DATABASE AND BIGDATA PROCESSING SYSTEM FOR ANALYSIS OF AIS MESSAGES IN THE NETBALTIC RESEARCH PROJECT

MICHAŁ LEWCZUK, PAWEŁ CICHOCKI
AND JÓZEF WOŹNIAK

Faculty of the Electronics, Telecommunications and Informatics
Gdansk University of Technology
Narutowicza 11/12, 80-233 Gdansk, Poland

(received: 2 August 2017; revised: 4 September 2017;
accepted: 11 September 2017; published online: 2 October 2017)

Abstract: A specialized database and a software tool for graphical and numerical presentation of maritime measurement results has been designed and implemented as part of the research conducted under the netBaltic project (Internet over the Baltic Sea – the implementation of a multi-system, self-organizing broadband communications network over the sea for enhancing navigation safety through the development of e-navigation services.) The developed software allows tracing graphs of radio-connections between shore stations and vessels (offshore units), based on historical data including the traffic of ships and their specific parameters collected on the Baltic Sea during the last four years. It also enables preparation of data for network simulation experiments using AIS (Automatic Identification of Ships) and GPS (Global Positioning System) loggers installed on shore stations and vessels, taking into account a number of input parameters, such as: time range, coast station selection, ship flags based on MMSI numbers and types and ranges of possible communication technologies used (WiFi, WiMax, Radwin, LTE, *etc.*).

The created tool has a multi-layer architecture that utilizes the MariaDB SQL database, the Apache2 WEB server, and a number of PHP applications. The runtime environment has been built on Linux Debian version 8 and the HP C7000 cluster of the 16 CPU x86_64 architecture. The modularity of the application allows parallel processing and, therefore, optimization of the computing cluster.

The database contains more than 70 million records which enables simulation of various topologies (with multi-hop transmissions) and network operations depending on the transmission techniques being used. The database is fully scalable, and allows easy adding of further data collected during subsequent measurement sessions. Additionally, the use of virtualization tools facilitates the future migration to more efficient processing environments, in case of a significant increase in the volume of data.

The data recorded in the database allows calculation of statistics for the surveyed networks, and determining the incidence of potential network nodes (*e.g.* by flag) complete with their available communication techniques – information which is important in determining structures of possible multi-hop networks and their performance. The software finds routes for datagrams according to accepted criteria and exports results to a network traffic simulator, and as such is an important part of the framework used for planning next measurement campaigns and determining which communications equipment would be more suitable for vessels.

Keywords: AIS, NMEA data, MariaDB SQL database, cache mechanism, virtualization, net-Baltic project

DOI: https://doi.org/10.17466/tq2017/21.4/a

1. Introduction

Maritime communication systems are expected to support e-navigation services, enable voice/video transmission, as well as exchange of other information (e.g. emails or web browsing) via the global information infrastructure. In order to support faster deployment of e-navigation and other services and applications useful in maritime environments, the IMO has proposed a concept known as e-navigation [1, 2]. This concept harmonizes the collection, integration, exchange, presentation and analysis of maritime information onboard and ashore by electronic means to enhance berth-to-berth navigation and related services, for safety and security at sea and protection of the marine environment. High-speed and cost-effective maritime wireless communications are essential for the success of the above concept as the deployment of a wide range of e-navigation solutions requires a broadband or semi-broadband transmission capability. Since, as pointed out in [3], about 80 percent of the world trade is transported by maritime units, it makes the importance of maritime communications even more evident. Additionally, a number of other marine activities including, *e.g.* fishing, sea exploitation or tourism increase the significance of marine communication systems and the necessity of broadband communications even more.

Due to this, many organizations and research working groups, including but not limited to ITU (International Telecommunication Union), ETSI (European Telecommunications Standards Institute) as well as IMO (International Maritime Organization) and IALA (International Association of Marine Aids to Navigation and Lighthouse Authorities) are working towards improvements in communication between vessels and different onshore and/or offshore installations and upgrading utilities and services offered to their crews and passengers travelling all over the world (including for example entertainment and broadband access to global networks for everybody) [4, 5].

In this situation also researchers from the Gdansk University of Technology, in cooperation with several Polish partners, including the National Institute of Telecommunications, the Institute of Oceanology of the Polish Academy of Sciences, and the companies: DGT-LAB and NavSim, are currently working on deployment of a heterogeneous maritime communication system, capable of transparently utilizing a diverse range of communication technologies. Such capability will allow the system to use any and all communication capabilities of a maritime vessel in the process of dynamically creating a data transmission infrastructure. The system is designed to make extensive use of self-organization and autoconfiguration principles to maximize the area from where digital communication is possible [6, 7].

The main aim of the netBaltic project is to develop and deploy a broadband wireless communication system capable of providing connectivity in a heterogeneous wireless environment, able to meet the requirements of e-navigation services. The lack of reliable high-throughput communications is currently a major barrier in e-navigation implementations. At the same time different, currently available communication technologies [2, 8–13] show many limitations that can be summarized as follows:

- HF and VHF technologies, although capable of long-range communication, are able to provide only a very limited bandwidth;
- higher frequency WLAN and WMAN technologies are severely range-limited;
- satellite communications technologies are often too expensive, especially for smaller vessels.

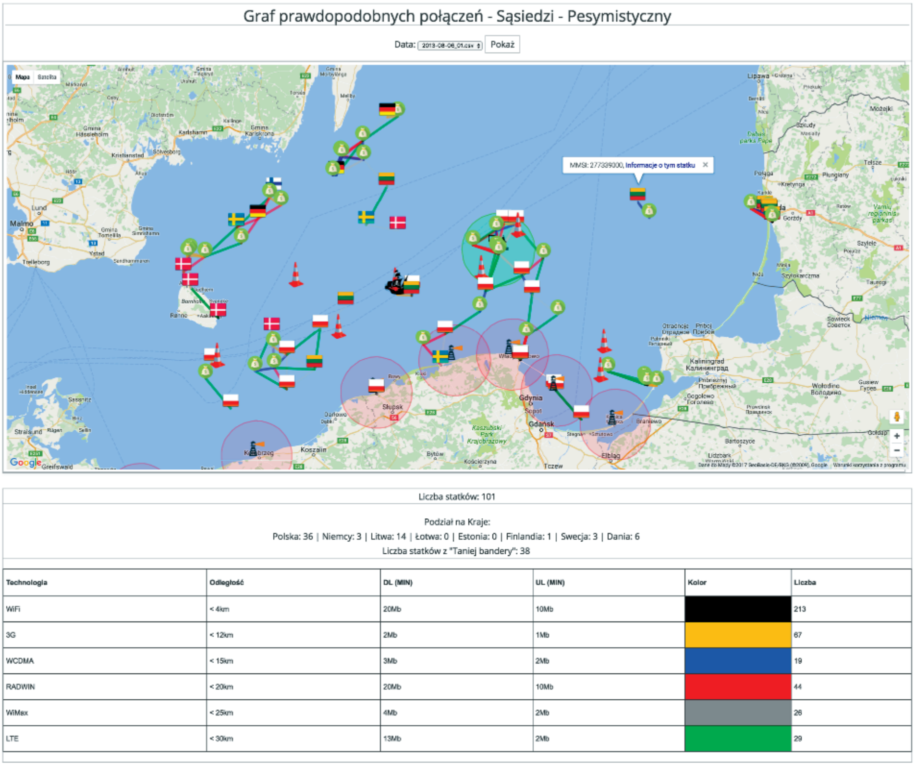
Thus, optimizing a chosen technology to operate in the maritime environment will bring beneficial results, however, it will not provide what can be called a general or universal solution. In this situation, a heterogeneous system, automatically making use of the most efficient available technology could be seen as a possible answer to the requirements of modern e-navigation services [2, 6].

As part of the research carried out under the netBaltic project (Internet over the Baltic Sea – The implementation of a multi-system, self-organizing broadband communications network over the sea for enhancing navigation safety through the development of e-navigation services) a dedicated database and tools for graphical and numerical presentation of maritime measurement results have been designed and implemented. The software allows, *inter alia*, tracking the route graph of connections between shore stations and vessels (offshore units) based on historical traffic of ships (and their parameters) moving over the Baltic Sea. The data has been collected (for the purposes if this research) during the last four years [14, 15].

The paper is organized as follows. First we describe the software that we have developed. Then, we show efficiency problems occurring with the big data processing in terms of the netBaltic measurements. Finally, we present the chosen solution for database acceleration using cache mechanisms together with example results referring to generation of reports by the designed and implemented application.

2. Software dedicated to NetBaltic project

The designed and implemented software allows gathering and processing of AIS and GPS messages recorded by trackers installed at shore stations and on ships (vessels) for network simulation purposes, assuming input constraints such as the presence of specific information elements (time, coast station selection, ship ownership basing on MMSI numbers and types/ranges of wireless technology) are observed. An example graph of vessels connections is presented in Figure 1.



Graf prawdopodobnych połączeń - Sąsiedzi - Pesymistyczny

Data: 2013-08-06_01.csv Pokaż

Liczba statków: 101

Podział na Kraje:
Polska: 36 | Niemcy: 3 | Litwa: 14 | Łotwa: 0 | Estonia: 0 | Finlandia: 1 | Swecja: 3 | Dania: 6
Liczba statków z "Taniej bandery": 38

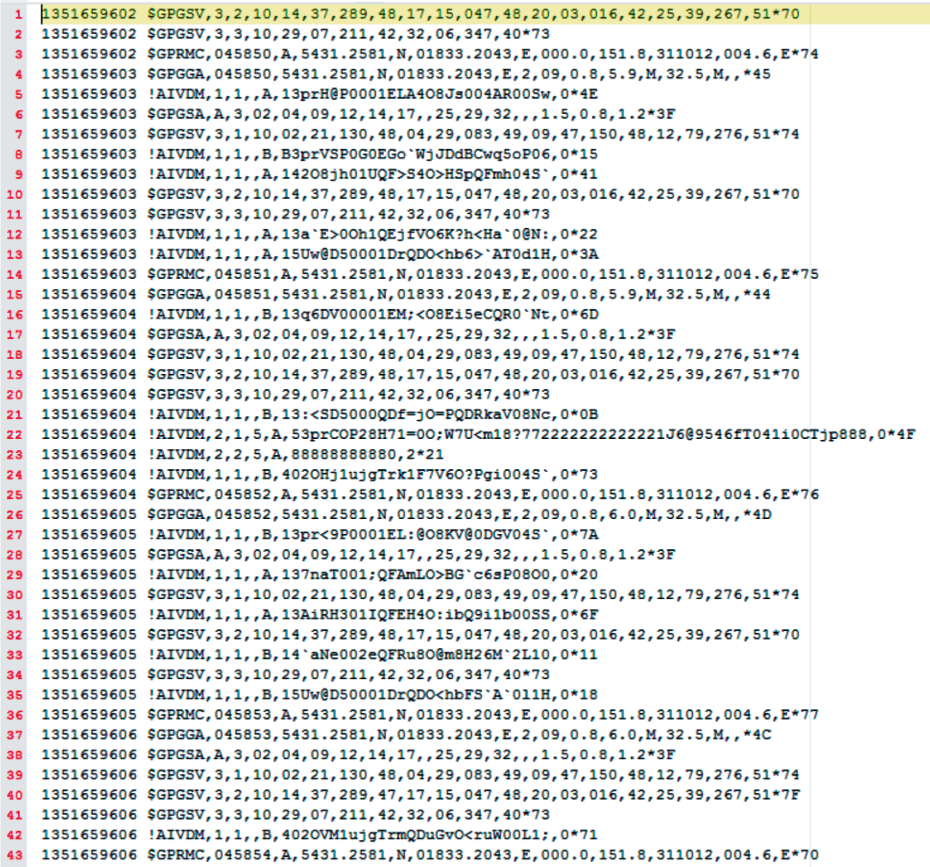
Technologia	Odległość	DL (MIN)	UL (MIN)	Kolor	Liczba
WiFi	< 4km	20Mb	10Mb	⬛	213
3G	< 12km	2Mb	1Mb	🟨	67
WCDMA	< 15km	3Mb	2Mb	🟦	19
RADWIN	< 20km	20Mb	10Mb	🟥	44
WiMax	< 25km	4Mb	2Mb	⬜	26
LTE	< 30km	13Mb	2Mb	🟩	29

Figure 1. Example graph of connections for pessimistic variant

3. Data from mobile and stationary AIS data loggers

Collecting reliable information from the maritime environment is necessary to perform advanced netBaltic project network simulation experiments including, e.g. vessel movement visualization, presentation of graphs – connections depending on the transmission technique used, tracing the optimal and redundant paths for effective routing purposes, etc. For this reason, AIS and NMEA GPS data loggers were built and installed, both at shore stations and on some vessels for the project purposes. All of them were equipped with external antennas capable of withstand harsh marine environmental conditions. Compact shipborne as well as standard, stationary receivers were made. Data streams from the AIS and GPS receivers were recorded in standard NMEA text files (see Table 1). Every single received message was stored and appended with a local timestamp (as illustrated in Figure 2). The files from the recorders containing raw data were split into hourly intervals and compressed day by day. A single hour file contained nearly 40,000 lines (where one line represented one message for later decomposition and analysis). Recorders compacted the daily data into separate zip files for flash memory saving and stored into two independent drives for redundancy.

The data collected by every single logger during one year in zipped files was about 2 GB. After decompression and deduplication (if necessary) we received



```
1351659602 $GPGSV,3,2,10,14,37,289,48,17,15,047,48,20,03,016,42,25,39,267,51*70
1351659602 $GPGSV,3,3,10,29,07,211,42,32,06,347,40*73
1351659602 $GPRMC,045850,A,5431.2581,N,01833.2043,E,000.0,151.8,311012,004.6,E*74
1351659603 $GPGGA,045850,5431.2581,N,01833.2043,E,2,09,0.8,5.9,M,32.5,M,,*45
1351659603 !AIVDM,1,1,,A,13prH@P0001ELA408Js004AR00Sw,0*4E
1351659603 $GPGSA,A,3,02,04,09,12,14,17,,25,29,32,,,1.5,0.8,1.2*3F
1351659603 $GPGSV,3,1,10,02,21,130,48,04,29,083,49,09,47,150,48,12,79,276,51*74
1351659603 !AIVDM,1,1,,B,B3prVSP0G0EGo`WjJDdBCwq5oP06,0*15
1351659603 !AIVDM,1,1,,A,142O8jh01UQF>S4O>HSpQFmh04S`,0*41
1351659603 $GPGSV,3,2,10,14,37,289,48,17,15,047,48,20,03,016,42,25,39,267,51*70
1351659603 $GPGSV,3,3,10,29,07,211,42,32,06,347,40*73
1351659603 !AIVDM,1,1,,A,13a`E>0Oh1QEjfVO6K?h<Ha`0@N:,0*22
1351659603 !AIVDM,1,1,,A,15Uw@D50001DrQDO<hb6>`AT0d1H,0*3A
1351659603 $GPRMC,045851,A,5431.2581,N,01833.2043,E,000.0,151.8,311012,004.6,E*75
1351659604 $GPGGA,045851,5431.2581,N,01833.2043,E,2,09,0.8,5.9,M,32.5,M,,*44
1351659604 !AIVDM,1,1,,B,13q6DV00001EM;<O8Ei5eCQR0`Nt,0*6D
1351659604 $GPGSA,A,3,02,04,09,12,14,17,,25,29,32,,,1.5,0.8,1.2*3F
1351659604 $GPGSV,3,1,10,02,21,130,48,04,29,083,49,09,47,150,48,12,79,276,51*74
1351659604 $GPGSV,3,2,10,14,37,289,48,17,15,047,48,20,03,016,42,25,39,267,51*70
1351659604 $GPGSV,3,3,10,29,07,211,42,32,06,347,40*73
1351659604 !AIVDM,1,1,,B,13:<SD5000QDf=jO=PQDRkaV08Nc,0*0B
1351659604 !AIVDM,2,1,5,A,53prCOP28H71=0O;W7U<m18?77222222222221J6@9546fT041i0CTjp888,0*4F
1351659604 !AIVDM,2,2,5,A,88888888880,2*21
1351659604 !AIVDM,1,1,,B,402OHj1ujgTrk1F7V6O?Pgi004S`,0*73
1351659604 $GPRMC,045852,A,5431.2581,N,01833.2043,E,000.0,151.8,311012,004.6,E*76
1351659605 $GPGGA,045852,5431.2581,N,01833.2043,E,2,09,0.8,6.0,M,32.5,M,,*4D
1351659605 !AIVDM,1,1,,B,13pr<9P0001EL:@O8KV@0DGV04S`,0*7A
1351659605 $GPGSA,A,3,02,04,09,12,14,17,,25,29,32,,,1.5,0.8,1.2*3F
1351659605 !AIVDM,1,1,,A,137naT001;QFAmLO>BG`c6sP08O0,0*20
1351659605 $GPGSV,3,1,10,02,21,130,48,04,29,083,49,09,47,150,48,12,79,276,51*74
1351659605 !AIVDM,1,1,,A,13AiRH301IQFEH4O:ibQ9i1b00SS,0*6F
1351659605 $GPGSV,3,2,10,14,37,289,48,17,15,047,48,20,03,016,42,25,39,267,51*70
1351659605 !AIVDM,1,1,,B,14`aNe002eQFRu8O@m8H26M`2L10,0*11
1351659605 $GPGSV,3,3,10,29,07,211,42,32,06,347,40*73
1351659605 !AIVDM,1,1,,B,15Uw@D50001DrQDO<hbFS`A`011H,0*18
1351659605 $GPRMC,045853,A,5431.2581,N,01833.2043,E,000.0,151.8,311012,004.6,E*77
1351659606 $GPGGA,045853,5431.2581,N,01833.2043,E,2,09,0.8,6.0,M,32.5,M,,*4C
1351659606 $GPGSA,A,3,02,04,09,12,14,17,,25,29,32,,,1.5,0.8,1.2*3F
1351659606 $GPGSV,3,1,10,02,21,130,48,04,29,083,49,09,47,150,48,12,79,276,51*74
1351659606 $GPGSV,3,2,10,14,37,289,47,17,15,047,48,20,03,016,42,25,39,267,51*7F
1351659606 $GPGSV,3,3,10,29,07,211,42,32,06,347,40*73
1351659606 !AIVDM,1,1,,B,402OVM1ujgTrmQDuGvO<ruW00L1;,0*71
1351659606 $GPRMC,045854,A,5431.2581,N,01833.2043,E,000.0,151.8,311012,004.6,E*70
```

Figure 2. Example of AIS and GPS data stream recorded, appended with local timestamps

about 25GB of raw data per year for one logger. Having considered the amount of data collected we decided to use only two recorders and a four year period for limiting the number of records used for analysis. During our simulation about half a terabyte of NMEA data was processed and imported into an SQL database for further computations.

Table 1. Example of raw NMEA data stored into raw logger files

TimeStamp	Delimiter	NMEA DATA (AIS, GPS)
1351659603	SPACE	$GPGGA,045850,5431.2581,N,01833.2043,E,2,09,0.8,5.9,M,32.5,M,,*45
1351659603	SPACE	!AIVDM,1,1,,A,13prH@P0001ELA408Js004AR00Sw,0*4E

4. Virtualization

The application was designed keeping in mind its further dynamic development along with the latest trends in building large data warehouses. Due to this the application was prepared originally as a virtual machine and then converted into the output template called "Virtual Appliance". The use of virtualization has also enabled a more efficient way of versioning and backups creation, which benefited the deployment and testing of new software editions. If after the implementation of a new version of the underlying application modules they would show unstable operation or data corruption, it would be no problem to retract the faulty test version and restore the data to the original state from before the deployment. The virtualization host had originally been based on the VMware environment, but with the migration to the new server room, the machine was converted to Microsoft HyperV and worked in a much more efficient manner, confirming the validity of the chosen development direction.

The virtual machine is based on the Debian Linux 8 operating system, which includes additional development environment components and a database. The main application for its proper operation used a multi-layered architecture based on a free SQL database – MariaDB, Apache2 WEB server and a number of script applications written in PHP 5.5. The virtual machine virtualization environment was a computing cluster based on the HP C7000 blade server with physical servers based on the Intel Xeon 16 CPU and x86 64-bits architecture.

5. Dedicated software

The data warehouse application consists of many independent components. Components were built with multithreading in mind so that they could work simultaneously while achieving a uniform cluster load in parallel work – resulting in efficient processing of large amounts of complex data.

An independent process additionally monitors the activity of individual application tasks and dynamically adjusts the number of threads to the actual disposal of the processor. The benefit of such a solution is that the processor does not wait for threads independently of each other, but operates throughout the entire processing period evenly.

The dedicated software was divided into three main components:

- Backend (*mechanisms which operate on the data in background*);
- Presentation Layer (*User Interface*);
- API REST (*interface which allows external applications to be connected*).

5.1. Backend

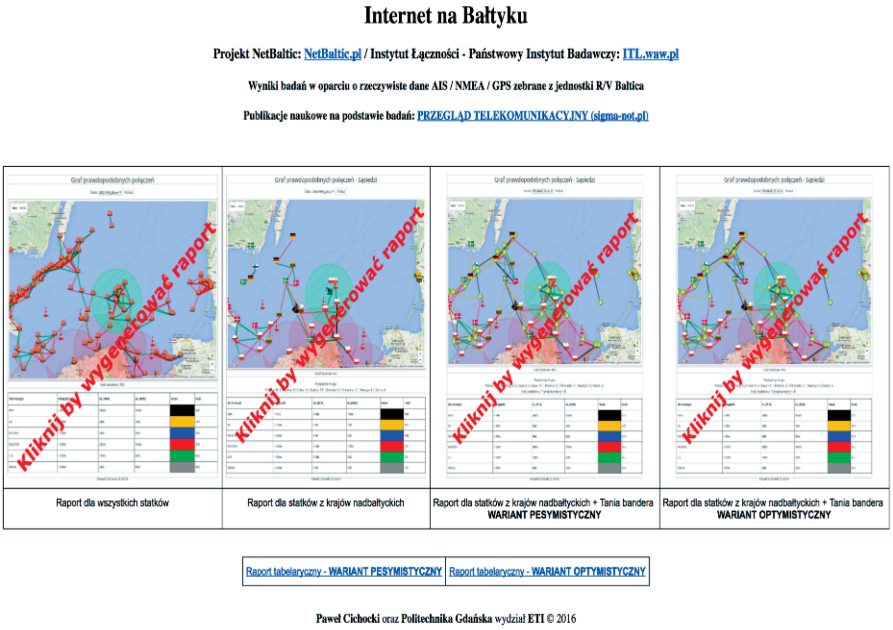
For effective operation the application uses a number of independent mechanisms – such an approach is referred to as modular software. The operation of application modules does not affect other independent modules. This is not the case for data processing dependent modules where some data is passed through several modules in the correct order (*e.g.* decoder and accelerator modules – the accelerator will not start unless it has processed all the decoder data).

Mechanisms operating in the background and not directly interacting with the user's application are based on multithreading. The user uploads raw data from the recorders into the application. The application, in turn, automatically retrieves raw text files, decodes them, parses and analyzes them in complex ways. Then, the accelerator modifies the final data files that are uploaded into the database. In the final phase, when the processed data is in the database, reports and visualizations are made available for users by means of a WWW interface.

The software has been designed to automatically collect extra information that cannot be retrieved with MMSI AIS receiving units by accessing external information sources. For example, a network robotic engine has been created that searches through public registers of vessels (*e.g.* MarineTraffic.com) based on regular expressions (RegEx). This mechanism visits selected subpages with information about the current name, dimensions, categories, destination of the unit. The robot then processes the downloaded data, verifies the content extracted from the AIS raw files and inserts it into the local database, and if a photo of the vessel is available, it downloads it to the server and links it to the data for later comprehensive presentation.

5.2. *Visualization and presentation*

The user results module (part of the application) has been created to generate multiple independent reports. Ready-made reports (predefined – with parameter values already supplied) and custom reports for custom values entered by the user can be created on request (see Figure 3).



Internet na Bałtyku

Projekt NetBaltic: NetBaltic.pl / Instytut Łączności - Państwowy Instytut Badawczy: ITL.waw.pl

Wyniki badań w oparciu o rzeczywiste dane AIS / NMEA / GPS zebrane z jednostki R/V Baltica

Publikacje naukowe na podstawie badań: PRZEGLĄD TELEKOMUNIKACYJNY (sigma-not.pl)

Graf prawdopodobnych połączeń
Graf prawdopodobnych połączeń - Sąsiedzi
Graf prawdopodobnych połączeń - Sąsiedzi
Graf prawdopodobnych połączeń - Sąsiedzi
Kliknij by wygenerować raport
Kliknij by wygenerować raport
Kliknij by wygenerować raport
Kliknij by wygenerować raport

Raport dla wszystkich statków	Raport dla statków z krajów nadbałtyckich	Raport dla statków z krajów nadbałtyckich + Tania bandera **WARIANT PESYMISTYCZNY**	Raport dla statków z krajów nadbałtyckich + Tania bandera **WARIANT OPTYMISTYCZNY**

Raport tabelaryczny - **WARIANT PESYMISTYCZNY**	Raport tabelaryczny - **WARIANT OPTYMISTYCZNY**

Paweł Cichocki oraz **Politechnika Gdańska** wydział **ETI** © 2016

Figure 3. Main page of NetBaltic application GUI

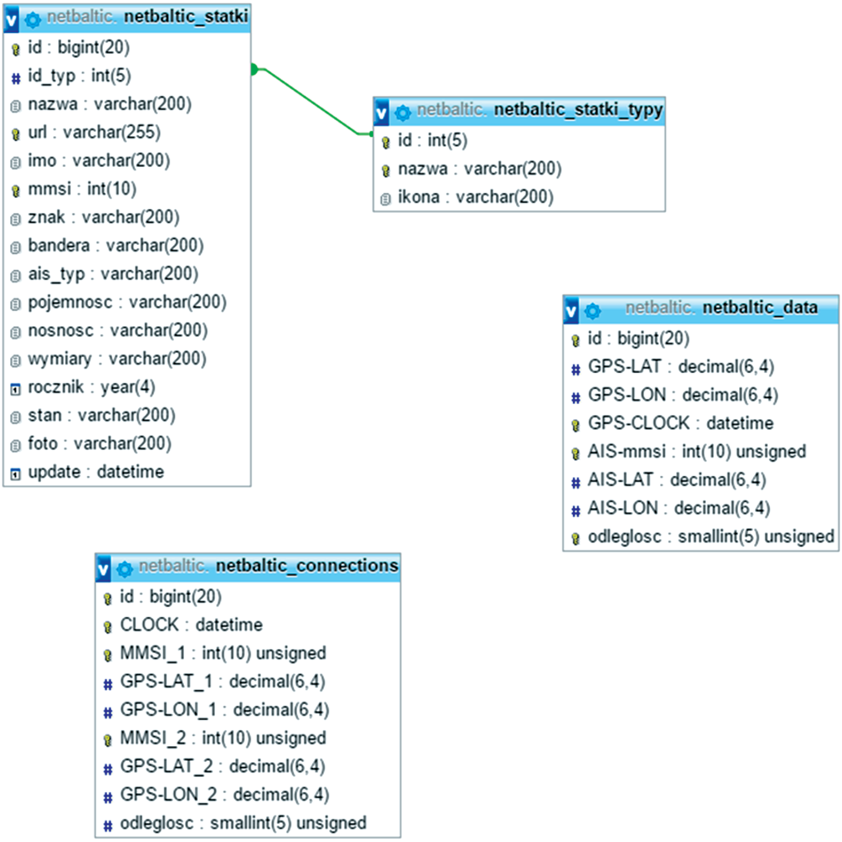
Reports generated on request include both tabular reports and graphs with a probable set of network connections for vessels and base stations based on four years of collected data. Example reports include:

- general reports for all vessels registered by the recorder in a selected time interval;
- reports for vessels registered in specific Baltic Sea states (optimistic / pessimistic / parametric) and so-called “cheap flag” vessels – depending on the communication technique used;
- tabular reports (optimistic / pessimistic / parametric) for all defined fixed measurement points on the Baltic Sea – depending on the communication technique used.

6. Warehouse database and CACHE mechanism

Currently, the database contains more than 70 million unique records, enabling simulations of various topologies including both onshore stations and mobile stations (vessels). The database is scalable and more data collected as a part of subsequent measurement campaigns can be easily added.

The database consists of four tables (Figure 4). All tables have defined indices and triggers to speed up queries generated by the software and to validate new data input (independently of the software – a two-step verification).



netbaltic. netbaltic_statki
id : bigint(20)
id_typ : int(5)
nazwa : varchar(200)
url : varchar(255)
imo : varchar(200)
mmsi : int(10)
znak : varchar(200)
bandera : varchar(200)
ais_typ : varchar(200)
pojemnosc : varchar(200)
nosnosc : varchar(200)
wymiary : varchar(200)
rocznik : year(4)
stan : varchar(200)
foto : varchar(200)
update : datetime

netbaltic. netbaltic_statki_typy
id : int(5)
nazwa : varchar(200)
ikona : varchar(200)

netbaltic. netbaltic_data
id : bigint(20)
GPS-LAT : decimal(6,4)
GPS-LON : decimal(6,4)
GPS-CLOCK : datetime
AIS-mmsi : int(10) unsigned
AIS-LAT : decimal(6,4)
AIS-LON : decimal(6,4)
odleglosc : smallint(5) unsigned

netbaltic. netbaltic_connections
id : bigint(20)
CLOCK : datetime
MMSI_1 : int(10) unsigned
GPS-LAT_1 : decimal(6,4)
GPS-LON_1 : decimal(6,4)
MMSI_2 : int(10) unsigned
GPS-LAT_2 : decimal(6,4)
GPS-LON_2 : decimal(6,4)
odleglosc : smallint(5) unsigned

Figure 4. Database structure

A user defining a new report using an application accessible throughout a web page interacts directly with the database server. However, the data used for presentation and enumeration is retrieved from the CACHE server's cache instead.

The cache is cyclically rebuilt in the background independently of users' activity. The time needed to rebuild CACHE from the current database is almost 24 hours of continuous processing. Caching has accelerated the generation of reports from minutes to about 2–15 seconds.

If a user needs to use up-to-date data directly from the database, bypassing the accelerator mechanisms, it is possible for him to use a dedicated REST API, available at the appropriate URI. The API returns an URI to an XML data file that can be processed by any external application.

The introduction of the API has allowed us to provide support for desktop applications created by external developers and to do so in an efficient and secure manner – the programmer does not need to know the database structure specifics or modify the server's source code.

7. Problems encountered in the design phase

Many software problems were encountered during the software development. The most common problem to be dealt with was the limited hardware resources and too many data sets to process. The amount of data to be processed from a four-year period in the first phase of the application (4-threaded direct database processing) gave the optimistic prediction: the processing time of recorded data would last more than 8 months of continuous computations. The time period was directly proportional to the number of records in the database.

Query optimization and additional indexing of selected columns did not solve these issues. The solution which allowed the results to become available to users with only a small latency on an average server was to abandon the current application builder policy and create dedicated middleware to import data.

Initially, the application worked on a well-known principle:

Application Presentation Layer + Decoder ↔ Database Connector ↔ Database

The problem was an inefficient database (with a high number of records) that was written on a fast disk array. In order to eliminate performance problems, the existing database-based logic – duplicate detection and data validation – was added to the intermediate application (accelerator) and was based on cache and ORM applications running on high-speed RAM.

The principle of operation of the application in the final version is as follows:

Application Presentation Layer → CACHE ↔ Database ↔ Accelerator + ORM ↔ Cache ← Decoder

Multi-threaded mechanisms, namely the accelerator and CACHE were created for faster data processing, based on our earlier experience. The operation of these mechanisms is based on building database structures on RAM using ORM and application popup files. The implementation of these solutions resulted in several times faster execution of applications and possibility of processing even much larger data sets. The time consumed by the data processing was reduced to two weeks of multi-threaded server operation. During that time, all data from the recorder was processed (historical data from a period of four years). Introducing Caching algorithms helped us to shorten the processing time from about 10 minutes to less than 1 second (usually less than a half of second) for drawing a single graph, and to below 10 seconds for preparation of full reports. Example comparative times obtained with and without using the CACHE module are shown in Table 2.

Table 2. Illustration of benefits brought by caching mechanisms – loading time values with and without the CACHE module

Visualization	Loading time without CACHE	Loading time with CACHE
Graph connections – All Vessels	more than 10 min.	below 1 s (usually around 0.5 s)
Graph connections – Baltic countries (Neighborhoods)	more than 10 min.	below 1 s (usually around 0.5 s)
Graph connections – Baltic countries (Neighborhoods) + Cheap flags	more than 10 min.	below 1 s (usually around 0.5 s)
Graph connections – user defined report	more than 10 min.	below 2 s (usually around 1 s)
Table with calculation results – user defined report	more than 10 min.	below 10 s (usually around 5 s)
Table with calculation results – standard report	more than 10 min.	below 10 s (usually around 5 s)

8. Conclusions

Using the information obtained from mobile and stationary recorders, pre-processed and uploaded into the database, it was possible to generate statistics for wireless networks surveyed in the real maritime environment and in particular to determine the incidence of potential network participants (nodes) in terms of the use of particular communication techniques. It was really important for determining the suitability, performance and application of NetBaltic project.

Dedicated software is capable of finding routes from a specific source to a set of destinations for data packets according to a set of criteria and enables the export of results to network traffic simulators (via API). This software has become an essential part of the theoretical framework to plan further measurement campaigns within the project and determine which communication devices should be installed in order to achieve the expected results.

Acknowledgements

This work was 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] IMO MSC 81/23/10 2005 *Work Programme. Development of an e-Navigation strategy*, submitted by Japan, Marshall Islands, the Netherlands, Norway, Singapore, the United Kingdom and the United States, International Maritime Organization

[2] IMO NCSR Annex 7 2014 *Draft e-navigation strategy implementation plan* http://www.imo.org/en/OurWork/Safety/Navigation/Documents/enavigation/SIP

[3] Zaidi K S, Jeoti V, Awang A 2013 *Wireless backhaul for broadband communication over sea*, Proc. IEEE 11th Malaysia International Conference on Comm. 298

[4] Report on IALA Workshop 2016 *Development of VHF Data Exchange System (VDES)*

[5] ITU RSG: IALA, e-NAV14-10.3.6 2013 *Working Document toward a draft new Report Maritime Radiocommunication Systems and Requirements*

[6] Wozniak J, Gierlowski K, Hoeft M 2017 *Broadband Communication Solutions for Maritime ITSs: Wider and Faster Deployment of New e-Navigation Services*, 16^{th} Intenational Conference on Intelligent Transportation System (IEEE Xplore)

[7] Hoeft M, Gierłowski K, Nowicki K, *et al.* 2016 *netBaltic: Enabling Non-Satellite Wireless Communications over the Baltic Sea*, *IEEE Communications Magazine* **05/2016** 2

[8] Kdouh H, Brousseau C, Zaharia G, *et al.* 2011 *Measurements and path loss models for shipboard environments at 2.4 GHz*, 41st European Microwave Conference 408

[9] Yang K, Roste T, Bekkadal F, *et al.* 2011 *Long-Distance Propagation Measurements of Mobile Radio Channel over Sea at 2 GHz*, IEEE Vehicular Technology Conference (VTC Fall) 1

[10] Roux Y M Le, Ménard J, Toquin C, *et al.* 2009 *Experimental measurements of propagation characteristics for maritime radio links*, 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST) 364

[11] ITU-R M. 2092 2015 *Technical characteristics for a VHF data exchange system in the VHF maritime mobile band*

[12] Forouzan B 2006 *Data Communications and Networking (Satellite Networks), 4th ed.*, The McGraw-Hill Companies

[13] Zhou M T, *et al.* 2013 *IEEE Wireless Communications* **20** (5) 134

[14] Lewczuk M, Cichocki P, Wozniak J 2016 *Telecom. Review + Telecom. News* **12** 1326 (in Polish)

[15] Lewczuk M, Cichocki P, Wozniak J 2016 *Telecom. Review + Telecom. News* **8-9** 734 (in Polish)