



Proceedings of the Combined workshop on
Self-organizing, Adaptive, and Context-
Sensitive Distributed Systems
and
Self-organized Communication in Disaster Scenarios
(SACS/SoCoDiS 2013)

Evaluation of self-organizing communication mechanisms within a
communication platform for disaster management

Volkmar Schau, Christian Erfurth and Wilhelm Rossak

12 Pages

Evaluation of self-organizing communication mechanisms within a communication platform for disaster management

¹Volkmar Schau, ²Christian Erfurth and ¹Wilhelm Rossak

¹Friedrich-Schiller-University Jena, Department of Computer Science
Ernst-Abbe-Platz 2-4, D-07740 Jena, Germany
{volkmar.schau|wilhelm.rossak}@uni-jena.de

²University of Applied Sciences Jena
Carl-Zeiss-Promenade 2, D-07745 Jena, Germany
christian.erfurth@fh-jena.de

Abstract: The request for IT support in rescue missions is getting stronger. In complex situations like Mass Casualty Incidents it is hard to get a good situation report which is essential for proper decisions. The requirements on IT are manifold and the development of reliable systems with networked devices is difficult and long. Additionally the evaluation of a system or its components cannot be limited on tests in some training. This paper describes an evaluation of the communication layer within a prototypical rescue management system developed by the SpeedUp project. The evaluation is done with the help of a simulation where professional and technical aspects are taken into account. Different classical routing mechanisms are compared with an approach using mobile software agents.

Keywords: Mass Casualty Incidents, Rescue Management Systems, Mobile Software Agents, Autonomous Communication

1 Motivation and Scenario

Imagine the following scenario: You are waked up by a phone call at 3 o'clock in the morning. You are the chief emergency doctor at call. The control centre is calling. In a wooded area a train accident is reported. This is the only information you got. An InterCityExpress train is crashed with an oncoming local train probably. The place of accident is not known exactly. The approach to the place is managed by the control centre. It is raining since yesterday evening.

During the approach you can call the organizational chief of the rescue service with your cellular merely. The information he has is not better. He has tried to reach the first approached emergency doctor by phone without success - perhaps busy.

A look at the map shows the place of accident is reachable by passing small roads only. A multitude of private cars, police cars, and cars of fire workers block the roads already. An agreement on the parking area with the directors of operation from the police and the fireworks is mandatory. The closer you get to the accident place the more exciting, run around people you encounter. They will be illuminated by the headlamps. It is dark outside.

The last 200 m you have to walk since vehicles and people block the road. The exact situation is not yet reported to you. From the length of the two trains the dimension of the accident area can be estimated of about 600 m roughly. Due to the local environment you are not able to get a better overview of the area by yourself. In your field of vision you see some crashed railway cars partly 20 m beside the track.

This sample scenario of a Mass Casualty Incident (MCI) will lead us through this paper. In situations as described above one can easily follow that the earliest possible knowledge of the situation in general is essential for proper decisions. Structured and coordinated handling requires (1) data collection, (2) information forwarding and (3) hierarchical organization [SEE+11]. Fire workers, police, and rescue service units often drive from different directions to the area of operation. A combination and consolidation of every single information part during arrival would be extremely helpful.

Nowadays information exchange is typical done with the help of IT systems. An IT support of forces in rescue missions is challenging:

- The multitude of tasks from rescue persons within different roles,
- diverse organizations with differing structures and cultures
- regional distinctions due to Germany's federalism,
- the compatibility of technical solutions with processes and persons,
- and special requirements regarding the technical equipment,

as well as further aspects of the equipment usage (e.g. usability, communication limitations) result in an extensive and complex set of requirements. However, in some cases small solutions lead to huge improvements already.

2 The SpeedUp Demonstrator

The conceptual design of an IT based communication system for data exchange even across organizational borders has been the focus of the SpeedUp project¹ [Spe13]. Industry partners, police, fire workers, and emergency rescue service have collaborated with research partners to develop a proof of concept. The project has been supported by a social science research team to discover differences between the organizations and to investigate communication processes with the goal to identify points for a useful integration of IT [GKM+11, WMY11]. For the objectives within the project a 3-layer approach has been the working basis (see Figure 1). *SpeedUp Scenario* is the topmost layer which defines 2 scenarios with different challenges for the proof of concept: Within MANV Scenario a mass casualty incident is considered. The GuT Scenario is placed within buildings and tunnels. *SpeedUp Practice* deals with organizational structures, communication flows, cultures of organizations, etc. The lowermost layer *SpeedUp Technology* puts the focus on the development and integration of emerging technologies to fit the defined requirements from above layers.

¹ The work is part of the SpeedUp project which is funded within the German Federal Government's program "Research for Civil Security" (call "Rescue and protection of people") by the Federal Ministry of Education and Research (duration: 1 May 2009 - 30 April 2012).

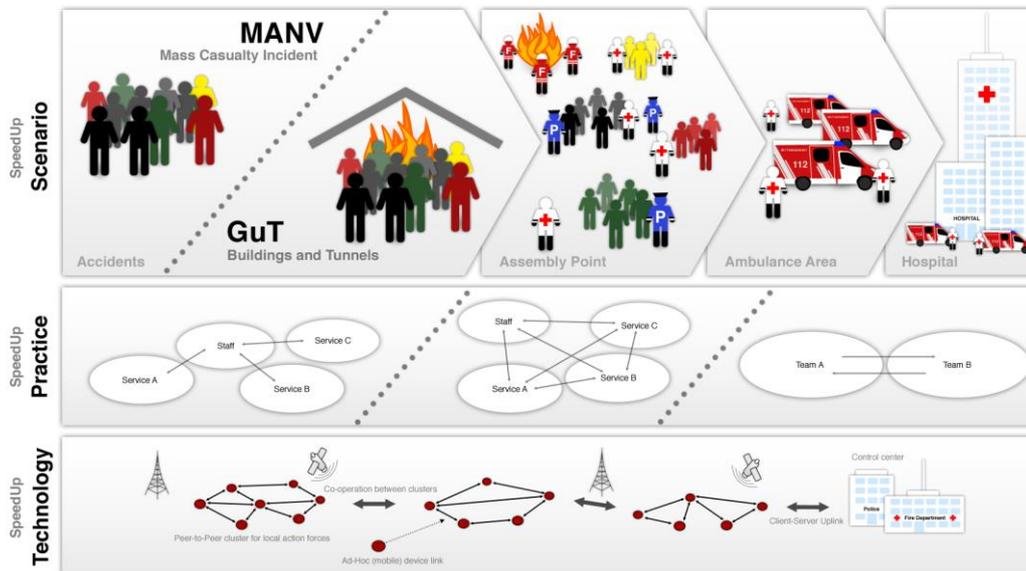


Figure 1: Overview of SpeedUp layers with different working focus.

On the technology layer one result is the SpeedUp Demonstrator (see left side of Figure 2) – a software system which includes the following functionalities in an experimental way:

- Triage of injured persons (support the triage process, store and forward data)
- Map information
- Localization of rescue team members
- Management of forces and vehicles
- Forces communication (chat)
- Mission order hierarchy
- Regional policy
- Mission log
- Access to relevant data
- Wireless data exchange

The demonstrator can be used on different mobile devices. The set of features and the presentation at the GUI depends on the role of the acting person. Whereby the user of a device shall not be stressed with underlying communication aspects (e.g. exchange of data, infrastructure). One objective of the project was to manage basic communication in a self-organizing manner.

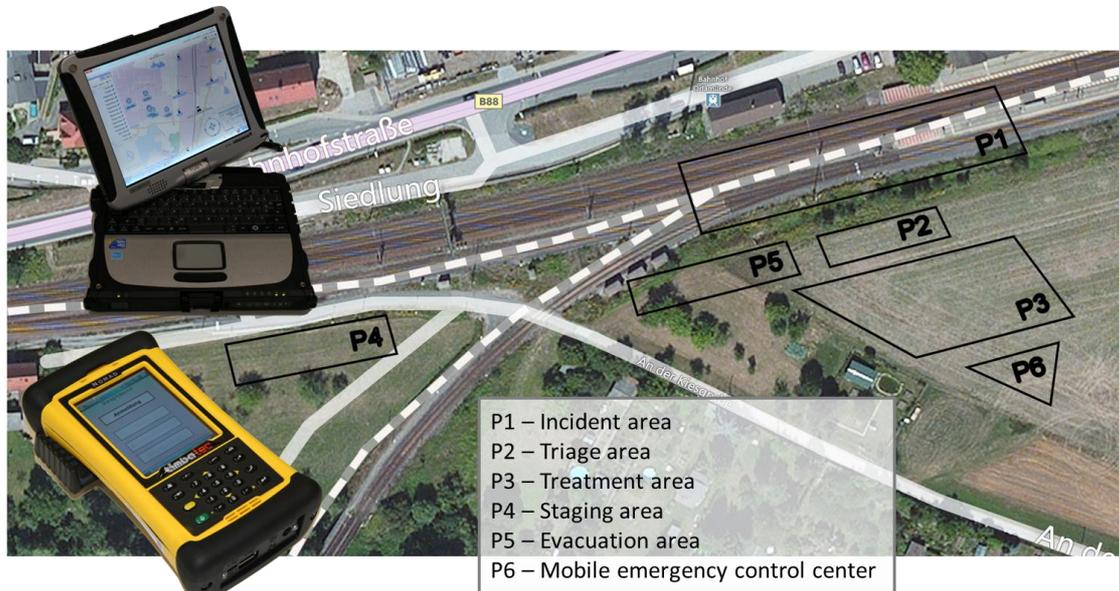


Figure 2: SpeedUp Demonstrator on different devices and an aerial view (taken from Bing) from the sample scenario.

The SpeedUp Demonstrator has been tested and stepwise adapted in different rescue practices and during events with rescue service safeguarding. However, a real evaluation is hardly to achieve. Trainings for rescue forces are good to be prepared but real situations are more challenging. So how can reliability and robustness of such a special system are evaluated? Tests only in the lab do not cover environmental effects. From field tests a very good feedback could be derived using the collected data. But the demonstrator could not be used in real field test (in regular missions) since it is not yet approved. Hence, one way for an evaluation of such a system is a simulation on basis of a good simulation model. In this paper the simulation focusses on data transmission while acting persons move with their devices during the mission. Therefore a sequence of the mission (the scenario from above) is prepared for the simulation.

3 The Simulation

For the simulation a set of rescue force actions has been arranged like in a storyboard. In addition typical tasks of emergency service and fire workers with relevant actions has been collected and some of them integrated in the simulation system. In this way the professional part of the simulation has been designed. For the technical part the storyboard was supplemented with coordinates of rescue forces and movement information as well as connectivity information.

The exchange of data between devices is message based using mobile software agents [BR05]. Mobile software agents go along with the data package and try to route the package while collecting and exchanging environmental and professional information with other agents on devices. We assumed that there is no reliable global communication infrastructure available. Devices try to exchange data (agents) e.g. via WLAN or Bluetooth when other devices are in

communication distance. Within the simulation different communication mechanisms are compared while the scenario procedures identical.

Based on the detail scenario in section 1 the paper describes in the next part a general view on MCI. The first layer outlines all action forces, processes and specific subtasks around - so called professional layer. For all action force's infrastructure a second level - calling technical layer - is set up.

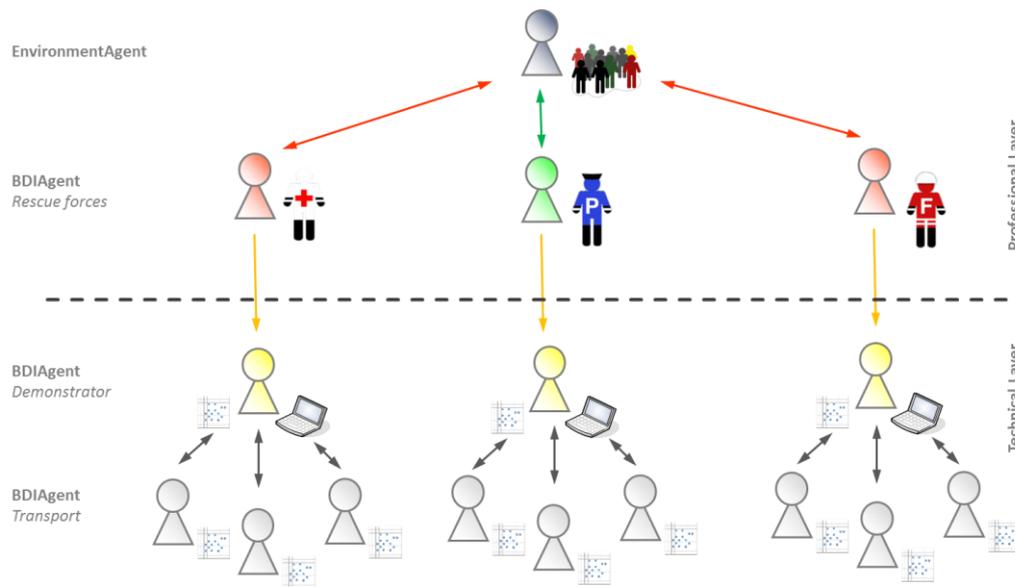


Figure 3: professional and technical layer

Figure 3 gives an overview about the context of professional and technical layer. We act on the assumption that parts of action forces have a mobile device. For inter-communication of mobile device our model uses mobile agents.

Especially, a MCI site is structured in several places (see Figure 2) with well-known responsibilities. In Figure 4 we depict a schematic drawing based on the scenario Figure 2.

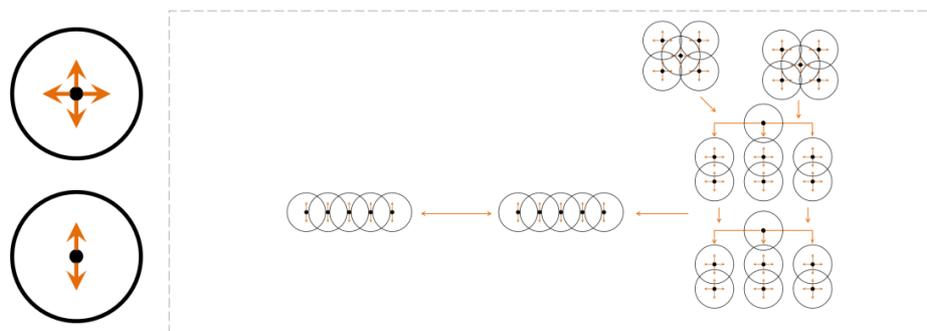


Figure 4: schematic areas from scenario (based on Figure 2)

Thereon, we see a distribution of action forces working in several places as well as the freedom degrees for each member. In P1 there is an incident area. All action forces are spread in scattered configuration caused by the specific disaster. P2 is called triage area. All person injured from the incident area are handed over to medical units for classification according their injury (triage). This place is outside of the incident area. According the classifications as to the further medical handling patients are transferred to different parts of treatment area (P3) labeled by special (triage) colors. The treatment area serves as patient-centered care attaining a transportable state. Reaching that state a patient is ready for transfer into a hospital. The transfer starts from evacuation area (P5). A standby rescue vehicle comes from the staging area (P4) picking up the patient in P5.

By comparing incident area (P1) with triage (P2) and treatment area (P3) we see different arrangements. For the simulation model P2 and P3 have a similar configuration caused by structured settings. A similar approach we find in evacuation (P4) and staging area (P5). All rescue forces and vehicles are organized in a regular line.

Based on the previous simplification our simulation model is transformed into the EVOS simulation system [Mul12, SKS+12]. EVOS is an agent-based discreet event simulation system able to simulate several mission scenarios. Parts of the simulation system are three types of dedicated agents: (1) SimulationAgent, (2) EnvironmentAgent and (3) BDIAgent. Starting with the SimulationAgent that agent is responsible for simulation supervision and open loop controlling. The EnvironmentAgent re-enact a simulation environment. Environment elements are vehicles and patients acting as passive parts of the simulation model. Action forces are represented as active objects mapped by BDIagents. Thereby, a simulation consists of 1+1+N running agents (SimulationAgent + EnvironmentAgent + BDIAgent).

First step to run a simulation is done by starting a SimulationAgent. As a central element the SimulationAgent is responsible to start and finish a simulation run. Additionally, he also manages all BDIagents representing active objects. Let's assume there is a new member of action forces supporting the incident. For the simulation the SimulationAgent starts one BDIagent per new member of action forces. The EnvironmentAgent simulates a specific location by managing simulation time, environment state, active objects and defined events. For example the correlation between mobile device connections and data transfers at the technical layer is expressed by events.

All active objects (forces, demonstrator devices and transport agents) are represented by BDIagents. Any specific role is set up by own plan and action libraries plus special beliefs about known environment elements. Plan library contains role depended actions and actions rules. Demands on how to run an action are expressed by action rules. When an action is prepared to fire a specific action implementation is done by action libraries. An action library models elementary basic functions. A plan library structure is given in Figure 5.

A plan library affects the role for a BDIagent within a simulation. Any role is defined by specific goals and plans which are parts of a concrete plan library. A goal defines role's objectives and plans how to reach a goal. In Figure 5 there is a general plan structure (plan

library). Each plan has a distinct name. Achievable goals for a plan define the statement „<achievableGoals> .. </achievable>“ which assigns objectives for a plan. Whether a plan is able to run it is checked by an action rule modeled by “<precommand> .. </precommand>” statement. Depending on BDI Agent’s beliefs this rule term fires and the plan is potentially running respectively.

```

1 <planLibrary name="ParamedicTransportPlanLibrary">
2   <goal>
3     <abstractGoal name="transport">
4       <priority>10</priority>
5     </abstractGoal>
6   </goal>
7
8   <plan name="driveToLocation">
9     <achievableGoals>
10      <abstractGoalName>
11        atTheLocation
12      </abstractGoalName>
13    </achievableGoals>
14    <precommand>
15      <!-- pre-condition -->
16    </precommand>
17    <priority>10</priority>
18    <goal>
19      <abstractGoal name="definedDirections">
20        <priority>10</priority>
21      </abstractGoal>
22    </goal>
23    <postcommand>
24      <!-- postcondition -->
25    </postcommand>
26  </plan>
27 </planLibrary>

```

Figure 5: Example for transport plan library

For some goals there is more than one plan. In a given situation a selection of the proper plan is done by current beliefs (derived from situation and environment). The choice of plans is then managed by a priority function “<priority> .. </priority>” that ranges from a numeric number to a complex formula. In general plan libraries distinguish between abstract goals and action goals. Abstract goals are resolved into plans as mentioned before.

Action goals are not broken down into plans. These goals run elementary basic functions which are parts of agent’s role. Transport is one example depict in figure 6 (in a higher stage to catch the steps and conditions broken down into plan and action libraries). As an elementary basic function it means to attend to a specific place on site (environment). Such basic behavior is deployed inside agent’s action library. Inside a plan library the term “transport” exists both as abstract goal and action goal.

Task	Action	Communication	Rescue force
transport <patient> to patient positioning place	<i>[patient not transportable]</i> ensure_transportability <patient> <i>[Patient transportable]</i>		
	<i>[<patient> transportable]</i> organize_orderly <patient> <patient positioning place> <i>[enough <orderlies> available]</i>		
	<i>[<patient> transportable & enough <orderlies> available]</i> give_command <orderly> „ transport <patient> to <patient positioning place>“ <i>[<orderly> and <patient> arrived]</i>	transfer command, transfer confirmation	paramedic, rescue aid man
escort <patient> to patient positioning place	<i>[patient can go alone]</i> walk with <patient> to patient positioning place <i>[arrived]</i>		

Figure 6: Example for transport task

3.1 Professional Layer

The professional layer contains all elements around all action force's work like positioning, movement and tasks on the site. Based on simulation structure mentioned before the layer covers all BDIagents as rescue units (active objects) which are related to the acting of rescue forces and hierarchies to solve the situation (we refer to figure 1 of [SSE10] by combining the two layers "SpeedUp Szenario" and "SpeedUp Praxis").

3.2 Technical Layer

The technical layer includes the entire technical IT-infrastructure used by rescue forces. In the context of the SpeedUp project parts of rescue units are equipped with mobile devices (hard- and software demonstrators). All collected data can be entered by mobile device and distributed to all other nodes. Inter-mobile device communication is done by mobile agents [BR05, NSR12]. Therefor exists for our simulation model two entities: (1) demonstrator agent representing a hard- and software demonstrator and (2) transport agent responsible for inter-mobile device communication. A demonstrator agent is able to accept data entered by rescue forces (inside the simulation located on the professional layer) as well as transport agents coming from other nodes. For new data collected from rescue forces a demonstrator agent hands over information to a transport agent who transfers the data to other nodes.

3.3 Simulation procedure

The sequence of actions (incoming and leaving rescue units, performed actions, etc.) – the formalized storyboard (O-1bc in Figure 7) – is used as an input from the professional layer for the simulation run. Derived from the scenario described by the actions is the generalized MCI scenario from Figure 4 which serves as another pre-condition for the simulation. It contains both, positions and movement vectors of rescue forces' communication devices (nodes) (I-1ab, Figure 7) over a time of 1000 sec. Each node sends regularly messages to be distributed to its neighbours to reach all nodes. The sending interval between two messages is 1 sec, which allows us to distribute 1000 information per node. These characteristics are captured by the *Network Simulator ns2* (I-2b) in stage I. Within stage I the positional relationship of nodes is discovered. So communication relationships over the chosen time period are pre-calculated by ns2.

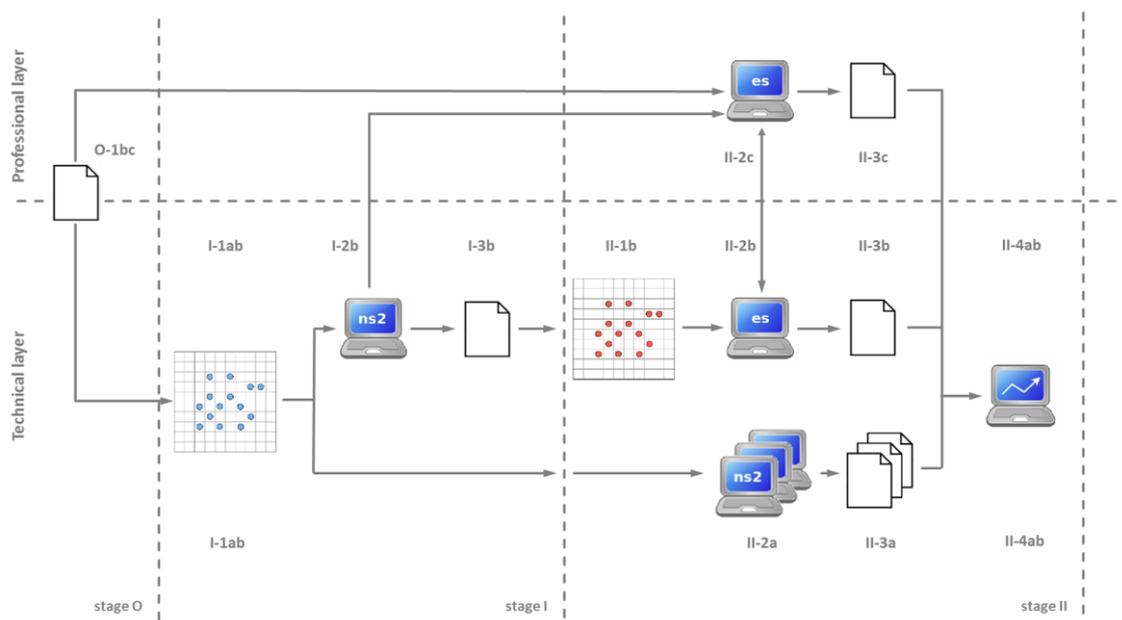


Figure 7: Three-stage-two-paths simulation procedure.

For our purpose we consider two ways in stage II. The *Network Simulator ns2* (II-2a, Figure 7) is used to simulate classical routing protocols (Destination Sequenced Distance Vector – DSDV [PB94], Ad Hoc On-Demand Distance Vector – AODV [PR99] and Dynamic Source Routing – DSR [JM96]) and node movements (line II-2a up to II-3a) based on the position and movement vector data (I-1ab). The *Ellipsis multi-agent system es* [Mul12] (II-2b, Figure 7) examines agent based routing (line I-2b up to II-3b) by *mobile shared map and cloning agents (SMAC)* [SEE+11] based on pre-processed node connectivity (II-1b). The pre-processed data provides the direct connectivity between two nodes generated from the ns2 run in stage I (I-2b up to I-3b). For the simulation 50 % of the connections are regarded as not successful i.e. the data transmission is not finished.

The professional part of the simulation with the Ellipsis multi agent system exchanges data with the technical layer. Each time when an action from the storyboard (O-1bc) leads to a data input by a rescue service member (II-2c) the data packet is overhanded to the technical layer for transmission in the simulation (II-2b). The overall result of the two-way run is then consolidated and analysed (II-4ab) by utilizing the trace files (II-3a, II-3b). From the professional layer log data (II-3c) is also generated.

3.4 Simulation Results

To compare different classic routing protocols with the SMAC approach, simulations of different dynamic levels were set up. The results for a different number of rescue people (with less than, exactly, and more than 40 devices in the MCI scenario) are compared in Figures 8. The ordinate shows the completeness of information distribution within the entire MANET as well as dedicated operation areas (y-ordinate in %).

The agent approach works inefficient within sparse areas, e.g. a staging area, for the simulations due to its large migration overhead. In the context of a highly-dynamic scenario missing alternative paths are a heavy drawback for agents, if compared to other routing protocols. Data packets of routing protocols are using the limited time frames for existing links more efficient due a smaller packet overhead. Therefore, more packets are able to pass.

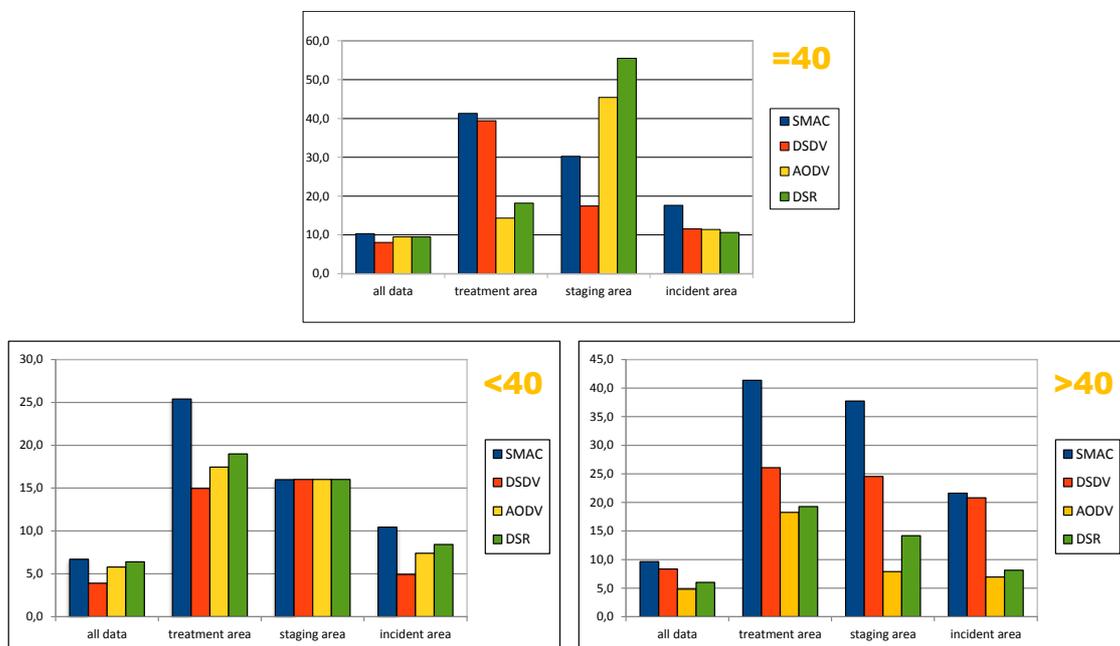


Figure 8: Robustness and Sufficiency for dynamics with less than, exactly, and more than 40 devices.

The incident area with more devices shows similar behaviour of agents and routing as in the staging area, while agents are more successful at dense node distribution. In well-structured

areas, like the treatment area, the advantage of agents is clearly visible. High node density and many different stable link options support even a very large amount of data. Taking all data into consideration, a slight advantage of the agent approach can be recognized (all data, Figure 8).

4 Conclusion

The development of such specialized systems like rescue management systems is challenging. Beside other aspects robustness and reliability are essential to reach a productive state of such systems. This cannot be tested in training missions solely. In this paper we have presented a simulation for the communication layer of a rescue management system on basis of the network simulator ns2 combined with the Ellipsis multi agent system. We compared different routing mechanisms to get a relation between traditional routing protocols in ad-hoc networks and our mobile agent based approach SMAC. The simulation has shown that in areas of an MCI with more connected devices SMAC is more efficient than traditional techniques. In sparse areas traditional techniques work more efficient due to smaller data packages. However, the agent has further potentials in routing packages since the professional layer can be taken into account for instance the typical distribution of rescue peoples in an MCI and movement patterns of types (or roles) of rescue peoples or packages could be prioritized on basis of the type of data package (triage data, order, etc.).

Acknowledgment

Many thanks to Dr. Krohn for giving an insight in rescue missions by providing us with the scenario description. Thanks to Stefan Hellfritsch for preparing the simulation and Sebastian Scharf for the simulation runs and the measurements.

References

- [BR05] Peter Braun and Wilhelm Rossak: *Mobile Agents: Basic Concepts, Mobility Models and the “Tracy” Toolkit*. Morgan Kaufman, 2005.
- [GKM+11] Aygul Gabdulkhakova, Birgitta König-Ries, Mareike Mähler, Yeliz Yildirim-Krannig and Fabian Wucholt: *Identifying and Supporting Information Needs in Mass Casualty Incidents – an Interdisciplinary Approach*. Proceedings of the 8th International ISCRAM Conference. Lisbon, Portugal, 2011.
- [JM96] David B. Johnson and David A. Maltz: *Dynamic Source Routing in Ad Hoc Wireless Networks*. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.
- [Mul12] Multiagentsystem Ellipsis sourceforge Homepage.
<http://sourceforge.net/projects/masellipsis/> [Online, accessed 13.02.2013]

- [NSR12] Phuong T. Nguyen, Volkmar Schau, and Wilhelm Rossak: An Adaptive Communication Model for Mobile Agents inspired by the Honey Bee Colony: Theory and Evaluation. Proceedings 10th European Workshop on Multiagent Systems (EUMAS), Dublin, Ireland, 2012.
- [PB94] C.E. Perkins and P. Bhagwat: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. ACM, pp.234 – 244, 1994.
- [PR99] Charles E. Perkins and Elizabeth M. Royer: Ad Hoc On-Demand Distance Vector Routing. In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, 1999.
- [SEE+11] Volkmar Schau, Christian Erfurth, Gerald Eichler, Steffen Späthe and Wilhelm Rossak: Geolocated Communication Support in Rescue Management, Proceedings of the 8th International ISCRAM Conference. Lisbon, Portugal, 2011.
- [SKS+12] Volkmar Schau, Kathrin Kirchner, Steffen Späthe, Sebastian Scharf, Stefan Hellfritsch, Gerald Eichler, Christian Erfurth, Wilhelm Rossak and Jens Reichel: Simulation of Rescue Force Communities in Mass Casualty Incident Situations. Proceedings 10th International Conference on Innovative Internet Community Systems (I2CS), Trondheim, Norwegian, GI LNI, Volume P-204, 2012.
- [SSE10] Volkmar Schau, Steffen Späthe and Gerald Eichler: SpeedUp: Mobile und selbstorganisierende Kommunikations- und Datenplattform für komplexe Großlagen, Tagungsband zum 7. GI/KuVS-Fachgespräch „Ortsbezogene Anwendungen und Dienste“, Logos-Verlag, Berlin, 2010.
- [Spe13] SpeedUp Consortium: SpeedUp Homepage. <http://www.speedup-projekt.de/>
[Online, accessed 13.02.2013]
- [WMY11] Fabian Wucholt, Mareike Mähler and Yeliz Yildirim-Krannig: Framework conditions and fields of application of an IT-Rescue Management Support System (IT-RMSS) for authorities and organizations with safety responsibilities (BOS) in mass casualty incident (MCI). In: Lecture Notes in Informatics (LNI) – Proceedings. Informatik 2011, Workshop IT-Unterstützung für Rettungskräfte, Berlin, 2011.