# Full Network Coverage Monitoring Solutions – The netBaltic System Case

Damian Karpowicz, Tomasz Gierszewski(✉), and Krzysztof Nowicki

Faculty of ETI, Gdańsk University of Technology, Gdańsk, Poland
tomag@pg.edu.pl
https://eti.pg.edu.pl/

**Abstract.** This paper defines the problem of monitoring a specific network, and more precisely – part of reporting process, which is responsible for the transport of data collected from network devices to station managers. The environment requires additional assumptions, as a specific network related to the netBaltic Project is to be monitored. Two new monitoring methods (EHBMPvU and EHBMPvF) are proposed, which priority is full network coverage. Both are based on employing additional IPv6 headers. Methods have been ana-lyzed in dedicated simulation environment and compared to classic monitoring solution – SNMP. The results show, that one of the proposed solutions outperform the current standard, but depend on traffic characteristics of the network.

**Keywords:** Network monitoring · IPv6 · netBaltic · Extension headers

## 1 Introduction

Network monitoring is a term used to describe [1] systematic, continuous control the behavior of the network and all elements included in this network. The process of monitoring network consists of five logical parts [2]: data collection, data representation, reporting, data analysis and presentation of the results. Network monitoring is the key process in the implementation of network management tasks and supports, among other things like network functioning analysis, problem and defects identification and the correctness of network configuration changes verification.

By collecting data more frequently and gathering more and more various types of data, more problems can be identified, but it also increases the network load. This problem becomes more significant with the increasing bandwidth consumption, rising IPv6 protocol popularity and the popularity of the Internet in general [2] due to network traffic growth – and the volume of measurement data grows with it. New system solutions for network monitoring should evolve towards better scalability and performance provision [3], and this is why organizations are still developing better network monitoring systems.

One way to reduce both the consumption of network bandwidth and computational overhead in management stations (centers) is to increase the intelligence in


P. Gaj et al. (Eds.): CN 2018, CCIS 860, pp. 89–102, 2018.
https://doi.org/10.1007/978-3-319-92459-5_8

devices, which gather monitoring data [4–6]. The first concept is to do some preliminary analysis just in the data collecting devices. In this way these devices send the partially analyzed data to management stations, instead of the entire set of monitoring information. The increased intelligence refers also to replacing periodical information sending with reporting performed only upon predefined events occurrence [2].

Along with the mentioned device intelligence increase, changes in the way of collected data reporting are also being proposed. An example of such a solution is the method of Intrinsic monitoring [7–9]. Its major feature is that it can significantly reduce the generated network traffic [7]. The principle of this method involves the use of an additional IPv6 header and the transfer functionality to decide about sending measurement data to network devices. Their task is to make autonomous decisions in order to send the measurement data, e.g. through the use of existing traffic (packets) in the network.

Section 2 describes the character of network monitoring, which is carried out within the framework of netBaltic project [10,11]. The essence of this network is that wireless links with a relatively low bit rates and high error rate are the only available. Section 3 contains a description of analyzed solutions about how to transport the collected monitoring data. In this section also the EHBMPvU and EHBMPvF methods are presented. Section 4 is devoted to the comparison of the developed methods and the classic solution to transport monitoring data. Finally, Sect. 5 presents the results of the analysis along with the indicators when and in what conditions should the particular method of reporting be used.

## 2 The Problem of Monitoring Specific Mesh Network Topology

The netBaltic Project can be characterized as a heterogeneous, wireless, self-organizing network with multi-hop transmission. In general there are wireless links with relatively low bitrates and high error rates mainly due to wavy water surface interacting with electromagnetic waves. Therefore, an important element is to ensure the least possible bandwidth.

Reporting of measurements, that is, transporting of the collected data via the network to the management stations, requires an additional traffic, which affects the behavior of the network. In case of netBaltic system, it is important that the monitoring system should provide supervision of all the network elements. This means that the entire network should be monitored. Another important thing is that the monitoring solution implementation should pose the least possible overhead. Such assumptions stem from the nature of the network.

The aim of the netBaltic Project is to develop innovative mechanisms for self-organizing heterogeneous wireless networks, allowing possibly fast data transfer between vessels, ships and centers of storage and data processing, as well as the public Internet access. Wireless data transmission at sea will improve the safety of navigation through the realization of e-navigation services [10]. This project

is being implemented by a consortium whose leader is Gdansk University of Technology. The netBaltic system is intended to provide constant connectivity with ships and vessels from the Internet. In case this is not possible, to provide certain services operating in delay tolerant mode (e.g. maps updates, weather forecast delivery, e-mail). It is important that the construction of the network structure will be based on IPv6 protocol.

Due to the nature of the network and the characteristics of sea physics it is essential to employ appropriate mechanisms for routing between nodes. The priority is to minimize the consumption of bandwidth resources by both the signaling (routing) and user data selection strategy, to avoid unnecessary transmission. Hence, each netBaltic system node functions as a specialized router.

The routing solution proposed by the netBaltic Project consortium, is to use two complementary routing protocols, i.e. proactive: TBRP (Tree-Based Routing Protocol) [12] and reactive: RM-AODV (Radio Metric Ad-hoc On Demand Distance Vector) [12]. This combination of two complementary types of routing can be called a hybrid routing. Such combination, known from the IEEE 802.11s standard [13], is called Hybrid Wireless Mesh Protocol. TBRP routing protocol employs a tree structure and does not provide full knowledge about the network topology to each node. When TBRP routing does not guarantee knowledge of the route between nodes, it is necessary to use the RM-AODV method, which is a reactive routing protocol, used on demand.

## 3 Network Monitoring System Proposal

The bandwidth of NetBaltic system is extremely valuable and should be used reasonably. For this purpose, this paper contains two different proprietary solutions, both based on the mechanism of the additional IPv6 headers.

The first proposal for the monitoring system is called EHBMPvF (Extension Header Based Monitoring Protocol version Forced), while the second proposal is EHBMPvU (Extension Header Based Monitoring Protocol version Unforced). Both solutions of transport monitoring data guarantee full network coverage [14].

### 3.1 EHBMPvF Method

EHBMPvF monitoring works by sending special monitoring packets, which aim is to visit specified list of nodes in a predefined sequence. The last address in the list is the node from which the monitoring packet was sent. Single so-prepared package allows to collect data from a certain number of nodes, depending on the data link layer payload size and the size of monitoring data acquired from each node.

Monitoring messages can be sent at a specified time interval or on demand. Monitoring is initiated from the network node that is monitoring center. The principle of monitoring EHBMPvF is based on the additional routing header (type 0) available in IPv6. This type of additional header, employing IPv6 addresses, allows to define which network nodes packet has to traverse along its route.

The mechanism of formatting lists of nodes' addresses for packet monitoring works in a similar way to the BFS (Breadth-First Search) tree (graph) – first the nodes closest to the monitoring center are visited. As the TBRP proactive routing protocol produces routing tables with just the number of hops to reach each node (network prefix in general), this information is used to prepare the list of nodes to visit in the specified order – from the closest to the most distant ones.

All of these routers' addresses in the header must be visited in the specified order, but other nodes may be visited on the way. To specify the list of nodes, through which a monitoring packet is expected to traverse, it is necessary to use information kept by the node that is the root of the tree, as it has information about how to get to all the nodes assigned to it.

When a node receives such packet, it takes action aimed at checking whether the node is the recipient of this message, i.e. the IPv6 datagram's destination address.

If it is not, the package is sent further, according to the route determined by the routing mechanism. As there is the possibility that the node would not know the route to the next node specified, the reactive routing protocol RM-AODV is used to find the route.

If the receiving node belongs to the list of specified addresses, it adds own monitoring data to the received packet. The data is structured as in Fig. 1 and belongs to the datagram payload.


**Fig. 1.** The data structure placed in a package by the monitored node. In parentheses is the length of the data fields in bits.

The fields' meaning is the following:

- TimeStamp – this field contains the time of transmitted data. Using a 32-bit the information about the year (12 bits), month (4 bits), day (5 bits), hour (5 bits) and minutes (6 bits) can be stored;
- Fragment Length – length in bytes of Data field (16 bits);
- Fragment ID – this is a field informing that the Data field carries only part of the whole monitoring information. The value identifies the number of the fragment. The value of 0 indicates that the data placed in the structure is not fragmented (16 bits)
- Data – contains the collected data from a single node.

Along path traversal each such structure is added, successively one after another, reflecting the order of nodes along the path specified in the header.

**Changing Network Conditions Mitigation.** The EHBMPvF monitoring uses the information contained in the TBRP root's routing table of the tree, which is updated in discrete time periods. Therefore there is possible situation in which the node being on the list to visit, either loses the possibility to communicate, or fails completely, leading to the loss of even indirect communication with off-shore infrastructure. This means that both the TBRP and RM AODV protocols won't be able to find the route. In such situation a modification of the routing header should be made, by removing the node from the list and sending the packet to the next node in the list.

Another undesirable situation that may occur, is when one of the nodes loses packet with monitoring data as a result of an accident. To overcome this, the TCP protocol is used for sending the monitoring packets for reliability. In case of damaged link used in the created TBRP tree, besides using tree reconstruction mechanisms, the other option is to perform RM AODV routing.

All these mechanisms are employed to ensure that the EHBMPvF monitoring solution is resistant to changes in the network, i.e. upon a node failure, or topology change, the monitoring will continue reacting to current conditions.

**Network Coverage in EHBMPvF.** Each special packet sent by the monitoring EHBMPvF contains the list of nodes to visit and to collect data from. The network transfers this packet until each node listed adds its monitoring data, and then the packet sent back to the monitoring center. The TCP protocol is used for each step, which ensures that the transfer of packet is successful. In this way the monitoring center obtains data from the specified part of the network – the number of nodes is equal to the size of the address list.

Therefore, using the rule of deduction, it can be concluded that if one packet allows for partial coverage of the network, then sending $P$ packets with unique node addresses on the list, the full network coverage can be achieved. This number is determined by the following formula:

$$P = \left\lceil \frac{x}{\left\lfloor \frac{w}{z} \right\rfloor} \right\rceil, \tag{1}$$

where:

$P$ – the required number of monitoring packets,
$x$ – total number of nodes to be monitored (value representing the size of the routing table TBRP at the root of the tree),
$w$ – the maximum size of payload data in the application layer of a single datagram in bytes,
$z$ – the size of data collected from each node in bytes (the size should be shared by all the nodes).

The $P$ value is calculated based on the current number of available network nodes, taken from the routing table. Hence, the full network monitoring coverage by the EHBMPvF method can be obtained, if and only if the monitoring center sends P special packages.

### 3.2 EHBMPvU Method

Collecting data from the nodes in the network using the EHBMPvU monitoring method involves the use of existing users' packets as the mean of transport. The packet have to be addressed to the monitoring center's IP address. One packet allows to collect data from a certain number of nodes, depending on the size of the data itself and available application layer payload space.

The rate of sending data depends on the frequency of nodes' applications communication. Monitoring is initiated by each node independently. The principle of operation in monitoring EHBMPvU is based on an additional new header, named Monitoring Header, which aims to transport the data required for monitoring.

Adding data to the package is possible if and only of the three conditions are met:

- packet must have a valid recipient. To reach the target data, it can only be added to a packet, which will meet on its path the monitoring center node;
- the packet have enough space to add an additional header structure either with the data or the data itself, if the header has been added by some previous node along route;
- there must be the need to send monitoring data.

Monitoring Header and method of its placement in the packet is shown in Fig. 2.

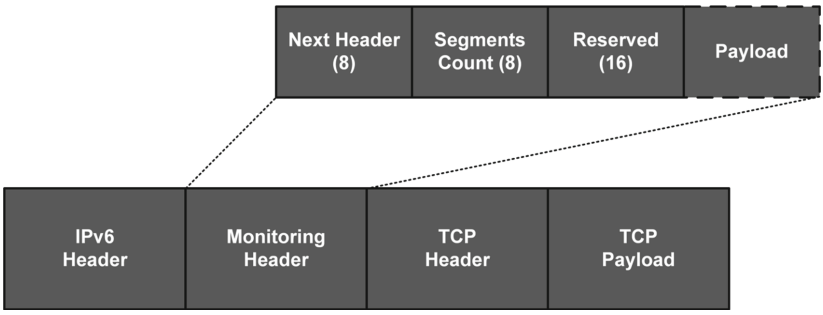


**Fig. 2.** Additional Monitoring Header and scheme of placing it in the datagram. In parentheses is the lengths of the fields in bits.

The proposed Monitoring Header, which is shown in Fig. 2 has the four following fields:

- Next Header – identifies the next header following just after the Monitoring Header. This field allows for compliance with the mechanism of the additional headers in IPv6 (8 bits);
- Segments Count – determines the number of nodes, which added data to the Monitoring Header. When adding data, increase this value (8 bits);

– Reserved – field that was introduced to take into account possible future modifications to this header. Currently it is being skipped during processing (16 bits);
– Payload – field in which structure are added – contain data added by the node, including monitoring data.

Figure 3 shows the data structure which is placed in the Monitoring Header. Each new structure is included consecutively one by one.

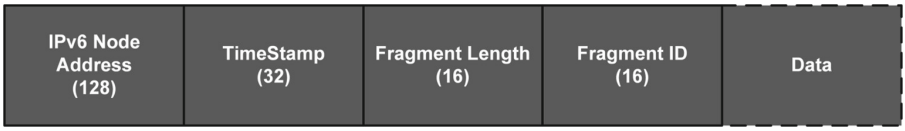


**Fig. 3.** The structure of the data placed in the Monitoring Header by the node. In parentheses is the length of the data fields in bits.

Meaning of the fields of this structure is similar to the structure shown in Fig. 1 used by the EHBCMPvF method. There is an additional IPv6 Node Address field at the beginning in the length of a single IPv6 address (128 bits), which is used to identify the origin of the data. This field contains the IPv6 address of the node which added the data to the packet.

After adding the collected data to the packet, the node sends it towards the monitoring center. Each packet which has an additional header can carry the monitoring data from at least one node. The size of monitoring data depends on meeting the three previously described conditions.

**Changing Network Conditions Mitigation.** The EHBMPvU monitoring method is resistant to changes in the network, i.e. when a node failure occurs, or it loses even indirect communication with offshore infrastructure, monitoring will continue without this node. This is due to the fact that a damaged node has no influence on the monitoring process of the other nodes in the network.

The only undesirable situation that may occur is when one of the intermediate nodes on the packet route towards the monitoring center lose packet with monitoring data as a result of an accident. Such situation, however, is addressed by the use of only TCP application protocol for piggybacking purposes. In case of damaged link used in the created TBRP tree, besides using tree reconstruction mechanisms, the other option is to perform RM AODV routing.

**Network Coverage in EHBMPvU.** Each node in the network operating the EHBMPvU monitoring method is obligated to make an attempt to send monitoring data in preconfigured periods of time. Prior sending the data all the required datagram conditions have to be met, especially the destination address and the available payload space.

The use of just TCP segments ensures that the transfer of packet between the nodes will be reliable. Finally, the packet reaches the target and monitoring center receives monitoring data from at least one node. Using the rule of deduction, it may be concluded that if each node in the network behaves as described above, full network coverage can be achieved. The worst case scenario requires using the number of application packets equal to the number of nodes in the network. The necessary number of packets can be specified as follows:

$$P \leq x \tag{2}$$

where:

$P$ – required number of monitoring packets,
$x$ – the number of nodes in the network.

## 4 Comparison of the Presented Methods with the Classic Method of Monitoring

In order to determine the efficiency of two proposed methods for the netBaltic system, the simulations was performed by using the devoted simulator [14] and the results were compared to the typical monitoring use case – namely the SNMP protocol. The latter was assumed to work in the send request and wait for response mode of operation. Scalability, security and the overhead were taken under consideration in the evaluation.

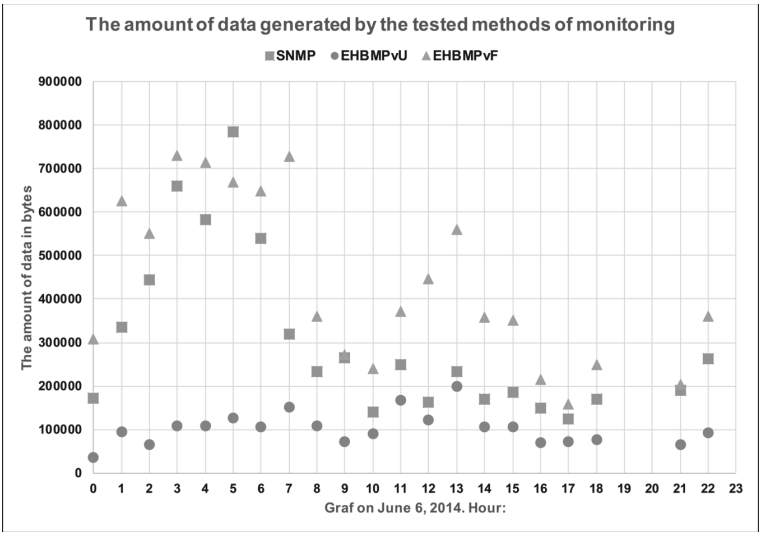


**Fig. 4.** Comparing the amount of data generated by the methods EHBMPvF, EHBMPvU and SNMP (classic method).

The experiment was performed on the data collected from the AIS system by Baltica ship, dated on 6th June, 2014 [15]. The simulation was performed for snapshots of ships' positions taken in one hour intervals. Summary, which provides information about numbers of nodes and links, as well as the sizes of the root's arrays of the trees (obtained by the TBRP protocol) for particular hours, are shown in Table 1. The data for hours 19, 20 and 23 was unavailable, because the node selected for the root of a tree in TBRP routing, did not contain any record in its routing table. This is due to the fact that the data collected by the AIS system by Baltica ship does not necessarily show all the ships in the Baltic Sea region, because this system has limited range.

For each network snapshot, 15 iterations of the simulation were run during which data transport methods used for monitoring were analyzed. The results, which represent the total amount of generated data on all links of the network are shown in Fig. 4.

**Table 1.** Hourly summary for number of vessels (nodes in the network) that have been registered by the AIS.

| Time (hour) | Number of nodes | Number of links | The size of root's routing table |
|---|---|---|---|
| 0 | 124 | 3024 | 99 |
| 1 | 120 | 2604 | 116 |
| 2 | 121 | 3079 | 112 |
| 3 | 136 | 3427 | 134 |
| 4 | 135 | 3566 | 134 |
| 5 | 138 | 3382 | 136 |
| 6 | 139 | 3613 | 130 |
| 7 | 133 | 2559 | 126 |
| 8 | 75 | 685 | 67 |
| 9 | 60 | 352 | 54 |
| 10 | 57 | 237 | 47 |
| 11 | 68 | 339 | 58 |
| 12 | 64 | 306 | 58 |
| 13 | 70 | 395 | 64 |
| 14 | 63 | 281 | 55 |
| 15 | 59 | 253 | 52 |
| 16 | 60 | 268 | 45 |
| 17 | 50 | 200 | 38 |
| 18 | 54 | 279 | 47 |
| 21 | 58 | 271 | 47 |
| 22 | 66 | 371 | 59 |

To perform monitoring using EHBMPvF method, the reactive routing RM-AODV was used, due to the fact that the TBRP routing did not provide the knowledge about the whole structure of connections in the network available at each node. Therefore, the results of EHBMPvF should be considered together with the RM AODV overhead.

The evaluation of EHBMPvF monitoring just without taking into account the effect of RM-AODV protocol resulted in generating about 354% more data on average in related to the EHBMPvU method. Comparing it with the classic method gave on average about 61% more data. On the other hand, the classic method generated on average about 233% more data comparing to the EHBM-PvU method. Worth mentioning here is the fact, that typically SNMP monitoring requires several send/receive operations for each OID separately. In our case it was assumed to be a single request and single bulk response in the size of 128 bytes with monitoring data, which was equal for the other methods too.

The results taking into consideration the amount of data generated also by the RM-AODV routing, it can be concluded that the EHBMPvF monitoring required to generate on average about 17,600% of the data compared to the EHBMPvU method and 4290% to the classic one. These are the worst case values, because in typical scenario the network would also employ the RM-AODV for application data routing, so it would reduce the number of on demand routing calls dedicated purely to monitoring requests, thereby reducing the total amount of data generated.

Scalability is a very important feature, which allows to assess whether the future system expansion has a chance to succeed or fail. Analyzing the results obtained in simulation environment, it was agreed that the EHBMPvU scales much better than EHBMPvF monitoring. The classic method for network monitoring, known as send a request and wait for the response, is a popular solution, but for the tested networks, it generated on average about 230% more data compared to the EHBMPvU.

The actual, precise time consumed for collecting data from the whole network (using presented monitoring methods) is impossible to determine, because of multitude of factors that may run in parallel in the real system. However, this time can be estimated.

In the case of EHBMPvF implementation, the duration of the monitoring process consists of the time required for propagation of special data packets over the network and delays related to the need for on demand RM-AODV routing. The more times the RM-AODV on demand routing is called, the more total time is required for EHBMPvF monitoring.

However, in the EHBMPvU method, besides the time associated with the propagation of packets from a node to the monitoring center, significant influence has the time to wait for the right TCP packet. The probability of occurrence an appropriate packet (which can be used to transport the collected measurement data) at the node at time T after the occurrence of the request specifying the need to send the collected data can be represented by the equation:

$$P(A) = P(Y) \int_0^T p(x)dx \int_1^R h(z)dz \tag{3}$$

where:

$P(A)$ – the probability of an event A, the occurrence of the proper packet;
$P(Y)$ – the empirical probability of (derived from statistical surveys) the event y, where y event states that the packet is either addressed to the monitoring center or has it along its route;
$p(x)$ – empirical probability distribution, which specifies the possibility of passing through or sending the packet by the node, where $x$ is the packet arrival time;
$T$ – time in which the proper packet occurs;
$h(z)$ – empirical probability distribution which define the probability of a packet that has a specified size of data field of link layer protocol, where z is the size of the data field;
$R$ – maximum size of the data field, decreased by size of added monitoring data.

The monitoring time in the classic version of the monitoring depends only on the delay associated with the propagation of data packets from the monitoring center to a node, and then back to the monitoring center.

The proposed EHBMPvU monitoring mechanism is not transparent to the devices in the network, as it requires all devices to support the use of the proposed additional IPv6 header mechanism. If all the nodes in the network have to be monitored, this means that all the nodes must be able to recognize and process the additional monitoring header. Otherwise the packet is discarded by the router, because the node will not know what to do with such header. In case of the EHBMPvF mechanism, these issues looks different, because the Routing Header (Type 0) is described in IPv6 [16], but for safety reasons, in the year 2007, the Routing Header (Type 0) was withdrawn and marked by status of disapproval (deprecated) [17]. Therefore, it may not be supported by the routers. This is due to the fact that this header can be used to perform DoS (Denial of Service) attacks. The issue has been solved by introducing Routing Header Type II [18].

In the context of the necessary requirements to perform monitoring, one should keep in mind the resources of devices in the network. The described monitoring solutions require the ability to parse, process and send monitoring data, so routers must have sufficient processing power to perform these tasks, along with the other functions implemented in router. The busiest router is node which is the monitoring center, because it collects data from all nodes in the network. In the EHBMPvF method it is also responsible for sending requests for collecting data. To send such packets, the router must perform additional calculations associated with the optimization of the number of required packets and must allocate a list of addresses to visit for each packet. The EHBMPvF mechanism uses the RM-AODV on demand routing, which generates additional traffic in the network and load to the nodes.

Hardware requirements for the classic method of monitoring are similar to what EHBMPvU requires – the processing of monitoring data.

Security of collected data is also important as they should be kept confidential and integral. To ensure these requirements, the use of IPsec (Internet Protocol Security [19–21]) security suite is recommended in the netBaltic Project.

## 5 Summary

To summarize the presented characteristics in Table 2, one can come to the conclusion that the best solution is to use EHBMPvU monitoring, which is well scalable, provides full coverage in a short time, has small requirements and is secured by the means for confidentiality and authenticity. The only problem which may occur is the case when the application traffic characteristics would a node require to wait long for the correct data packet. In the extreme case where the traffic would be strictly limited or the total lack of it, the realization of EHBMPvU monitoring would not be possible.

Knowing the characteristics of the traffic the probability of sending monitoring data by each node can be estimated – after an event triggering this process (at certain time or in the response to the identified event in the network).

EHBMPvU method should be used in networks where the characteristics of the existing traffic does not cause unacceptable delays in providing data used for monitoring. Longer delays may cause that the transmitted data can be regarded as outdated.

The classic method of monitoring is the best choice if the network to be monitored has an unacceptable traffic characteristics to use by the EHBMPvU solution.

**Table 2.** Comparison of the characteristics of the analyzed monitoring methods.

| | EHBMPvF | EHBMPvU | Classic monitoring |
|---|---|---|---|
| Scalability rating ($10^n$ nodes) | n = 1 | n = 1, 2, 3, 4, 5 | n = 1, 2, 3, 4 |
| Full network coverage | Yes | Yes | Yes |
| Real time to complete network coverage | From a few milliseconds to several minutes, hours – depending on the characteristics of the existing network traffic | From a few milliseconds to several minutes, hours – depending on the characteristics of the existing network traffic | From a few to hundreds of milliseconds, depending on the size of the network |
| Hardware requirements (the number of operations performed) | Support monitoring traffic and RM-AODV | Support monitoring traffic | Support monitoring traffic |
| The security of transmitted data | Yes | Yes | Yes |

Alternatively, you can use a hybrid solution using the classic method of monitoring wherever the characteristics of traffic does not allow for acceptable time to provide data to the monitoring center and EHBMPvU where the time is acceptable.

The choice of monitoring method for netBaltic system requires testing the characteristics of mobility. Then and only then it will be possible to unambiguously identify the best monitoring method for this type of network. However, yet at this early stage of testing, the EHBMPvF method can be rejected due to very large overhead related to monitoring.

**Acknowledgments.** This work has been partially supported by the Applied Research Program under the Grant: ID PBS3/A3/20/2015, founded by the National Centre for Research and Development.

## References

1. Silvestri, S., Urgaonkar, R., Zafer, M., Ko, B. J.: An online method for minimizing network monitoring overhead. In: 2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS), pp. 268–277. IEEE (2015)
2. Lee, S., Levanti, K., Kim, H.S.: Network monitoring: present and future. Comput. Netw. **65**, 84–98 (2014)
3. Mahkonen, H., Manghirmalani, R., Shirazipour, M., Xia, M., Takacs, A.: Elastic network monitoring with virtual probes. In: 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), pp. 1–3. IEEE (2015)
4. Prieto, A.G., Stadler, R.: A-GAP: an adaptive protocol for continuous network monitoring with accuracy objectives. IEEE Trans. Netw. Serv. Manage. **4**(1), 2–12 (2007)
5. Dilman, M., Raz, D.: Efficient reactive monitoring. IEEE J. Sel. Areas Commun. **20**(4), 668–676 (2002)
6. Jiao, J., Naqvi, S., Raz, D., Sugla, B.: Toward efficient monitoring. IEEE J. Sel. Areas Commun. **18**(5), 723–732 (2000)
7. Höfig, E., Coşkun, H.: Intrinsic monitoring using behaviour models in IPv6 networks. In: Strassner, J.C., Ghamri-Doudane, Y.M. (eds.) MACE 2009. LNCS, vol. 5844, pp. 86–99. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05006-0_7
8. Shi, L., Davy, A., Muldowney, D., Davy, S., Höfig, E., Fu, X.: Intrinsic monitoring within an IPv6 network: mapping node information to network paths. In: 2010 International Conference on Network and Service Management (CNSM), pp. 370–373. IEEE (2010)
9. Shi, L., Davy, A.: Intrinsic monitoring within an ipv6 network: relating traffic flows to network paths. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2010)
10. Hoeft, M., Gierłowski, K., Nowicki, K., Rak, J., Woźniak, J.: netBaltic: enabling non-satellite wireless communications over the baltic sea. IEEE Commun. Mag. **5** (2016). http://gcn.comsoc.org/netbaltic-enabling-non-satellite-wireless-communications-over-baltic-sea
11. Woźniak, J., Hoeft, M.: Aim and main research tasks of the netBaltic project (in Polish). Telecommun. Rev. Telecommun. **12**, 1301–1303 (2016)

12. Institute of Electrical and Electronics Engineers: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 802.11-2012 (2012)
13. Institute of Electrical and Electronics Engineers: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 10: Mesh Networking. 802.11-2011 (2011)
14. Karpowicz, D., Nowicki, K.: Implementation of database API for archival large-scale AIS data retrieval for netBaltic Project simulation purposes (in Polish). Scientific report, Gdańsk (2016)
15. Lewczuk, M., Hoeft, M., Cichocki, P., Woźniak, J., Nowicki, K.: Systems of AIS data acquisition, processing and visualization for the netBaltic project (in Polish). Telecommun. Rev. Telecommun. News **12**, 1326–1329 (2016)
16. Deering, S., Hinden, R.: RFC 2460, Internet Protocol, Version 6 (IPv6) Specification. Internet Engineering Task Force (1998)
17. Neville-Neil, G., Savola, P., Abley, J.: RFC 5095, Deprecation of Type 0 Routing Headers in IPv6 (2007)
18. Johnson, D.B., Arkko, J., Perkins, C.E.: RFC 6275, Mobility Support in IPv6 (2015)
19. Seo, K., Kent, S.T.: RFC 4301, Security Architecture for the Internet Protocol (2005)
20. Kent, S.T.: RFC 4303, IP Encapsulating Security Payload (ESP) (2005)
21. Gierszewski, T.: Transport security mechanisms for netBaltic system (in Polish). Telecommun. Rev. Telecommun. News **8–9**, 813–816 (2017)