Network Nodes for Smart Thermal Grids

Tobias Teich, Daniel Kretz, Tim Neumann, Oliver Scharf, Pierre Werner

University of Applied Sciences Zwickau, Germany
Tobias.Teich@fh-zwickau.de, Daniel.Kretz@fh-zwickau.de, Tim.Neumann@fh-zwickau.de, Oliver.Scharf@fh-zwickau.de, Pierre.Werner@fh-zwickau.de

***Abstract*—** *Analogous to the smart grid in electrical context, we have developed a smart thermal grid providing features to supply grid network partners either with heat or to consume superfluous thermal energy. A local communication and supply network connects several customers with each other. Depending on offer or demand, the transport medium can be transmitted bidirectional between interconnected network node stations. The network for information exchange and trading operates parallel to the thermal supply network.*

Our developed network nodes consist of a hydraulic module for transportation, measurement and control technology as well as a microcomputer for advanced tasks and communication. Furthermore, the microcomputer provides a swarm intelligence controller. Our controller is responsible for data communication, processing network events and realizing a strategy for optimized network operation. Network nodes are able to optimize the network in a decentralized manner. Network nodes are devices that manage all transport-, measuring and control tasks. This paper gives a brief insight into our development efforts to the hardware and software platform of our microcomputer as well as the routing strategy based on ant colony optimization.

Keywords*— *Smart Thermal Grid, Smart Grid, Swarm Intelligence Controller, Ant Colony Optimization

I. INTRODUCTION

In general, heat energy is one of the most important energy types especially because of a high cost impact. Thermal energy is required e.g. as process heat as well as in heating processes. Hence, technological processes are causing huge quantities of thermal energy consumption and there is a lot of wasted and not reused energy. Against this, we have countless consumers that could reuse excess energy. Therefore, we searched for innovative approaches providing consumers with unused and potentially wasted energy from producers. The resulting smart thermal grid provides a solution to serve either cooling or heating demands by exchanging wasted energy amounts.

Our basic intention is to reduce the amount of required primary energy and the related carbon dioxide emission by reducing wasted energy. Therefore, we need an efficient heat transmission and consumption using different temperature levels within a common grid and smart control. A sensor network gathers required grid state information e.g. temperature and pressure values, actor parameters, flow rates as well as consumer demands. For our smart thermal grid, we have developed a special distribution node. Those nodes manage energy- and data-flows within our grid and control the hydraulic and thermal as well as the communication level.

The approach of our smart thermal grid is a multidirectional energy transmission between consumers and node stations in form of bivalent and multivalent nodes using high and low temperature levels. Therefore, we have developed hydraulic devices for transportation of a thermal carrier medium in several directions with different temperatures. Distribution nodes include hardware for measuring and a microcontroller, as illustrated in Fig. 1. In cases of communication failure, our microcontrollers automatically switch to internal failsafe mode and keep the network running.

Our network controller (NC) is part of each distribution node and we have a connection between them to explore the entire network structure automatically. Each NC provides their network and grid state information as well as consumer demands and offers. Additionally they have an autonomous search algorithm to implement a decentralized control approach.

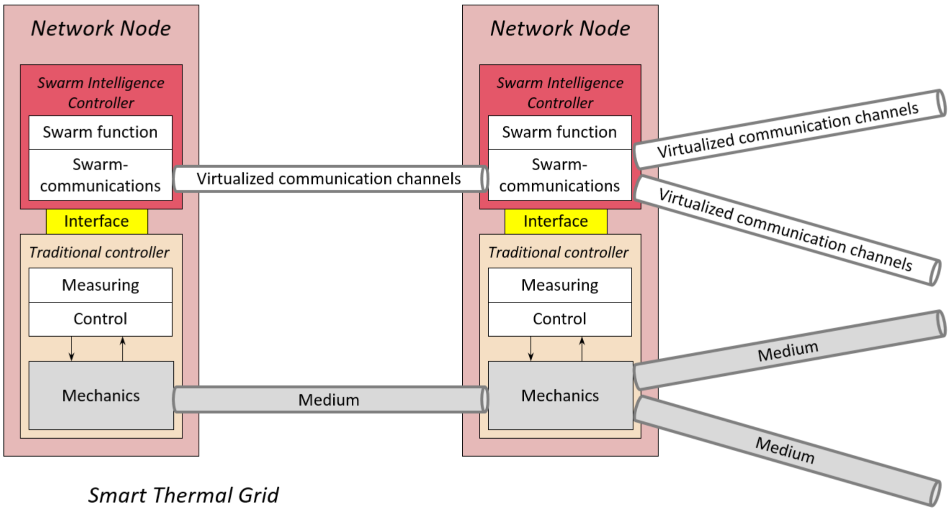

FIG. 1 ABSTRACT NODE SCHEME

II. SYSTEM ARCHITECTURE

2.1 Software stack

For the implementation of our NC devices, we have developed an optimized carrier board and implemented a software stack based on the Open Services Gateway initiative (OSGi) as shown in Fig. 2 [1].

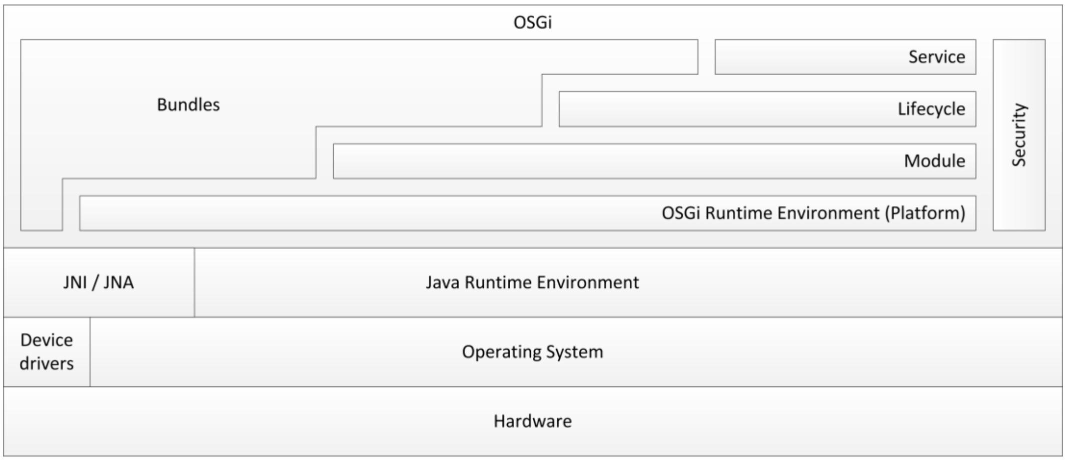

FIG 2. OSGI IMPLEMENTATION STACK

2.1.1 Hardware

The grid hardware installation requires bidirectional tubes, valves and pumps as well as the information technology. Each grid node ships with our network controller consisting of a microcomputer and an optimized carrier board. We use NXP Apalis iMX6 from Toradex as system on module (SOM) that exists either as dual- or quad-core version.

2.1.2 Operating system for the microcomputer

We have customized an individual Linux-based operating system (OS) for our controller. The operating system image provides required runtime software and optional developer and analysis tools. We created a customized embedded Linux operating system using yocto project [2] with open embedded layers and a self-patched version of toradex's board support package.

2.1.3 Java Runtime Environment

For several reasons, we have preferred Java runtime environment to implement the controller software:

- Broad availability of third-party libraries, especially for input and output tasks

- Portability, scalability and re-usability
- Efficient modular application development with OSGi

2.1.4 Open Services Gateway initiative

Based on recent work and experience, we use eclipse integrated development environment for OSGi bundle development and equinox as appropriate runtime.

2.2 Network communication

In general, we preferred TCP/IP-based communication to support a variety of communication channels. Hence, we can additionally use mobile network services like GPRS, UMTS, LTE, or use existing broadband networks to achieve maximum communication range. For more flexibility, we equipped our carrier board with two network interfaces and with interfaces for important industrial communication standards like serial RS232, RS485 as well as a Controller Area Network (CAN). The network controller uses the message queue telemetry transport protocol (MQTT) for communication [3]. MQTT is an industrial OASIS [4] standard. General advantages of MQTT are robustness and reliability in fragile connection scenarios.

MQTT uses a message broker to relay messages between clients through defined topics. In fact, in complex environments with many participants, it is impossible to define peer connections between all network partners or keep them up to date. Therefore, we define only adjacent message brokers to initialize the network communication. Afterwards, nodes can explore network structure via the message broker and locate each other.

In general, a central communication broker is a critical point of weakness because data exchange becomes impossible in any case of failure. A robust solution requires a shift from centralized to decentralized communication. Therefore, we provide multiple brokers within the network because each network controller can become a message broker. Communication is asynchronous and we publish messages efficiently in defined topics. We have implemented our MQTT services for client and server communication using eclipse moquette [5] as broker and eclipse paho as MQTT client. Additionally, we provide a communication library for easy third party integration using those services (Fig. 3).

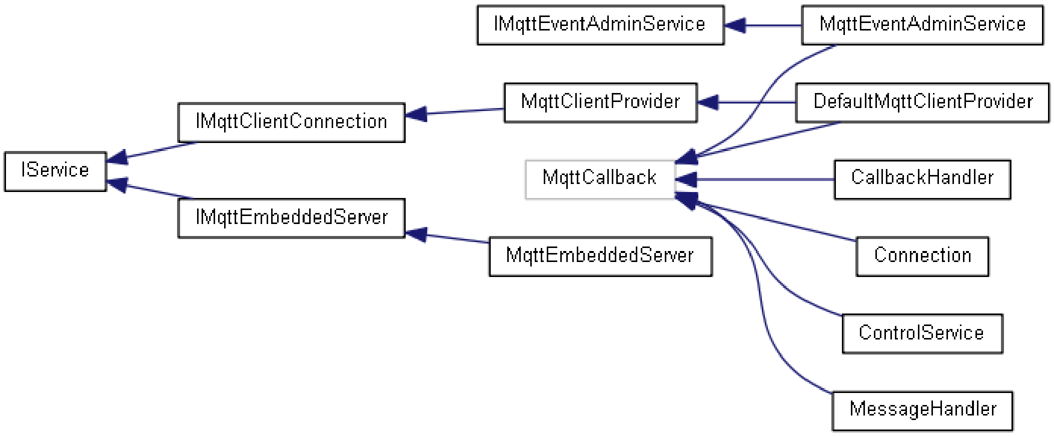

FIG. 3 MQTT SERVICE IMPLEMENTATION

2.3 Carrier board development

We equipped our device with an embedded computer where central components (CPU and RAM) are modular implemented as SOM-blocks. A system-on-module is a small single board computer, which contains the main components of a computer circuit: CPU, memory and additional chip sets, e.g. for power management and network. We connect our compatible carrier board to the SOM-board by plug or socket. This carrier board contains connectors and electronic circuits for required periphery, such as network and USB ports, power supply and many more. An advantage using a SOM/carrier board architecture against single board solutions is the reduction of development efforts. Another advantage is scalability because many manufacturers offer pin-compatible SOM families with graded performance and functionality.

As already mentioned, we use Toradex Apalis i.MX6 family as SOM platform which already provides several standard interfaces, such as Gigabit-Ethernet (GbE), USB, SATA, PCIe, SDIO, as well as several industrial interfaces like UART, I2C, SPI, CAN and several display standards like HDMI and LVDS. We implemented missing features like a second ethernet port, 1-Wire, RS485 and EnOcean as individual circuits on the carrier board (Fig. 4).

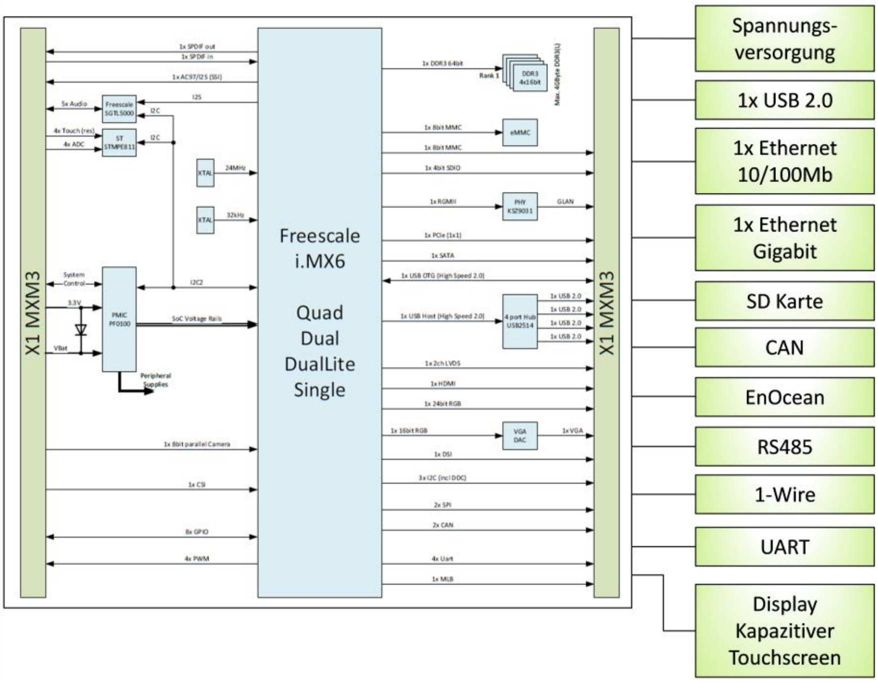

FIG. 4 CARRIER BOARD LAYOUT

2.4 Protocol definition

An important requirement for our communication protocol is to provide a flexible and extensible message format for arbitrary content, e.g. to communicate entirely different configured nodes with the same message structure. As previously discussed, the MQTT implementation provides required services for our message transport. A particular intention of MQTT is an efficient transmission even in fragile network environments with small bandwidth or high latency. Therefore, our message content should be as compact as possible, too.

XML is an established interchange format and provides the desired flexibility of message content. However, XML produces unjustifiable overhead caused by its syntax. Against this, we use JavaScript Object Notation (JSON) as an efficient alternative. JSON originates from JavaScript and is a standardized compact text-based data interchange format from the Internet Engineering Task Force (IETF, RFC 7159) [6]. The notation maps primitive data structures, such as characters, integers, digits and logical values. Additionally, JSON supports complex structures like strings, objects or arrays and object attributes with key-value pairs. The main advantage of key-value pairs is that we have neither fixed sequence structures nor hard content definitions and object contents are dynamic. We have adopted this approach in our protocol development and implemented an MQTT translation.

For dynamic message design, we describe a flexible structure: a message property object is associated to a message object. Property objects may contain any number of key-value pairs. Supported value types are limited to plain sequences of characters or any other message property object. Finally, nesting message properties supports the description of complex message structures.

III. SEARCH ALGORITHM

To illustrate our routing strategy within an undefined network graph, we have outlined an exemplary network with several bi- and multivalent nodes. Within this network, alternative supply routes and bi-directional transport directions exist (Fig. 5). For our network exploration, we use ant colony optimization (ACO). Simply summarized, the ACO strategy is based on the biological model of ant trails. Normally, ants visit known food sources on the shortest available trail. To retrace their routes, ants spread pheromone on their trace. Other ants follow this trace with high probability. If an existing trail is barred, after a short time ants will follow an efficient alternative route. The main reason is that ant are passing shorter routes more often and consequently the pheromone trace is getting more and more intensive on this route. A longer trail to the same destination loses its pheromone intensity because of evaporation [7].

Our strategy assumes that the structure of our smart thermal grid network is dynamic and requires exploration. In fact, nodes could be added or removed add any time or change their offers and demands. Our advantage using this strategy is the exploration of undefined dynamic networks. Each node controller knows only a partial definition of the and consequently our network is robust and tolerant against dynamic changes. Especially temporal offers and dynamic contracts can change our network because new partners may be attractive or connections changed their delivery direction.

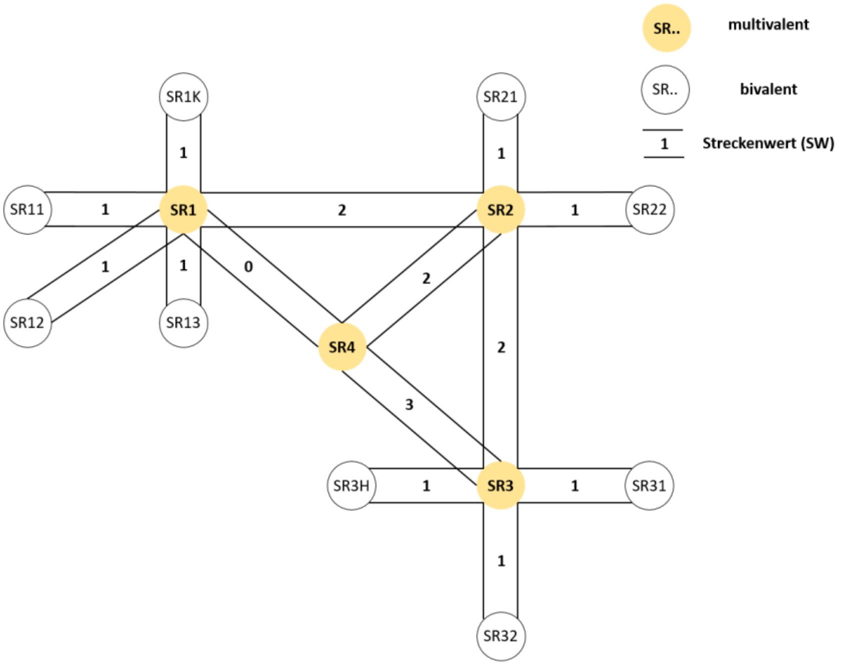

FIG. 5 SAMPLE NETWORK WITH ALTERNATIVE SUPPLY ROUTES

We have defined energy- and price-factors to abstract and relativize real world values. Positive energy-factor (EF) values define a possible emitted thermal energy output. Against this, negative values define an energy requirement or a consumable capacity. Positive price-factor (PF) values indicate an offer at a fixed rate, while negative values represent demands that allow expenditures up to this factor. Subsequently, thermal identical objectives like heat offers (HO) and cooling demands (CD) distinguish themselves by their price factor value (Table 1).

TABLE 1
DEMAND AND OFFER INTERPRETATION

	EF	PF
HO	+	+
HD	-	-
CO	-	+
CD	+	-

For our ACO strategy, we defined following constraints: Suppliers communicate their offers on request, so they have a passive role in the network. It is generally possible, that suppliers search solvent customers. However, this is contradictory to our network intention, where participants should search cost efficient partners. If valid solutions are available, partners make a contract update their network status. Additionally, we defined grid overheads as toll values (SW, Fig. 5). Toll values may include infrastructure charges, energy transport losses or required transport energy. Furthermore, edges are unidirectional or bidirectional. Finally activating a delivery process by making a contract causes directed edges.

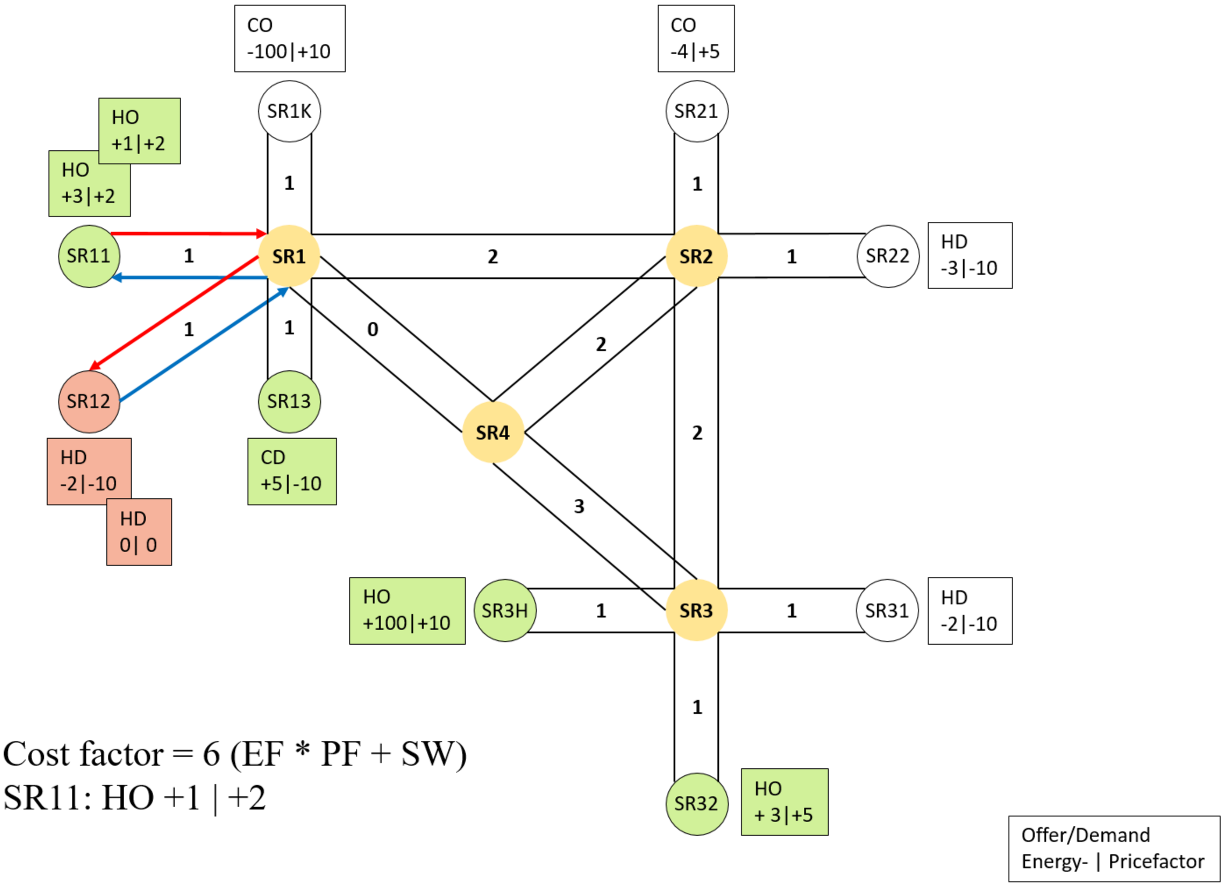

FIG. 6 EXAMPLE NETWORK

For simulating the ant system with our example grid, we require a definition of demand and offer, a search start point and available edges (available delivery routes of adjacent nodes), the number of ants (parallel search processes), an initial trail matrix with uniform distribution, and a visibility matrix from previous searches. Fig. 7 and Fig. 8 illustrate the probability calculation for selecting an adjacent node in the grid. If there is only a single route available, we define a selection probability of 100 %, otherwise we calculate the probability based on pheromone and visibility matrices.

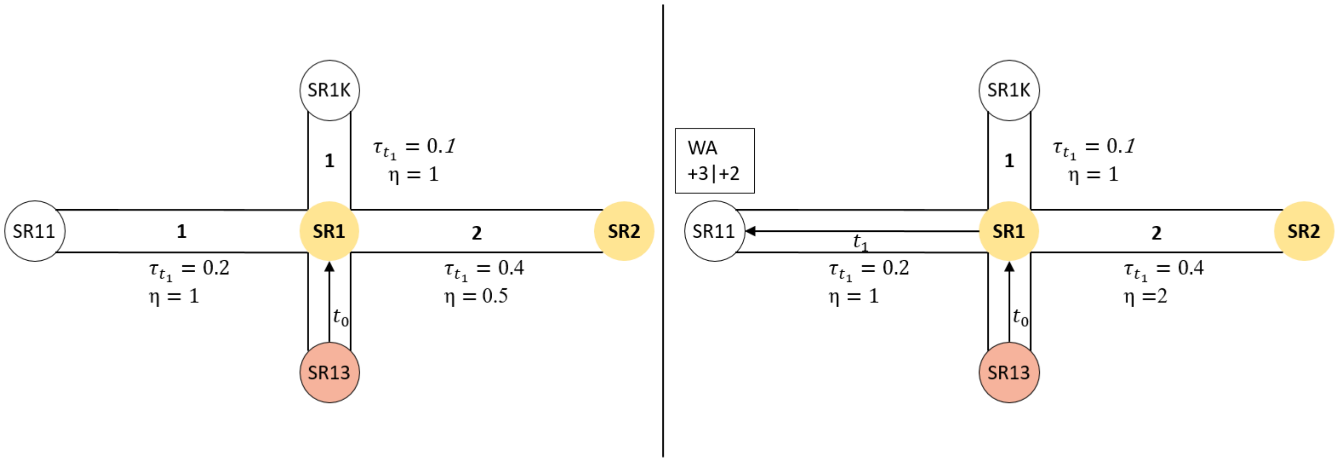

FIG. 7 INITIAL VALUES AND SELECTED ROUTE

$$
\begin{aligned}
s_0 &= SR_{13} \\
p_{SR_{13},SR_1} &= 1 \\
s_1 &= SR_1 \\
N_{t_1} &= \{SR_{1K}, SR_{11}, SR_2\} \\
p_{SR_1,SR_{13}} &= 0 \\
sum &= (0.4^{0.7} \cdot 0.5^{0.3}) + (0.2^{0.7} \cdot 1^{0.3}) + (0.1^{0.7} \cdot 1^{0.3}) \\
&= 0.9514 \\
p^k_{SR_1,SR_2} &= \frac{0.4^{0.7} \cdot 2^{0.3}}{0.9514} = 0.4496 \rightarrow 0.4496 \\
p^k_{SR_1,SR_{11}} &= \frac{0.2^{0.7} \cdot 1^{0.3}}{0.9514} = 0.3407 \rightarrow 0.7902 \\
p^k_{SR_1,SR_{1K}} &= \frac{0.1^{0.7} \cdot 1^{0.3}}{0.9514} = 0.2097 \rightarrow 1
\end{aligned}
$$

$$random(z) : 0 \leq z \leq 1 : z_{t_1} = 0.723 \rightarrow i_{t_2} = SR_{11}$$

$$p^k_{ij} = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N^k_i} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \forall_j \in N^k_i \\ 0 & \forall_j \notin N^k_i \end{cases}$$

τ : *Pheromone value (trail)*
ij : *edge*
k : *individual ant*
p : *probability*
N : *selectable nodes*

η : *heuristic value (visibility) with* $\eta = \frac{1}{SW}$ *for* $SW \geq 1$
$\eta = 1$ *const. for* $SW < 1$
$\alpha + \beta = 1$
$\alpha > \beta$: *rising pheromone value (collective knowledge)*
$\alpha < \beta$: *rising heuristic value (individual knowledge)*
$\alpha = 0.7$
$\beta = 0.3$
z : *uniform distributed random number*
s : *iteration step of an ant (starting point)*

$$
\begin{aligned}
L_k = \left(\sum_{ij}^{\forall (ij) \in tabu_k} SW_{ij} \right) + (|EF_Q| \cdot PF_S) | \forall (i,j) \in tabu_k \\
with |EF_Q + EF_S| < |EF_Q - EF_S| and PF_Q \leq -PF_S
\end{aligned}
$$

$|EF_Q + EF_S| \geq |EF_Q - EF_S|$: *solution does not match demand*
$PF_Q > -PF_S$: *cost limit exceeded*
$tabu_k$: *selected trails*
Q : *start node*
S : *end node*

FIG. 8 PROBABILITY CALCULATION FOR PATH SELECTION AND ROUTE EVALUATION

If there are no more paths available, the search of our ant ends and we evaluate the selected route by calculating a target function value using the subsequent calculation. An ant dies, if there is no solution on their route (no further paths or no suitable offer) and hence does not update the pheromone matrix. The main reason is to avoid infinite loops and ant select edges only once in a search run. If there is no solution available, the whole ant colony dies and existing routes evaporate. Consequently, the network is deeper explored after some search time.

IV. CONCLUSION

This paper provided a brief insight into our development efforts for creating smart thermal grids with focus on our developed network controller. Therefore, we discussed the technical architecture of our designed hardware platform, its software stack and some important implementation details. Furthermore, we presented our search algorithm based on ant colony optimization. We have implemented a simulation environment to combine existing grid nodes in a real test station with a virtual smart thermal grid for further researches and successfully tested our search strategy. Currently, we get optimized results at any time and we could solve the important aspect to deal with highly dynamic networks. Further work could deal with tests and efficiency evaluation of alternative search algorithms in different network structures.

REFERENCES

[1] OSGi Core. Hg. v. OSGi Alliance (6). Online available at https://osgi.org/download/r6/osgi.core-6.0.0.pdf, last verified: 03.02.2016

[2] Linux Foundation (2015): Yocto Project. Mega-Manual. Scott Rifenbark. Linux Foundation, Intel Corporation (2.0). Online available at: http://www.yoctoproject.org/docs/2.0/mega-manual/mega-manual.html, last access: 03.02.2016.

[3] Web-Address: http://www.mqtt.org, last access: 03.02.2016.

[4] OASIS (Organization for the Advancement of Structured Information Standards): Web-Address: https://www.oasis-open.org, last verified 03.02.2016.

[5] Web-Address: https://projects.eclipse.org/projects/iot.moquette, last verified 03.02.2016.

[6] Smith, Ben (2015): Beginning JSON. Berkeley, Calif.: Apress (The expert's voice in web development). Online available at http://dx.doi.org/10.1007/978-1-4842-0202-9.

[7] Dorigo, Marco: Stützle, Thomas (2004): Ant colony optimization, Cambridge, Mass.: MIT Press.