

Implications of Communication Constraints for the DLRP in Transport Logistics

Bernd-Ludwig Wenning, Christian Zabel, Henning Rekersbrink, Carmelita Görg, Bernd Scholz-Reiter

Autonomous control of logistic objects such as vehicles and goods in transport processes means that such objects act independently to find their best way through a logistic network. In order to make reasonable decisions, they have to interact with other objects, e.g. to determine the availability of transport demand or free transport capacity along a possible route. Interaction means the use of information and communication infrastructure, for example by using the Distributed Logistic Routing Protocol (DLRP) that was developed for this purpose.

The physical infrastructure, however, has direct technical constraints such as limited communication bandwidth or storage capacity and indirect constraints that are caused, for example, by the communication costs. This article discusses some of these constraints and their implications for the implementation and operation of DLRP in transport logistic processes.

Introduction

Autonomous control of logistics objects [3] is a new paradigm that changes the way the logistic challenges are handled, away from a centralized planning and dispatching towards considering the logistic objects as autonomous entities that determine their own path through a logistic network. In transportation logistics, this means that the logistic objects such as vehicles and goods become capable of making their own decisions about their route and taking action to follow them. As an autonomous control approach, the Distributed Logistic Routing Protocol (DLRP) has been proposed. Logistic objects that implement this approach use it to collect necessary information about the current status of the logistic network to be able to make sensible decisions. The DLRP specifies the messages and their flows between the logistic objects.

Naturally, resources such as bandwidth, energy or memory are not unlimited, and such limitations impose constraints to the communication and interaction of autonomous logistic objects. Communication costs impose another constraint. Therefore, it is necessary to investigate the use of autonomous control concepts such as DLRP under the presence of these constraints to identify the challenges that are encountered. This article focuses on the aspect of how much traffic is gen-

erated on the communication interface of an autonomous logistic object that uses DLRP as its autonomous control method.

Related work: The DLRP

As a concept for autonomous control of logistic objects, the Distributed Logistic Routing Protocol (DLRP) [4], [7] has been proposed. It is a decentralized routing method that is based on the assumption that the logistic objects (vehicles and goods) are equipped with devices capable of computing and communicating, or they are represented by software agents acting on behalf of them in a multi-agent system. Thus, there are either physical devices or software agents that enable the logistic objects to interact and decide autonomously. In addition to the vehicles and goods, the vertices (e.g., logistic distribution centers) also become participants in this interaction.

In contrast to the classical logistic routing scenarios where heuristic methods are applied to solve static optimization problems such as the Vehicle Routing Problem (VRP) or the Pickup and Delivery Problem (PDP), the DLRP and the scenarios where it is applied are different. First, the DLRP is not a concept to solve static planning problems and create a plan before the transportation process is executed, but it is a control concept that makes real time decisions during transportation process. This way, it is easier to integrate new transport orders into the ongoing process as and when they appear. Further, the scenario topologies are restricted on existing connections between locations (vertices) in the network to better represent existing infrastructure. Scenario topologies are not only defined by a set of vertices, but also by a graph connecting those. The example scenario described later illustrates this. The vertices within such topologies represent, e.g., logistic distribution centers, and the edges represent major road connections between them.

Vehicles and goods that use the DLRP determine their routes individually and subsequently match with goods and vehicles that share the same route. The routing is done by using a route discovery messaging that is similar to source routing methods in ad-hoc communication networks: The vehicles/goods send out a route request to the nearest vertex, which forwards it to the neighbor vertices, which in return do the same. Each vertex adds local knowledge about the current network status and transport demand to the request, so that by the time when the request reaches the destination vertex, it has collected information about the complete route that it has travelled. The destination vertex sends a reply to the vehicle or the good, which then can make a decision. After having made a decision, the vehicles and goods announce their intended routes to the involved vertices, where they can be used to create the relevant information for route discoveries from other vehicles and goods. Therefore, the vertices can be considered to act as informa-

tion brokers. This interaction among vehicles, vertices and goods is depicted in Figure 1.

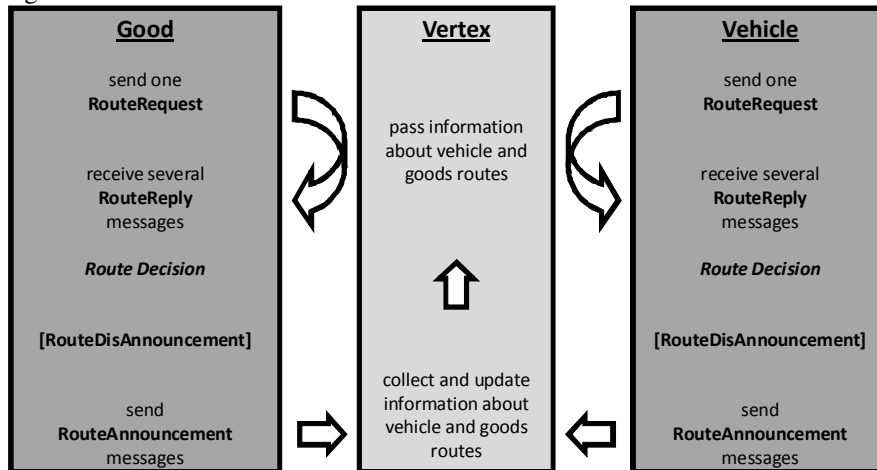


Figure 1: Interaction among goods, vertices and vehicles in DLRP [4]

Four main message types are present in DLRP:

- Route requests (route request messages), being sent to discover routes
- Route replies (route reply messages), reply messages returned by the destinations
- Route announcements, being sent to publish route decisions
- Route disannouncements, being sent to cancel route announcements when a decision is changed

Out of these four message types, the route request is the one that potentially has the largest contribution to the communication traffic created in DLRP as several route requests are forwarded among the vertices in the network in each route discovery.

Multi-criteria routing as it is done in DLRP significantly differs from ad-hoc routing protocols in communication networks such as AODV (Ad-hoc On-demand Distance Vector) [2] and DSR (Dynamic Source Routing) [1]: The quality of a route is not necessarily correlated with the sequence of arrival of route request messages at a specific location in the network. The sequence of route request arrivals just allows a statement about the communication path on which the route request has travelled, but not on the logistic route associated with it. Therefore, the common assumption in AODV that the first incoming route request is representing the best route, permitting to drop all the subsequent incoming route requests from the same route discovery, does not hold for logistic routing where the routed goods and vehicles travel in a network whose characteristics are different from those of the communication network that passes the routing messages. Consequently, multiple route requests may need to be processed and forwarded by a vertex during a single route discovery, and multiple route replies may need to be sent

back to the logistic object as well. This leads to a potentially high amount of routing messages.

Possible DLRP architectures

As mentioned before, the DLRP can be implemented in physical devices that are attached to the logistic objects, or in software agents that act on behalf of them. Different levels of distribution, and consequently, different architectures are possible, for example:

- Each logistic object is represented by an agent in a central agent system. All interactions among logistic objects and all decisions are being done within the agent system. The physical goods are only tagged with bar codes or RFID so that they can be tracked whenever they are handled (e.g., loaded or unloaded). The vehicles or their drivers receive the route decisions, e.g., on an onboard unit or handheld device. This setup has a low level of distribution as all decisions are being done on the agent system. Of course, it is also possible to have multiple interconnected agent systems, in which case the level of distribution is slightly higher.
- The maximum level of distribution would be implemented if each individual logistic object (goods as well as vehicles) is equipped with an individual device that interacts with other devices and makes decisions. Each of these devices acts according to the DLRP specifications.
- An intermediate level of distribution is implemented if vertices and vehicles run local agent systems. In this setup, the goods are represented by mobile agents that migrate between the agent systems. Decisions are made within the local agent systems on vehicles and vertices.

The different architectural options lead to different requirements on the technical side, such as high computational power of the agent system in the central agent system approach, and bandwidth requirements in the approaches with a high level of distribution.

In this publication, the focus is on the distributed approach where each logistic object carries its own device.

Message sizes

Constraints such as costs, bandwidth or memory requirements that may impose limitations on DLRP can be directly related to communication traffic. Therefore, the communication traffic created by DLRP has to be analyzed in terms of message numbers and message sizes. An indication on how much communication traffic is to be anticipated and acceptable is provided by the communication volume that current telemetry systems for trucks generate: Those systems, that usually communicate via GPRS, send data in the order of magnitude of some 100 Kilo-byte per month. If DLRP would generate significantly more traffic on the logistic object's communication link, this would probably not be considered economically reasonable. So the target volume would be in the range of several hundred Kilo-byte per month.

In the following, the sizes of the four mentioned DLRP message types are discussed. This discussion is based on the assumption that the logistic objects use a decision system as described in [6] and [8] with the multiplicative Multi-Criteria Context-based Decision function (MCCD) involving 3 appropriately scaled criteria.

The actual sizes of DLRP messages are dependent on implementation details and on how detailed the carried information should be. However, based on the minimal set of information that needs to be included, lower bound estimations are done in order to investigate traffic volumes that are created by DLRP. The following assumptions are made:

- addresses of logistic entities are 32-bit integers
- MCCD is used for the route decisions
- 3 context criteria are used in the decisions
- each criterion is described by an identifier, and there are weights and scaling functions for each criterion, as well as limits for acceptable ranges of each criterion
- 3 scaling parameters are required per criterion to describe the scaling function that is applied

Because of these assumptions, the calculated message sizes have to be regarded as examples. If more criteria or more complex scaling functions are used, the messages could become larger.

Route request messages

Route request messages have to contain at least the following information:

- vehicle/good flag to identify what type of logistic object is the origin of this message (boolean, 1 bit)
- DLRP message type (2 bits)
- sender address (integer, 32 bit)

- destination address (integer, 32 bit)
- sequence number (short integer, 8 bit)
- time to live (short integer, 8 bit)
- due time (integer timestamp, 32 bit)
- good size or vehicle capacity (floating point, 32 bit)
- hop list with multiple elements that consist of
 - hop address (integer, 32 bit)
 - expected arrival time (integer timestamp, 32 bit)
 - expected leave time (integer timestamp, 32 bit)
- context field, describing the three criteria and their values, with the following contents
 - context criteria identifier (3× 8 bit → 24 bit)
 - scaling type identifiers (3× 8 bit → 24 bit)
 - 3 scaling parameters per criterion (9× floating point, 32 bit → 288 bit)
 - context weights (3× floating point, 32 bit → 96 bit)
 - context limits, one for each criterion and one for the overall route metric (4× floating point, 32 bit → 128 bit)
 - context values (3× floating point, 32 bit → 96 bit)

When these numbers are summed up, the resulting message size is $803 + 96n$ bit, where n is the number of hops. Padding can be applied to create complete octets. Including the padding, the message size is $101 + 12n$ Byte. Of course, this is the size at application level, and lower layer protocol (e.g. TCP/IP) overhead is not included.

Route reply messages

If the destinations just copy the route request into a route reply, the message size remains the same, only the content of the message type field is changed, the time to live field is replaced by a hop count field and source and destination may be swapped. This, however, is not the optimum way as it can be assumed that the vehicle or package remembers with which parameters it has initiated the route discovery, so that the context field does not need to contain all information. If the route reply is cut down to the absolutely necessary information, the following contents remain:

- DLRP message type (2 bit)
- sender address (integer, 32 bit)
- destination address (integer, 32 bit)
- sequence number (short integer, 8 bit)
- hop count (short integer, 8 bit)
- hop list with multiple elements that consist of
 - hop address (integer, 32 bit)
 - expected arrival time (integer timestamp, 32 bit)
 - expected leave time (integer timestamp, 32 bit)

- context field with the following contents
 - context values ($3 \times$ floating point, 32 bit \rightarrow 96 bit)

In this case, the resulting message size is $178 + 96n$ bit, or (with padding) $23 + 12n$ Byte.

Route announcement messages

In the route announcements, the logistic objects inform a vertex about their planned arrival and departure. For this, the announcement has to contain at least the following fields:

- vehicle/good flag (Boolean, 1 bit)
- DLRP message type (2 bit)
- sender address (integer, 32 bit)
- announcement number (short integer, 8 bit)
- expected arrival time (integer timestamp, 32 bit)
- expected leave time (integer timestamp, 32 bit)
- next hop address (integer, 32 bit)
- good size or vehicle capacity (floating point, 32 bit)
- route preference (floating point, 32 bit)

The sum of these fields results in 203 bit, which corresponds to 26 Byte with padding.

Route disannouncement messages

A route disannouncement cancels a previous announcement, so it has to at least contain enough data to uniquely identify the announcement that it refers to. Therefore, its contents are:

- vehicle/package flag (Boolean, 1 bit)
- message type (2 bit)
- sender address (integer, 32 bit)
- announcement number (short integer, 8 bit)
- next hop address (integer, 32 bit)

This result in 75 bit, leading to 10 Byte with padding.

Communication volumes

In one route discovery, multiple route requests are propagating through the network. This amount is denoted as N_{RREQ} . However, only one of them is sent by the logistic object that initiates the route discovery, the others are forwardings among the vertices. The destination may send N_{RREP} route replies back to the logistic object. Additionally, route announcements are sent by the logistic object af-

ter the route decision (n announcements for a route with n hops), and each route is eventually disannounced with the same amount of disannouncements.

When the technical and cost constraints of communication are considered in a fully distributed architecture, the communication traffic has to be categorized in two categories: the first category is vertex-to-vertex communication. This can be considered to be done on wired broadband media. The second category is the communication between vertices and logistic objects. This would usually be done wirelessly, as the logistic object is mobile. Separating the route discovery traffic into these two categories results in the following:

Vertex \leftrightarrow Vertex:

- $N_{RREQ} - 1$ route requests

Vertex \leftrightarrow Logistic object:

- 1 route request
- N_{RREP} route replies
- n route announcements
- n route disannouncements

It can be seen that the amount of route requests mainly influences the vertex-to-vertex communication, while the other message types are only sent between logistic objects and vertices.

Previous work (e.g. [5], [8]) has shown that the route discovery in DLRP can create a large amount of communication traffic if no measures are taken to limit this traffic. However, the main focus there has been on the overall network load, where the route requests are the main contribution. Different approaches such as hop count limitation and intermediate route evaluation have been proposed to limit the amount of route requests.

Considering the categorization of traffic as shown above, the most constrained link is, however, most likely the logistic object's wireless link. Per route discovery, this link has to carry, assuming a length of n for all replied routes:

$$\begin{array}{rcl}
 & & 113 \text{ Byte} \quad (\textit{first hop route request}) \\
 + & & N_{RREP}(23 + 12n) \text{ Byte} \quad (\textit{route replies}) \\
 + & & 26n \text{ Byte} \quad (\textit{route announcements}) \\
 + & & 10n \text{ Byte} \quad (\textit{route disannouncements}) \\
 = & 113 + 23N_{RREP} + n(36 + 12N_{RREP}) \text{ Byte} & (\textit{total})
 \end{array}$$

It can be seen that the amount of route replies has a significant influence on the communication traffic that the logistic object has to handle. Therefore, an efficiently configured DLRP concept should not only limit the generated amount of route requests, but also the amount of route replies during a route discovery. The simulation results in the following section consider both.

Simulations

Previous work on the traffic volume in DLRP, as described in [5] and [8], utilized a scenario topology with 18 vertices based on a German road network. This scenario has now been extended to a topology with 40 vertices and a corresponding set of connections. This topology is still based on a highway network connecting German cities, but with a higher level of detail, as depicted in Figure 2. The vertices labeled as AD and AK are highway intersections which are neither sources nor sinks. All others are sources and sinks. The source and destination locations of transport goods are uniformly distributed among all those vertices, i.e. the mean rate of goods generation is the same in each vertex, and each vertex should also receive the same amount of goods. The overall generation rate is 25 goods per time unit. Each of the goods has to be delivered within 24 time units after its generation. An initial budget is assigned to each of the goods when it is generated. This initial budget is proportional to the shortest distance (in km) between source and destination. 20 vehicles, each with a capacity of 12 transport goods, are assumed to be present in the scenario. Each of these vehicles has the same capacity and travel at the speed of 100 km per time unit.

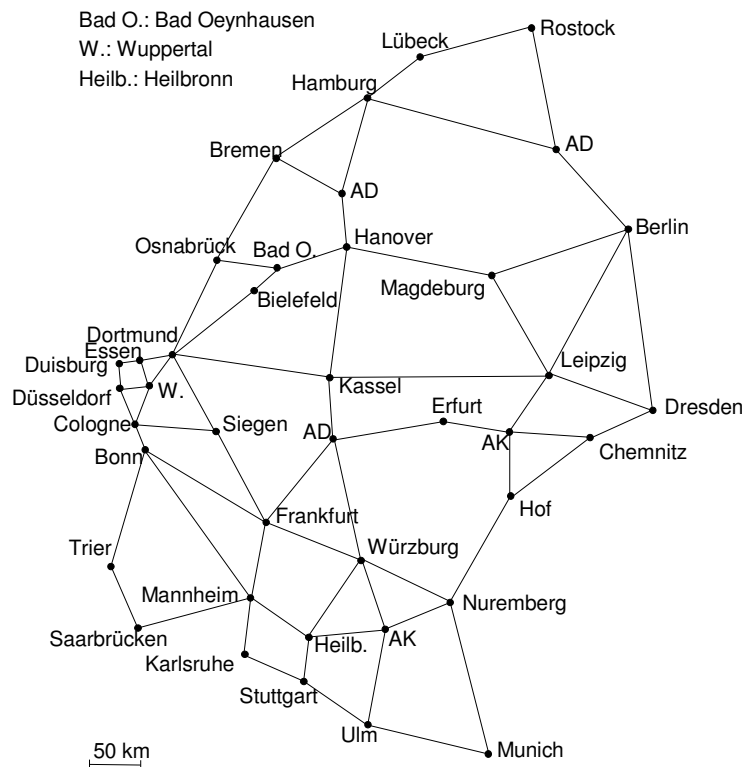


Figure 2: 40-vertex topology

In [5], two methods of restricting the amount of goods-initiated route requests were presented and investigated for the 18-vertex topology: forwarding restrictions based on intermediate route evaluation and the application of fixed, scenario-wide hop limits for the route request propagation. These fixed hop limits are now applied to the 40-vertex scenario. Additionally, the route request forwarding is also limited by the route costs. This limit is kept at 50% of the transport good's current budget. If a transport good does not receive any route reply after a timeout, the route discovery is tried once again with the cost limit dropped, but keeping the hop limit unchanged.

Hop limit	Delivered transport goods	Goods route discoveries	Goods route requests	Goods route replies	Goods route ann.
7	48565	$1.02 \cdot 10^6$	$8.64 \cdot 10^8$	$2.91 \cdot 10^6$	$2.10 \cdot 10^6$
8	49482	$6.43 \cdot 10^5$	$4.22 \cdot 10^8$	$4.78 \cdot 10^6$	$2.22 \cdot 10^6$
9	49479	$6.43 \cdot 10^5$	$6.33 \cdot 10^8$	$7.19 \cdot 10^6$	$2.23 \cdot 10^6$
10	49480	$6.38 \cdot 10^5$	$9.07 \cdot 10^8$	$9.61 \cdot 10^6$	$2.21 \cdot 10^6$

Table 1: Route request amounts with fixed, scenario-wide hop limits

Table 1 shows simulation results for this setup. Each row in this table shows average values of 5 simulation runs with different random seeds being used for the generation of the goods. In each simulation, 50000 transport goods were created. The simulation is stopped shortly after the generation of the last transport good. It can be seen that in this scenario, a hop limit of 8 hops is the lower limit. With a limit of 7 hops, less goods are delivered while the amount of route requests increases. The reason is that the network diameter is 8 hops, i.e. a minimum of 8 hops is needed for a connection between the most distant vertices in the topology. A hop limit that is greater than 8 just increases the communication traffic, particularly the amount of route requests propagating through the network and the amount of route replies sent back to the logistic objects, without much impact on the logistic performance, which is here measured by the amount of delivered goods and the vehicle utilization. So the hop limit of 8 is not only the minimum, but also the optimum if fixed hop limits are applied.

The next logical step is to introduce an adaptive hop limit. The adaptive limitation in the given case was chosen as follows: A transport good starts its route discovery with a small hop limit and the cost limit as described above. If the route discovery fails and a timeout occurs, the transport good initiates a retry, increasing the hop limit by one. In case of further failures, this is repeated until the upper limit of the hops is reached. In this setup, the upper limit was chosen to be 9 hops. This method is very similar to the expanding ring search done in AODV. The last retry is done without cost limit to ensure that a route will be found. For the results in the following table, 5 simulation runs were done again for each row, and the amount of created goods was 50000 again.

Hop limit	Delivered transport goods	Goods route discoveries	Goods route requests	Goods route replies	Goods route ann.
3 + retry counter	49987	$1.36 \cdot 10^6$	$5.94 \cdot 10^8$	$3.51 \cdot 10^6$	$1.78 \cdot 10^6$
4 + retry counter	49986	$8.56 \cdot 10^5$	$3.51 \cdot 10^8$	$2.29 \cdot 10^6$	$1.85 \cdot 10^6$
6 + retry counter	49820	$6.77 \cdot 10^5$	$2.96 \cdot 10^8$	$2.44 \cdot 10^6$	$2.13 \cdot 10^6$

Table 2: Results for adaptive limits

Table 2 leads to the following findings:

The trade-off between the number of trials and the communication volume per single discovery can clearly be seen. When the goods are starting with a short hop limit such as 3, they in average need many retries, which is visible from the amount of route discoveries. When they start with a limit of 6 or use a fixed limit of 8 as shown in Table 1, there is an overhead caused by increased volume in single route discoveries, some of which are unnecessarily long routes (as visible from the amount of route announcements). With respect to the amount of route replies, an initial hop limit of 4 has turned out to be the optimum, but it leads to slightly more route discoveries as some retries are needed to discover long routes.

Starting with a low hop limit leads to slightly more efficient goods routes, so that more goods have been delivered during the simulation time window. The reason for this is that a transport good makes a decision when it has received a certain amount of route replies without waiting for more. The lower limit eliminates long and inefficient routes, so that it is more likely that the transport good has received a reply with an efficient route within the hop limit.

For these reasons, the configuration with an initial hop limit of 4 is considered to be the optimum choice here. In this setup, the amount of messages still seems to be a high number, but this is the cumulative amount for all goods. Per good, this corresponds to 17.1 route discoveries, 45.8 route replies, and 37.0 route announcements (and the same amount of disannouncements). Based on above calculations, each of the route replies has 119 Byte or less as the network diameter is 8 hops. Each first-hop route request has 113 Byte, each route announcement 26 Byte and each route disannouncement 10 Byte. The cumulative communication volume per good in average is therefore:

$$\begin{aligned}
 & 17.1 \cdot 113 \text{ Byte} && (\text{first hop route request}) \\
 + & 45.8 \cdot 119 \text{ Byte} && (\text{route replies}) \\
 + & 37 \cdot 26 \text{ Byte} && (\text{route announcements}) \\
 + & 37 \cdot 10 \text{ Byte} && (\text{route disannouncements}) \\
 = & 8714.5 \text{ Byte} && (\text{total})
 \end{aligned}$$

This shows that the average communication volume that the individual goods have to handle is well below 10 kilobyte for this scenario. As mentioned earlier, this is of course the volume at application level, so lower layer communication protocols will add some overhead to it. Still, this volume can be considered to be reasonable in an autonomous logistics scenario. However, further evaluations have to be done on larger scenarios to check whether the traffic volume on the logistic object's link scales well with increasing scenario size.

Conclusion + Outlook

In a fully distributed architecture, DLRP needs to be designed efficiently in order to keep the costs and the resource consumption on an economically and technically reasonable level. This article has investigated the communication traffic that is generated, with a special focus on the logistic object's communication interface. It has been shown that in the investigated setup and logistic network topology, the communication volume on this interface is in the order of some kilobyte, which is a reasonable volume. On larger topologies, however, it is expected that more communication traffic will be necessary, so further investigations are required here.

In future work, this analysis will be done in more detail for different topology sizes to identify whether there are limitations to the size of the logistic network that can be handled reasonably with DLRP. Different architectural options will also be analyzed in detail with respect to the requirements on computational power, communication and memory.

Acknowledgment

This research was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes".

References

- [1] Johnson, D.; Maltz, D.: Dynamic source routing in ad-hoc wireless networks. In: Imielinski, T; Korth, H. (eds): *Mobile Computing*. 1996, pp. 153-181.

- [2] Perkins, C.; Belding-Royer, E.; Das, S.: Ad-hoc on-demand distance vector (AODV) routing. IETF RFC 3561, 2003.
- [3] Scholz-Reiter, B.; Windt, K.; Freitag, M.: Autonomous logistic processes – new demands and first approaches. Proceedings of the 37th CIRP international seminar on manufacturing systems, 2004, pp. 357-362.
- [4] Scholz-Reiter, B.; Rekersbrink, H.; Freitag, M.: Internet routing protocols as an autonomous control approach for transport networks. In: Proceedings of the 5th CIRP international seminar on intelligent computation in manufacturing engineering, 2006, pp. 341-345.
- [5] Wenning, B.-L.; Rekersbrink, H.; Görg, C.: Scalability investigations on communication traffic in distributed routing of autonomous logistic objects. In: 9th international conference on ITS telecommunication. Lille, France, 2009, pp. 8-12.
- [6] Wenning, B.-L.; Rekersbrink, H.; Timm-Giel, A.; Görg, C.: Weighted multiplicative decision function for distributed routing in transport logistics. In: 2. International conference for dynamics in logistics (LDIC 2009). 2009, in print.
- [7] Wenning, B.-L.; Rekersbrink, H.; Timm-Giel, A.; Görg, C.; Scholz-Reiter, B.: Autonomous control by means of distributed routing. In: Hülsmann, M; Windt, K. (eds): Understanding autonomous cooperation and control in logistics – The impact on management, information and communication and material flow. Springer, 2007, pp. 325-335.
- [8] Wenning, B.-L.: Context-based routing in dynamic networks. (Dissertation) In: Advanced Studies Mobile Research Center Bremen. Vieweg+Teubner, 2010.