tilde{v}. \tag{2}$$

Rules for processes are simple and they not need deeper explanations.

Table 5 contains the LTS for networks. Transitions are of the form $M \xrightarrow{\gamma} M'$, where the grammar for γ is:

$$\gamma ::= c?\tilde{v}@l \mid c_L!\tilde{v}[l,r] \mid c!\tilde{v}@K \mid \tau. \tag{3}$$

Rule (Snd) models the sending, with transmission radius r, of the tuple $\tilde{v}$ of values via channel c to the set L of receivers, while rule (Rcv) models the reception of $\tilde{v}$ at l via channel c. Rule (Bcast) models the propagation of broadcast.

All the nodes lying within the transmission cell of the transmitter may hear the communication, regardless of the fact that the node hearing the communication is a member of L. Rule (Obs) models the observability of a transmission: every output action may be detected (and hence *observed*) by any node located within the transmission cell of the sender. The action $c!\tilde{v}@K$ represents the transmission of tuple $\tilde{v}$ of messages via c to the set of recipients of L whose locations lie within the transmission cell of the transmitter ($K = \{k : k \in L \wedge d(k,l) \leqslant r\}$ where l and r are respectively the location and the transmission radius of the message sender). If $K \neq \emptyset$ this is an observable action corresponding to the barb $\downarrow_c$. Rule (Lose) models both message loss and a local activity of the network which an observer is not party to. τ-actions are used, as commonly in process calculi, to denote non-observable actions. Rule (Move) models migration of a mobile node from a location k to a new location l; again δ represents the maximum distance that a node can cover in a single computational step. The possibility of a node to log on and log out arbitrarily is modelled by rules (m-Conn) and (m-Disc), while rules (s-Conn) and (s-Disc) model arbitrary and unpredictable connections and disconnections of stationary nodes. We use different rules for mobile and stationary nodes, because a mobile node that disconnects, can reconnects with the network in any other location, but a stationary node is bound to its specific location. Finally (Par) and (Res) are standard in process calculi. Notice that since we do not transmit channels, there is no scope extrusion.

On end we prove that LTS-based semantics coincides with the reduction semantics and the notion of observability given in the previous section.

Lemma 1.

1. *If* $M \xrightarrow{c?\tilde{v}@l} M'$, *then there are* n, P, μ, l, r, M_1 *and* $\tilde{d}$, *with* $c \notin \tilde{d}$, *such that*
$$M \equiv (\nu\tilde{d})(n[c(\tilde{x}).P]^{\mu}_{l,r} | M_1)$$
and
$$M' \equiv (\nu\tilde{d})(n[\{\tilde{v}/\tilde{x}\}P]^{\mu}_{l,r} | M_1).$$

2. *If* $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$, *then there are* n, P, μ, l, r, M_1, I *(possibly empty), and* $\tilde{d}$, *with* $c \notin \tilde{d}$, *and* n_i, P_i, μ_i, l_i, r_i, *with* $d(l,l_i) \leq r$ *for all* $i \in I$, *such that:*
$$M \equiv (\nu\tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | \textstyle\prod_{i\in I} n_i[c(\tilde{x_i}).P_i]^{\mu_i}_{l_i,r_i} | M_1)$$
and
$$M' \equiv (\nu\tilde{d})(n[P]^{\mu}_{l,r} | \textstyle\prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i} | M_1).$$

Proof. Proof can be obtained by induction on the transition rules of Table 5.
Case 1: $M \xrightarrow{c?\tilde{v}@l} M'$

(Rcv) Let $M \xrightarrow{c_L!\tilde{v}[l,r]} M'|N'$, then there exist n, P, μ such that $M \equiv n[P]^{\mu}_{l,r}$ and $M \equiv n[P]^{\mu}_{l,r}$. Since $P \xrightarrow{c\tilde{v}} P'$ then there must be Q such that $P = c(\tilde{v}).Q$ and $P' = \{\tilde{v}/\tilde{x}\}Q$, then, if we suppose $\tilde{d}$ and M_1 empty, lemma is proved because $M \equiv n[c(\tilde{v}).Q]^{\mu}_{l,r}$ and $M' \equiv n[\{\tilde{v}/\tilde{x}\}Q]^{\mu}_{l,r}$.

(Par) If $M|N \xrightarrow{c?\tilde{v}@l} M'|N$, then $M \xrightarrow{c?\tilde{v}@l} M'$. Then, by induction hypothesis we have $M \equiv (\nu\tilde{d})(n[c(\tilde{x}).P]^{\mu}_{l,r}|M_1)$ and $M' \equiv (\nu\tilde{d})(n[\{\tilde{v}/\tilde{x}\}P]^{\mu}_{l,r}|M_1)$. Since $\tilde{d}$ is a tuple of new names in M, they can be chosen so that they are new also for N (we consider in this calculus the equivalence with respect to α-conversion). Then, since by hypothesis $c \notin \tilde{d}$, by applying rule (Struct-Res-Par) of structural congruence we can write:
$M|N \equiv (\nu\tilde{d})(n[c(\tilde{x}).P]^{\mu}_{l,r}|M_1|N)$ and $M'|N \equiv (\nu\tilde{d})(n[\{\tilde{v}/\tilde{x}\}P]^{\mu}_{l,r}|M_1|N)$.

(Res) Suppose $(\nu d')M \xrightarrow{c?\tilde{v}@l} (\nu d')M'$, then $M \xrightarrow{c?\tilde{v}@l} M'$. Then, by induction hypothesis $M \equiv (\nu\tilde{d})(n[c(\tilde{x}).P]^{\mu}_{l,r}|M_1)$ and $M' \equiv (\nu\tilde{d})(n[\{\tilde{v}/\tilde{x}\}P]^{\mu}_{l,r}|M_1)$. Since by hypothesis $d' \notin fc(c?\tilde{v}@l)$ then $c \neq d'$. Since $\tilde{d''} = \tilde{d} \bigcup \{d'\}$, then we can write:
$(\nu d')M \equiv (\nu\tilde{d''})(n[c(\tilde{x}).P]^{\mu}_{l,r}|M_1)$ and $(\nu d')M' \equiv (\nu\tilde{d''})(n[\{\tilde{v}/\tilde{x}\}P]^{\mu}_{l,r}|M_1)$.
The other cases follow straightforwardly from congruence rules of the reduction relation.

Case 2: $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$

(Snd) Let $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$, then there exist n, P, μ such that $M \equiv n[P]^{\mu}_{l,r}$ and $M \equiv n[P]^{\mu}_{l,r}$. Since $P \xrightarrow{\bar{c}\tilde{v}} P'$, then there must be Q such that
$P = \bar{c}_L\langle\tilde{v}\rangle.Q$ and $P' = Q$, then, if we suppose $\tilde{d}$ and M_1 empty, lemma is proved because
$M \equiv (\nu\tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r}|M_1)$

(Bcast) Let $M|N \xrightarrow{c_L!\tilde{v}[l,r]} M'|N'$ because $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$ and $N \xrightarrow{c?\tilde{v}@l'} N'$, with $d(l.l') \leq r$. By inducion hypotesis:

$$M \equiv (\nu\tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r}|\prod_{i\in I} n_i[c(\tilde{x_i}).P_i]^{\mu_i}_{l_i,r_i}|M_1)$$

and

$$M' \equiv (\nu\tilde{d})(n[P]^{\mu}_{l,r}|\prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i}|M_1).$$

for some n, P, μ, l, r, M_1, I (possibly empty), and $\tilde{d}$, with $c \notin \tilde{d}$, and n_i, P_i, μ_i, l_i, r_i, with $d(l, l_i) \leq r$ for all $i \in I$, and

$$N \equiv (\nu\tilde{d'})(n'[c(\tilde{x}).Q]^{\mu'}_{l',r'}|N_1)$$

and

$$N' \equiv (\nu\tilde{d'})(n'[\{\tilde{v}/\tilde{x}\}Q]^{\mu'}_{l',r'}|N_1).$$

for some n', Q, μ', l', r', N_1 and $\tilde{d'}$, with $c \notin \tilde{d'}$

then we can write:

$$M|N \equiv (\nu\tilde{d''})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r}|\prod_{i\in I} n_i[c(\tilde{x_i}).P_i]^{\mu_i}_{l_i,r_i}|n'[c(\tilde{x}).Q]^{\mu'}_{l',r'}|M_1|N_1)$$

and

$$M'|N' \equiv (\nu \tilde{d''})(n[P]^{\mu}_{l,r}|\prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i}|n'[\{\tilde{v}/\tilde{x}\}Q]^{\mu'}_{l',r'}|M_1|N_1).$$

with $\tilde{d''} = \tilde{d} \cup \tilde{d'}$ Proof of the other cases is analogous to the first part of the lemma.

Lemma 2 (≡ respects transitions). *If $M \xrightarrow{\gamma} M'$ and $M \equiv N$ then there exists N' such that $N \xrightarrow{\gamma} N'$ and $M' \equiv N'$*

Proof. Proof proceeds by induction on the depth of the inference $M \longrightarrow M'$. The full proof must treat all possible cases for the final step of the inference $M \xrightarrow{\gamma} M'$. Here we consider just some cases.

(Par) Suppose $M|N \xrightarrow{\gamma} M'|N$, inferred by rule (Par), where $M \xrightarrow{\gamma} M'$; Many are the possible structural congruence rules to apply. For example we can consider the rule (Struct Par Comm), then if $M|N \equiv N|M$, by applying (Par) we obtain $N|M \xrightarrow{\gamma} N|M'$, and, by applying (Struct Par Comm) we finally obtain $N|M' \equiv M'|N$. Considering rule (Struct Cxt Par), if $M_1 \equiv M$ and $Q = M_1|N$, then, for induction hypothesis $M \xrightarrow{\gamma} M'$ and $M \equiv M_1$ implies $M_1 \xrightarrow{\gamma} M_1'$ and $M' \equiv M_1'$. We can so deduce $M_1|N \xrightarrow{\gamma} M_1'|N$ and $M'|N \equiv M_1'|N$, by using the rule (Struct Cxt Par).

(Res) Suppose $(\nu c)M \xrightarrow{\gamma} (\nu c)M'$, inferred by $M \xrightarrow{\gamma} M'$ (with $c \notin fc(\gamma)$); now there are many ways in which $(\nu c)M \equiv Q$ due to a single use of a structural congruence rule; we will confine ourselves to considering just some cases.
If $M \equiv N$ for some N, using the rule (Struct Cxt Res) we deduce $(\nu c)M \equiv (\nu c)N$. By induction hypothesis, since $M \xrightarrow{\gamma} M'$ and $M \equiv N$, then there exists N' such that $N \xrightarrow{\gamma} N'$ and $M' \equiv N'$. Using the rule (Res) in $N \xrightarrow{\gamma} N'$, considering that $c \notin fc(\gamma)$, we obtain $(\nu c)N \xrightarrow{\gamma} (\nu c)N'$. Finally, by applying again the rule (Struct Cxt Res) to $M' \equiv N'$ we obtain $(\nu c)M' \equiv (\nu c)N'$.

The other cases are proved in analogous way.

Theorem 1 (Harmony theorem).

1. *$M \downarrow_c$ iff $M \xrightarrow{c!\tilde{v}@K}$ for some value $\tilde{v}$ and some set K of locations.*
2. *If $M \xrightarrow{\tau} M'$ then $M \longrightarrow M'$.*
3. *If $M \longrightarrow M'$ then $M \xrightarrow{\tau} \equiv M'$.*

Proof.

1. The first part follows from definition of barb and from Lemma 1.
2. The second part is proved by induction on the derivation $M \xrightarrow{\tau} M'$.
   Suppose that the τ-action has been generated by an application of the rule (Lose). In this case we have $\dfrac{M \xrightarrow{c_L!\tilde{v}[l,r]} M'}{M \xrightarrow{\tau} M'}$, then, by an application of Lemma 1 we have:

$$M \equiv (\nu \tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | \prod_{i\in I} n_i[c(\tilde{x}).P_i]^{\mu_i}_{l_i,r_i} | M_1)$$

and

$$M' \equiv (\nu \tilde{d})(n[P]^{\mu}_{l,r} | \prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i} | M_1).$$

for some n, P, l, r, μ, $\tilde{d}$, with $c \notin \tilde{d}$, and some n_i, P_i, l_i, r_i, μ_i, such that $d(l, l_i) \leq r \forall i \in I$. By applying rules (R-Bcast), (R-Par), (R-Res) we get

$$\begin{array}{c}(\nu \tilde{d})(n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | \prod_{i\in I} n_i[c(\tilde{x}).P_i]^{\mu_i}_{l_i,r_i} | M_1) \longrightarrow \\ (\nu \tilde{d})(n[P]^{\mu}_{l,r} | \prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i} | M_1)\end{array}$$

and, by applying (R-Struct), we obtain $M \longrightarrow M'$, as required. Suppose now that the τ-action has been generated by an application of rule (Move):

$$\frac{d(k,l) \leq r}{n[P]^{\mathtt{m}}_{l,r} \xrightarrow{\tau} n[P]^{\mathtt{m}}_{k,r}}$$

then, by an application of (R-Move) we get

$$\frac{d(k,l) \leq r}{n[P]^{\mathtt{m}}_{l,r} \longrightarrow n[P]^{\mathtt{m}}_{k,r}}.$$

The other cases, as the rule (Move), follow straightforwardly from congruence rules of the reduction relation.

3. The third part of the theorem is proved by induction on the derivation $M \rightarrow M'$. If we consider the rules where a τ-action in LTS semantics corresponds to a reduction (for example (Move) and (R-Move)), proof can be omitted. We're going to describe the other cases.
Suppose that the derivation $M \longrightarrow M'$ has been generated by an application of rule (R-Bcast)

$$\frac{\forall i \in I.d(l, l_i) \leq r}{n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | \prod_{i\in I} n_i[c(\tilde{x_i}).P_i]^{\mu_i}_{l_i,r_i} \rightarrow n[P]^{\mu}_{l,r} | \prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i}}$$

Then, by applying rules (Snd), (Rcv) and (Bcast) we obtain:

$$\frac{\dfrac{\bar{c}_L\langle\tilde{v}\rangle.P \xrightarrow{\bar{c}_L\tilde{v}} P}{n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} \xrightarrow{c_L!\tilde{v}[l,r]} n[P]^{\mu}_{l,r}} \quad \dfrac{c(\tilde{x_1}).P_1 \xrightarrow{c\tilde{v}} \{\tilde{v}/\tilde{x_1}\}.P_1}{n_1[c(\tilde{x_1}).P_1]^{\mu_1}_{l_1,r_1} \xrightarrow{c?\tilde{v}@l_1} n_1[\{\tilde{v}/\tilde{x_1}\}P_1]^{\mu_1}_{l_1,r_1}} \; d(l, l_1) \leq r}{n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | n_1[c(\tilde{x_1}).P_1]^{\mu_1}_{l_1,r_1} \xrightarrow{c_L!\tilde{v}[l,r]} n[P]^{\mu}_{l,r} | n_1[\{\tilde{v}/\tilde{x_1}\}P_1]^{\mu_1}_{l_1,r_1}}.$$

By applying $|I| - 1$ times rule (Bcast) and one time rule (Lose) we get

$$n[\bar{c}_L\langle\tilde{v}\rangle.P]^{\mu}_{l,r} | \prod_{i\in I} n_i[c(\tilde{x_i}).P_i]^{\mu_i}_{l_i,r_i} \xrightarrow{\tau} n[P]^{\mu}_{l,r} | \prod_{i\in I} n_i[\{\tilde{v}/\tilde{x_i}\}P_i]^{\mu_i}_{l_i,r_i}$$

as required. Suppose that the derivation $M \longrightarrow M'$ has been generated by an application of rule (R-Struct)

Table 4: **LTS**-Processes

$$\text{(Input)}\ \frac{-}{c(\tilde{x}).P \xrightarrow{c\tilde{v}} \{\tilde{v}/\tilde{x}\}P} \qquad \text{(Output)}\ \frac{-}{\bar{c}_L\langle\tilde{v}\rangle.P \xrightarrow{\bar{c}_L\tilde{v}} P}$$

$$\text{(Then)}\ \frac{P \xrightarrow{\eta} P'}{[\tilde{v} = \tilde{v}]P, Q \xrightarrow{\eta} P'} \qquad \text{(Else)}\ \frac{Q \xrightarrow{\eta} Q'\ \tilde{v_1} \neq \tilde{v_2}}{[\tilde{v_1} = \tilde{v_2}]P, Q \xrightarrow{\eta} Q'}$$

$$\text{(Rec)}\ \frac{\{\tilde{v}/\tilde{x}\}P \xrightarrow{\eta} P'\ A(\tilde{x}) \stackrel{\text{def}}{=} P}{A\langle\tilde{v}\rangle \xrightarrow{\eta} P'}$$

$$\frac{M \equiv N\ N \longrightarrow N'\ N' \equiv M'}{M \longrightarrow M'}$$

By induction hypothesis $N \xrightarrow{\tau} \equiv N'$, then there exists N'' such that $N \xrightarrow{\tau} N''$ and $N'' \equiv N'$. Then, using Lemma 2 there exists M'' such that $M \xrightarrow{\tau} \equiv M''$ and $M'' \equiv N''$. By transitivity of $\equiv$ it follows that $M'' \equiv M'$, then $M \xrightarrow{\tau} \equiv M'$ as required. Finally, as both the τ-transitions and structural congruence are preserved by network contexts, the cases when the reduction $M \longrightarrow M'$ is derived either by rules (R-Par) or rule (R-res) are straightforward.

4 Simulation and Bisimulation

In this section, using our LTS, we define notions of simulation and bisimulation. Then we prove that bisimulation is a complete characterization of *reduction barbed congruence*, and hence represents a valid method for proving that two networks are reduction barbed congruent. This property let as deal with all issues that do not permit the correct behavior of mobile ad hoc networks. We then have to prove both that $\cong \subseteq \approx$ and that $\approx \subseteq \cong$.

For convenience we use metavariable α to range over those actions that will be used in the definition of bisimulation.

$$\alpha\ ::=\ c?\tilde{v}@l\ \mid\ c!\tilde{v}@K\ \mid\ \tau.$$

Since we are interested in *weak behavioral equivalences*, that abstract over τ-actions, we introduce the notion of *weak action*.

- $\Rightarrow$ denotes reflexive and transitive closure of $\xrightarrow{\tau}$
- $\overset{c?\tilde{v}@F}{\Longrightarrow}$ denotes $\Rightarrow \xrightarrow{c?\tilde{v}@l_1} \Rightarrow \ldots \Rightarrow \xrightarrow{c?\tilde{v}@l_n} \Rightarrow$ for $F = \{l_1, ..., l_n\}$
- $\overset{c!\tilde{v}@K}{\Longrightarrow}$ denotes $\overset{c?\tilde{v}@F_1}{\Longrightarrow} \xrightarrow{c!\tilde{v}@K_1} \overset{c?\tilde{v}@F'_1}{\Longrightarrow} \ldots \overset{c?\tilde{v}@F_n}{\Longrightarrow} \xrightarrow{c!\tilde{v}@K_n} \overset{c?\tilde{v}@F'_n}{\Longrightarrow}$, for $\bigcup_{i=1}^{n} K_i = K$, $\bigcup_{j=1}^{n}(F_1 \cup F'_1) = F \wedge F \cap K = \emptyset$;

Table 5: **LTS**-Networks

$$\text{(Snd)}\ \frac{P \xrightarrow{\bar{c}_L \tilde{v}} P'}{n[P]^{\mu}_{l,r} \xrightarrow{c_L!\tilde{v}[l,r]} n[P']^{\mu}_{l,r}} \qquad \text{(Rcv)}\ \frac{P \xrightarrow{c\tilde{v}} P'}{n[P]^{\mu}_{l,r} \xrightarrow{c?\tilde{v}@l} n[P']^{\mu}_{l,r}}$$

$$\text{(B-cast)}\ \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} M'\ N \xrightarrow{c?\tilde{v}@l'} N'\ d(l,l') \leq r}{\begin{array}{c} M|N \xrightarrow{c_L!\tilde{v}[l,r]} M'|N' \\ N|M \xrightarrow{c_L!\tilde{v}[l,r]} N'|M' \end{array}}$$

$$\text{(Obs)}\ \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} M'\ K \subsetneq \{k : d(l,k) \leq r \wedge k \in L\}, K \neq \emptyset}{M \xrightarrow{c!\tilde{v}@K} M'}$$

$$\text{(Lose)}\ \frac{M \xrightarrow{c_L!\tilde{v}[l,r]} M'}{M \xrightarrow{\tau} M'} \qquad \text{(Move)}\ \frac{d(l,k) \leq \delta}{n[P]^{\mathtt{m}}_{l,r} \xrightarrow{\tau} n[P]^{\mathtt{m}}_{k,r}}$$

$$\text{(s-Disc)}\ \frac{-}{n[P]^{\mathtt{s}}_{l,r} \xrightarrow{\tau} n[P]^{\mathtt{s}}_{nil(l),r}} \qquad \text{(s-Conn)}\ \frac{-}{n[P]^{\mathtt{s}}_{nil(l),r} \xrightarrow{\tau} n[P]^{\mathtt{s}}_{l,r}}$$

$$\text{(m-Disc)}\ \frac{-}{n[P]^{\mathtt{m}}_{l,r} \xrightarrow{\tau} n[P]^{\mathtt{m}}_{nil,r}} \qquad \text{(m-Conn)}\ \frac{-}{n[P]^{\mathtt{m}}_{nil,r} \xrightarrow{\tau} n[P]^{\mathtt{m}}_{l,r}}$$

$$\text{(Par)}\ \frac{M \xrightarrow{\gamma} M'}{\begin{array}{c} M|N \xrightarrow{\gamma} M'|N \\ N|M \xrightarrow{\gamma} N|M' \end{array}} \qquad \text{(Res)}\ \frac{M \xrightarrow{\gamma} M'\ c \notin fc(\gamma)}{(\nu c)M \xrightarrow{\gamma} (\nu c)M'}$$

– $\stackrel{\hat{\alpha}}{\Longrightarrow}$ denotes $\Rightarrow$ if $\alpha = \tau$ and $\stackrel{\alpha}{\Longrightarrow}$ otherwise;

Notice that the third point of this definition means that a distributed observer receiving an instance of message $\tilde{v}$, at each location in K, in several computational steps, cannot assume that those messages belong to the same broadcast transmission, but they may be different transmissions of the same message. The presence of the weak input actions are due to the fact that we want to ignore all the input executed by each location which is not included in the set of receivers. Let understand with a little example the importance of abstract from this type of input actions.

Example 1. Let M be a process such that

$$M \xrightarrow{c!\tilde{v}@K} M' \text{ because } M \xrightarrow{c_L!\tilde{v}[l,r]} M' \ .$$

Then

$$\forall l' \notin K \; M|n[c(\tilde{x}).P]^{\mu}_{l',r'} \stackrel{c!\tilde{v}@K}{\Longrightarrow} M|n[\{\tilde{v}/\tilde{x}\}P]^{\mu}_{l',r'}.$$

In other words if a node sends a message to a given set L of receivers, we will not mind if other nodes, lying at some location not included in L and listening to channel c, can perform a synchronisation with the node having sent the message or they do not.

Definition 6 (Bisimilarity). *A binary relation $\mathcal{R}$ over networks is a* simulation *if $M\mathcal{R}N$ implies:*

- *If $M \xrightarrow{\alpha} M'$, $\alpha \neq c?\tilde{v}@l$, then there exists N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and $M'\mathcal{R}N'$*
- *If $M \xrightarrow{c?\tilde{v}@l} M'$ then there exists N' such that:*
  - *$N \stackrel{c?\tilde{v}@l}{\Longrightarrow} N'$ and $M'\mathcal{R}N'$*
  - *or $N \Rightarrow N'$ and $M'\mathcal{R}N'$.*

We say that N simulates *M if there is some simulation $\mathcal{R}$ such that $M\mathcal{R}N$. A relation $\mathcal{R}$ is a* bisimulation *if both $\mathcal{R}$ and its converse are simulations. We say that M and N are bisimilar, written $M \approx N$, if there exists some bisimulation $\mathcal{R}$ such that $M\mathcal{R}N$.*

It is easy now to prove that bisimulation is an equivalence relation, because reflexivity and symmetry are trivial, and transitivity follows from definition of $\stackrel{\hat{\alpha}}{\Longrightarrow}$. Notice that there are other important properties of bisimulation, here we will prove closure under contexts.

Lemma 3 ($\approx$ is contextual). *Let M and N be two networks such that $M \approx N$, then:*

1. *$M|O \approx N|O$, for all networks O;*
2. *$(\nu c)M \approx (\nu c)N$, for all channels c.*

Proof. As regards the first item we have to prove that the relation

$$\mathcal{S} = \{(M|O, N|O) \mid \forall O, M \approx N\}$$

is a bisimulation. To prove it we do a case analysis on the transition $M|O \xrightarrow{\alpha} \hat{M}$. The interesting cases are when the transition is due to an interaction between M and O, and this happens by an application of rule (Bcast). Let $M|O \xrightarrow{c!\tilde{v}@K} \hat{M}$ because $M|O \xrightarrow{c_L!\tilde{v}[l,r]} \hat{M}$ for some r, l, with $k \in L \wedge d(l,k) \leq r$, $\forall k \in K$, due to an application of (Bcast). There are then two possibilities:

1. $M|O \xrightarrow{c_L!\tilde{v}[l,r]} \hat{M}$ because $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$ and $O \xrightarrow{c?\tilde{v}@l'} O'$ with $d(l,l') \leq r$ and $\hat{M} = M'|O'$
2. $M|O \xrightarrow{c_L!\tilde{v}[l,r]} \hat{M}$ because $M \xrightarrow{c?\tilde{v}@l'} M'$ and $O \xrightarrow{c_L!\tilde{v}[l,r]} O'$, with $d(l,l') \leq r$ and $\hat{M} = M'|O'$

Case 1. Now we have to consider two different cases, that depend by the presence or the absence of l' in the set L of receivers.

- If $l' \in L$, by an application of rule (Obs) $M \xrightarrow{c!\tilde{v}@K'} M'$, with $K' = K \cup \{l'\}$. As $M \approx N$ then there exists N' such that $N \overset{c!\tilde{v}@K'}{\Longrightarrow} N'$ with $M' \approx N'$. By applying rule (Obs) backward there must be $K_1, ..., K_n$ such that

$$N \Rightarrow \xrightarrow{c!\tilde{v}@K_1} ... \xrightarrow{c!\tilde{v}@K_n} \Rightarrow N'$$

with $\bigcup_{i=1}^{n} K_i = K'$ and $l' \in K_j$ for some $1 \leq k \leq n$. This implies

$$N \Rightarrow \xrightarrow{c!\tilde{v}@K_1} ... \Rightarrow \xrightarrow{c_L!\tilde{v}[l_j,r_j]} \Rightarrow ... \xrightarrow{c!\tilde{v}@K_n} \Rightarrow N'$$

with $l_j \in L \wedge d(l_j, k) \leq r_j \; \forall k \in K_j$. Hence we can apply rule (Bcast):

$$N|O \Rightarrow \xrightarrow{c!\tilde{v}@K_1} ... \Rightarrow \xrightarrow{c_L!\tilde{v}[l_j,r_j]} \Rightarrow ... \xrightarrow{c!\tilde{v}@K_n} \Rightarrow N'|O'$$

Finally, by applying rule (Obs) we can turn transition $\xrightarrow{c_L!\tilde{v}[l_j,r_j]}$ into $\xrightarrow{c!\tilde{v}@K_j}$. This implies $N|O \overset{c!\tilde{v}@K}{\Longrightarrow} N'|O'$, with $(M'|O', N'|O') \in \mathcal{S}$, as required.

- If $l' \notin L$, as $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$, by applying rule (Par) we have $M|O \xrightarrow{c_L!\tilde{v}[l,r]} M'|O$, and, as $O \xrightarrow{c?\tilde{v}@l'} O'$, by applying again rule (Par) we obtain $M'|O \xrightarrow{c?\tilde{v}@l'} M'|O'$. As $M \approx N$ we can that $N \overset{c!\tilde{v}@K}{\Longrightarrow} N'$, and we can apply again rule (Par) obtaing $N|O \overset{c!\tilde{v}@K}{\Longrightarrow} N'|O$, and then $N'|O \xrightarrow{c?\tilde{v}@l'} N'|O'$. As $l' \notin K$ we deduce finally $N|O \overset{c!\tilde{v}@K}{\Longrightarrow} N'|O'$ as required.
  Case 2 $M|O \xrightarrow{c_L!\tilde{v}[l,r]} \hat{M}$ because $M \xrightarrow{c?\tilde{v}@l'} M'$ and $O \xrightarrow{c_L!\tilde{v}[l,r]} O'$, with $d(l, l') \leq r$ and $\hat{M} = M'|O'$. As $M \approx N$ then there exists N' such that:
  - $N \overset{c?\tilde{v}@l'}{\Longrightarrow} N'$, with $M' \approx N'$; in this case
    $$N|O \Rightarrow \xrightarrow{c_L!\tilde{v}[l,r]} \Rightarrow N'|O'$$
    and, by an application of rule (Obs), also $N|O \overset{c!\tilde{v}@K}{\Longrightarrow} N'|O'$, with $(M'|O', N'|O') \in \mathcal{S}$, as required.
  - or $N \Longrightarrow N'$, with $M' \approx N'$; in this case by applying rule (Par) we obtain
    $$N|O \Rightarrow \xrightarrow{c_L!\tilde{v}[l,r]} \Rightarrow N'|O'$$
    and, by applying rule (Obs) also $N|O \overset{c!\tilde{v}@K}{\Longrightarrow} N'|O'$, with $(M'|O', N'|O') \in \mathcal{S}$, as required.

Cases where there is no interaction between M and O are easy to deal with. In order to prove the second item of the lemma it suffices to show that the relation

$$\mathcal{S} \overset{\text{def}}{=} \{((\nu c)M, (\nu c)N) : M \approx N, \forall c\}$$

is a bisimulation. We do a case analysis on the transition $(\nu c)M \xrightarrow{\alpha} O$. The proof is straightforward as channels cannot be transmitted, hence there is no *scope extrusion*.

We can now demonstrate that our bisimulation is a valid proof method for *reduction barbed congruence*.

Theorem 2 (Soundness). *Let M and N be two arbitrary networks, such that $M \approx N$. Then $M \cong N$*

Proof. By *Harmony theorem* we prove that bisimulation is reduction closed (items 2 and 3) and barb preserving (item 1), and by Lemma 3 we prove that bisimulation is contextual, thus $\approx \subseteq \cong$.

By Soundness theorem let us prove that bisimulation is a complete characterization of *Reduction Barbed Congruence*, we are going now to prove that *Reduction Barbed Congruence* is a complete characterization of bisimulation.

Proposition 1. *If $M \cong N$ then*

- *$M \Downarrow_c$ iff $N \Downarrow_c$*
- *$M \Longrightarrow M'$ implies that there is N' such that $N \Longrightarrow N'$ and $M' \cong N'$*

Lemma 4 (Completeness). *Let M and N be two arbitrary networks, such that $M \cong N$. Then $M \approx N$*

Proof. We prove that the relation $\mathcal{R} = \{(M, N) \mid M \cong N\}$ is a bisimulation. The result will follow by co-induction.

- Suppose that $M \mathcal{R} N$ and $M \xrightarrow{\tau} M'$. By applying harmony theorem $M \rightarrow M'$. Then there is N' such that $N \rightarrow^* N'$, hence $N \Longrightarrow N'$.
- Suppose $M \mathcal{R} N$ and $M \xrightarrow{c!\tilde{v}@K} M'$, with $K = \{k_1, ..., k_n\}$. As the action $c!\tilde{v}@K$ can only be generated by an application of rule (Obs), it follows that $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$ for some l, r such that $k \in L \wedge d(l,k) \leq r \forall k \in K$. Let us build a context which mimics the effect of the action $c!v@K$ and also allows us to subsequently compare the residuals of the two systems under consideration. Our context has the form $\mathcal{C}[\cdot] \stackrel{\text{def}}{=} [\cdot] \mid \prod_{i=1}^{n}(m_i[c(\tilde{x}).[\tilde{x} = \tilde{v}]\bar{\mathtt{f}}_{\infty}^{(i)}\langle\tilde{x}\rangle]_{k_i,r_i}^{\mathtt{s}} \mid n_i[\mathtt{f}^{(i)}(\tilde{x}).\bar{\mathtt{ok}}_{\infty}^{(i)}\langle\tilde{x}\rangle]_{k_i,r_i}^{\mathtt{s}})$ with names m_i, n_i for $1 \leq i \leq n$ and channels names $\mathtt{f}^{(i)}$, $\mathtt{ok}^{(i)}$ for $1 \leq i \leq n$ fresh. Intuitively, the existence of the barbs on the fresh channels $\mathtt{f}^{(i)}$ indicates that the action has not yet happened, whereas the presence of the barbs on channels $\mathtt{ok}^{(i)}$, together with the absence of the barbs on channels $\mathtt{f}^{(i)}$ ensures that the action has been performed.
  As $\cong$ is preserved by network contexts, $M \cong N$ implies $\mathcal{C}[M] \cong \mathcal{C}[N]$. As $M \xrightarrow{c_L!\tilde{v}[l,r]} M'$ it follows that
  $\mathcal{C}[M] \implies M' | \prod_{i=1}^{n}(m_i[\mathbf{0}]_{k_i,r_i}^{\mathtt{s}} | n_i[\bar{\mathtt{ok}}_{\infty}^{(i)}\langle\tilde{x}\rangle]_{k_i,r_i}^{\mathtt{s}}) = \hat{M}$, with $\hat{M} \not\Downarrow_{\mathtt{f}^{(i)}}$ and $\hat{M} \Downarrow_{\mathtt{ok}^{(i)}}$, for $1 \leq i \leq n$.
  The reduction sequence must be matched by a corresponding reduction sequence $\mathcal{C}[N] \Longrightarrow \hat{N}$ with $\hat{M} \cong \hat{N}$, $\hat{N} \not\Downarrow_{\mathtt{f}^{(i)}}$ and $\hat{N} \Downarrow_{\mathtt{ok}^{(i)}}$ for $1 \leq i \leq n$. The contrains on the barbs allow us to deduce the structure of the above reduction sequence

$\mathcal{C}[N] \Longrightarrow N' | \prod_{i=1}^{n} (m_i[\mathbf{0}]^{\mathtt{s}}_{k_i,r_i} | n_i[\bar{\mathtt{ok}}^{(i)}_{\infty}\langle\tilde{x}\rangle]^{\mathtt{s}}_{k_i,r_i}) = \hat{N}$.

This implies $N \stackrel{c!\tilde{v}@K'}{\Longrightarrow} N'$ with $K \subseteq K'$. More precisely, the derivative N' might be reached performing several outputs of the message $\tilde{v}$ along the same channel c. We can prove that K' contains K because we are sure that all nodes m_i hare reached by a transmission along channel c from N. It's easy now to show that $N \stackrel{c!\tilde{v}@K}{\Longrightarrow} N'$, by considering in the composition of the action only on those outputs addressed to the locations in K, and turning the other in τ-actions, using rule (Lose).

As $\hat{M} \cong \hat{N}$, and as Reduction Barbed Congruence is preserved by restriction, we have $(\nu \mathtt{f}, \mathtt{ok})\hat{M} \cong (\nu \mathtt{f}, \mathtt{ok})\hat{N}$. As $\mathtt{f}^{(i)}$ and $\mathtt{ok}^{(i)}$ for all $1 \leq i \leq n$ fresh, by applying structural congruence we have

$(\nu \mathtt{f}, \mathtt{ok})\hat{M} \equiv M' | (\nu \mathtt{f}, \mathtt{ok})(m_i[\mathbf{0}]^{\mathtt{s}}_{k_i,r_i} | n_i[\bar{\mathtt{ok}}^{(i)}_{\infty}\langle\tilde{x}\rangle]^{\mathtt{s}}_{k_i,r_i})$

$(\nu \mathtt{f}, \mathtt{ok})\hat{N} \equiv N' | (\nu \mathtt{f}, \mathtt{ok})(m_i[\mathbf{0}]^{\mathtt{s}}_{k_i,r_i} | n_i[\bar{\mathtt{ok}}^{(i)}_{\infty}\langle\tilde{x}\rangle]^{\mathtt{s}}_{k_i,r_i})$.

Using bisimilarity definition and soundness theorem we can easily prove that $(\nu \mathtt{f}, \mathtt{ok})(m_i[\mathbf{0}]^{\mathtt{s}}_{k_i,r_i} | n_i[\bar{\mathtt{ok}}^{(i)}_{\infty}\langle\tilde{x}\rangle]^{\mathtt{s}}_{k_i,r_i}) \cong \mathbf{0}$.

A a consequence, it follows that $M' \cong N'$, as required.

– Suppose that $M \mathcal{R} N$ and $M \xrightarrow{c?\tilde{v}@l} M'$.

The reception of a message cannot be directly observed. So we have to build a context which let the action be observable.

A context associated to the action $M \xrightarrow{c?\tilde{v}@l} M'$ could be:

$\mathcal{C}[\cdot] \stackrel{\text{def}}{=} [\cdot] | n[\bar{c}_l\langle\tilde{v}\rangle.\bar{\mathtt{f}}_{\infty}\langle\tilde{v}\rangle.\bar{\mathtt{ok}}_{\infty}\langle\tilde{v}\rangle]^{\mathtt{s}}_{k,r}$

with $\mathtt{f}$ and $\mathtt{ok}$ fresh channels, and $d(l,k) \leq r$. As $\cong$ is preserved by network contexts, $\mathcal{C}[M] \cong \mathcal{C}[N]$. As $M \xrightarrow{c?\tilde{v}@l} M'$ it follows that

$\mathcal{C}[M] \Longrightarrow M' | n[\bar{\mathtt{ok}}_{\infty}\langle\tilde{v}\rangle]^{\mathtt{s}}_{k,r} = \hat{M}$

with $\hat{M} \not\Downarrow_{\mathtt{f}}$ and $\hat{M} \Downarrow_{\mathtt{ok}}$. The reduction sequence must be matched by a corresponding reduction sequence $\mathcal{C}[N]$, so we have $\mathcal{C}[N] \Longrightarrow \hat{N}$ and $\hat{M} \cong \hat{N}$, with $N \not\Downarrow_{\mathtt{f}}$ and $N \Downarrow_{\mathtt{ok}}$. The contrains on the barb ensure that the action $c?\tilde{v}@l$ has been performed, so there exists N' such that $N \stackrel{c?\tilde{v}@l}{\Longrightarrow} N'$, or $N \Longrightarrow N'$, in case rule (Lose) has been applied to the node n. As $\hat{M} \cong \hat{N}$, and $\cong$ is preserved by restriction, it follows that $(\nu \mathtt{ok})\hat{M} \cong (\nu \mathtt{ok})\hat{N}$, from which we can easily derive

$(\nu \mathtt{ok})\hat{M} \equiv M' | (\nu \mathtt{ok})(n[\bar{\mathtt{ok}}_{\infty}\langle\tilde{v}\rangle]^{\mathtt{s}}_{k,r})$

$(\nu \mathtt{ok})\hat{N} \equiv N' | (\nu \mathtt{ok})(n[\bar{\mathtt{ok}}_{\infty}\langle\tilde{v}\rangle]^{\mathtt{s}}_{k,r})$

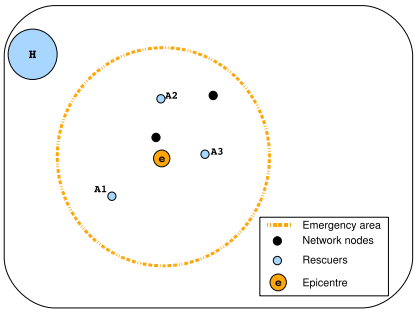
As $(\nu \mathtt{ok})(n[\bar{\mathtt{ok}}_{\infty}\langle\tilde{v}\rangle]^{\mathtt{s}}_{k,r}) \equiv \mathbf{0}$ we obtain $M' \cong N'$, as required.

We have proved that $\cong \subseteq \approx$.

5 Properties

In this section we use CMN$^{\#}$ to define and prove some properties of mobile ad hoc networks. First we review the properties for MANETs described in [11] and then we study some new properties which cannot be dealt with in the original CMN model. We use some examples to better describe the properties

Fig. 2: Installation of a mobile Ad-Hoc network after an earthquake



analyzed. We use a running example depicted in Figure 2, describing the case of an emergency due to an earthquake. The hospital sends tree ambulances to the emergency area (in figure A1, A2, A3). Then an ad hoc network is installed to manage the communication between the ambulances, placing a router near the epicenter of the earthquake. (in figure the orange circle).

We have to make an assumption before starting to deal with the properties of CMN$^{\#}$, about the possible receivers of the whole set of transmissions of a network. Given a process P, we denote by $\texttt{rcv}(P)$ the minimum set of locations ensuring that for each output action $\bar{c}_L\langle\tilde{v}\rangle$ performed by P it holds that $L \subseteq \texttt{rcv}(P)$. Indeed, the tag L associated to an output action occurring in P can be either a variable or a set of locations, then we are not able to statically calculate $\texttt{rcv}(P)$. However. since an ad hoc network is usually designed to guarantee the communications within a specific area, we can reasonably assume that the underling protocol will always multicast messages to recipients located within the interested area and we can abstractly represent them by a finite set of locations.

5.1 Ubiquity of nodes

We now provide to demonstrate that the position of a node in the network has no effect on its behaviour. This characteristic allows us to consider movements, connections and disconnections of the nodes as weak actions, and give us insights on how to analyze the problem of *Location Confidentiality*, because a node can hide information about its location without modifying the process it is executing. This property is important especially in those cases where information confidentiality is important to protect: the best example is in case of war. In our example the ubiquity of nodes ensures that communication between ambulances is not compromised by arbitrary movements of the nodes within the transmission cell of the router.

Theorem 3 (Ubiquity of nodes). *Let P be a process, $\lambda_1, \lambda_2 \neq nil(l)$ locations of mobile nodes, k a location and r a transmission radius. Then:*

1. $n[P]^{\mathfrak{m}}_{\lambda_1,r} \approx n[P]^{\mathfrak{m}}_{\lambda_2,r}$

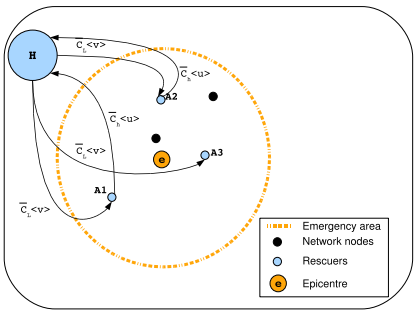
2. $n[P]^{\mathtt{s}}_{k,r} \approx n[P]^{\mathtt{s}}_{nil(k),r}$

Proof.

1. We show that
   $\mathcal{S} = \{(n[P]^{\mathtt{m}}_{\lambda_1,r}, n[P]^{\mathtt{m}}_{\lambda_2,r}) : \forall P, \lambda_1, \lambda_2 \neq nil(k), r\} \bigcup \mathcal{I}$
   is a bisimulation (where $\mathcal{I}$ is the identity relation). It is easy to prove because, if for some α we have
   $$n[P]^{\mathtt{m}}_{\lambda_1,r} \xrightarrow{\alpha} M,$$
   then, by applying rule (Move) if $\lambda_1, \lambda_2 \neq nil$, (m-Disc) or (m-Conn) otherwise, we obtain
   $$n[P]^{\mathtt{m}}_{\lambda_2,r} \xRightarrow{\hat{\alpha}} M.$$
2. We prove that
   $\mathcal{S} = \{(n[P]^{\mathtt{s}}_{k,r}, n[P]^{\mathtt{s}}_{nil(k),r}) : \forall P, k, r\} \bigcup \mathcal{I}$
   is a bisimulation (where $\mathcal{I}$ is the identity relation). Suppose, for any α
   $$n[P]^{\mathtt{s}}_{k,r} \xrightarrow{\alpha} M,$$
   then, by applying (s-Conn) we obtain
   $$n[P]^{\mathtt{s}}_{nil(k),r} \xRightarrow{\hat{\alpha}} M.$$
   In the second case $n[P]^{\mathtt{s}}_{nil(k),r}$ can only log on, so
   $$n[P]^{\mathtt{s}}_{nil(k),r} \xrightarrow{\tau} n[P]^{\mathtt{s}}_{k,r}.$$

5.2 Silent nodes cannot be observed

Fig. 3: Messages exchange between H and the ambulances



What is interesting about the ad hoc networks are the interactions between nodes. If a node sends no messages, it does not interact with the network, then

an external observer cannot know if this node is connected. Consider now the interactions between the hospital and the ambulances (Figure 3). Suppose that A1 and A2 are facing critical situations, and communicate with the hospital to prepare the acceptance of patients in hospital, while A3 have no patients to be accepted so no communications are sent from A3 to the hospital. The hospital broadcasts emergency messages to update the network about the general situation. An observer listening the communication between the nodes cannot know if A3 is connected, because it does not receive anything from that node.

Theorem 4 (Silent nodes are not observable). *If the process P does not contain output constructs, then*
$n[P]^{\mu}_{\lambda,r} \approx \mathbf{0}\ \forall \mu, \lambda, r.$

Proof. It follows from the definition of bisimilarity in which it is possible to match both τ-actions and input actions with weak τ-actions.

5.3 Obfuscating message transmission

We are going to propose a way to obfuscate messages transmission. This property is very important to reason about security problems and information and location confidentiality. We first have to prove that, alternating infinite output sequences, the order of the transmissions has got no effect to the behaviour of the node. We prove the theorem for a set of two messages (u and v), but the result can be generalized to an arbitrary set $V = \{v_1, ..., v_n\}$ of messages.

Theorem 5 (Mixing up infinite output sequences). *Let*
$ALT(a, L_1, b, L_2) \stackrel{\text{def}}{=} \bar{c}_{L_1}\langle a\rangle.\bar{c}_{L_2}\langle b\rangle.ALT\langle a, L_1, b, L_2\rangle$*, then, for any* n, r, u, v *it holds that*

1. $n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_l,r} \approx n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{s}}_{\sigma'_l,r}$*, where* $\sigma_l, \sigma'_l = l \vee nil(l)$*, with* l *location.*
2. $n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_1,r} \approx n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_2,r}$*, where* λ_1, λ_2 *are locations or nil.*

Proof.
1. We prove that

$$\mathcal{S} \stackrel{\text{def}}{=} \{(n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_l,r}, n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{s}}_{\sigma_l,r}) : \forall r, u, v, \sigma_l, \sigma'_l\} \bigcup \mathcal{I}$$

   is a bisimulation with respect to $\equiv$.
   Since a node can connect or disconnect with weak actions, we directly consider connected devices, so $\sigma_l, \sigma'_l = l$. Suppose

$$n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{l,r} \xrightarrow{c!u@K} \equiv n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{s}}_{l,r}$$

   for some set K of locations, then, using (Lose) to discard v and (Obs) to transmit u, we obtain

$$n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{s}}_{l,r} \stackrel{c!u@K}{\Longrightarrow} \equiv n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{s}}_{l,r}$$

With the same procedure we can show that, if

$$n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{s}}_{l,r} \xrightarrow{c!v@K} \equiv n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{l,r},$$

then:

$$n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{l,r} \stackrel{c!v@K}{\Longrightarrow} \equiv n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{l,r}.$$

2. We prove that

$$\mathcal{S} \stackrel{\text{def}}{=} \{(n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_1,r}, n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_2,r}) : \forall r, u, v, \lambda_1, \lambda_2\} \bigcup \mathcal{I}$$

is a bisimulation up to $\equiv$.
Suppose that

$$n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_1,r} \xrightarrow{c!u@K} \equiv n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_1,r}$$

for some set K of locations, then

$$n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_2,r} \xrightarrow{\tau} n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_1,r}$$

by applying rule (Move), (m-Disc) or (m-Conn), in accordance with the value of λ_1 and λ_2; by applying (Lose) to discard v and (Obs) to transmit u, we deduce

$$n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_2,r} \stackrel{c!u@K}{\Longrightarrow} \equiv n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_1,r}.$$

With the same procedure we can prove that, if

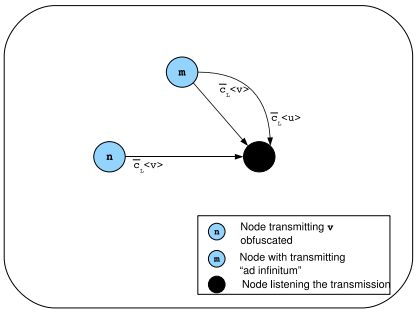
$$n[ALT\langle v, L_2, u, L_1\rangle]^{\mathtt{m}}_{\lambda_2,r} \xrightarrow{c!v@K} \equiv n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_2,r},$$

then:

$$n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_1,r} \stackrel{c!v@K}{\Longrightarrow} \equiv n[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_2,r}.$$

Now we have got all the notions to introduce a method to obfuscate messages transmission. A node `m` transmitting "ad infinitum" a message sequence, may obfuscate the transmission activity of nodes which are transmitting the same messages within the same transmission cell. this happens because nodes that receive a message cannot distinguish the sender, but only the channel of transmission. At a first analysis this property seems to be too much restrictive to be used in concrete situations, but there are many situations where packages transmitted in a network are standard. Consider the example of the network installed for the earthquake area: the ambulances and the hospital can communicate with standard messages. Consider for example the message `v`: "`patient with cardiac arrest`", suppose that `A1` transmits `v` "ad infinitum" to be sure that the hospital receives the message correctly. Then also `A2` transmits the same package to the server, using the same channel: it could be result

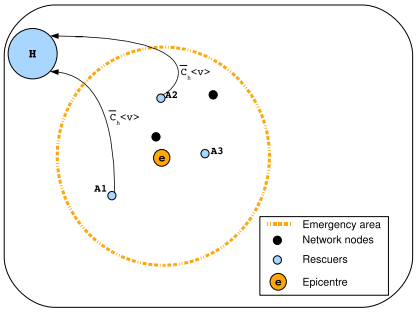
Fig. 4: Obfuscation of the transmission of v from the node n



that the hospital prepares the acceptance for only one patient, but the patients with cardiac arrest are two. We choose this example to show how a deep analysis of these problems in transmission management are particularly important in mobile ad hoc networks, because this kind of network is used in critical situations.

In our example message obfuscation results to be an obstacle for the good management of networks, a problem rather then a property. In other cases we can prove that the possibility of a node to hide its transmission could be a positive future (as in case of war). A formal model as CMN$^{\#}$, has the purpose of analyzing deeply the characteristics of the networks, with both their positive and negative consequences for the communication management.

Fig. 5: Example of obfuscation in transmission of v by A1



Theorem 6 (Obfuscating message transmission). *Let P and Q be two processes such that $fc(P,Q) \subseteq \{c\}$ for some channel c, where all the output actions of P and Q are in the set $\{\bar{c}_{L_1}\langle u\rangle, \bar{c}_{L_2}\langle v\rangle\}$. Then:*

1. $n[P]^{\mathtt{s}}_{\sigma_k,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r} \approx n[Q]^{\mathtt{s}}_{\sigma_k,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r}$, *where* $\sigma_k = k \vee nil(k)$

2. $n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_2,r} \approx n[Q]^{\mathtt{m}}_{\lambda_3,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{m}}_{\lambda_4,r}$, *where* $\lambda_i \neq nil(k) \forall 1 \leq i \leq 4$ *for any location* k.

Proof.

1. As $\approx$ is transitive, We have only to show

$$n[P]^{\mathtt{s}}_{\sigma_k,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r} \approx m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r}$$

because we will use the same procedure to prove that

$$n[Q]^{\mathtt{s}}_{\sigma_k,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r} \approx m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r}$$

then we can deduce

$$n[P]^{\mathtt{s}}_{\sigma_k,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r} \approx m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r} \approx n[Q]^{\mathtt{s}}_{\sigma_k,r}|m[ALT\langle u, L_1, v, L_2\rangle]^{\mathtt{s}}_{\sigma_k,r}.$$

Since the nodes can disconnect an connect arbitrarily in the network simply executing weak actions, we can directly consider the case in which n and m are connected, so $\sigma_k = k$. We then prove the relation

$$\begin{aligned}\mathcal{S} = \{(n[P]^{\mathtt{s}}_{k,r}|m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r}, m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r}) : \\ \forall u, v, k, r, \\ \forall P. fc(P) \subseteq \{c\} \text{ where all output actions of } P \text{ are} \\ \{\bar{c}_{L_1}\langle u\rangle, \bar{c}_{L_2}\langle v\rangle\}\} \bigcup \mathcal{I}\end{aligned}$$

is a bisimulation up to$\equiv$. Suppose then

$$n[P]^{\mathtt{s}}_{k,r}|m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r} \xrightarrow{c!u@K} \equiv M$$

for some set K of nodes. if m is the sender of u we have

$$M \equiv n[P]^{\mathtt{s}}_{k,r}|m[ALT\langle v, u\rangle]^{\mathtt{s}}_{k,r}.$$

If n is the sender of message we have

$$M \equiv n[P']^{\mathtt{s}}_{k,r}|m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r},$$

where $n[P]^{\mathtt{s}}_{k,r} \xrightarrow{c!u@K} n[P']^{\mathtt{s}}_{k,r}$. In the first case it simply suffices an application of (Obs) also to $m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r}$, obtaining $m[ALT\langle v, u\rangle]^{\mathtt{s}}_{k,r}$, whereas in the second case, after an application of (Obs) we use (Lose) to discard v obtaining $m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r}$ again. it is easy now to prove that

$$(n[P]^{\mathtt{s}}_{k,r}|m[ALT\langle v, u\rangle]^{\mathtt{s}}_{k,r}, m[ALT\langle v, u\rangle]^{\mathtt{s}}_{k,r})$$

and

$$(n[P']^{\mathtt{s}}_{k,r}|m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r}, m[ALT\langle u, v\rangle]^{\mathtt{s}}_{k,r})$$

are members of $\mathcal{S}$, because the same properties of the initial couple are still valid ($fc(P') \subseteq \{c\}$ and all output actions of P' are $\{\bar{c}_{L_1}\langle u\rangle, \bar{c}_{L_2}\langle v\rangle\}$), so it suffices the application of the same procedure.

2. As $\approx$ is transitive, it suffices proving

$$n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r} \approx m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda,r},$$

because we then use the same procedure to demonstrate

$$n[Q]^{\mathtt{m}}_{\lambda_3,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_4,r} \approx m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda,r},$$

we then deduce

$$n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r} \approx m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda,r} \approx n[Q]^{\mathtt{m}}_{\lambda_3,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_4,r}.$$

We now prove that the relation

$$\begin{aligned}\mathcal{S} = \{(n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r}, m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda,r}) :\\ \forall u,v,\lambda,\lambda_1,\lambda_2,r,\\ \forall P. fc(P) \subseteq \{c\} \text{ where all output actions of P are}\\ \{\bar{c}_{L_1}\langle u\rangle, \bar{c}_{L_2}\langle v\rangle\}\} \bigcup \mathcal{I}\end{aligned}$$

is a bisimulation up to $\equiv$. Suppose

$$n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r} \xrightarrow{c!u@K} \equiv M.$$

if m is the sender of u, we have

$$M \equiv n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle v,u\rangle]^{\mathtt{m}}_{\lambda_2,r},$$

if n is the sender of the message we have

$$M \equiv n[P']^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r},$$

where $n[P]^{\mathtt{m}}_{\lambda_1,r} \xrightarrow{c!u@K} n[P']^{\mathtt{m}}_{\lambda_1,r}$. In both cases we use the rule (Move) ((m-Disc) if $\lambda_2 = nil$, (m-Conn) if $\lambda = nil$) and we obtain $m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda,r} \xrightarrow{\tau} m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r}$. Then, in the first case it simply suffices an application of rule (Obs) also to $m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r}$ obtaining $m[ALT\langle v,u\rangle]^{\mathtt{m}}_{\lambda_2,r}$, in the second case, after an application of rule (Obs) we use (Lose) to discard v obtaining $m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r}$ again. it is is easy now to prove that

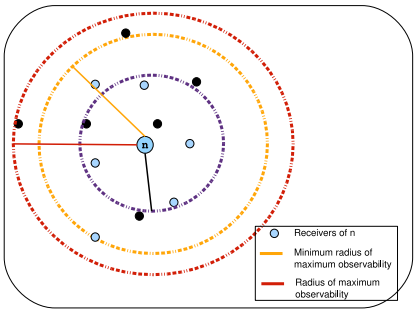
$$(n[P]^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle v,u\rangle]^{\mathtt{m}}_{\lambda_2,r}, m[ALT\langle v,u\rangle]^{\mathtt{m}}_{\lambda_2,r})$$

and

$$(n[P']^{\mathtt{m}}_{\lambda_1,r}|m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r}, m[ALT\langle u,v\rangle]^{\mathtt{m}}_{\lambda_2,r})$$

are in $\mathcal{S}$ because the same properties of the initial couple are still valid ($fc(P) \subseteq \{c\}$ where all output actions of P are $\{\bar{c}_{L_1}\langle u\rangle, \bar{c}_{L_2}\langle v\rangle\}$), so it suffices applying the same procedure.

Fig. 6: Radii of maximum observability



5.4 Radius of maximum observability

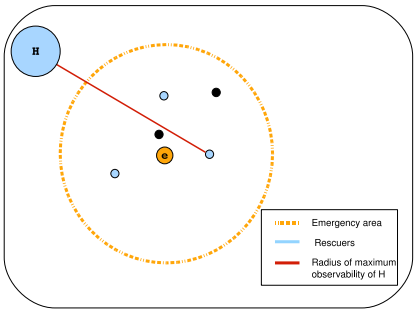
All the properties described above can be proved also in the original CMN [11]; now we introduce new properties which cannot be dealt with in CMN.

The first important characteristic we are going to describe is about the transmission radius of the nodes. Each transmission is associated to a set of locations including the receivers of the message. Then we can define a "*radius of maximum observability*", that is a radius such to allow a correct reception of a message to all the locations of the receivers set. In particular we can define the "*minimum radius of maximum observability*", which corresponds to the distance between the sender of the message and the most distant receiver. Figure 6 illustrates three different cases of a transmission effect in a network: the black radius (the smallest) does not allow the reception of the message by the set of the receivers (light nodes) while the other two radii reach at least one of the receivers. The orange radius is in particular the distance between the transmitter and the most distant receiver: This characteristic means that this is the *minimum radius of maximum observability*.

To understand how this property is important let consider the example depicted in Figure 7. The earthquake has damaged a defined area; we can than deduce that rescuers need to communicate only within the emergency area (in figure the area delimited by the orange circle). We can then determine the minimum transmission radius which ensures the central server of the hospital to be able to communicate with the ambulances sent for assistance in the disaster area. This property is important only for stationary nodes, whereas a mobile node can move within the transmission cell of the transmitter to receive the communication. In our example the central server needs to communicate with the ambulances, which are all mobile nodes, then instead of determining the minimum radius of maximum observability, which change arbitrarily, it is better considering a radius enough large to ensure the correct result of the transmission within all the emergency area.

Although we can define a minimum radius to reach all the intended receivers of a transmission, we cannot be sure of the correct reception of the message

Fig. 7: determination of the minimum radius of maximum observability



by the whole set of intended receivers; There are indeed various obstacles (as connection failures, ore physical barriers to transmissions...) which can obstruct the correct outcome of a transmission in the network; This is the reason why the output actions are always observable, while the input actions are not.

Theorem 7 (Radius of maximum observability). *Let P be a process such that $\mathtt{rcv}(P) = L$, if, for some r, $\sigma_l = l \vee \sigma_l = nil(l), d(l,k) \leqslant r \ \forall k \in L$, then:*

$$n[P]^{\mathtt{s}}_{\sigma'_l,r'} \approx n[P]^{\mathtt{s}}_{\sigma_l,r} \ \forall r' \geq r \ \textit{and} \ \forall \sigma'_l = l \vee \sigma'_l = nil(l).$$

In this case we say that r is a radius of maximum observability for the stationary process P located at σ_l.

Proof. We prove that the relation

$$\mathcal{S} = \{(n[P]^{\mathtt{s}}_{\sigma_l,r}, n[P]^{\mathtt{s}}_{\sigma'_l,r'}) : \\ \forall r' \geq r, \ \forall \sigma_l, \sigma'_l = l \vee nil(l), \\ \forall P.\mathtt{rcv}(P) = L\} \bigcup \mathcal{I}$$

is a bisimulation.
Consider

$$n[P]^{\mathtt{s}}_{\sigma_l,r} \xrightarrow{c!v@L_i} n[P']^{\mathtt{s}}_{\sigma_l,r}, \text{ with } L_i \subseteq L.$$

Then, by applying rule (Obs) backward it must be Then we have $n[P]^{\mathtt{s}}_{\sigma_l,r} \xrightarrow{c_{L_i}!v[l,r]} n[P']^{\mathtt{s}}_{\sigma_l,r}$, as $d(l,l_i) \leq r \forall l_i \in L_i \subseteq L$, and by applying rule (Snd) backward we finally obtain $P \xrightarrow{\bar{c}_{L_i} v} P'$. So we can write

$$n[P]^{\mathtt{s}}_{\sigma'_l,r'} \overset{c!v@K}{\Longrightarrow} n[P']^{\mathtt{s}}_{\sigma_l,r'}$$

where $K = \{k : k \in L_i \wedge d(l,k) \leq r'\}$. Then $L_i \subseteq K$ because, if $d(l,k) \leq r$ necessarily $d(l,k) \leq r'$, recalling that $r \leq r'$; We can also write $K \subseteq L_i$ by definition of K, then $K = L$, and we deduce

$$n[P]^{\mathtt{s}}_{\sigma_l,r'} \stackrel{c!v@L_i}{\Longrightarrow} n[P']^{\mathtt{s}}_{\sigma_l,r'}$$

Now it is sufficient to prove that

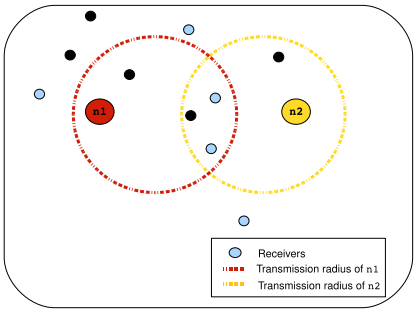
$$n[P']^{\mathtt{s}}_{\sigma_l,r} \mathcal{S} n[P']^{\mathtt{s}}_{\sigma_l,r'}$$

We know that for P' the same properties of P are valid ($\mathtt{rcv}(P') \subseteq L$ for definition of the function $\mathtt{rcv}(\cdot)$), we can then apply to P' the same procedure used for P.

Definition 7 (Minimum Radius of Maximum observability). *Let consider a process P such that $\mathtt{rcv}(P) = L$, we say that r is the minimum radius of maximum observability for the stationary node P located at σ_l if r is a radius of maximum observability, and if for all $r' < r$ it holds that r' is not a radius of maximum observability for the stationary node P located at σ_l.*

5.5 Simulation of stationary nodes in different locations

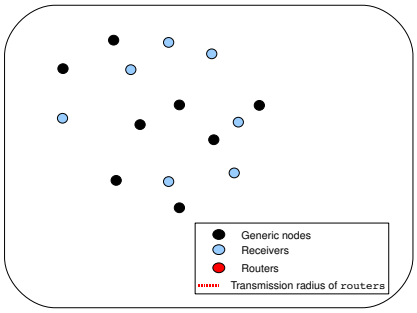
Fig. 8: Example of simulation of stationary nodes in different locations



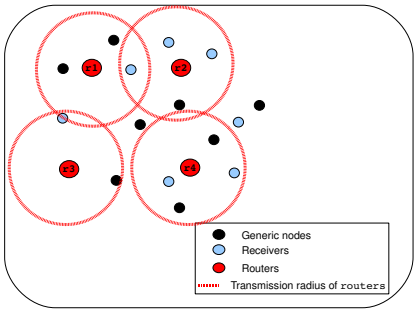
The tag L associated to each output action has been introduced, not only in order to find the minimum radius ensuring each transmission to reach all its receivers, but its introduction allows us to define other important properties for our calculus. If we pay attention to the behaviour of stationary nodes, we realize that the tag L enables, under particular prerequisites, the simulation of stationary devices in different locations. In our calculus the barb of a transmission is not determined by a set of locations, but by the intersection of two sets: more precisely the barb of a transmission is given by the set of locations which are members of L, lying in the transmission cell of the sender. By these assumptions, we can now consider the case where two stationary nodes, placed in different locations (with therefore different neighbours), but communicating with the same set of locations, result to have the same barb. A practical example of the usefulness of this property is the case we have to use the lowest number of routers within an area in order to ensure the communication with a

Fig. 9: Example of optimizing routers allocation

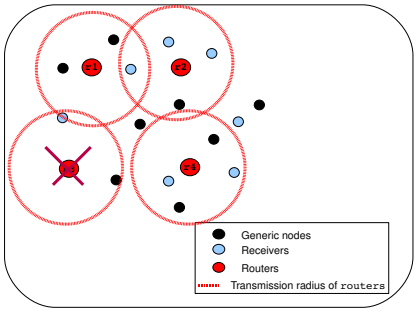
(a) initial network topology



(b) First routers allocation



(c) Optimal routers allocation



given set of locations. If we realise that two different routers result to have the same behaviour, one of them can be then turn off, allowing us to save power and physical resources. Looking at Figure 9 we can see an example of optimizing the allocation of the routers, by turning off the router `r3`, which is not usefull, because it can be simulated by `r1`.

Theorem 8 (Simulation of stationary nodes in different locations). *Let P be a process with $\texttt{rcv}(P) = L$. Suppose:*
$n[P]^{\mathtt{s}}_{\sigma_l,r}$ with $\sigma_l = l \vee nil(l)$ and $K = \{k : k \in L \wedge d(l,k) \leq r\}$
$m[P]^{\mathtt{s}}_{\sigma_{l'},r'}$ with $\sigma_{l'} = l' \vee nil(l')$ and $K' = \{k' : k' \in L \wedge d(l',k') \leq r'\}$. It holds that:

1. *If $K' \subseteq K$, then $n[P]^{\mathtt{s}}_{\sigma_l,r}$ simulates $m[P]^{\mathtt{s}}_{\sigma_{l'},r'}$;*
2. *If $K = K'$, then $n[P]^{\mathtt{s}}_{\sigma_l,r} \approx m[P]^{\mathtt{s}}_{\sigma_{l'},r'}$.*

Proof.

1. the interesting case to analyse is the output action. Consider $P = \bar{c}_{L_i}\langle v\rangle.P'$ for some $L_i \subseteq L$:

   $$m[P]^{\mathtt{s}}_{\sigma_{l'},r'} \xrightarrow{c!v@K''} m[P']^{\mathtt{s}}_{\sigma_{l'},r'}, \text{ where } K'' = K' \cap L_i$$

   Then we can write

   $$n[P]^{\mathtt{s}}_{\sigma_l,r} \stackrel{c!v@K'''}{\Longrightarrow} n[P']^{\mathtt{s}}_{\sigma_l,r} \text{ where } K''' = K \cap L_i.$$

   But, as $K' \subseteq K$, we deduce $K' \cap L_i = K'' \subseteq K''' = K \cap L_i$ and, by applying rules (Obs) we can also write

   $$n[P]^{\mathtt{s}}_{\sigma_l,r} \stackrel{c!v@K''}{\Longrightarrow} n[P']^{\mathtt{s}}_{\sigma_l,r}.$$

   As $\texttt{rcv}(P') \subseteq L$, by applying the same procedures to P' we can prove that $n[P']^{\mathtt{s}}_{\sigma_l,r}$ simulates $m[P']^{\mathtt{s}}_{\sigma_{l'},r'}$.
2. If $K = K'$ then $K \subseteq K'$ and $K' \subseteq K$; so, by applying the same procedure used to prove the first item of this theorem, we demonstrate that the relation

   $$\begin{aligned}\mathcal{S} = \{(n[P]^{\mathtt{s}}_{\sigma_l,r}, m[P]^{\mathtt{s}}_{\sigma_{l'},r'}) : \\ \forall \sigma_l = l \vee nil(l), \forall \sigma_{l'} = l' \vee nil(l') \\ \forall P.\texttt{rcv}(P) \subseteq L\} \bigcup \mathcal{I}\end{aligned}$$

   is a bisimulation.

5.6 Range repeaters

We are going to define the *Range Repeaters*, that are devices which regenerate a network signal in order to extend the range of the existing network infrastructure. A definition of repeater was already given in [11]. Here we give a new definition of range repeater for CMN$^{\#}$, and we use both range repeaters with one and two channels to prove new important properties.

Definition 8. *Let a, b be channels, l a location, r a transmission radius and L a set of locations. A repeater with two channels on L is a stationary device, denoted $rr[a \hookrightarrow_L b]^{\mathtt{s}}_{l,r}$, where $a \hookrightarrow_L b$ is a process whose general recursive definition is:*

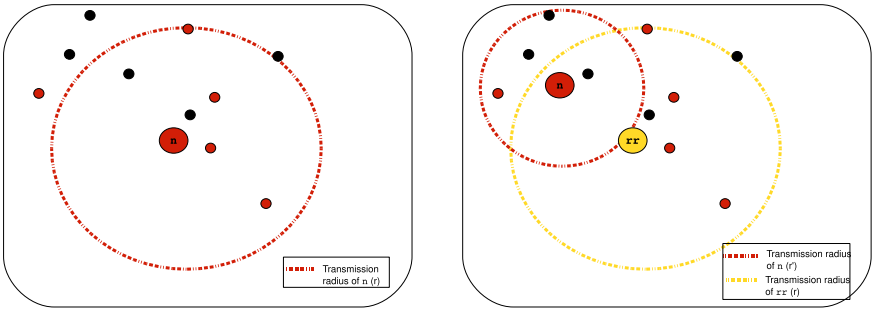
$$a \hookrightarrow_L b \stackrel{\text{def}}{=} a(x).\bar{b}_L\langle x\rangle.a \hookrightarrow_L b.$$

A range repeater with two channels receive values through the input channel and retransmits them through the output channels, to the message receivers. A range repeater with one channel operates analogously, but input and output channels coincide.

Definition 9. *Let c be a channel, l a location, r a transmission radius and L a set of locations. A range repeater with one channel on L is a stationary device, denoted $rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$.*

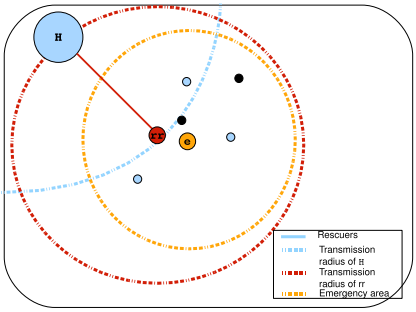
Fig. 10: Range repeater with one channel

(a) $n[P]^{\mathtt{s}}_{k,r}$ (b) $n[P]^{\mathtt{s}}_{k,r'}|rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$



Range repeaters are usually exploited to enlarge transmission cell of a stationary node and, if such a node always communicates with the same set of devices, each time through the same channel, by using a range repeater we can simulate the presence of the sender in the location of the repeater. The employment of range repeaters is particularly interesting for Ad-Hoc Networks, because it allows us to analyze and try to solve the problems of devices mobility. Consider once more the example used in this section. Describing the situation we did not consider distance between the hospital and the earthquake area, which may be too large to guarantee the communication with the ambulances running up in the emergency area. It will be then necessary employing a range repeater enough powerfull to cover all the area and, at the same time, reachable by the central server of the hospital. If the earthquake epicenter is too distance from the hospital we can install a series of consecutive repeaters, which will connect the central server to the disaster area.

Fig. 11: Example of repeaters use



Theorem 9 (Range repeaters with one channel). *Let P be a process such that $fc(P) \subseteq \{c\}$ for some channel c. Let $\texttt{rcv}(P) = L$. Let k, l be physical locations, r, and r' radii such that $d(k,l) \leq r$ and $d(k,l) \leq r'$. Then:*

$$n[P]^{\mathtt{s}}_{k,r'} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \text{ simulates } n[P]^{\mathtt{s}}_{l,r}.$$

Proof. We have to prove that the relation

$$\begin{aligned}\mathcal{S} \stackrel{\text{def}}{=} \{&(n[P]^{\mathtt{s}}_{l,r}, n[P]^{\mathtt{s}}_{k,r'} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}) : \\ &\forall l,k,r,r'.d(k,l) \leq r \ \wedge \ d(l,k) \leq r' \\ &\forall P.fc(P) \subseteq \{c\} \ \wedge \ \texttt{rcv}(P) = L\}\end{aligned}$$

is a simulation.
Suppose:

$$n[P]^{\mathtt{s}}_{l,r} \xrightarrow{c!v@K} n[P']^{\mathtt{s}}_{l,r}$$

for $K = \{k_1, ..., k_n : k_i \in L_i \wedge d(l,k_i) \leq r \ \forall 1 \leq i \leq n\}$, because $P \xrightarrow{\bar{c}_{L_i} v} P'$ for some $L_i \subseteq L$. Knowing that $d(k,l) \leq r'$, by applying rule (Bcast) we obtain:

$$n[P]^{\mathtt{s}}_{k,r'} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \xrightarrow{c_{L_i}!v[k,r']} n[P']^{\mathtt{s}}_{k,r'} | rr[\bar{c}_L\langle v\rangle.c \hookrightarrow_L c]^{\mathtt{s}}_{l,r},$$

and, by applying rule (Obs),

$$n[P']^{\mathtt{s}}_{k,r'} | rr[\bar{c}_L\langle v\rangle.c \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \xrightarrow{c!v@K'} n[P']^{\mathtt{s}}_{k,r} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}, \text{ where}$$
$$K' = \{k'_1, ..., k'_n : k'_i \in L \wedge d(l,k'_i) \leq r \ \forall 1 \leq i \leq n\}$$

that means:

$$n[P]^{\mathtt{s}}_{k,r'} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \stackrel{c!v@K'}{\Longrightarrow} n[P']^{\mathtt{s}}_{k,r} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$$

But, as $L_i \subseteq L$ then $K \subseteq K'$, by using rule (Lose) to hide the communications with nodes at locations in K' which are not in K we can finally write:

$$n[P]^{\mathtt{s}}_{k,r'} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \stackrel{c!v@K}{\Longrightarrow} n[P']^{\mathtt{s}}_{k,r} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$$

We have now only to prove that

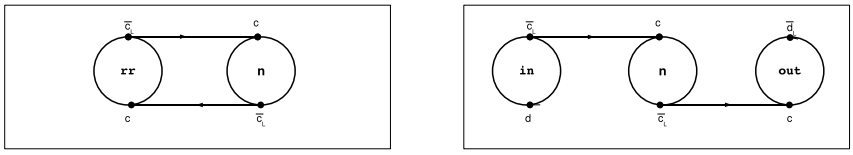
$$(n[P']^{\mathtt{s}}_{l,r}, n[P']^{\mathtt{s}}_{k,r'} | rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}) \in \mathcal{S}$$

Since for P' the same properties of P are still valid ($fc(P') \subseteq \{c\} \wedge \mathtt{rcv}(P') \subseteq L$) we can apply the same procedure and prove in this way the simulation.

We are going now to prove that the simulation just described can be realized also with a range repeater with two channels. Using two channels however we need two range repeaters, respectively for input ($\mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$) and output ($\mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r}$) management. The diagram in figure 12 illustrates the employment of the channels and the interaction between the nodes in the realization of range repeaters with one or two channels. This figure is inspired to the diagrams of Milner [13] in his introduction to CCS, describing the behaviour of agents and the use of channels for data input and output. We emphasise that this kind of diagrams gives no information about physical position of nodes or about network topology, but it only shows the connections through which devices can exchange data.

Fig. 12: Range repeaters: diagram of the interaction between the nodes

(a) $rr[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$ (b) $\mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} | n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r}$



Theorem 10 (Range repeaters with two channels). *Let P be a process such that $fc(P) \subseteq \{c\}$ and $\mathtt{rcv}(P) = L$, let l and k be physical locations, r, r' transmission radii such that $d(k,l) \leq r$ and $d(l,k) \leq r'$. Then, for any channel d:*
$n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$ *simulates* $n[\{d/c\}P]^{\mathtt{s}}_{l,r}$.

Proof. Proof is given with the same method used for the previous theorem, so we prove that the relation

$$\begin{aligned}\mathcal{S} \stackrel{\text{def}}{=} \{(n[\{d/c\}P]^{\mathtt{s}}_{l,r}, n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}) : \\ \forall l,k,r,r'.d(k,l) \leq r \ \wedge \ d(l,k) \leq r' \\ \forall P.fc(P) \subseteq \{c\} \ \wedge \ \mathtt{rcv}(P) = L\}\end{aligned}$$

is a simulation. Suppose:

$$n[\{d/c\}P]^{\mathtt{s}}_{l,r} \xrightarrow{d!v@K} n[\{d/c\}P']^{\mathtt{s}}_{l,r}$$

for $K = \{k_1, ..., k_n : k_i \in L_i \wedge d(l, k_i) \leq r \, \forall \, 1 \leq i \leq n\}$ because $P \xrightarrow{\bar{c}_{L_i} v} P'$ for some $L_i \subseteq L$. knowing that $d(k, l) \leq r'$, by applying rule (Bcast) we obtain

$$\begin{array}{c} n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \xrightarrow{c_{L_i}!v[k,r']} \\ n[P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[\bar{d}_L\langle v\rangle.c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}. \end{array}$$

Now we can exploit the repeater `out`, created on purpose for output actions. By applying firs rule (Bcast) and later rule (Obs) we obtain:

$$\begin{array}{c} n[P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[\bar{d}_L\langle v\rangle.c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \stackrel{d!v@K'}{\Longrightarrow} \\ n[P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \end{array}$$

with $K' = \{k'_1, ..., k'_m : k'_i \in L \wedge d(l, k'_i) \leq r \, \forall \, 1 \leq i \leq m\}$. Then:

$$\begin{array}{c} n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \stackrel{d!v@K'}{\Longrightarrow} \\ n[P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}. \end{array}$$

As $L_i \subseteq L$ we deduce that also $K \subseteq K'$ and, by applying rule (Lose) to all the nodes with locations in K' which are not in K we can also write:

$$\begin{array}{c} n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \stackrel{d!v@K}{\Longrightarrow} \\ n[P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \end{array}$$

Now it is sufficient to prove that

$$(n[P']^{\mathtt{s}}_{l,r}, n[P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}) \in \mathcal{S}$$

Since for P' the same properties of P are still valid ($fc(P') \subseteq \{c\} \wedge \mathtt{rcv}(P') \subseteq L$), by applying the same procedure used for P we can prove the simulation.
Suppose now that an input action is executed, then we use `in`. Suppose:

$$n[\{d/c\}P]^{\mathtt{s}}_{l,r} \xrightarrow{d?v@l} n[\{d/c\}\{v/x\}P']^{\mathtt{s}}_{l,r}$$

because $P \xrightarrow{cv} \{v/x\}P'$. Since it lies in location l, also `in` can receive v through channel d. In particular we have:

$$\begin{array}{c} n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \xrightarrow{d?v@l} \\ n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[\bar{c}_L\langle v\rangle.d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \end{array}$$

Now, by applying rule (Bcast), recalling that $P = c(x).P'$, we can write:

$$\begin{array}{c} n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[\bar{c}_L\langle v\rangle.d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \xrightarrow{c_L!v[l,r]} \\ n[\{v/x\}P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \end{array}$$

We can then deduce:

$$\begin{array}{c} n[P]^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \stackrel{d?v@l}{\Longrightarrow} \\ n[\{v/x\}P']^{\mathtt{s}}_{k,r'} | \mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r} | \mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r} \end{array}$$

Now it is sufficient to prove that

$$(n[\{v/x\}P']^{\mathtt{s}}_{l,r}, n[\{v/x\}P']^{\mathtt{s}}_{k,r'}|\mathtt{out}[c \hookrightarrow_L d]^{\mathtt{s}}_{l,r}|\mathtt{in}[d \hookrightarrow_L c]^{\mathtt{s}}_{l,r}) \in \mathcal{S}$$

Since for P' the same properties of P are still valid ($fc(P') \subseteq \{c\} \wedge \mathtt{rcv}(P') \subseteq L$), by applying the same procedure we can finally prove the simulation.

Defining the *Minimum Radius of Maximum Observability*, we have emphasised the importance of ensuring that a message transmission can reach the whole set of intended receivers. By applying this concept to the definition of range repeater, we can introduce the notion of *complete range repeater*, that is a repeater which has a radius enough large to reach all the locations in the set of receivers.

Definition 10 (Complete range repeater). *We define the repeater complete on L as a repeater $rc[c \hookrightarrow_L c]^{\mathtt{s}}_{l,r}$ whose barb coincides with L, i.e., $rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \Downarrow_c$ and $d(l,k) \leqslant r \forall k \in L$.*

Looking at the example in Figure 11, where we suppose that a repeater is installed to allow the central server of the hospital to communicate with the ambulances, we can notice how the repeater has been chosen such as to guarantee that its transmission radius (the red segment in figure) covers the complete area of the disaster (orange line in figure). This is also an example of a complete range repeater, whose radius is a radius of maximum observability for the entire earthquake area. The use of a complete range repeater reduces the problem of ensuring the communication between the central server and a set of locations, to the communication between only two devices: H will be then sure that, in whatever locations the ambulances will lie, they will be always reachable by rr.

Theorem 11 (Complete range repeaters). *Let c be a channel, l and k locations (in particular $\sigma_k = k \vee nil(k)$ and $\sigma_l = l \vee nil(l)$), r and r' be transmission radii. Let P be a process such that $fc(P) \subseteq \{c\}$ and $\mathtt{rcv}(P) = L$. Finally let $rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r}$ be a complete range repeater on L, lying at location l, with radius r. Then:*
$n[P]^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \Downarrow_c \forall k, r'.d(k,l) \leq r'$, and all the recipients of L will be reachable by the node n. It means that, for each fragment of the process P executing an output action $\bar{c}_{L_i}\langle v_i\rangle.Q$ such that $P = P'.\bar{c}_{L_i}\langle v_i\rangle.Q$ it holds $n[\bar{c}_{L_i}\langle v_i\rangle.Q]^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \xrightarrow{c!v@L_i} n[Q]^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r}$

Proof. Let consider $P = P_1.\bar{c}_{L_1}\langle v_1\rangle.Q$, and $\bar{c}_{L_1}\langle v_1\rangle$ is the first output action in process P (we can write $P \stackrel{\tau}{\Longrightarrow} \bar{c}_{L_1}\langle v_1\rangle.Q$). We can write

$$n[P]^{\mathtt{s}}_{\sigma_k,r'} \stackrel{c_{L_1}!v_1[k,r']}{\Longrightarrow} n[Q]^{\mathtt{s}}_{\sigma_k,r'}$$

since $d(l,k) \leq r'$ we have

$$n[P]^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \stackrel{c_{L_1}!v_1[k,r']}{\Longrightarrow} n[P']^{\mathtt{s}}_{\sigma_k,r'}|rc[\bar{c}_L\langle v\rangle.c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r}$$

By applying rule (Bcast) we can write

$$n[P']^{\mathtt{s}}_{\sigma_k,r'}|rc[\bar{c}_L\langle v\rangle.c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \xrightarrow{c_L!v_1[l,r]} n[P']^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r}$$

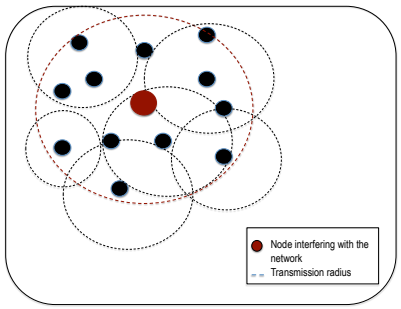
By applying rule (Obs), knowing that $L_1 \subseteq L$ and consequently $d(l,k) \leq r \; \forall k \in L_1$ we can write

$$n[P]^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \overset{c!v_1@L_1}{\Longrightarrow} n[P']^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r}$$

By our definition of Barb, using Harmony theorem we can deduce also $n[P]^{\mathtt{s}}_{\sigma_k,r'}|rc[c \hookrightarrow_L c]^{\mathtt{s}}_{\sigma_l,r} \Downarrow_c$ and the set of receivers is L because $d(k,l) \leq r \; \forall k \in L$. For the successive output actions of the process P the proof is similar because we know that each $L_i \subseteq L$ is completely reachable by the range repeater used.

5.7 The problem of *Interference*

Fig. 13: Example of Interference: a node with a too large transmission energy may disturb the other transmission within the network



One of the most critical problems in managing an ad hoc network is the power consumption, since it can be constituted of different kinds of nodes which may be powered with weak battery. The concept of *topology control* is the technique used in order to reduce the initial topology of the network to save energy and to extend the lifetime of the network. This can be then considered as a trade-off between power saving and network connectivity. In other words when we have to choose the transmission power of each node, we know that choosing a low transmission power for a node we will reduce its connectivity within the network, but we also reduce its power consumption. The main goal of topology control is therefore the choice, for each node, of a minimum transmission power which guarantees network connectivity. There is an other problem depending on the transmission power we chose for a node sending data within the network: a too large transmission radius will reach a large number of nodes, and there is a higher probability to disturb devices not interested in receiving the data transmitted, and to congest the network.

Following the definition of interference introduced in [5], the notion of *interference* arises from a natural question: How many other nodes a given network

node can disturb? Consider a node transmitting a message, we can then define the interference as the number of nodes listening to the message, but not interested in receiving it.

Definition 11 (Interference). *Let consider an output action* $c_L!\tilde{v}[l,r]$*, and* $K = \{k : d(l,k) \leq r\}$*, then the* level of interference *with respect to this output is defined as:*

$$I(c_L!\tilde{v}[l,r]) = |K - L|$$

For example, looking at Figure 13 we can observe that the red node has got a transmission radius large enough to reach all the nodes of the network: this is not a positive characteristic because means a large energy consumption, together with a higher probability of collisions with other transmission and a consequent network congestion.

Once we have explained the notion of interference, given a transmission, if the set of nodes not interested in receiving the transmission is empty, we can affirm that there is not interference in the communication , i.e. if $I(c_L!\tilde{v}[l,r]) = 0$, then we say that there is no interference in the communication.

Using CMN$^{\#}$ in order to describe the case in which a transmission reaches only the receivers, without any "interference" we can compare the behaviour of a node communicating with a given set L of recipients, with the same node broadcasting the same communication to the whole network. If what results from this comparison is a bisimulation, this means that there is no interference in the transmission considered, in other words this action will not "disturb" any other node of the network.

Let us first define the broadcasting version of a process P, denoted by $brd(P)$, as follows:

- if $P = \mathbf{0}$ then $brd(P) = \mathbf{0}$;
- if $P = c(\tilde{x}).P'$ then $brd(P) = c(\tilde{x}).brd(P')$;
- if $P = \bar{c}_L\langle\tilde{v}\rangle.P'$ then $brd(P) = \bar{c}_\infty\langle\tilde{v}\rangle.brd(P')$.
- if $P = [w_1 = w_2]Q, R$ then $brd(P) = [w_1 = w_2]brd(Q), brd(R)$.

Definition 12 (Interference-freeness). *We say that a node* $n[P]^\mu_{l,r}$ *is* interference-free *whenever*

$$n[P]^\mu_{l,r} \cong n[brd(P)]^\mu_{l,r}.$$

Theorem 12. *If* $n[P]^\mu_{l,r}$ *is* interference-free *then for all output actions* $c_L!\tilde{v}[l,r]$ *performed by* $n[P]^\mu_{l,r}$ *it holds that* $I(c_L!\tilde{v}[l,r]) = \emptyset$.

Proof. We will give a proof by contradiction. Since $n[P]^\mu_{l,r}$ is *interference-free* we can write:

$$n[P]^\mu_{l,r} \cong n[brd(P)]^\mu_{l,r}$$

Let consider a derivative $n[P']^\mu_{l',r}$ of $n[P]^\mu_{l,r}$ such that $n[P']^\mu_{l',r} \xrightarrow{c_L!v[l,r]} n[P'']^\mu_{l',r}$. Since $n[P]^\mu_{l,r} \cong n[brd(P)]^\mu_{l,r}$ there exists a derivative $n[Q']^\mu_{l'',r}$ of $n[brd(P)]^\mu_{l,r}$ such

that $n[P']^{\mu}_{l',r} \cong n[Q']^{\mu}_{l'',r}$. Now suppose by contradiction that $I(c_L!v[l',r]) > 0$. Then, by definition of Interference it holds that there exists k such that $d(l',k) \leq r$ and $k \notin L$. If we apply rule (Obs) to $n[Q']^{\mu}_{l'',r}$, assuming that only k could receive $\tilde{v}$, then we will write:

$$n[Q']^{\mu}_{l'',r} \xrightarrow{c!\tilde{v}@k} n[Q'']^{\mu}_{l'',r}$$

But we cannot write:

$$n[P']^{\mu}_{l',r} \stackrel{c!\tilde{v}@k}{\Longrightarrow} n[P'']^{\mu}_{l',r}$$

because $k \notin L$. Since we assumed that $n[P']^{\mu}_{l',r} \cong n[Q']^{\mu}_{l'',r}$, we reached a contradiction.

6 An example of CMN$^{\#}$ employment: the AODV

6.1 Routing protocols for mobile ad hoc networks

One of the most critical problems in the development of a mobile ad hoc network is the packets routing. When choosing the routing protocol we have to pay particular attention to the operational cost, since devices may be power limited (ad hoc networks are often composed of notebooks, mobile phones...); we have also to consider that there can be cases of non-cooperation of two types of nodes: selfish ones that want to save power, and malicious nodes that are interested in attacking the network.

We can classify routing protocols in three different categories:

- *Proactive protocols*: this kind of protocol maintain a list of destinations with the routes to reach them, and periodically refresh it. The advantage of these protocols is that every time a node wants to communicate with some other device, it has got the information it needs immediately; otherwise there are disadvantage due to the cost of data freshness maintaining.
- *Reactive protocols*: this kind of protocol find a route only on demand. The advantage that protocol search only the useful data, otherwise, when a node send a request, there can be a high latency time for the response to be received.
- *Hybrid routing*: there are some routing protocols that combine both reactive and proactive routing (in hierarchical protocols for example the choice of using a reactive or a proactive routing depends on the hierarchical level of a node looking for a route).

One of the most used routing protocols in a MANET's implementation is the **AODV** (*Ad hoc On-demand Distance Vector Routing*) [6], [21], [1], [23], which has been chosen to illustrate the usefulness of our calculus, just because of its common employment in this kind of situations.

The AODV is a reactive protocol which provides the best routes between the nodes only by request, and uses a set of sequence numbers to guarantee

received data to be always updated. The older CMN appeared unsuitable to represent this protocol, because it supports only broadcast communication, while the AODV needs also unicast and multicast transmissions. The main packets of this protocol are three: RREQ (route request), that is usually transmitted via broadcast, RREP (route reply), that is instead transmitted to the requester with an unicast transmission, and RERR (packet transmitted by a node after a connection failure); the best kind of transmission for RERR packets is the multicast communication, because not all the nodes of the network are interested in receiving this information, but only the devices that need to refresh their route table. Figure 14 describes the behavior of RREQ and RREP propagation: notice that a request broadcast generates an explosion of messages (Figure 14.a), while the reply, using a unicast communication, will be propagated to a limited number of locations (Figure 14.b).

AODV is one of the most used protocols for mobile ad hoc networks, otherwise there are also other protocols used in this kind of contexts. For example in some cases we can find the proactive version of the AODV, that is DSDV (Destination-Sequenced Distance Vector routing) [16], that uses the same technique of its reactive version, but it finds all possible routes within the network regardless of the fact that they will be useful or not; An other used reactive protocol for MANETs is DSR (Dynamic Source Routing) [18], which is similar to AODV, but it uses source routing instead of relying on the routing table at each intermediate device; as concerns the hybrid protocols we can find for example ZRP (Zone Routing Protocol) [10] or GSR (Global State Routing) [7] which in particular is a hierarchical routing protocol which is based on link state vectors (nodes exchange vectors of link states among their neighbors during routing information exchange).

In this section we are going to give an implementation of AODV to show the usefulness of our calculus for the analysis of concretes problems in the implementation of a mobile ad hoc network. Even though we consider only one of a great number of possible choices for the routes management, since the most routing protocols used in mobile ad hoc networks implementations have some common characteristics, we are persuaded that even changing the routing protocol, this calculus could be anyway able to capture properties and problems in the network management.

6.2 Details of the algorithm

Each node maintains tables to store information about its neighbors. In particular a *route table* is associated to each node, with an IP address k as key. it contains the following information:

- $seq\#_k$ denotes the sequence number of the node k;
- `nexthop` denotes the address of the first node in the route from the source to k;
- `hopcount` denotes the number of steps which separate the source from k;
- `lifetime` denotes the validity of the record in the table;

Fig. 14: Example of route request and reply from S to D

(a) RREQ broadcast (b) RREP transmission

– `List of precursors` denotes the list of nodes that use the source as the `nexthop` towards the destination.

The protocol manages route requests to connect the nodes of the network through the three packets mentioned before; more precisely:

RREQ (Route Request): A node looking for a route broadcasts an RREQ packet (Figure 14.a), then waits. It will choose the best route (in terms of costs, of reply time, or route length).
RREP (Route Response): A node receiving an RREQ makes a first control to verify if it is an intended recipient of the message, if the condition is verified it immediately replies, otherwise it looks in its *route table* if there exists a valid route for that destination; finally, if it has got no information to generate the RREP, it propagates the RREQ before answering (Figure 14.b).
RERR (Route Error): the error message is used for warning the network that a node is no more reachable (this situation happens for several reasons, as the disconnection of a node, or a software/hardware damage)

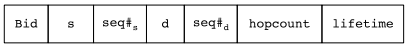
As we just told about before, a "sequence number"($seq\#_l$) has been associated to each node; it has been increased every time that a node decides to broadcast an RREQ, or a node that receives an RREQ, and that is exactly the destination of the route request, have to send back the reply. In this second case the sequence number to be increased will be chosen between the current value and the value written in the RREQ packet.

RREQ in detail

As Figure 15 illustrates, a packet RREQ contains:

– the field `Bid` (Broadcast ID), which identifies the packet; it is incremented every time a new copy of the same request is broadcasted;

Fig. 15: RREQ packet structure



- The couple $(s, seq\#_s)$, denoting the IP address and the sequence number of the route source;
- The couple $(d, seq\#_d)$, denoting the IP address and the sequence number of the route destination;
- The field `hopcount`, initialized to zero;
- The field `lifetime`, denoting the validity of the information contained in the packet.

Each time a node sends a request, it waits the reply and then refreshes its route table with the new information acquired.

A node after having received an RREQ:

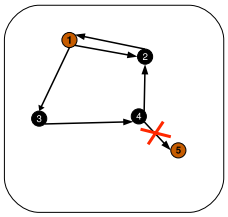
1. Controls if the received RREQ is a new information or a duplicate;
2. If the node itself is the destination of the request it immediately sends back the reply;
3. Otherwise controls the information contained in the route table. If it has got the required information, and the sequence number of the destination is at least the same of the request sequence number, it immediately answers with an RREP packet;
4. If the entries of its route table are too old, it increments `Bid` and `hopcount`, propagates the request and, once the reply is received, it sends back to the source the RREP required.

The field `hopcount` is necessary for the creation of a *Reverse route*, that is an entry in the route table of the node having received the request, made using the information contained in the RREQ, modelling a path from the current node to the source of the RREQ transmission.

Notice that `Bid` is a progressive number increased each time a request is transmitted again; this field is necessary for the good operating of the protocol, because it prevents the network from loops in RREQ propagations.

Figure 16 emphasizes the importance of using a Broadcast ID in a route request. In particular we show what can happen by omitting this field in the RREQ: a node could receive continuously the same packet without recognizing

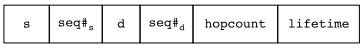
Fig. 16: Loop caused by an RREQ transmission without duplicates control



it, and repropagates it "ad infinitum". There can be generated loops, causing network obstruction and power dissipation.

RREP in detail

Fig. 17: RREP packet structure



As Figure 17 illustrates, RREP packet contains:

- The couple $(s, seq\#_s)$, with the sequence number updated;
- The couple $(d, seq\#_d)$, with the sequence number updated;
- The field `hopcount` with the number of the steps needed by s to reach d;
- The field "`lifetime`" updated.

As we just told in describing RREQ packet, a node answers with an RREP when it is itself the destination or when it has got the required information updated. If the node is the destination of the request:

1. if the RREQ sequence number is greater then the one of the node having received the packet, the receiver updates its sequence number with the RREQ's one, then it increments it.

2. it builds and broadcasts the RREP with the new sequence number of the destination, and `hopcount` incremented.

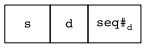
If the node having received the RREQ is not the destination, but it has got the required information:

1. it updates the field `hopcount` of the destination substituting it with the one contained in the table;
2. it updates the field `lifetime`;
3. it builds and sends the RREP with the new sequence number and the field `hopcount` updated.

Once having received an RREP, a node is ready to update its own route table; however the update can be executed only if the sequence number of the destination in the RREP is greater then the one in the entry, or if the two fields contain the same value, but the `hopcount` of the reply is less then the one in the entry. Finally also an eventually existing list of precursors will be updated.

RERR in detail

Fig. 18: RERR packet structure



When a node discovers a connection failure with another device of the network, it broadcasts an error message containing the address and the sequence number of the disconnected node, so that the neighbors can be advised that the disconnected node is no more reachable. As Figure 18 illustrates, the packet RERR contains:

– s, denoting the source of the invalidated path;
– The couple $(d, seq\#_d)$, with the sequence number updated.

Every time a node, sending a packet, realizes that a connection failure has happened, or when it receives an RERR:

1. it detects all the no more reachable destinations;

2. it updates the route table, invalidating the routes towards the no more reachable destinations and updating all their sequence numbers;
3. broadcasts the RERR packet;
4. If there exists a list of precursors the broadcast of RERR is no more necessary, but it suffices to send the packet to the precursors of the invalidated paths.

For a regular control of the routes validity it should be useful to establish time interval t; after each timeout every node broadcasts a "HELLO" packet containing the location of the source, so that the network knows that it is still alive. Thanks to the "HELLO" packets monitoring, the nodes can constantly control their *route tables* correctness. We will omit the management of these messages to simplify the implementation of the protocol.

6.3 A rappresentation of the AODV through CMN$^{\#}$

Now we are going to give a practical example of the usefullness of our calculus. We have chosen the AODV as the object of our example to justify our choice to extend CMN, proving that those extensions really improved this calculus. The routing protocol considered can manage broadcast, multicast and unicast transmission. In CMN only broadcast communication is permitted, whereas CMN$^{\#}$ allows all the transmissions necessary for modelling this routing protocol.

Assumptions

Before defining the semantics it is necessary the introduction of some terms, in order to simplify the writing of the message exchanges. First we decide to omit the field `lifetime`. Using the field `hopcount` in the RREQ management, the route chosen from the source to the destination of the request will be the one with the lower number of steps. In the real environments where Mobile Ad-Hoc Networks are adopted the way of choosing the best route to reach a node is not based on the number of steps. In the Ad-Hoc networks implementation the first property we want to guarantee is the power saving, and this characteristic influences also the routing protocol: the best routes will be the ones reachable with the lowest power cost [2]. In our model we will introduce the function `is better` to manage the choice of the best route without specifying the adopted method. In the request management a node will use the information about its own location. We then introduce the terms `self_node` and `self_loc`, that will be substituted respectively by the name and the location of the device executing the process. The name will represent the IP address which allows one to identify each single node of a network; an IP address cannot be indeed assigned to two distinct nodes. In our model it will be necessary increasing some numerical values; let v a value, then $v{+}{+}$ means its increased.

Now we are going to illustrate the data structures necessary to the nodes for a correct transmission and the reception of the route requests in the network:

RT = route table of the node executing the protocol. The entries of RT are in the form $\langle d, seq\#_d, \texttt{nexthop}_d, \texttt{hopcount}_d, LP_d\rangle$; d denotes the IP address of a destination, and it is the unambiguous key of the table which allows one to distinguish the different entries of the table;
RQT = request table. This table protects the network from loops in the packets broadcasts. (Consider the case in Figure 16), and it is constituted by entries in the form $\langle s, d, \texttt{Bid}_{s,d}\rangle$.

The tables RT and RQT are not associated to the locations, but to the names of the nodes, because they represent the physical store of the devices executing the process. The use of data structures emphasizes the importance of distinguishing between the name of a device executing a process, and its location. A mobile node actually moving in the network however maintains the information stored in its data structures. Using the same tag to identify both name and location of a node, after a location changing we will loose the data structures associated to that node. Another solution could be the one adopted by the CBS$^{\#}$ [15], where the store is associated to the location, but this choice, as already explained in the introduction (describing Nanz and Hankin work), will make the model unnecessarily heavy. The information stored in each entry of a route table is another proof of the importance of distinguishing between the name and the physical location of a node: each entry contains the route to reach a certain device (so the key of the entry is the name of the destination), however the next hop (the first step to reach the destination) is not a device, but a location, because the route from a source to a destination has got a precise physical direction.

As already told, the AODV is constituted of three packets, which will be represented by three tuples, where the first value is a constant allowing the nodes to recognize the packet nature and to identify the information that has been sent. We add an ack packet, to update the list of precursors of route tables each time new routes are found.

$(\texttt{rreq}, \texttt{Bid}_{s,d}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d})$ = packet to require the best route from s to d;
$(\texttt{rrep}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d})$ = reply packet containing the route from s to d;
$(\texttt{rerr}, s, d, seq\#_d)$ = packet informing the network of a connection failure between s and d.
$(\texttt{ack}, d, seq\#_d)$ = packet sent by s to confirm the reception of a path towards d.

In order to make the protocol more legible we introduce some functions describing the behavior of the nodes in particular conditions. Our choice does not compromise the correctness of the model, because these functions can be read as boolean constants.

- $\texttt{geq}(u, v)$ = function that returns true if u is greater than v, or if u and v are equal;

- `is better`($entry_d$) = function that returns true if, comparing $entry_d$ with the entry associated to d in the route table of the current node, it will result that it does not exist a route towards that destination, or that already exists an entry for that destination, but the sequence number of the new entry is greater than the one in the route table, or the sequence numbers are equal, but the new route is better than the one contained in RT;
- `refresh`($entry_d$) = function that inserts $entry_d$ in RT, or overwrites the entry, if already inserted in the route table;
- `unable`(d) = function that eliminates the entry relative to d (if existent) from RT ;
- `new`($\langle s, d, \texttt{Bid}_{s,d}\rangle$) = function that inserts the entry $\langle s, d, \texttt{Bid}_{s,d}\rangle$ in RQT;
- `isNew`($s, d, \texttt{Bid}_{s,d}$) = function that controls if a node has already received the request with the Broadcast ID $\texttt{Bid}_{s,d}$, comparing it with the one in the RQT.

Protocol development

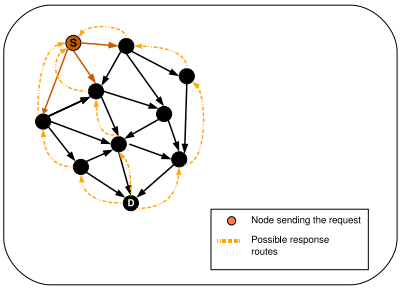
In order to make the protocol more legible the functions will be written in a compact way, ignoring some rules of the CMN$^{\#}$ sintax. We will write [*cond*] instead of [$cond = \texttt{true}$]. As concerning the values researches in the data structures it should be necessary to have a condition for each entry of the table visited. We reduce this series of conditions with only one summarizing them, exploiting the logical symbol $\exists$, that returns true if the object we are looking for exists in the table. We introduce the symbols for the logical AND ($\wedge$) and OR ($\vee$): the logical AND stands for a set of consecutive conditions, whereas an easy way to solve the OR is writing it as a series of AND.

CMN# has got no stores associated to the nodes of the network. We could extend the calculus inserting a data store, but this modification will make the model too much heavy in vain, because the store actions are internal to a node, then they are not observable. In our example we will simply consider two tables associated to a node n of the network: RT_n (route table) and REQ_n (requests table).

Before illustrating the behavior of the nodes in sending and receiving route requests, we introduce the auxiliary function $REVERSE\_ROUTE$, which exploits the information in the RREQ to refresh the route table. If there are no paths towards the request source the function will generate a new entry; if a route towards the source already exists, the function will compare the new data received with the ones contained in the table and it will choose the ones to store. Notice that, creating the new entry, the list of precursors does not change, because this set will be modified only after connection failures or after the reception of a route acceptance. LP_d represents the set of precursors relative to d in the route table of the current node.

Figure 19 shows how useful is the function *REVERSE ROUTE*: the orange arrows indicate the possible replies to the *RREQ* sent by the source; using a function that is able to create reverse routes, we can guarantee that, when

Fig. 19: Example of reverse route possible use



propagating the response, all the nodes involved in the transmission have the information required to let the message arrive to the destination.

$$REVERSE\_ROUTE\langle(d, seq\#_d), next, count\rangle \stackrel{\text{def}}{=} [\texttt{is better}(\langle d, seq\#_d, next, count, LP_d\rangle)]\ \texttt{refresh}(\langle d, seq\#_d, next, count, LP_d\rangle)$$

A node looking for a route to d executes:

$$RREQ\_SND\langle d\rangle \stackrel{\text{def}}{=} \texttt{new}(\langle \texttt{self\_node}, d, \texttt{Bid}_{\texttt{self\_node},d}\rangle).\ \bar{c}_\infty\langle(\texttt{rreq}, \texttt{Bid}_{\texttt{self\_node},d}, (\texttt{self\_node}, seq\#_{\texttt{self\_node}}), (d, seq\#_d), 0), \texttt{self\_loc}\rangle$$

A node, after receiving a request, controls the absence of loops and then it can behaves in three ways: if it is the destination itself, it immediately replies, otherwise it controls its route table; if it has got the necessary information it sends back the reply, else it repropagates the request to the neighbors. The insertion of information about the list of precursors in the $RREP$ is not necessary, because, if the route in the table has been refreshed, this list does not change, whereas if the entry has been created for the first time the list of precursors will be empty.

In order to simplify the understanding of the protocol we denote the variables with names corresponding with the entities represented (for example the third value received in a request is the source, then we will directly write s instead of x_3).

$$\begin{aligned} RREQ\_RCV \stackrel{\text{def}}{=}\ & c(\tilde{x}).[x_1 = \texttt{rreq}](\\ & [\texttt{isNew}(s, d, \texttt{Bid}_{s,d})](\texttt{hopcount}_{s,d}{+}{+}.[d = \texttt{self\_node}](\\ & seq\#_{\texttt{self\_node}}{+}{+}.\bar{c}_p\langle(\texttt{rrep}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d}), \texttt{self\_loc}\rangle),\\ & [\exists d' \in RT.(d' = d \wedge \texttt{geq}(seq\#_{d'}, seq\#_d))](\\ & \bar{c}_p\langle(\texttt{rrep}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d}{+}\texttt{hopcount}_{s,d'}), \texttt{self\_loc}\rangle),\\ & (REVERSE\_ROUTE((s, seq\#_s), p, \texttt{hopcount}_{s,d}).\\ & \texttt{Bid}_{s,d}{+}{+}.\bar{c}_\infty\langle(\texttt{rreq}, \texttt{Bid}_{s,d}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d}), \texttt{self\_loc}\rangle))) \end{aligned}$$

A node after a reply reception, if it is the destination itself, it updates its route table with the new information, otherwise it propagates the reply towards the real destination:

$$
\begin{aligned}
RREP\_RCV \stackrel{\text{def}}{=} c(\tilde{x}).[x_1 = \texttt{rrep}](& \\
& [s = \texttt{self\_node}].[\texttt{is better}(\langle d, seq\#_d, p, \texttt{hopcount}_{s,d}, LP_d\rangle)] \\
& (\texttt{refresh}(\langle d, seq\#_d, p, \texttt{hopcount}_{s,d}, LP_d\rangle).\bar{c}_p\langle(\texttt{ack}, d, seq\#_d), \texttt{self\_loc}\rangle), \\
& \bar{c}_{\texttt{nexthop}_s}\langle(\texttt{rrep}, (s, seq\#_s), (d, seq\#_d), \texttt{hopcount}_{s,d}), \texttt{self\_loc}\rangle)
\end{aligned}
$$

Each time a node receives an acknowledgment it updates its list of precursors in this way:

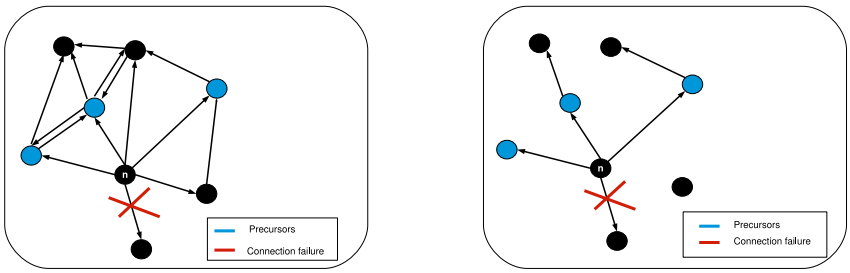
$$ACK\_RCV \stackrel{\text{def}}{=} c(\tilde{x}).[x_1 = \texttt{ack}](LP_d = LP_d \cup \{p\})$$

When a node looses the connection with one of its neighbors, it broadcasts an error message; We are going to exploit the list of precursors in order to optimize the management of the $RERR$ transmissions, preventing the network from an useless power cost caused by the broadcast of the error message all over the network. With this solution we actually reduce the set of the destinations of the `rerr` packet to the list of precursors associated to the disconnected node. Figure 20 illustrates the optimization produced by our solution in the development of the protocol: notice how using a simple broadcast (Figure 20.a), the propagation of RERR packet generates an explosion of messages, while using the list of precursors (Figure 20.b), there are less transmissions, because the message will be repropagated only by those nodes interested in invalidating the route.

Fig. 20: Propagation of an error message

(a) Propagation without using a list of precursors

(b) Propagation using the list of precursors



$$RERR\_SND\langle d\rangle \stackrel{\text{def}}{=} \bar{c}_{LP_d}\langle(\texttt{rerr}, \texttt{self\_node}, d, seq\#_d), \texttt{self\_loc}\rangle$$

$$RERR\_RCV \stackrel{\text{def}}{=} c(\tilde{x}).[x_1 = \texttt{rerr}]($$
$$[\exists d' \in RT.(d' = d \wedge \texttt{nexthop}_{d'} = l_s \wedge \texttt{geq}(seq\#_d, seq\#_{d'})]$$
$$(\texttt{unable}(d).\bar{c}_{LP_d}\langle(\texttt{rerr}, \texttt{self\_node}, d, seq\#_d), \texttt{self\_loc}\rangle))$$

In the development of the protocol, a strict loyalty to the sintact rules of the calculus should compromise the understanding of the functions; we then use some abstractions in order to make the functions more compact, but every function can be written using the correct sintax of CMN$^{\#}$. Following we will prove this assertion for one of the functions, in particular we decided to write $RREQ\_RCV$. Let $d_1, ..., d_n$ be the n primary key associated to the table RT of the node executing the action.

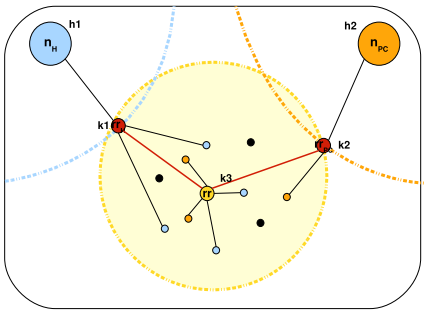
$$RREQ\_RCV \stackrel{\text{def}}{=} c(\tilde{x}).[x_1 = \texttt{rreq}]($$
$$[\texttt{isNew}(x_3, x_5, x_2)](x_7{+}{+}.[x_5 = \texttt{self\_node}]($$
$$seq\#_{\texttt{self\_node}}{+}{+}.\bar{c}_{x_8}\langle(\texttt{rrep}, (x_3, x_4), (x_5, x_6), x_7), \texttt{self\_loc}\rangle),$$
$$[d_1 = x_5][\texttt{geq}(seq\#_{d_1}, x_6) = true]$$
$$(\bar{c}_{x_8}\langle(\texttt{rrep}, (x_3, x_4), (x_5, x_6), x_7 + \texttt{hopcount}_{x_3,d_1}), \texttt{self\_loc}\rangle),$$

$$...$$

$$[d_n = x_5][\texttt{geq}(seq\#_{d_n}, x_6) = true]$$
$$(\bar{c}_{x_8}\langle(\texttt{rrep}, (x_3, x_4), (x_5, x_6), x_7 + \texttt{hopcount}_{x_3,d_1}), \texttt{self\_loc}\rangle),$$
$$(REVERSE\_ROUTE((x_3, x_4), x_8, x_7).$$
$$x_2{+}{+}.\bar{c}_{\infty}\langle(\texttt{rreq}, x_2, (x_3, x_4), (x_5, x_6), x_7), \texttt{self\_loc}\rangle))))$$

Example of using AODV protocol

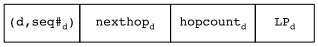
Fig. 21: Development of a network in the rescues management



Following we are going to describe a practical example of the AODV employment. Let consider the case of an earthquake, and then the development of a MANET to manage communications between the rescue's entities. In particular we assume that the hospital and the civil defense administer their rescuers with

an access point located in their central station. Figure 21 depicts the situation we have supposed to be happened. We then write n_H to denote the central server of the hospital, that is a stationary node located at $h1$, with transmission radius r_{h1}. The central server of the civil defense will be represented by the node n_{PC}, located at h_2, with transmission radius r_{h2}. The hospital and the civil defense are too distant from the disaster area (in figure represented by the yellow circle); it will be necessary the installation of range repeaters to ensure the ambulances and the other rescuers of a correct communication with the central servers, once they arrive in the emergency area. A range repeater (brr) will be placed near the epicenter, covering all the area. There will be also placed two range repeaters at middle way between the epicentre and the two central servers. This development tries to ensure the complete cover of the area damaged by the earthquake.

Fig. 22: Structure of a route table entry



At the beginning the *route table* of the nodes we have introduced above are not empty and we are going to describe them in detail. Notice that the route table stores the routes towards the other nodes (usually the unambiguous name distinguishing one node to each other is the IP address), but the intermediate steps of the routes are defined by the physical locations; this fact evidences the importance of a distinction between name and location of a node. The name of a node is actually associated to the physical device executing the process, while the location denotes its physical position in the network. The route table is a data structure associated to a device, but when communicating messages to other devices a node has to know the physical locations of the neighbour which will repropagate its message. The entries of this table are described in Figure 22. The beginning situation of our example will be following described in detail:

RT_{n_H}

$(rr_H,1)$	k1	1	$\emptyset$
$(rr,1)$	k1	2	$\emptyset$
$(n_{PC},1)$	k1	4	$\emptyset$

$RT_{n_{PC}}$

$(rr_{PC},1)$	k2	1	$\emptyset$
$(rr,1)$	k2	2	$\emptyset$
$(n_H,1)$	k2	4	$\emptyset$

RT_{rr_H}

$(n_H,1)$	h1	1	$\{k3\}$
$(rr,1)$	k3	1	$\{h1\}$
$(n_{PC},1)$	k3	3	$\{h1\}$

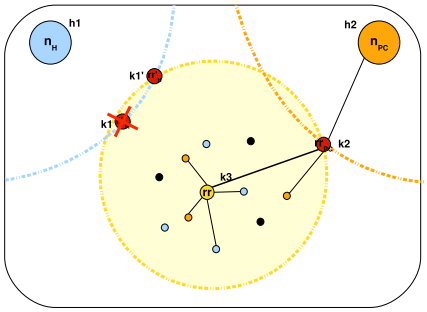
$RT_{rr_{PC}}$

$(n_{PC},1)$	h2	1	$\{k3\}$
$(rr,1)$	k3	1	$\{h2\}$
$(n_H,1)$	k3	3	$\{h2\}$

RT_{rr}

$(n_H,1)$	k1	2	$\{k2\}$
$(rr_H,1)$	k1	1	$\{\emptyset\}$
$(n_{PC},1)$	k2	2	$\{k1\}$
$(rr_{PC},1)$	k2	1	$\{\emptyset\}$

Fig. 23: New network topology after the disconnection of rr_H



Now consider the situation depicted by Figure 23, which supposes the disconnection of the range repeater located at the border of the hospital's transmission cell, probably caused by a hardware damage.

$$\text{(s-Disc)}\ \frac{-}{rr_H[P]^{\mathtt{s}}_{k1,r_{k1}} \xrightarrow{\tau} rr_H[P]^{\mathtt{s}}_{nil(k1),r_{k1}}}$$

After this disconnection both n_H and rr loose their connection with rr_H, and when they discover that, they must update their route table and send an error message to all the nodes that are members of the list of predecessors in the routes which have $k1$ as `nexthop`. For each destination d reachable through $k1$, the following actions are to be executed:

$$\mathtt{unable}(d).RERR\_SND(d)$$

Notice that, looking at the tables of n_H and rr, the routes to be invalidated are four; after the propagation of the message in the whole network, at the same way there will be invalidated all the routes using $k1$ as hop through any destination. The final situation will be:

RT_{n_H} $\emptyset$

$RT_{n_{PC}}$

$(rr_{PC},1)$	k2	1	$\emptyset$
$(rr,1)$	k2	2	$\emptyset$

RT_{rr_H} $\emptyset$

$RT_{rr_{PC}}$

$(n_{PC},1)$	h2	1	$\{k3\}$
$(rr,1)$	k3	1	$\{h2\}$

RT_{rr}

$(n_{PC},1)$	k2	2	$\{\emptyset\}$
$(rr_{PC},1)$	k2	1	$\{\emptyset\}$

After a first attempt of establishing once again the connection with the network without success, the staff decides to install a new range repeater near rr_H, placed in a location that enables it to reach both the hospital and rr. There will be then placed rr'_H at the location k'_1; suppose now for simplicity the route table of rr'_H not to be empty, but including the same entries of rr_H route table.

$RT_{rr'_H}$

$(n_H,1)$	h1	1	$\{k3\}$
$(rr,1)$	k3	1	$\{h1\}$
$(n_{PC},1)$	k3	3	$\{h1\}$

Suppose now that rr needs to communicate with n_H, that is a stationary node placed at location $h1$. rr will broadcast a request to reach n_H:

$$rr[RREQ\_SND(n_H).RREP\_RCV.P]^{\mathtt{s}}_{k3,r_{k3}} \Longrightarrow rr[RREP\_RCV.P]^{\mathtt{s}}_{k3,r_{k3}}$$

The message broadcasted will be received also by rr'_H, located at $k1'$. rr'_H has got the information requested, then it can immediately send back a reply to rr, which consequently updates its route table. If n_{PC} and rr_{PC} will send a route request for n_H, they will obtain a reply directly by rr. Analogously n_H, broadcasting request packets in order to obtain new routes towards the other locations that it is interested to reach, will obtain a reply by rr'_H, that is placed inside the n_H transmission cell. As regards the other nodes which were directly connected to rr_H, we know that rr is a complete range repeater (meaning that its transmission radius covers the whole area interested in communicating with the hospital or with the civil defense), and this characteristic ensures that all the nodes inside this area will be however reachable by n_H, also if not reachable by rr'_H. We are now going to write in detail the code describing the communication between rr and rr'_H. Let P_{rr} and $P_{rr'_H}$ be the processes executed respectively by rr and rr'_H:

$P_{rr} = RREQ\_SND(n_H).RREP\_RCV.Q_1$
$P_{rr'_H} = RREQ\_RCV.Q_2$

Let v be a route request packet broadcasted by rr to find the best route to reach n_H:

$v = (\mathtt{rreq}, \mathtt{Bid}_{rr,n_H}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H}), k3$
where $\mathtt{Bid}_{rr,n_H}$, $seq\#_{rr}$, $seq\#_{n_H}$ are constants initialized to 1, while $\mathtt{hopcount}_{rr,n_H}$ is 0.

rr'_H is placed inside the rr's transmission cell, then it will receive v.

$$rr[P_{rr}]^{\mathtt{s}}_{k3,r_{k3}}|rr'_H[P_{rr'_H}]^{\mathtt{s}}_{k1',r_{k1'}} \xrightarrow{c_\infty!v[k3,r_{k3}]} rr[P'_{rr}]^{\mathtt{s}}_{k3,r_{k3}}|rr'_H[P'_{rr'_H}]^{\mathtt{s}}_{k1',r_{k1'}}$$

Both the processes evolve:

$P'_{rr} = RREP\_RCV.Q1$
$P'_{rr'_H} = [\mathtt{isNew}(rr, n_H, \mathtt{Bid}_{rr,n_H})](\mathtt{hopcount}_{rr,n_H}{+}{+}.[n_H = \mathtt{self\_node}]$
$(seq\#_{\mathtt{self\_node}}{+}{+}.$
$\bar{c}_{k3}\langle(\mathtt{rrep}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H}), \mathtt{self\_loc}\rangle),$
$[\exists d \in RT.(d = n_H \wedge \mathtt{geq}(seq\#_d, seq\#_{n_H}))]$
$(\bar{c}_{k3}\langle(\mathtt{rrep}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H} +$
$\mathtt{hopcount}_{rr'_H,n_H}), \mathtt{self\_loc}\rangle),$
$(REVERSE\_ROUTE((rr, seq\#_{rr}), k3, \mathtt{hopcount}_{rr,n_H}).$
$\mathtt{Bid}_{rr,n_H}{+}{+}.$
$\bar{c}_\infty\langle(\mathtt{rreq}, \mathtt{Bid}_{rr,n_H}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H}), \mathtt{self\_loc}\rangle))).Q2$

rr'_H executes a series of τ-actions, elaborating the RREQ broadcasted by rr.

$$rr'_H[P'_{rr'_H}]^{\mathtt{s}}_{k1',r_{k1'}} \Longrightarrow rr'_H[P''_{rr'_H}]^{\mathtt{s}}_{k1',r_{k1'}} \Longrightarrow rr'_H[P'''_{rr'_H}]^{\mathtt{s}}_{k1',r_{k1'}}$$

rr'_H controls if it is the real destination of the request. Once it discovers that the request is addressed to n_H it looks at all the entries of its route table looking for the required information.

$P''_{rr'_H} = [\exists d \in RT.(d = n_H \wedge \mathtt{geq}(seq\#_d, seq\#_{n_H}))]$
$(\bar{c}_{k3}\langle(\mathtt{rrep}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H} +$
$\mathtt{hopcount}_{rr'_H,n_H}), \mathtt{self\_loc}\rangle),$
$(REVERSE\_ROUTE((rr, seq\#_{rr}), k3, \mathtt{hopcount}_{rr,n_H}).$
$\mathtt{Bid}_{rr,n_H}{+}{+}.$
$\bar{c}_\infty\langle(\mathtt{rreq}, \mathtt{Bid}_{rr,n_H}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H}), \mathtt{self\_loc}\rangle))).Q2$

rr'_H discovers that it has got the information for rr, then it does not need to propagate the request, but it can directly generate the RREP packet, and send it (via unicast channel) to rr:

$P'''_{rr'_H} = \bar{c}_{k3}\langle(\mathtt{rrep}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H} +$
$\mathtt{hopcount}_{rr'_H,n_H}), \mathtt{self\_loc}\rangle.Q2$

Let v' be the message answering to the request in v:

$$v' = (\mathtt{rrep}, (rr, seq\#_{rr}), (n_H, seq\#_{n_H}), \mathtt{hopcount}_{rr,n_H} + \mathtt{hopcount}_{rr'_H,n_H}), k1'$$

rr is placed inside the transmission cell of rr'_H, then it will be able to receive correctly the RREP, and the processes executed will evolve:

$$rr[P'_{rr}]^{\mathtt{s}}_{k3,r_{k3}}|rr_H[P'''_{rr_H}]^{\mathtt{m}}_{k1',r_{k1'}} \xrightarrow{c_{k3}!v'[k3,r_{k3}]} rr[Q_1]^{\mathtt{s}}_{k3,r_{k3}}|rr_H[Q_2]^{\mathtt{m}}_{k1',r_{k1'}}$$

Fig. 24: Network condition after the application of the AODV protocol

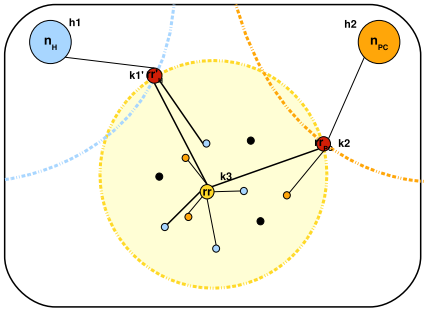


Figure 24 depicts the situation after messages exchange between rr and n_H in order to correctly update their route table. The updated route table are following reported:

RT_{n_H}

$(rr,1)$	k1'	2	$\emptyset$

RT_{rr}

$(n_H,1)$	k1'	2	$\emptyset$
$(n_{PC},1)$	k2	2	$\{k1\}$
$(rr_{PC},1)$	k2	1	$\emptyset$

In this example we used range repeaters in order to underline the properties of observability introduced with the CMN$^{\#}$. In this example, thanks to the theorems 7 (Radius of maximum observability) and 11 (Complete Range Repeaters), we can affirm that the mobile nodes moving inside the area damaged by the earthquake are always reachable by the central stations managing the rescuers (unless one of the range repeaters suffers a hardware or software damage).

7 Related and future work

This work is the result of an accurate bibliographic research, which allows us to consider and evaluate a number of calculi for the analysis of Mobile Ad-Hoc Networks. We have then chosen the CMN created by Merro [11] as a calculus to be extended, in order to join together all the positive characteristics of all the calculi we have studied. The reason why we choose CMN is the way it represents the topology of the network: Merro associates a location and a transmission radius to each node. Everytime a node broadcasts a message, it can be received only by the devices lying within the transmission cell of that node; this is the best solution for a good analysis of the observable actions of the nodes in the network. Nevertheless CMN did not result the best calculus for many other aspects, so we choose to modify it in order to increase the number of properties representable. By following the ω-calculus of Singh, Ramakrishnan and Smolka [22], we introduced the possibility of a node to transmit a message not only via broadcast, but also via unicast or multicast communication. Since Ad-Hoc

networks use radio-frequencies to manage a message communication to a finite and specific set of nodes, channels cannot be private. Even though a node cannot hide a message transmission, we believe that a formal declaration of the receivers set of a packet transmitted is any case useful. By associating a tag to each transmission, containing a set of nodes (message receivers), enabled us to prove some important properties concerning the observability of transmissions. For example we have defined the *Minimum Radius of Maximum Observability*, that is the minimum radius necessary to make a packet reachable by all its receivers: finding a way of calculating the minimum radius. which allows one to reach all the intended recipients of a message, is important for the problem of reducing power consumption without loosing network connectivity; moreover we have been able to prove that, under some specific conditions, stationary nodes lying in different locations might have the same behavior (this property in particular is a good starting point for the analysis of the *Location Confidentiality* [19]). in the original CMN a node was free to move arbitrarily within the network, but not to connect or disconnect. We chose to add some rule to allow all these actions, to give a more precise and realistic representation of a node's behavior. By adding these new rules we have made no particular modifications to the behavior of mobile nodes, because a connection or disconnection can be seen as a movement of the node inside or outside the network area, but these new rules are really important for the stationary nodes, because we allow them to arbitrarily connect and disconnect.

Future work

Merro, realizing CMN, has chosen to extend CCS, and not π-calculus, especially because CCS does not allow the creation of new names, then *scope extrusion* is not permitted. The *scope extrusion*, in the process algebra, is the ability of an entity to create a new name, using it to share a secret with another entity. This property allows the representation of cryptographic messages (if two nodes share a secret key), or, for calculi allowing channels transmission, it enables the representation of communication by means of a private channel. Nevertheless, introducing *scope extrusion* in a calculus, complicates the proof of all the properties about observability of the networks we introduced, then we choose to avoid the ability of nodes to create new names. It could be yet interesting an extension of CMN which allows *scope extrusion*. This modification will be important for the analysis of security problems, and in particular for the study of cryptographic communication between two nodes: the Mobile Ad-Hoc networks are actually used in situations that need data confidentiality, but packets transmissions cannot be made using private channels.

In the sixth section we used CMN$^{\#}$ to develop a routing protocol (AODV), often used in realization of mobile Ad-Hoc Networks. To simplify our work we used the parameter `hopcount`, that indicates the number of hops necessary to reach the destination, we will then consider as the best route to choose between two nodes, the one that is constituted of the minimum number of hops. This is the simplest, but not the best solution in mobile ad hoc routing: when dealing with

this particular kind of networks the most important parameter to consider in choosing a route is the power cost of the transmissions. MANETS are constituted of various devices, as notebooks and mobile phones, which have got different performances and power capacity, so power saving is one of the most important properties to guarantee. In managing messages transmission we have also to consider the different power capacities of the various devices: packets have to be sent in a way which allows any node to receive them. Recently some modifications to AODV protocol have been proposed, in order to minimize the power cost of transmissions. [2]. The standard protocol has been modified by adding fields, that indicate the sending and receiving power, so that we can evaluate the minimum power necessary to communicate with all the nodes of the network. Another consequence of the problem of power saving, is the selfish behavior of some nodes, that could refuse to cooperate with the network in order to save power. We propose to extend CMN$^{\#}$ by adding information about sending and receiving power of the devices of an Ad-Hoc network. A such extension might allow a more precise development of routing protocols and a deep and detailed analysis of the power saving in MANETS management.

Mobile Ad-Hoc Networks are wireless networks, realized through the protocol IEEE 802.11, which exploits radio frequencies to manage the data transmissions. These networks are constituted of *half-duplex* channels: each device cannot send and receive at the same time through a given channel, but it can execute only on action at a time. Consequently we need a procedure to gather, correct or avoid collisions. Massimo Merro, after the realization of CMN, together with Eleonora Sibilio, has proposed a new calculus for the study of wireless networks: the TCWS [12], where each action is associated to the instant of its execution. We can then manage the problem of processes synchronization and transmission execution time. Developping this new calculus Merro has considered the structure of wireless networks, which are based on MAC level (Medium Access Control) to avoid interferences and collisions in communications between the nodes. In particular, in order to reduce the number of collisions, there has been used a function of distributed coordination: CSMA/CA (Carrier Sense Multiple Action with Collision Avoidance) [23]. In TCWS devices are based on the CSMA scheme in order to allow transmission only when the channel results to be free. The calculus proposed has been developed considering networks with a static topology, but the authors affirm that an extension can be made in order to trait mobile networks; this modification could make the calculus adapted for MANETS modelling. In literature it is difficult to find calculi analyzing Ad-Hoc networks with temporal tags. Nevertheless it could be interesting a temporal analysis of MANETS, treating in detail the problem of interferences and collisions. For example, developping AODV, we have omitted the timeouts and the field `lifetime` of the packets transmitted, because there are no temporal tag associated to the nodes of the network. By adding a temporal tag we could obtain a more complete and precise representation of the networks we are interested to study.

References

1. Ad hoc on-demand distance vector routing protocol. http://moment.cs.ucsb.edu/AODV.
2. P. Barsocchi, N. Celandroniand E. Ferro, A. Gotta, F. Davoli, G. Giambene, F. J. Gonzlez Castao, J. I. Moreno, and P. Todorova. Radio resource management across multiple protocol layers in satellite networks: a tutorial overview. *IJSCN International Journal of Satellite Communication and Networks*, 23:265 305, 2005.
3. A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Verifying Persistent Security Properties. *Computer Languages, Systems and Structures*, 30(3-4):231–258, 2004.
4. S. Bucchegger and J.Y. Le Boudec. Cooperative routing in mobile ad-hoc networks: Current efforts against malice and selfishness. In *In Proc. of Mobile Internet Workshop*, 2002.
5. M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference? In *Peoceedings of the 5 Symposium on mobile Ad-hoc Networking and Computing*, volume 623, pages 9–19, 2004.
6. Ian D. Chakeres and E. M. Belding-Royer. Aodv routing protocol implementation design. In *In Proc. of 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, volume 7. IEEE press, 2004.
7. Tsu-Wei Chen and Mario Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks, 1998.
8. R. Focardi and S. Rossi. Information flow security in dynamic contexts. *Journal of Computer Security*, 14:65–110, 2006.
9. J.C. Godskesen. A calculus for mobile ad hoc networks. In *Proc. of the 10th Int. Conference on Coordination Models and Languages (COORDINATION'07)*, volume 3653 of *LNCS*, pages 132–150. Springer-Verlag, Berlin, 2007.
10. J. Haas. A new routing protocol for the reconfigurable wireless networks, 1997.
11. M. Merro. An observational theory for mobile ad hoc networks. *Information and Computation*, 207(2):194–208, 2009.
12. M. Merro and E. Sibilio. A timed calculus for wireless systems. Research Report 75/2009, Department of Computer Science, University of Verona, 2009.
13. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
14. R. Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.
15. S. Nanz and C. Hankin. A framework for security analysis of mobile wireless networks. *Theoretical Computer Science*, 367(1):203 – 227, 2006.
16. C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance vector (dsdv) for mobile computers, 2004.
17. K. V. S. Prasad. A calculus of broadcasting systems. *Science of Computer Programming*, 25(2-3):285–327, 1995.
18. V. Narasimha Raghavan, T. Peer Meera Labbai, N. Bhalaji, and Suvitha Kesavan. Extended dynamic source routing protocol for the non co-operating nodes in mobile adhoc networks. *International Journal of Applied Mathematics and Computer Sciences*, 3(1):12 – 17, 2007.
19. V. Narasimha Raghavan and Suvitha Kesavan T. Peer Meera Labbai, N. Bhalaji. Extended dynamic source routing protocol for the non co-operating nodes in mobile ad-hoc networks. *International Journal of Applied Mathematics and Computer Sciences*, 3, 2002.
20. R.Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. of International Colloquium on Automata, Languages and Programming (ICALP'92)*, volume 623 of *LNCS*, pages 685–695. Springer-Verlag, Berlin, 1992.

21. E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. *In Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999.
22. A. Singh, C.R. Ramakrishnan, and S.A. Smolka. A process calculus for mobile ad hoc networks. In *Proc. of the 10th Int. Conference on Coordination Models and Languages (COORDINATION'08)*, volume 5052 of *LNCS*, pages 296–314. Springer-Verlag, Berlin, 2008.
23. Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 2003.
24. Ieee 802.11 official website. http://www.ieee802.org/11.
25. L. Zhou and Z.J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 1999.