Information Sciences and Technologies Bulletin of the ACM Slovakia

Association for Computing Machinery

Advancing Computing as a Science & Profession

June 2019
Volume 11, Number 1

Aim and Scope of the Information Sciences and Technologies Bulletin of the ACM Slovakia

ACM Slovakia offers a forum for rapid dissemination of research results in the area of computing/informatics and more broadly of information and communication sciences and technologies. It is primarily a web based bulletin publishing results of dissertations submitted at any university in Slovakia or elsewhere, perhaps also results of outstanding master theses. Besides that, conferences that meet bulletin's expectations with regard to scientific rigor are invited to consider publishing their papers in the bulletin in form of special issues. Besides the web version of the bulletin, a paper version is available, too.

The Bulletin aims:

- To advance and to increase knowledge and interest in the science, design, development, construction, languages, management and applications of modern computing a.k.a. informatics, and more broadly of information and communication sciences and technologies.
- To facilitate a communication between persons having an interest in information and communication sciences and technologies by providing a forum for rapid dissemination of scholarly articles.

Scope of the Bulletin is:

- original research in an area within the broader family of information sciences and technologies, with a particular focus on computer science, computer engineering, software engineering and information systems, and also other similarly well established fields such as artificial intelligence or information science.

Types of contributions:

- **Extended abstracts of doctoral dissertations.** This is the primary type of article in the Bulletin. It presents main contributions of the dissertation in form of a journal paper together with separate section with list of published works of the author. In Slovakia and the Czech Republic, it corresponds to typical length of so called *autoreferat*. In fact, it is envisaged that publishing the extended abstract in the Bulletin makes *autoreferat* obsolete and eventually can replace it completely. It should be noted that by publishing it in the Bulletin, the extended abstract will receive a much wider dissemination. Exceptionally, at the discretion of the Editorial Board, the Bulletin may accept extended abstracts of other than doctoral theses, e.g. Master theses, when research results reported are sufficiently worthy of publishing in this forum. Rules and procedures of publishing are similar.
- **Conference papers.** The Bulletin offers organizers of interesting scientific events in some area within the scope of the Bulletin to consider publishing papers of the Conference in the Bulletin as its special issue. Any such proposal will be subject of discussion with the Editorial Board which will ultimately decide. From the scientific merit point of view, method of peer reviewing, acceptance ratio etc. are issues that will be raised in the discussion.

Besides that the Bulletin may include other types of contributions that will contribute to fulfilling its aims, so that it best serves the professional community in the area of information and communication sciences and technologies. There are four regular issues annually.

Editorial Board

Executive Editor: *Dominik Macko*
Cover Design: *Peter Lacko*
Typeset in LaTeX using style based on ACM SIG Proceedings Template.

User Management Architecture Based on Software-Defined Networking

Rastislav Bencel*
Institute of Computer Engineering and Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
rastislav.bencel@stuba.sk

Abstract

WLAN networks are widely spread today and represent one of the main accesses to the Internet. Their major benefit is the ability to make mobility of the end-station possible and that is why the goal is to reach an effective user management. In our research we deal with the deficiencies in IEEE 802.11 standard which is aimed at the mobility of the client. There are many ways to deal with these drawbacks. In order to address these shortcomings, it is clear that the new trend in the Software-Defined Networking network is being advantageous. SDN allows more effective management of the network and the research shows that this is true for the primary area of wired networks but also for the wireless. We have extended the SDN architecture for the wireless network based on SDN principles on the basis of the actual research. We have defined new components for encrypting and decrypting WPA2 security in our architecture. We have verified this architecture using Colored Petri Nets and experimental verification in real environment.

Categories and Subject Descriptors

C.2.1 [**Computer- Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.3 [**Computer-Communication Networks**]: Network Operations

Keywords

Network architecture, handover, OpenFlow, IEEE 802.11 standard, Software-Defined Networking

1. Introduction

The original IEEE 802.11 standard [1] was not focused on the mobility of the client but primarily on providing of the connection. The legacy IEEE 802.11 standard uses 4-way handshake within handover process. Moreover, the handover requires also a scanning process before the 4-way handshake. This scanning process takes a long time and its duration is depended on the used method (active or passive). It is not enough secure for wireless medium to extend the IEEE 802.11 standard to increase security (IEEE 802.11i) due to WEP security . The IEEE 802.11i adds additional messages to the original 4-way handshake in the IEEE 802.11 standard. These messages increase the handover time.

The handover time consists of:

- Scanning time - finding a new available access point.
- Authentication time - legacy authentication and new authentication for WPA2.
- Reasociation time - time for exchange reassociation request and response between mobile station and new access point.

The family of IEEE 802.11 standards were extended by standards 802.11k and 802.11r. These standards mitigate drawbacks of a long handover process. The IEEE 802.11k focuses on the decrease of scanning time. The IEEE 802.11r uses only 4-way handshake during handover. Decreasing messages to 4-way handshake is reached by special hierarchy encryption keys and by exchange information before handover. Usage protocols IEEE 802.11k and 802.11r can significantly improve handover time. Although the time is decreased by IEEE 802.11 standard family handover is still decided on the side of a mobile station. The mobile station decision is not efficient for the network because of the limited view of network.

The Software-Defined Networking (SDN) architecture [9] is a new trend in the network area. SDN architecture is centralized and decouples control and data plane. The most important advantage of SDN is more efficient network management than in traditional networks. The primary target of SDN are wired networks but recent research [4] [3] shows that SDN can be used in wireless networks as well.

*Recommended by thesis supervisor: Prof. Ivan Kotuliak Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on August 22, 2018.

Bencel, R. User Management Architecture Based on Software-Defined Networking. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 11, No. 1 (2019) 1-5

The structure of this extended abstract is as follows. The second section describes existing solutions and their drawbacks. The architecture design is presented in the third section and the fourth section describes the evaluation of design. The last fifth section is the conclusion.

2. Related Work

A lot of solutions exist for usage mobility nowadays in the IEEE 802.11 networks. Some of them are not intended for the SDN architecture but principles and ideas can be applied. A good example for this solution is a personal access point [13]. This solution uses a migration of a virtual access point for user mobility. Every mobile station has its own virtual access point. This virtual access point is called a personal access point and is generated during an association process. The handover is performed by removing a personal access point from an old access point and adding it to a new access point. The mobile station does not recognize handover and does not have to send any additional message for performing handover. A decision of handover is made in an access control component which represents a centralized component in traditional network.

The examples of solutions interconnect SDN architecture and IEEE 802.11 standards are [12] [6]. Moreover two main principles using in SDN architecture are recognized for client mobility without its modification. These principles are:

- Personal Access Point.
- All network has the same SSID and BSSID.

The personal access point in SDN architecture is represented by the Odin solution [11]. It uses a light virtual access point (LVAP) which is generated by the Odin master. It is a part of the SDN controller and a the Odin Agent is part of an access point. The Odin master communicates with the Odin Agent via custom protocol. The Odin solution has separated control channels for wired and wireless part. The similar solution to Odin is SWAN [5]. A lot of projects use this Odin solution as a foundation for their research [10]. These extensions take over drawbacks from Odin solutions in the form of separated control channels. A security area is not well defined and it is only mentioned in the Odin solution.

The second principle is defined by a fact that all network has the same SSID and BSSID. The mobile station connects to the network and data are transferred via an available physical access point. The handover is performed by data flow change to a new physical access point. The old physical access point does not send any frame to wireless medium for the mobile station. Drawbacks of this solution are interferences due to overlapping cells in the same radio frequencies. This principle has also limitations in interchannel handover. Representers of this principle are [14][7].

The state of the art shows a lack of architecture design for user management usage SDN concept. On the basis of the acquired knowledge we defined two goals for our thesis. These goals are:

- Modification of centralized architecture for user management without modification of mobile station - handover can not have significant impact on the quality of the connection.
- Proposed protocols to control virtual access points in proposed architecture.

3. Architecture Design

The architecture design is built on a personal access point which is represented by a virtual access point context (VAP context). The parameters of VAP context are:

- Mobile station MAC address.
- Mobile station IP address.
- BSSID.
- SSID.

The VAP context does not contain encryption keys for the WPA2 security as LVAP in the Odin solution [11]. The encryption keys are moved to a specialized component. The architecture use only one control channel between a control layer and an infrastructure layer for the wired and wireless parts of the network. Common control channels have following advantages:

- Clear design and transparency - using standardized SDN interfaces with purpose they were defined. There are no additional interfaces or new protocols implemented into our architecture.
- Higher security due to encryption between the SDN Controller and the SDN forwarding device using OpenFlow protocol. The OpenFlow protocol assumes usage underlay security protocols for increasing security of SDN network management.

3.1 High-Level Architecture

The high-level architecture is depicted in Figure 1 and this figure contains all key components for ensuring IEEE 802.11 functionality. Colored parts of the figure are new or modified components in the architecture. The architecture extends the SDN architecture and contains three main layers:

- Application layer - contains applications for providing network services e.g. authentication, user mobility. Communication between this layer and control layer is ensure via API which depends on control software implementation.
- Control layer - represents the main control element in the architecture. This layer ensures control plane and controls the infrastructure layer. It has to support all IEEE 802.11 extensions of OpenFlow protocol and applications for correct managing IEEE 802.11 functionality.
- Infrastructure layer - is divided into two sublayers due to the fact that architecture contains two transfer mediums. These two sublayers are linked by WTP component. These sublayers are:
  - Transport - represents wired part of the network utilization Ethernet protocol.

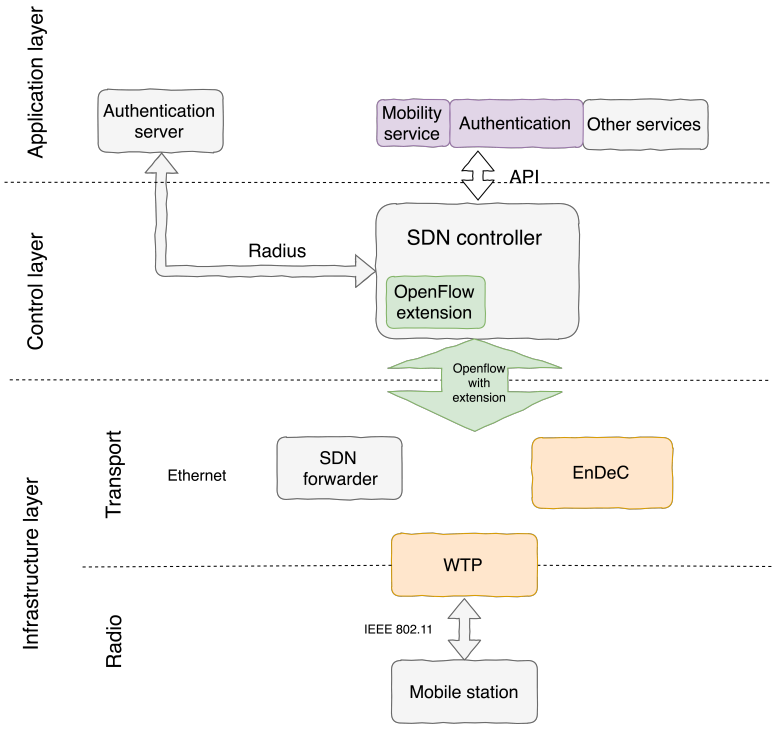

Figure 1: High-level architecture.

– Radio - represents wireless part of the network utilization IEEE 802.11 standard.

The main architecture components are:

- Authentication server - a standard architecture element that performs WPA2 Enterprise authentication.
- SDN controller - represents the main central component that performs network management. It contains applications which perform network services. These application are on the top of the SDN controller. The SDN controller is dependent on the implementation.
- SDN forwarder - represents a standard OpenFlow forwarder without modifications.
- Encryption and Decryption component - specialized hardware that encrypts and decrypts WPA2 Enterprise security.
- Wireless Termination Point - represents an access point that contains an extended OpenFlow protocol and does not have full logic of standard access point logic in IEEE 802.11 networks. Split-MAC access is used.
- Mobile station - represents a terminal station that is not modified in any way.

3.2 Communication Protocol

The communication protocol between the control layer and the infrastructure layer has to ensure correct management of the VAP context in the individual WTP components. We have identified the basics scenarios for proper management of a mobile station:

- Mobile station connection to the network - involves reaching 802.11 associated state, performs WPA2 authentication and assigned IP address for the user station. The user station is available to access to network resources after assigned IP address. The control layer generates the VAP context during this process.
- Handover - add VAP context to a new WTP and remove VAP context from an old WTP.
- Mobile station disconnection - disconnect a mobile station from the network. The control layer removes VAP context from the WTP.

The control layer must also correctly manage the encryption and decryption component in addition to managing VAP contexts. The mobile data without correct decryption are not usable. The control layer must be able to add, remove, update, and delete all encryption keys, depending on the event that may occur.

The communication protocol is implemented using the extension of OpenFlow protocol following specification for extension design [8]. The extension parts are called Experimenter. Within our extension, we have defined new OpenFlow messages, and a new packet type, matches, and a new instruction for generating control messages based on the received IEEE 802.11 frames as part of the WTP Component.

3.3 Association and Handover Process

The WTP sends information about mobile station to SDN Controller after receiving the probe request from the mobile station. This probe request must have the same SSID as the WTP has associated networks. The SDN Controller receives the information about mobile station and

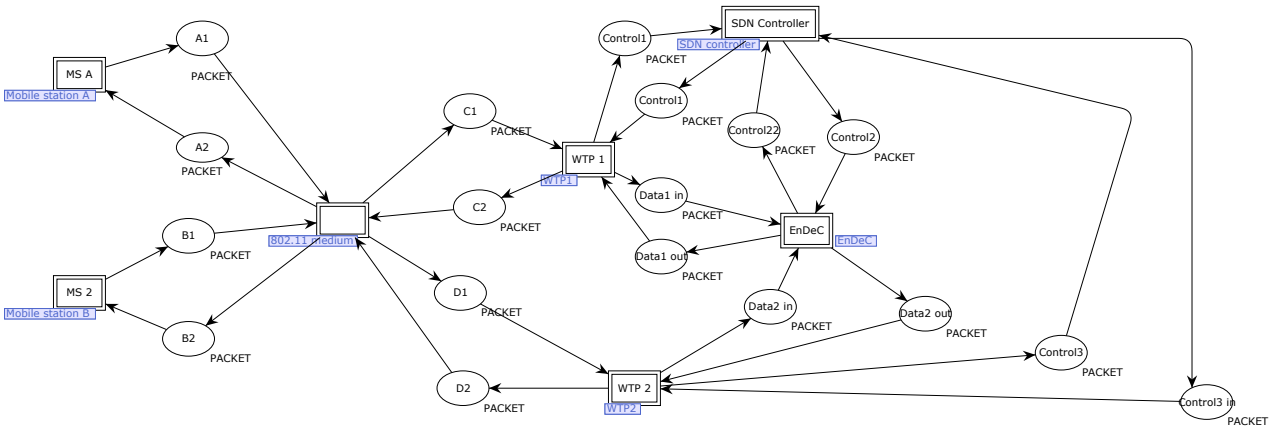

Figure 2: Architecture model in Colored Petri nets.

decides about the next step. If the association process is continuing, so the SDN Controller sends message for adding VAP context to the WTP. On the other hand, the SDN Controller ignores information about the received probe request frame on the WTP. The message for adding VAP context does not contain user station IP address in this step. The IP address is assigned by DHCP server immediately after mobile station perform WPA2 enterprise authentication. All DHCP and EAPOL messages are transferred via the SDN Controller to the DHCP or Authentication server. The WTP and the SDN Controller add the IP address to the user VAP context. The encryption keys are stored in the SDN controller. Performing handover is done on a basis of reported statistic from the WTP. After SDN controller decides to perform handover, It is sends messages for adding VAP context to the new WTP and removing VAP context from the old WTP. The SDN controller has to change traffic flow of mobile station from the old WTP to the new WTP.

3.4 Data Communication

Data communication from the radio sublayer has to pass through an encryption and decryption component that contains encryption keys. The rules for forwarding traffic have to be set to the correct WTP. They change during handover from an old WTP to a new WTP. Data transferred between WTP and an encryption and decryption component has to be tunnelled with Additional Authentication Data header. Tunneled data does not need contain information about used modulation, channel, RSSI etc. The encrypted data has to be transferred to an encryption component also if data communication is used between mobile stations which are connected to the same WTP.

4. Evaluation

We used two methods to evaluate the accuracy of the proposed architecture whose implementation is mutually independent. These two ways are:

- Colored Petri nets.
- Experimental verification of a common control channel.

4.1 Colored Petri Nets

Colored Petri nets were used to verify the entire architecture design to simulate concurrent systems and are therefore a suitable tool for verifying the architecture proposed. The model topology shown in Figure 2 was used to verify the architecture and allowed to verify the main processes within the architecture:

- Mobile station association - the verification was performed on one and two connecting mobile stations.
- Mobile station handover - the handover of the station between WTP during which we monitored the effect of the handover on the transmitted frames.
- Mobile station disconnection - is related to the disconnection from the network.

Individual processes were verified by both methods: using and without the use of the time extension of the colorful Petri nets. Liveness, reachability and boundedness of places have been achieved in both cases. We have evaluated the correctness of the proposed architecture. We monitored the effect on frame rate in time-based measurements. Packet loss and duplicity were affected by the time difference of adding and removing the VAP context to WTP components.

4.2 Experimental Solution

As part of this verification we focused on the ability to use a common management channel effectively to manage the wired and wireless parts of the network. We have found out that it is possible to use one control channel for both parts of the network as part of our measurements in real environment that have been published [2]. We have not found any significant worsening of the parameters due to the handover of the station between the WTP components in our solution compared with the Odin solution. The packet loss from one measurement is shown in Figure 3. In addition, the maximum throughput was not affected due to the station handover.

5. Conclusion

This thesis presents an architecture that allow user management in IEEE 802.11 standard networks using SDN architecture. Interfaces between parts are well defined in the extended SDN architecture. In addition, a control protocol is defined between the SDN controller and

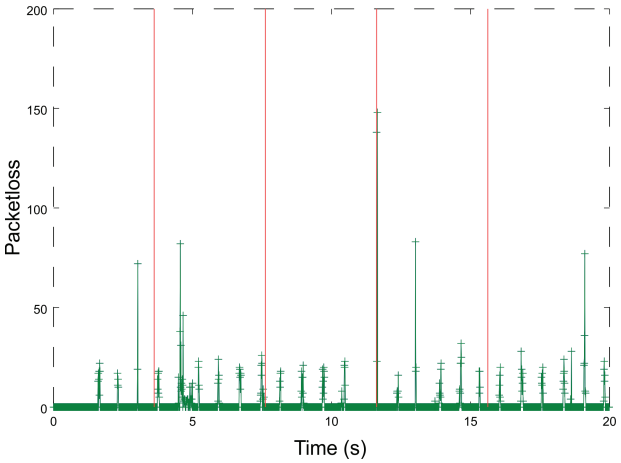

Figure 3: Packet loss in experimental solution with common control channel.

the WTP component, the SDN controller and the cryptographic component. The control protocol was implemented by extended the OpenFlow protocol according to the its specification.

The verification of the architecture was done in two ways. The first method was to verify the whole concept and was realized by using colorful Petri nets. The second method verified the possibility of effective using a common control channel for controlling the wired and wireless part of the network. We came to the conclusion after deeper analysis of both approaches that the design of the architecture is correct.

The proposed architecture presents a good basis for next extensions which can be e.g. to optimize a decision about the handover, using interchannel handover or usage of the newest security standard WPA3.

Acknowledgements. This work is a partial result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF. It is also a part of APVV-15-0731 project and VEGA 1/0836/16, KEGA 011STU-4/2017. The authors would like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

References

[1] IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, 2016.

[2] R. Bencel, K. Kost'al, I. Kotuliak, and M. Ries. Common SDN control channel for seamless handover in 802.11. In *2018 Wireless Days (WD)*, pages 34–36. IEEE, apr 2018.

[3] I. T. Haque and N. Abu-Ghazaleh. Wireless Software Defined Networking: A Survey and Taxonomy. *IEEE Communications Surveys and Tutorials*, 18(4):2713–2737, 2016.

[4] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, jan 2015.

[5] T. Lei, Z. Lu, X. Wen, X. Zhao, and L. Wang. SWAN: An SDN based campus WLAN framework. In *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, pages 1–5. IEEE, may 2014.

[6] S. Monin, A. Shalimov, and R. Smeliansky. Chandelle: Smooth and Fast WiFi Roaming with SDN/OpenFlow. *A Poster Presented at the US Ignite*, pages 31–32, 2014.

[7] K. Nakauchi and Y. Shoji. WiFi network virtualization to control the connectivity of a target service. *Journal of the National Institute of Information and Communications Technology*, 62(2):55–62, 2015.

[8] Open Networking Foundation. OpenFlow Switch Specification Version 1.5.1. *OpenFlow Switch Specification*, pages 1–283, 2015.

[9] O. N. F. W. Paper. Software-Defined Networking : The New Norm for Networks. 2012.

[10] A. K. Rangisetti, H. B. Baldaniya, P. K. B, and B. R. Tamma. Load-aware hand-offs in software defined wireless LANs. In *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 685–690. IEEE, oct 2014.

[11] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao. Towards programmable enterprise WLANS with Odin. In *Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12*, volume HotSDN '12, pages 115–120, New York, New York, USA, 2012. ACM Press.

[12] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo. CloudMAC. In *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12*, page 393, New York, New York, USA, 2012. ACM Press.

[13] L. Zan, J. Wang, and L. Bao. Personal AP protocol for mobility management in IEEE 802.11 systems. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 418–425. IEEE, 2005.

[14] D. Zhao, M. Zhu, and M. Xu. Supporting "One Big AP" illusion in enterprise WLAN: An SDN-based solution. In *2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2014.

Selected Papers by the Author

J. Balažia, R. Bencel, I. Kotuliak. Architecture proposal for seamless handover in 802.11 network. In *2016 9th IFIP Wireless and Mobile Networking Conference (WMNC)*, 95–102, Colmar, France, 2016. IEEE.

R. Bencel, K. Košťál, I. Kotuliak, M. Ries. Common SDN control channel for seamless handover in 802.11. In *WD 2018. 10th Wireless Days Conference*,34–36, Dubai, UAE, 2018. IEEE.

T. Kováčik, R. Bencel, J. Maťo, R. Broniš, P. Trúchly, I. Kotuliak. Enhanced hybrid TV platform with multiscreen, advanced EPG and recommendation enablers. *Journal of Electrical Engineering*,68, pages 224–234.

Effective Multiplatform Firmware Update Process for Embedded Low-Power Devices

Ondrej Kachman*
Department of Design and Diagnostics of Digital Systems
Institute of Informatics
Slovak Academy of Sciences
Dúbravska cesta 9, 845 07 Bratislava, Slovakia
ondrej.kachman@savba.sk

Abstract

Low-power devices can nowadays be found in many systems of collaborating computational devices. They are used in wireless sensor networks, cyber-physical systems, smart systems, etc., and their numbers can reach hundreds. Each system may include devices based on many different platforms in their structure. The devices may be often battery powered and physically inaccessible. It is almost a necessity to enable firmware updates for these devices. Firmware updates are used to add features to firmware, fix problems or change its functionality completely. Battery powered devices with constrained resources require energy efficient firmware update process. We have developed a novel multiplatform process for differential updates of embedded low-power devices. It is independent of the network protocols used, reduces data shared between devices during an update and saves energy on the memory operations required by target devices to finish the update. It supports multiple configurations to adapt to different devices and platforms. The presented solution is suitable for modern intelligent systems that use low-power devices.

Categories and Subject Descriptors

C.3 [**Special-purpose and Application-based Systems**]: Microprocessor/microcomputer applications; D.1.1 [**Programming techniques**]: Applicative (Functional) Programming; D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*portability, version control*

*Recommended by thesis supervisor: Assoc. Prof. Ladislav Hluchý
Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on December 12, 2018.

Kachman, O. Effective Multiplatform Firmware Update Process for Embedded Low-Power Devices. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 11, No. 1 (2019) 6-11

Keywords

firmware, remote update, reprogramming, low-power, embedded

1. Introduction

Modern systems with low-power sensor and actuator devices in their structure may be composed of hundreds of such devices. The devices usually communicate wirelessly as they may be physically inaccessible. Firmware of these devices developed under test conditions may fail once deployed in the field. In such cases, physically collecting each device and reprogramming it with a programming device is undesirable. This can be solved by inclusion of an update module in the original firmware version. The update module can be a part of firmware or a device's bootloader and is responsible for reprogramming of firmware. Firmware updates do not have to be used only for bug fixes, they can also be used to add, remove or modify firmware features. Frequent, incremental reprogramming of a battery powered device could deplete its battery fast and shorten its lifespan significantly. Low-power devices in some systems are expected to run for years on a single battery. An update mechanism developed for these devices should aim to reduce energy consumption of the update process.

This extended abstract presents a multiplatform firmware update process for low-power devices. It uses differential updates to reduce the amount of data shared on a network during an update. Differential updates use delta files, binary files that encode differences between two files. Wireless interfaces of the low-power devices are usually their most energy-hungry components, so the update data reduction helps to save energy. The delta file format is designed to be decoded with linear complexity so the CPU of a device does not waste energy on some complex decompression. Another reduction to energy consumption is achieved by the proposed update module. Erase is the most expensive operation for NAND flash memories. Our update module, Patch module, rewrites altered pages at most once per update. This reduces energy consumption and preserves flash memories that are limited to approx. 10 000 erases. Multiplatform nature of the process is achieved through focus on the object files and no alterations to the code and optimizations generated by compilers.

The rest of the extended abstract is organized as follows. Section 2 describes existing solutions in the problem area. Short description of the design of a novel multiplatform firmware update process is in the section 3. Evaluation

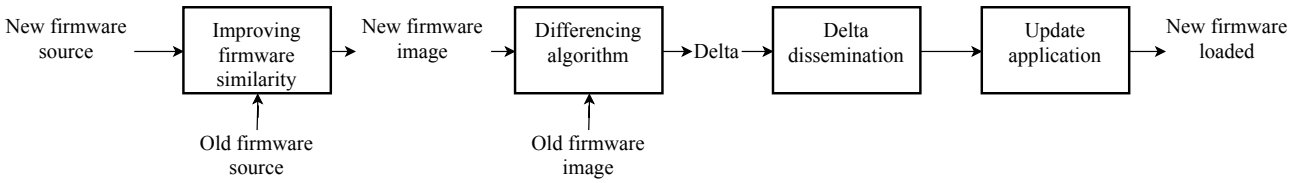

Figure 1: Four main stages of an update process [9].

of the designed update process is described in section 4. Section 5 provides a short conclusion to this paper.

2. Related Work

The area of remote firmware updates can be split to 4 main stages. These stages are shown in Figure 1. Some works in this area are focused on the whole process while some target only some of the stages. The research started with the development of wireless sensor networks (WSNs) and TinyOS sensor operating system in early 2000's. One of the most important early works is the Deluge [6] dissemination and reprogramming protocol. It is primarily focused on the network dissemination stage to reduce number of retransmissions and control messages shared between sensor devices. It detects changed blocks of fixed-size blocks and disseminates them through the whole network. Many works in the area built on this and improved the other stages of the update process.

2.1 Improving Firmware Similarity

Similarity improvement stage aims to produce as similar firmware binaries as possible. More similar binaries for old and new firmware will later result in a smaller delta file, thus less update data. There are multiple approaches to this problem and some of the existing solutions may use only one while others may consist of multiple. The main approaches are:

- **Changes to compilers** - Compiler can be edited to preserve register allocations for variables present in multiple firmware versions [11, 5]. The disadvantages of this approach are that it is highly platform specific and counters compiler optimizations.
- **Handling memory shifts** - New or modified sections can be placed to a new memory location and referenced using proper instructions [13, 16, 15]. Functions that grow will not cause memory shifts. The approach is also platform specific as it inserts CALL or JUMP instructions into the source code.
- **Handling data shifts** - Data in RAM memory can also be reorganized in every firmware version. It is possible to change layout of the RAM sections to counter these changes [12, 15]. However, these works cannot guarantee that data will not shift.
- **Handling relocatable code** - Instructions that reference various memory locations (relocatable entries) can be set to the same value and later filled in by a loader [2, 3]. Making these entries the same helps to detect more common code sequences.
- **Memory fragmentation** - Memory can be fragmented to provide sections with space to grow and shrink [10]. The space is called a slop region. The disadvantage of this approach is less data space left on a device.

2.2 Differencing Algorithm

A differencing algorithm compares an old firmware binary with a new firmware binary. It detects common sequences and new sequences. Using these sequences, it encodes a delta file. Delta files consist of delta operations that are translated to a physical memory operations on a target devices. Lighter differencing algorithms are block based - easier to implement but require more update data [6, 14]. More complex differencing algorithms are byte or word-based [3, 7, 4]. Basic operation types for delta files are:

- COPY - moves existing sequences
- ADD - adds new sequences
- INSERT - inserts a single data unit
- PAD - pads memory with a specified data unit

2.3 Delta Dissemination

The dissemination stage is heavily network focused and a research area of its own. The task is to transfer current delta file to all target devices in the network securely and reliably. A lot of research went into design of the protocols specialized for firmware reprogramming like Deluge [6, 13, 12]. In recent years, technology standards and standard protocols have been developed even for the systems that use low-power devices. The research shifted towards upper layers and determination of the most effective update trees [1, 17]. The modern research for the other three stages of the firmware update process should be agnostic to the networking technologies and protocols used to propagate the update data.

2.4 Update Application

The update module on a target device is responsible for application of an update. The operation should not take long and waste energy on memory operations. Also, rewriting the same page of a flash memory multiple times during an update reduces the memory's durability. Standard approach is to rebuild firmware in an external memory, reset the device and load the firmware from the external memory in bootloader [2, 3]. Modern devices may not have an external memory and must use their internal program memory to rebuild the firmware image. This is sometimes referenced as on-the-fly update [15, 8]. During on-the-fly update, firmware can be rebuilt completely in another part of the program memory or pieced together right in its location using swap space. While the first approach is more reliable, the latter requires less memory space and can be more suitable for devices with limited-size memories.

3. Design of a Multiplatform Solution

This section describes our multiplatform firmware update process. It is based on the state-of-the-art analysis and introduces new configurations and optimizations. During the analysis stage, it was identified that there are no

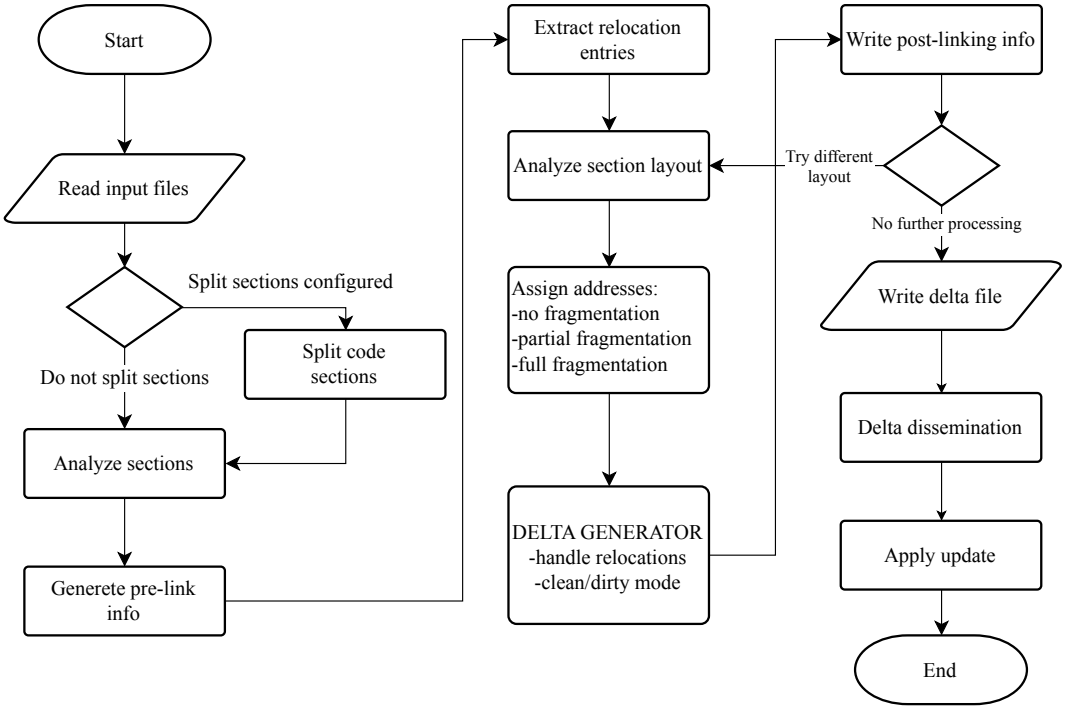

Figure 2: General flowchart of the designed firmware update process.

multiplatform solutions. The main objective was to design and evaluate a multiplatform approach that would include all the best practices and ideally improve them. The main hypothesis regarding the multiplatform nature of the designed solution was:

If the remote firmware update process does not alter the source code of a firmware but rather operates on object files and various other data, then it has the full potential for multiplatform use.

The general flowchart of the designed solution is shown in Figure 2 and the following subsections describe the processes, configurations and decisions that are included in the solution.

3.1 Reading Input Files and Analyzing Code Sections

The process starts with analytical algorithms. The first one reads, analyzes and possibly slightly alters input files. The input files are all firmware object files produced by the used compiler. These files include optimized code and data sections along with additional sections that contain various information. The algorithm is responsible for identification of the code sections and their organisation. It supports two configurations:

- **Default sections** - The algorithm identifies default sections. Some of these sections may be defined in source code by firmware developers. They can be split to modules and group multiple functions into the same sections. This configuration gives the developers more control over defined sections and modules.
- **Split sections** - The algorithm places every firmware function into its own section that can be later put to a specified physical address by the linker. This configuration automates the process and does not require edits to the source code of the firmware.

3.2 Collecting Pre-linking Information

This algorithm works with the code sections prepared in the previous step and with the post-linking information of an old firmware version if there is one. If there are no previous versions, the only data required are the section sizes. If there is a previous firmware version, the following additional data are collected:

- New, removed, modified sections
- Size change of the modified sections
- Memory shifts caused by modified sections
- Section order changes
- Slop region sizes

3.3 Extracting Relocation Entries

This algorithm is adopted from [2] and modified. In order to make firmware images more similar, relocation entries that reference memory locations can be set to the same value. Our algorithm only resets the changed entries and encodes them as INSERT delta operations before delta optimization. In [2], the entries were compressed and added as metadata to delta files. Our algorithm managed to reduce the data required but the approach did not show positive results.

3.4 Analyzing Section Layout

Based on the pre-linking information, this algorithm detects any changes in the section order compared to a previous version. Changes can be caused by the source code

refactoring or addition of new sections. The order is corrected to counter the memory shifts of the unchanged data.

3.5 Assigning Section Addresses and Linking

This algorithm assigns physical addresses to the code sections, generates a linker command and runs the linker. While assigning the section addresses, three methods are supported [8]. The algorithm can be configured to use any of these methods:

- **No fragmentation** - There are no slop regions provided to sections
- **Partial fragmentation** - The most edited sections are provided with a slop region
- **Full fragmentation** - All sections are provided with a slop region

Slop regions significantly reduce memory shifts. However, if the firmware will not be updated many times, they are undesirable as they fragment memory and reduce data space. If there are many incremental updates, firmware can be defragmented once the updates are done. It costs additional operations but the firmware is cleaned up and data space expanded.

3.6 Delta Generator

Delta generator (DG) [7] is our original differencing algorithm. It is a word-based algorithm that supports variable word size. NAND flash memories are currently the most used program memories for low-power devices. Word is the smallest write unit of these memories. Word size can vary between different hardware platforms so the variable size word support is necessary for a multiplatform solution.

DG detects matching and non-matching segments of a firmware. Matching segments are left in their memory locations. For each non-matching segment, DG attempts to find matching sequences in the different locations of an old firmware version. Once complete, DG prepares the lists of common and changed code sequences. These are encoded as delta operations. DG supports following operations:

- SKIP - skips over an unmoved matching sequence
- COPY - moves matching sequence to a new location
- ADD - inserts a new sequence into the memory
- INSERT - inserts a single word or a relocation entry

After the operations are generated, DG tries to optimize out some small and redundant operations by merging them into ADD operations. When finished, the operations are encoded into their binary form from the lowest target address to the highest address. Merging these binary data, calculating a CRC code for them and appending the code to the end results in the final binary file - the delta file (Figure 3). This file is ready for the network dissemination stage.

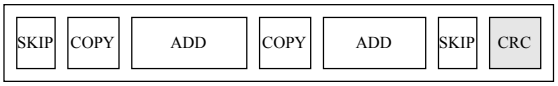

Figure 3: Simple representation of the delta file structure.

3.7 Write Post-linking Information

With the delta file prepared for the network dissemination, it is important to store some data about the current firmware version for future updates. Physical addresses assigned to the sections are stored along with the info about current size of the assigned slop regions. At this point, the process can return to the section order analysis algorithm and try out a different addressing method. This is used to compare, how the different configurations influence the delta file size. The delta file is then passed to the application that starts the update dissemination stage.

3.8 Delta Dissemination

Our process was designed to be agnostic to a networking protocol used. Delta files generated by DG can be disseminated using any standard protocol. It requires a server application that will connect to the target devices and transfer the delta files to them. Target devices must implement the necessary communication and support the used protocol. Dissemination stage is out of scope of this work.

3.9 Application of an Update

Patch module is our update module for target devices. It is designed to decode the delta files and to apply delta operations in the program memory. Decoding process has linear complexity so the Patch module does not waste CPU time and device's energy. It also rewrites each program memory page at most once per update to save energy on erase operations. Patch module has two stages:

1. Handle overlapping delta operations
2. Process delta operations

Our process is designed to apply an update on-the-fly and right in the space reserved for the firmware. This means that some data could be rewritten before a COPY operation would move them to another location. During the first stage of the Patch module, the algorithm iterates through the delta file and lists any COPY operation that has its data rewritten by any of the previous operations. The data that these COPY operations use are copied into a swap space that must be reserved on the target devices. The affected COPY operations are then rerouted through the swap space.

During the second stage, the Patch module starts to rebuild the firmware. It executes operations as they are stored in the delta file, but the pages are first reconstructed in the RAM memory. Once a page has been reconstructed, it is physically erased and written back from the RAM. No page has to be erased twice this way. Overlapping operations use data from the swap space. Once the update is finished, swap space must also be erased. After the update is finished, the Patch module can start the firmware itself or reset the device and let bootloader start the ex-

ecution of the new firmware version. This concludes the whole firmware update process.

4. Evaluation

We evaluated the designed firmware update process on three different hardware platforms. All of the tested platforms are used in low-power systems but have very different specifications. The experiments aimed to show that our solution can be applied to these platforms without implementation of any platform specific code. Also, the successful results confirm the main hypothesis. We used object files in ELF object file format and GCC compiler ported for each platform. The tested platforms are:

- 8-bit ATmega32u4 microcontroller from Atmel
- 16-bit MSP430 microcontroller from Texas Instruments
- 32-bit ARM Cortex-M4 processor on a bluetooth low-power system-on-chip from Nordic Semiconductor

For ATmega platform, we used a simple custom firmware. For MSP430 and ARM platform, we used example firmware from software development kits provided by the manufacturers. The changes made to firmware were:

1. Changed value of a constant
2. Added a new section
3. Changed a section
4. Removed a section
5. Added multiple sections
6. Changed multiple sections
7. Removed multiple sections
8. Changed order of some sections
9. Added and removed multiple sections
10. Added, changed, removed and reordered multiple sections

These test cases are the same for each tested platform. We compared our DG delta file size to differencing algorithms RMTD [4] and R3diff [3]. Delta file size percentages compared to the full firmware image size for all platforms are shown in Figures 4 - 6. DG can reduce delta file size by more than 90% compared to the full firmware image and also improves over existing methods by 5-50%. Additional algorithms for improved firmware similarity and various configurations further shrink the delta size by up to another 50%. Delta generator is able to produce delta files with just 25% of the size when compared to other methods (for example case 5 on ARM, Figure 6). For all of the change cases, the delta files successfully reconstructed the desired firmware image and did not erase any memory page more than once per update. This reduces energy consumption during an update significantly.

5. Conclusion

This extended abstract describes a novel multiplatform firmware update process. Its universal use for different hardware platforms is its main contribution. It also introduces various new configurations, new methods to handle and organize firmware sections, an update module that preserves memory durability of the low-power devices and a new differencing algorithm. With all of its algorithms, the designed solution improves over existing solutions. It was tested on three different platforms and showed mostly positive, in some cases even significant results. The presented update process is suitable for the modern networked systems that use low-power devices in their structure.

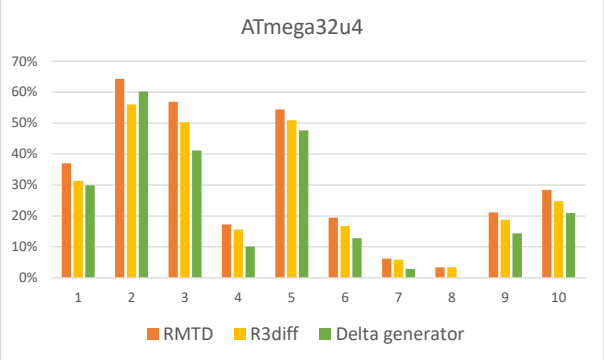

Figure 4: Delta file size percentages for the tests on the ATmega.

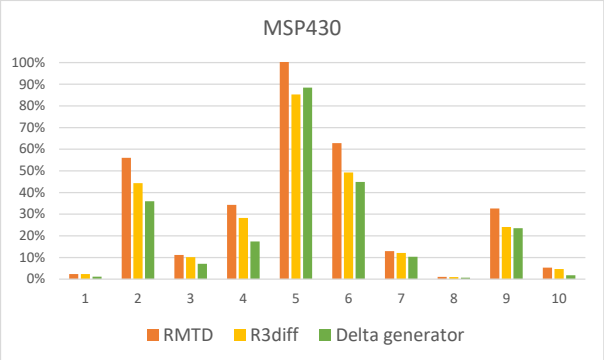

Figure 5: Delta file size percentages for the tests on the MSP430.

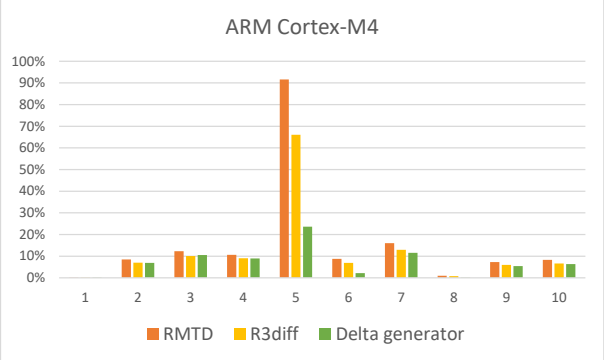

Figure 6: Delta file size percentages for the tests on the ARM Cortex-M4.

Acknowledgements. This work has been supported by Slovak national project VEGA 2/0192/15 and by the ECSEL Joint Undertaking (JU) under grant agreement No 737434.

References

[1] H. Asahina, I. Sasase, and H. Yamamoto. Efficient tree based code dissemination and search protocol for small subset of sensors. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 283–288, May 2017.

[2] W. Dong, Y. Liu, C. Chen, J. Bu, C. Huang, and Z. Zhao. R2: Incremental reprogramming using relocatable code in networked embedded systems. *IEEE Transactions on Computers*, 62(9):1837–1849, Sept 2013.

[3] W. Dong, B. Mo, C. Huang, Y. Liu, and C. Chen. R3: Optimizing relocatable code for efficient reprogramming in networked embedded systems. In *2013 Proceedings IEEE INFOCOM*, pages 315–319, April 2013.

[4] J. Hu, C. J. Xue, Y. He, and E. H. . Sha. Reprogramming with minimal transferred data on wireless sensor network. In *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pages 160–167, Oct 2009.

[5] Y. Huang, M. Zhao, and C. J. Xue. Wucc: Joint wcet and update conscious compilation for cyber-physical systems. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 65–70, Jan 2013.

[6] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 81–94, New York, NY, USA, 2004. ACM.

[7] O. Kachman and M. Balaz. Optimized differencing algorithm for firmware updates of low-power devices. In *2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, pages 1–4, April 2016.

[8] O. Kachman and M. Balaz. Configurable reprogramming methodology for embedded low-power devices. *IFIP Advances in Information and Communication Technology*, 499:211–219, 2017.

[9] O. Kachman and M. Balaz. Firmware update manager: A remote firmware reprogramming tool for low-power devices. In *2017 IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, pages 88–91, April 2017.

[10] J. Koshy and R. Pandey. Remote incremental linking for energy-efficient reprogramming of sensor networks. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.*, pages 354–365, Feb 2005.

[11] W. Li, Y. Zhang, J. Yang, and J. Zheng. Ucc: Update-conscious compilation for energy efficiency. In *Wireless Sensor Networks, ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI)*, 2007.

[12] R. K. Panta and S. Bagchi. Hermes: Fast and energy efficient incremental code updates for wireless sensor networks. In *IEEE INFOCOM 2009*, pages 639–647, April 2009.

[13] R. K. Panta, S. Bagchi, and S. P. Midkiff. Zephyr: Efficient incremental reprogramming of sensor nodes using function call indirections and difference computation. In *Proc. of the 2009 Conf. on USENIX Annual Technical Conf.*, USENIX'09, pages 32–32, Berkeley, CA, USA, 2009. USENIX Association.

[14] J. Qiu, S. Li, and B. Cao. Repage: A novel over-air reprogramming approach based on paging mechanism applied in fog computing. 2018, 2018.

[15] N. B. Shafi, K. Ali, and H. S. Hassanein. No-reboot and zero-flash over-the-air programming for wireless sensor networks. In *2012 9th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 371–379, June 2012.

[16] C. Zhang, W. Ahn, Y. Zhang, and B. R. Childers. Live code update for iot devices in energy harvesting environments. In *2016 5th Non-Volatile Memory Systems and Applications Symp. (NVMSA)*, pages 1–6, Aug 2016.

[17] Z. Zhao, J. Bu, W. Dong, T. Gu, and X. Xu. Coco+: Exploiting correlated core for energy efficient dissemination in wireless sensor networks. *Ad Hoc Networks*, 37:404 – 417, 2016.

Selected Papers by the Author

O. Kachman, M. Baláž. Optimized differencing algorithm for firmware updates of low-power devices. In *Formal Proc. of the 2016 IEEE 19th Int. Symp. on Design and Diagnostics of Electronic Circuits and Systems, DDECS 2016*, 2016. IEEE.

O. Kachman, M. Baláž. Firmware Update Manager: A remote firmware reprogramming tool for low-power devices. In *Proc. - 2017 IEEE 20th Int. Symp. on Design and Diagnostics of Electronic Circuit and Systems, DDECS 2017*, 2017. IEEE.

O. Kachman, M. Baláž. Configurable reprogramming methodology for embedded low-power devices. In *Technological Innovation for Smart systems*, volume 499 of *IFIP Advances in Information and Communication Technology*, pages 211-219. Springer, 2017.

Software Defined Networking in Wireless Mobile Networks

Martin Nagy*
Institute of Computer Engineering and Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
martinko.nagy@gmail.com

Abstract

Thesis focuses on the topic of Software Defined Networking (SDN) in context of 3GPP (Third Generation Partnership Project) track mobile networks. Software defined networking is a trend that is slowly making its way to most areas of computer networking. Nowadays it is visible mainly in datacenter networking, private clouds and 5G. However other technologies, such as 2G, 3G, LTE or Wi-Fi can benefit from it as well. We proposed a new architecture for GPRS (General Packet Radio Service) delivery (packet based 2G data). Architecture is designed to be backwards compatible with the radio access network which shall simplify the deployment. 2G may seem as an obsolete technology, however it is still widely used worldwide, thanks to its maturity and low production costs. Moreover it is a great fit for some of today's IoT (Internet of Things) use cases. Integral part of the new architecture is a new tunneling approach called MAC tunneling, which replaces GTP (GPRS Tunneling Protocol) tunneling. The architecture is deployable not only with GPRS, but is generalized to be used with any other access technology. Solution proof of concept is built and tested with standard, not-modified 2G base station and terminal in order to practically evaluate the architecture.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; C.2.3 [**Network Operations**]: Network management

*Recommended by thesis supervisor: Prof. Ivan Kotuliak Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on April 29, 2019.

Nagy, M. Software Defined Networking in Wireless Mobile Networks. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 11, No. 1 (2019) 12-20

Keywords

GPRS, SDN, NFV, Network Functions Virtualization, Software Defined Networking, UnifyCore, OpenFlow, mobile networks, 3GPP networks, wireless networks

1. Introduction

SDN (Software Defined Networking) is mostly visible in areas of fixed networks, mainly core networks and datacenter networking to name a few. However, in the area of cellular mobile networks, the SDN is just getting traction, mostly thanks to the advent of 5G networks, which promise high utilization of SDN related technologies, for example to provide network slicing [8]. While first standard 5G networks are nowadays being deployed, our thesis focuses on legacy cellular networking - networks which are already widely deployed and used by the majority of the population.

Our SDN approach was developed on the basis of 2G data networks (GPRS – General Packet Radio Service) and later extended/generalized to other, even non-cellular access technologies.

OpenFlow protocol was used to bring SDN capabilities to our concept. We chose this protocol since it is open, well standardized and moreover, plethora of open-source OpenFlow implementations exist. This enabled practical experiments and evaluation of the whole concept.

2. Mobile Networks

GPRS network is an add-on to the ubiquitous 2G voice network (GSM – Global System for Mobile Communications). Voice services were not the focus are of our thesis, therefore we will omit them going forward with this document. For further information on the GSM architecture and technologies, we encourage reader to study publicly available 3GPP (Third Generation Partnership Project) standards [5].

The 2G packet network added two new elements to the core network of already existing GSM network and required minor, mostly software changes in the radio access network [3, 2]. These nodes are SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node). SGSN is responsible for mobility and session management of the connected mobile stations / devices (MS). These include procedures such as authentication, location tracking, even traffic encryption and decryption. SGSN directly interfaces with the radio access network (RAN)

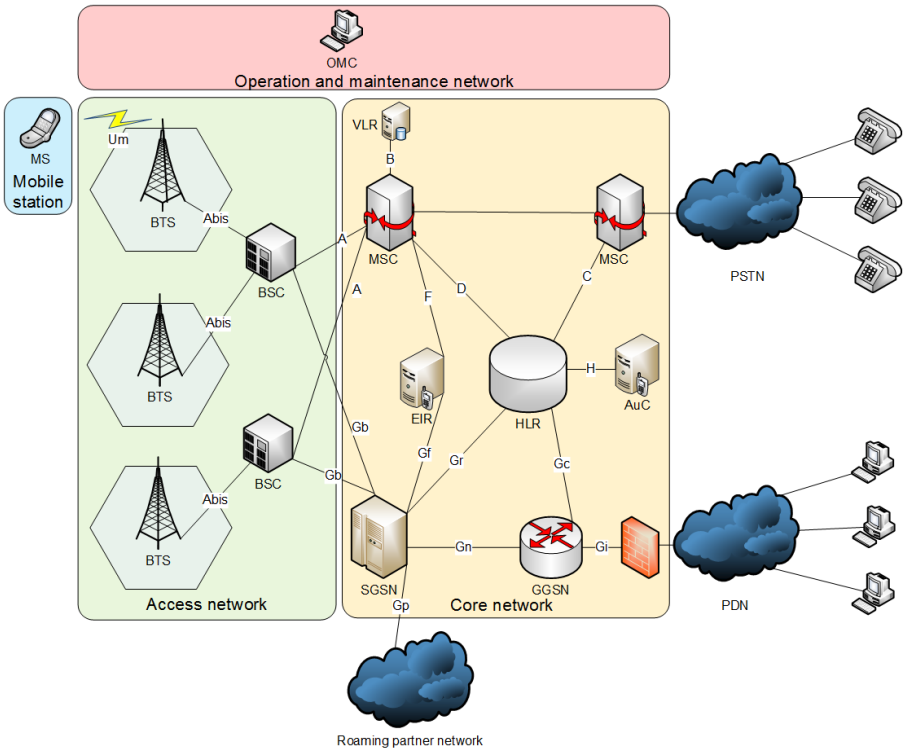

Figure 1: Standard 3GPP GSM/GPRS architecture [2].

by Gb interface, which connects SGSN and BSC (Base Station Controller), the brain of the 2G access network.

GGSN connects to the SGSN on one side and to the PDNs (Packet Data Networks) on the other side. These can be for example various corporate intranets, public Internet or even various dedicated application servers (e.g. MMS server, operator specific platforms). GGSN exchanges information about the mobile station's session state with SGSN. This is mostly session establishment, modification and teardown. In a nutshell, GGSN acts as protocol translator between the "mobile" world and the external networks. On top of that, since the mobile station's connection is terminated on the GGSN, it serves as a connectivity anchor and assigns IP address to the connected mobile station.

Connection type (or service type in other words) is defined in mobile networks by its APN (Access Point Name). APN determines what service does the mobile station want to access, what are the connection parameters (e.g. dynamic or static IP address allocation) etc. The connection/session in mobile networks language is denoted as PDP Context (Packet Data Protocol Context). It is basically a virtual connection from the mobile station, through the whole mobile network to the service, identified by various entities at different protocol levels (e.g. Tunnel End Point ID for the GPRS Tunneling Protocol level, Temporary Logical Link Identifier on the Base Station Subsystem GPRS Protocol level, etc.). High level architecture of GSM/GPRS network is depicted on Figure 1.

Since the days of 2G data networks, network operators have deployed 3G (UMTS – Universal Mobile Telecommunications System) and its evolutions (HSPA – High Speed Packet Access, HSPA+ – Evolved High Speed Packet Access), 4G networks (LTE – Long Term Evolution) and its evolutions (LTE-A – LTE Advanced, LTE-A Pro – LTE Advanced Pro). Currently first 5G networks are being deployed across the globe [17, 9].

3. Software Defined Networking

SDN is in contrast with what is nowadays usually present in production communication networks. The network gear (e.g. routers, switches or application gateways) combine the network logic, management and data plane in one box. These boxes are usually managed though vendor specific command line interfaces (CLI) or by management application of the given vendor.

SDN decouples control plane and user plane of the networking gear and introduces standardized interfaces between the two. By using feature set which is configurable by a well-defined protocol, it enables high vendor interoperability and on top accelerates innovation, since data plane and control plane can evolve each at its own pace (don't need to evolve together as in case of traditional network gear).

The SDN approach used in the thesis is OpenFlow [1]. It introduces simple model of packet processing. Each OpenFlow switch carries multiple OpenFlow tables (Figure 2). Tables are filled with OpenFlow rules as instructed by the SDN controller. The rules are composed of match field, against which the packet content is being compared (e.g. destination IP address value, UDP port, etc.), actions and instruction which shall be executed on the packet if it is matched.

When a packet is received by the switch, it enters the first table and is being compared with the rules within

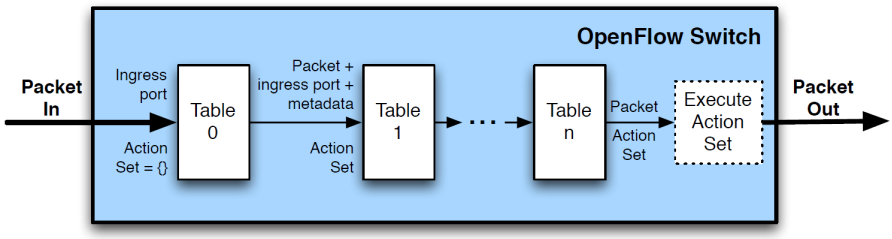

Figure 2: OpenFlow packet processing pipeline [1].

the table. If the packet matches given rule, action set is being added to its metadata and finally being executed. If the packet is not matched in any table, OpenFlow switch either drops the packet or forwards the packet to the controller, where controller logic may decide what to do with it. Please note that OpenFlow switches do not carry any of the logic that standard switch has (i.e. no MAC address learning or flooding mechanisms). OpenFlow switch simply executes only the rules that are installed by the controller (and indeed in cooperation with the SDN controller can mimic behavior of standard switches or routers).

In addition to match rules, actions and instructions, there are also counters, which are incremented by rule hit and also a possibility to use metering tables. OpenFlow features are being extended by every OpenFlow standard release and also allow to use so called experimenter actions, which anybody can use while experimenting with new, not yet standardized network approaches.

It needs to be noted, that there do exist many other SDN approaches and OpenFlow is not the only one on the market, however it seems to be the frontrunner.

For example there is PCE (Path Computation Element) approach [24], which centralizes route computation and optimization in the distributed environment of computer networks. It basically offloads routers from CPU intensive task of route computation and optimization, while using also traffic engineering. Path computation clients (e.g. routers) communicate with the path computation element by standardized Path Computation Element Protocol (PCEP).

Perhaps a competitor to OpenFlow in OpenStack based internal clouds, there is Juniper's Contrail solution [16], which uses XMPP (eXtensive Messaging and Presence Protocol). In OpenStack environments, Juniper's vRouter replaces vanilla OpenVSwitch and Contrail controller replaces standard Neutron (OpenStack default networking) module. Contrail and vRouter communicate via XMPP, while OpenVSwitch and Neutron though OpenFlow (and OVSDB – OpenVSwitch DataBase).

There are also SDN approaches, which did not gain commercial traction over the time (unlike OpenFlow). For example ForCES (Forwarding and Control Element Separation) [4] is one of the oldest standards, which consists of several RFC documents. Forces formally introduces the notion of forwarding elements, control elements, forwarding element managers and control element managers and their interfaces. Also, it decouples forwarding elements to logical functional blocks. Due to its complexity, there exist only few projects, which implemented ForCES and it was finally overcome by much simpler, however not that elaborative approaches (e.g. OpenFlow).

4. Related Work

Research and development in the area of 3GPP mobile networks has been led by network infrastructure and chipset vendors such as Ericsson, Nokia, Huawei, Qualcomm, Cisco to name a few, partially in cooperation with mobile network operators.

This is due to the fact that mobile networks ecosystem, although with publicly available specifications is quite complex and up to now, there were almost no open-source implementations of mobile nodes that would enable academic community to execute practical evaluations of their novel and experimental approaches.

However nowadays (i.e. with advent of 5G networks), the situation seems to be changing and both operators and network equipment vendors are taking advantage of open-source projects and other innovative technologies rooted outside of their usual ecosystem.

For example Huawei introduced a MobileFlow protocol based architecture for 3GPP mobile networks. They invented this new protocol (althought most likely on the basis of OpenFlow) and built architecture around it (Figure 3), with MobileFlow forwarders similar to OpenFlow ones, but considerably more feature rich. Supporting for example also charging and GTP (GPRS Tunneling Protocol) processing [22].

Ericsson on the other hand took a more conservative approach with extending standard OpenFlow protocol and OpenFlow forwarders with GTP support. Ericsson argues that today's carrier networks rely on double routing and forwarding decisions – one taken on the underlay layer (IP/MPLS routing), second on the overlay layer (GTP routing). Centralized routing decision – i.e. IP/MPLS and GTP logic centralized in a single point shall simplify network design, optimize routing decisions and accelerate failover [18].

In the area of academic research, some papers are still being coauthored with the network gear vendors or operators [10]. As for purely academic research, there is for example an interesting paper from German researchers that builds a theoretical model of the mobile network and evaluates benefits of SDN and NFV (Network Functions Virtualization) [19]. Other papers focus on the related topics to SDN in mobile networks such as QoS [13] or its benefits to other technologies such as CDN (Content Delivery Networks) [11].

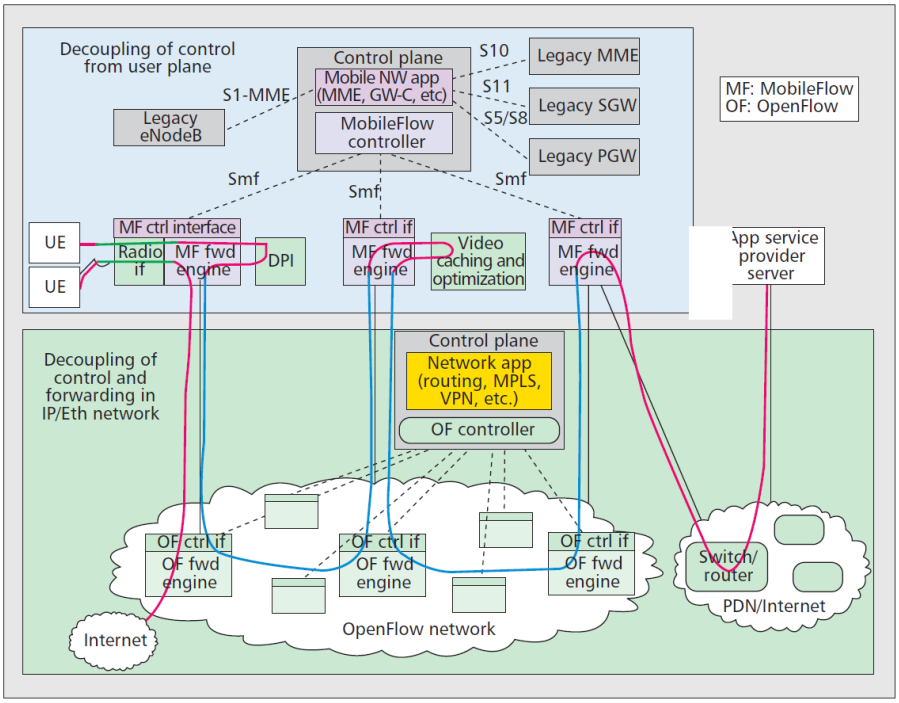

Figure 3: Huawei's MobileFlow architecture [22].

Going forward the number of academic papers focusing on mobile networks is growing, however with focus on 5G, possibly 4G. There is little traction for the 3G networks and no traction at all in the area of 2G networks. Despite that we think that applying SDN to 2G mobile network will be beneficial, as this network is still operational worldwide with many terminals in network supporting this technology exclusively. Although 2G, or GPRS to be precise does not fulfill the requirements of today's interactive multimedia applications (e.g. throughput, delay), it can be (and is) still used for example in context of IoT (Internet of Things) for latency and throughput non demanding use-cases, for example telemetry.

5. Thesis Goals

As pointed out in the previous chapter, there are many projects – both from the industry and academia that focus on SDN in the area of 5G networks. Also LTE (4G) networks are being well used to provide use cases for SDN deployment, which is visible also in related publications. We are not aware of any related work in regards to SDN in combination with GPRS (2G networks).

However in our opinion, second generation of mobile data networks is still relevant. According to the GSMA's Mobile Economy 2018 report and forecast, 40% of mobile connections globally (out of 7.79 billion) were with 2G technology. In developing regions, the number is even higher (e.g. Sub-Saharan Africa with 60% share of second generation network) [14] [15].

It may be a fact that most of the future investments in regards to mobile networks will go towards either LTE extensions or new deployment of 5G. Thus making SDN research and development in the area od legacy networks economically not feasible for equipment vendors. However GPRS network still remains in heavy use worldwide, serving low speed and high latency data. GPRS ubiquity, its low price and supported features can be and actually are used for data access where there is no other technology, or the connection features are good enough (e.g. already mentioned telemetry). Therefore our thesis will focus on SDN deployment in GPRS networks.

Thesis goals can be broken down into several incremental steps.

- *Design and proposal of the new method for GPRS control and data / user plane separation.* In order to deploy software defined networking in GPRS architecture, control and user plane need to be separated. Unlike UMTS or LTE, in GPRS are these information transported in a single stream of data.
- *Design and proposal of new GPRS architecture based on the SDN approach.* Once there are available separated streams of control and user plane data, software defined networking can utilized.
- *Design and proposal of a simplified architecture for delivery of GPRS services.* Having SDN integrated in the mobile network, architecture will be simplified with emphasis on the core network.
- *Enhancement / generalization of the overall architecture for other access technologies.* With SDN controller in place and rest of the network having programmable interfaces, architecture will be extended not to provide GPRS only data service, but also to serve other access technologies while maintaining common transport core and control.
- *Verification and evaluation of the new SDN based architecture.* After the concept is complete, one access technology will be selected and proof of concept for practical evaluation will be built. Also analytical evaluation of selected feature of the new architecture will take place.

6. Proposed Architecture

New SDN enabled architecture takes the existing standard 3GPP GPRS as a baseline. Architectural changes are made with backwards compatibility in mind. This applies to the new SDN enabled core compatibility with legacy and standard radio access network. The rationale behind this focus is to ease deployment of such solution as much as possible. If any changes in the radio access network would be required due to the SDN architecture deployment, it would be very hard to implement these changes cross whole radio access network, since it usually consist of multiple hundreds or even thousands base stations which are geographically spread (depending on the network operator footprint and also country terrain profile).

New architecture is depicted on Figure 4. GGSN and SGSN were removed from the architecture and their functions were spread cross new nodes in the architecture – ePCU (enhanced Packet Control Unit), SDN controller, vGSN (virtual GPRS Support Node) and OpenFlow based forwarding core. Since there is no GGSN or SGSN anymore, also GTP (GPRS Tunneling Protocol) is not used in the network. The only exception to GTP use is interfacing with legacy network operators for roaming purposes. However, if interfaced with SDN enabled domains/operators only, no GTP is needed. Instead of GTP, Ethernet II header is being reused for tunneling purposes – we call this approach MAC tunneling.

6.1 Signaling and User Plane Separation

As mentioned in previous section, in order to fulfil all subsequent thesis goals, first we need to extract signaling and user data plane from the joint stream of data. A new network element, which will execute this separation – ePCU was deployed on the interface between the radio access network and the core network - Gb.

Signaling messages on this interface are related either to SGSN-MS signaling (e.g. mobility or session management) or SGSN-BSC signaling (radio access network management). Separation of SGSN-BSC communication is done on the GPRS-NS (GPRS – Network Service) protocol level, where all messages except NS-UNIDATA are being forwarded to the network element responsible for signaling handling (in our case this is vGSN, in standard 3GPP architecture this is SGSN). MS-SGSN signaling – data separation is being done on a higher level – GPRS-LLC (GPRS – Logical Link Control) layer. LLC SAPI (Service Access Point Identifier) value determines the type of the payload – LL3, LL5, LL9 and LL11 being the user data related SAPIs and other values indicating signaling data.

6.2 New Nodes in the Architecture

ePCU is basically a special kind of OpenFlow forwarder, which understands GPRS protocols on the Gb interface and is able to identify and separate signaling from user plane data. Signaling is forwarded to another new node – vGSN, user plane traffic is handed over to OpenFlow transport core.

vGSN processes only signaling messages (unlike SGSN), either mobile station signaling or BSC signaling. vGSN interfaces also with SDN controller, which is responsible for whole core network, but also assists during authentication procedures or session management procedures. Actual session establishment and construction of the network path (towards Internet for example) is fully with the SDN controller.

SDN controller interfaces with the access network management and signaling node (vGSN), but also with the core transport network (OpenFlow forwarders and ePCU) and subscriber databases, in order to have subscription management logically centralized.

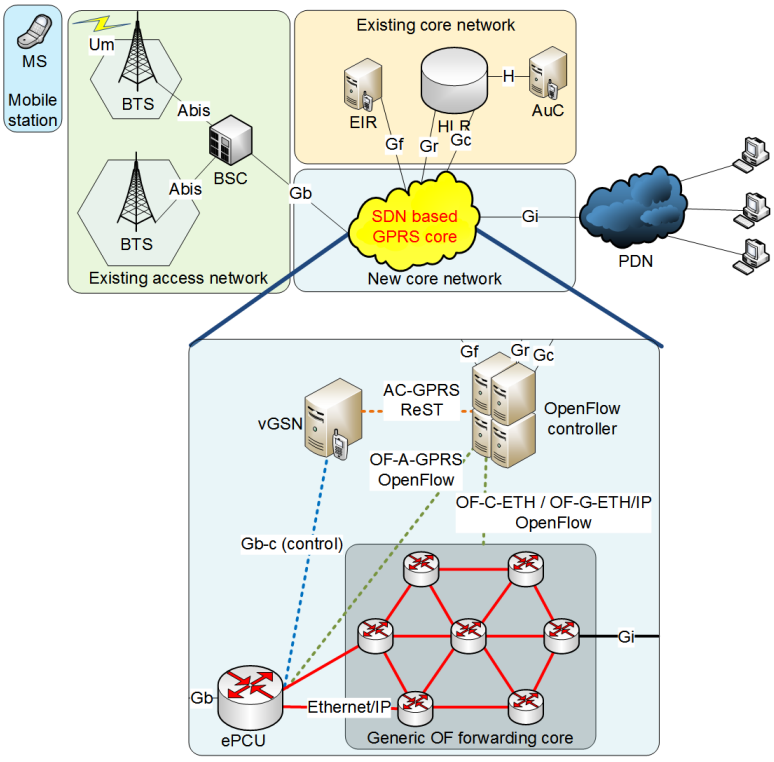

Figure 4: New SDN GPRS architecture.

Transport core is based on OpenFlow compliant forwarders and is controlled by OpenFlow controller. These forwarders execute MAC tunneling according to OpenFlow rules set by the controller. Inner core forwarders manipulate the Ethernet header only, however the access edge forwarders (ePCU) examine the IP header and the access specific header (e.g. GPRS specific protocols). The external networks edge (e.g. Internet uplink) also examine the IP header in order to select correct tunnel for particular mobile station.

Architectural changes had naturally impact also on the protocol stacks. Figure 5 depicts signaling plane in the standard 3GPP GPRS network and Figure 6 depicts user plane. For the new GPRS SDN architecture, the protocol stacks are depicted on Figure 7 and Figure 8.

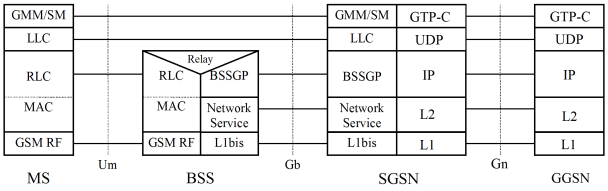

Figure 5: Control plane in standard 3GPP GPRS architecture.

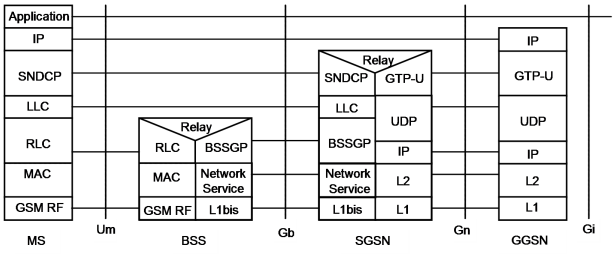

Figure 6: User plane in standard 3GPP GPRS architecture.

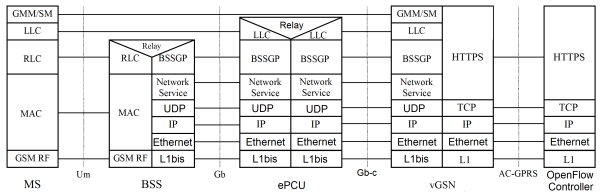

Figure 7: Control plane in GPRS SDN architecture.

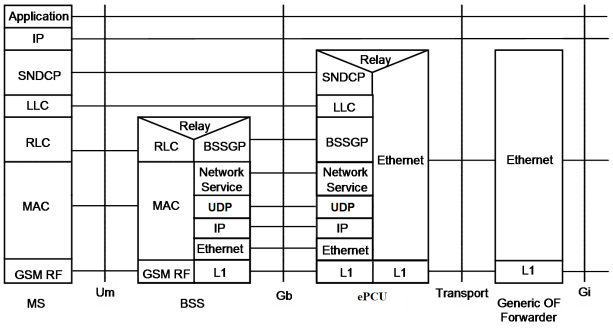

Figure 8: User plane in GPRS SDN architecture.

As mentioned before, standard 3GPP architecture employs GTP, both for the signaling (GTP-C) and for user plane tunneling (GTP-U). In GPRS SDN architecture the GPRS specific signaling is terminated on vGSN, processed there and if controller cooperation is needed, controller APIs are being called via ReST (Representational state transfer).

6.3 MAC Tunneling

As mentioned previously, GPRS SDN architecture does not use GTP as the core signaling and transport protocol. However, the need to separate users is still present.

At the beginning, we were considering to use something standard and well known to the networking community (inventing new tunneling approach was not really a goal of thesis). However we found out that protocols such as GRE (Generic Routing Encapsulation), VxLAN (Virtual extensible Local Area Network), EoGRE (Ethernet over Generic Routing Encapsulation), MPLSoGRE (Multiprotocol Label Switching over Generic Routing Encapsulation) or even GTP, are poorly supported cross the open-source ecosystem of OpenFlow forwarders and controllers. Moreover we realized, that there is no need for a feature rich tunneling protocol, such as any of the mentioned ones. Therefore we reused already present Ethernet II header for tunneling purposes (basically using the source MAC address as a tunnel ID. Having all traffic MAC tunneled, one can steer and breakout traffic at any node that can match and set Ethernet II header fields (basically any OpenFlow compliant forwarder).

6.4 Architecture Generalization for other Access Technologies

After SDN enabled GPRS architecture concept was finished, we moved on with generalization of this architecture for other access technologies. Following rules for application of the concept to other network types were proposed:

- Access specific protocols are terminated as close as possible to the access network, on access adaptor nodes.
- Access adaptor nodes extract access specific user plane data (ideally IP level) and forward it to transport core. If signaling is present, same node extracts it and forwards it to access network manager.
- Common transport core is based on MAC tunnels, which are enabled by OpenFlow forwarders and controlled by SDN controller.
- SDN controller is queried for session related procedures within access network, but also orchestrates procedures across different access network managers.

GPRS SDN architecture can be used for explanation of the evolution of the GPRS specific SDN concept to generalized SDN architecture (Figure 9). ePCU in GPRS SDN architecture is an access adaptor. Its responsibility is to terminate access specific protocols on the user plane(if needed, also to separate signaling from user plane data) and forward access specific signaling to access network manager – vGSN is the access network manager in GPRS SDN architecture.

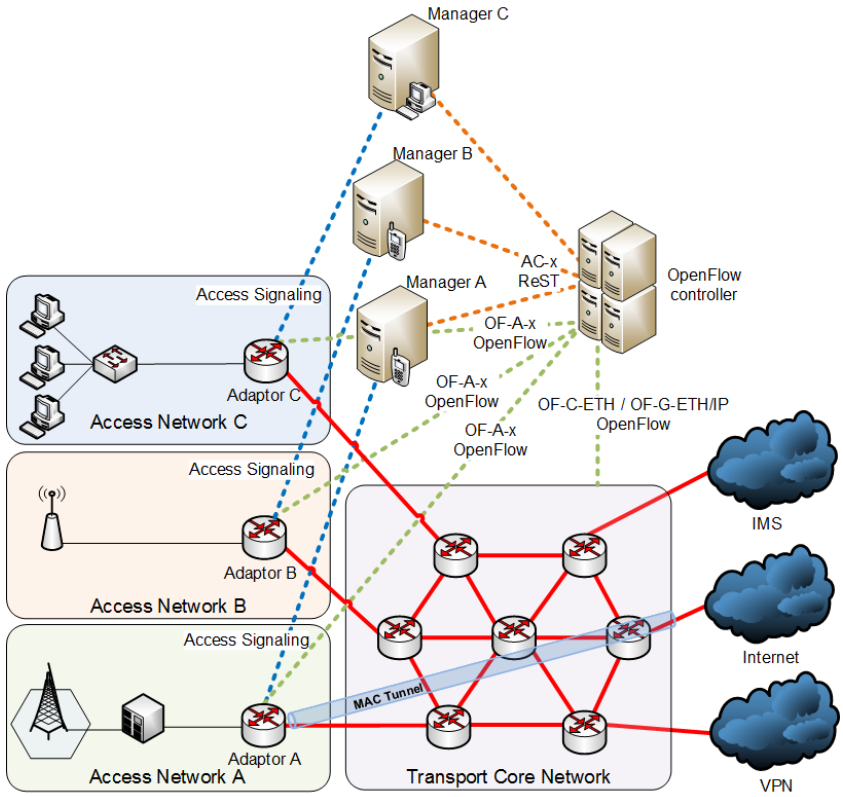

Figure 9: Access agnostic SDN architecture based on GPRS SDN architecture.

Access network manager is responsible for access network specific signaling towards the client connected in particular access network and for translation of requests to controller (if request is SDN controller related – such as session establishment or authentication).

Controller is there to provide core network control, but also orchestration cross multiple access managers and access adaptors in cases where client changes access technology. Controller also acts as a single authentication entity.

7. Architecture Verification

We verified our concept using both theoretical and experimental approach. In the theoretical verification, we examined the new architecture efficiency in terms of user plane data transport. In experimental approach, we built a proof of concept of the architecture.

7.1 Theoretical Verification

In theoretical evaluation part, we looked at the overhead of the MAC tunneling. It was compared to the popular tunneling approaches used in the commercial carrier networks and also to the GTP tunneling which it basically replaced in the new architecture.

Thus as the baseline, we took the protocol stacks as defined by 3GPP (Figure 5 and Figure 6). GTP is used here as the tunneling protocol, however in real deployment, it may be complemented with MPLS (MultiProtocol Label Switching) and perhaps also Ethernet VLANs (802.1q). As a sample packet distribution we used simple IMIX distribution of traffic (Table 1).

Table 1: Simple IMIX Definition

Packet size (IP level) [Bytes]	Ratio	Percentage of packets [%]	Percentage of traffic volume (at IP level) [%]
40B	7	58,3 %	6,8 %
576B	4	33,3 %	56,4 %
1500B	1	8.3 %	36,7 %

Since MAC tunneling does not need any additional protocol headers, except Ethernet II which is in our case anyways present, it proves itself to be the most efficient when compared to MPLS, 802.1q or even VxLAN.

7.2 Practical Verification

As for practical evaluation, we have built a proof of concept of SDN powered GPRS network using real 2G BTS hardware (i.e. no simulation or emulation used). This access technology was selected, due to the fact that at the time, no other 3GPP access technology node was available in the FIIT STU lab.

The setup was composed of Sysmocom SysmoBTS [23], which is a relatively inexpensive 2G (850/900/1800/1900 MHz) BTS (Figure 10). It is designed, build and sold by the community formed around Osmocom project. This project started as a network security research, focusing mainly on 2G network security issues [20]. Over the time, the group developed own baseband software for few models of 2G phones and continued developing software and hardware of other 3GPP defined network nodes, (e.g. BSC, MSC, HRL, etc.) which they used during the network security experiments.

Figure 10: SysmoBTS hardware base station.

SysmoBTS runs fully fledged ARM Linux, thus can host the whole mobile network and just connect its uplink to the Internet (data) or to SIP PBX for voice (Session Initiation Protocol Public Branch Exchange). Such setup is called network in the box, however we did not use it in this way. We ran only BTS and PCU applications on the SysmoBTS, so the SysmoBTS acts just as standard (3GPP compliant) BTS. To the core network it exposes 3GPP compliant Abis (voice/SMS) and Gb (data/SMS) interfaces. Both interfaces are logical and from the hardware point of view are terminated on the Ethernet port of the SysmoBTS unit. On the other side of the Ethernet cable we connected Linux PC running Ubuntu 14.04, 64bit version.

Since our thesis is focused on mobile data, we will omit Abis (voice) going forward and focus on Gb. This interface is terminated on ePCU, which executes the data plane – signaling separation function as described in the previous section. User data is then forwarded to the transport core, which is based on OpenFlow forwarders. Whole core, including the ePCU is based on experimental open-source OpenFlow forwarder implementation called ofsoftswitch13 [12]. Its code was modified by adding custom actions and match rules (for GPRS signaling – user plane separation), basically bringing GPRS awareness to the code. Ofsoftswitch13 is not designed to be high performance software OpenFlow forwarder implementation (unlike OpenVSwich), but rather focuses on easy extensibility to provide a ground for network experiments.

The egress part of core is basically "Internet" uplink. Last forwarder in the MAC tunnel sets correct source and destination MAC addresses and sends packet to the Linux kernel, where it is matched with iptables rules and NAPT (Network Address and Port Translation) is applied. Next the packet exits the Linux machine via second Ethernet interface (of Wi-Fi if applicable) towards the Internet.

Signaling data is forwarded from ePCU to the vGSN node, which is a combination of selected control functions of SGSN and GGSN. vGSN processes communication with the BTS and also with the mobile station. Osmosgsn [21] and openGGSN [6] code bases and surrounding libraries were used to build vGSN. These open-sources eased the proof of concept development, mainly thanks to the implemented mobile protocol stacks and prototypes of GPRS specific messages and state machines.

vGSN is on the north interfaced via ReST with the SDN controller. Controller holds full visibility of the network topology, state of the forwarders, basically is in charge of the whole transport network. Another open-source project was used here. We built controller application on top of the Ryu framework [7]. Ryu provides hooks to the OpenFlow processing – such as OpenFlow events handling, OpenFlow messages composition and parsing. However Ryu does not provide any controller, it is basically a framework for building controllers. Thus all the controller logic is purely custom product of ours.

8. Conclusion and Future Work

While looking at the thesis goals, its contribution can be broken down into following items:

- A new method of signaling-user data separation for the GPRS network was proposed. This is based on the SAPI information element and executed on the ePCU node. According to our research, there does not exist similar approach for GPRS.
- Thanks to this separation, SDN was deployed in the GPRS network.
- GPRS architecture was simplified by removing SGSN, GGSN nodes and the GTP protocol, which was used both for user data tunneling and also for signaling between the two nodes.
- New nodes were introduced to the architecture. As mentioned before, ePCU is providing signaling-user data separation, vGSN provides access network signaling processing, SDN controller is in charge of transport core and connectivity orchestration. OpenFlow forwarding core is used as an underlay and provides also overlay by using MAC tunneling.
- SDN enabled GPRS architecture was used as a basis for proposal of an access agnostic (generic) SDN architecture. This architecture can integrate various access technologies, while using the network concept from GPRS SDN (such as common MAC tunnels based core controlled from SDN controller, access control by dedicated access managers – such as vGSN and signaling-user data separation provided by access adaptor (enhanced OpenFlow forwarders) – such as ePCU.
- Architecture was verified by a proof of concept, using unmodified BTS hardware and MS, which proved that changes in the core network (implementation of SDN concept) did not have impact on the radio access network and are transparent to it. This will ease practical deployment of the concept. Moreover effectivity of MAC tunneling was evaluated using simple IMIX traffic model. Providing that MAC tunneling does not use any additional protocol headers, it was the most efficient out of the compared protocols.

Future work on the topic can be done in various ways. One of them is definitely other access technologies integration – for example LTE or WiFi.

Next there might be extension of the proposed GPRS SDN concept, where for example ciphering support can be

added. In the standard architecture, ciphering is executed on SGSN (for later network generations, this is being pushed more to radio access network nodes). Since there is no SGSN, we removed this feature entirely from the concept. The reason is, that encryption, if needed is provided by the application layers. If implemented in GPRS SDN architecture, this would imply major changes mostly in the ePCU (and also in the controller). ePCU would need to store ciphering keys and also temporary identities of the mobile stations (P-TMSI – Packet Temporary Mobile Station Identity). Moreover it would need to examine signaling messages, in order to keep track of P-TMSI reallocations and apply correct ciphering keys on the mobile station's traffic. In our opinion this goes far beyond the simplicity of the forwarding plane (one of the concepts within SDN), thus was excluded from the original proposal.

Also future work can be done in the area of MAC tunneling, which as pointed out provides means of offloading traffic in any point of the network (as far as that point is a OpenFlow compliant forwarder).

Last, usual topic within computer networks research area would be QoS (Quality of Service). MAC tunnels provide a basis for different QoS model implementations. Different traffic types, customers, access technologies can have dedicated tunnels and these can traverse different forwarders and links with different properties (if multiple paths are available between source and destination) and can be optimized for given traffic type.

Acknowledgements. This project was partially supported by the Tatra banka foundation under the contract No. 2012et011.

References

[1] Open networking foundation: Openflow switch specification 1.4.0, 2013.

[2] 3GPP: 23.002: Network architecture; 13.1.0, 2014.

[3] 3GPP: 23.060: General packet radio service (GPRS); service description; stage 2; 13.1.0, 2014.

[4] IETF: Forwarding and control element separation workgroup, 2015.

[5] 3GPP: Specifications, 2019.

[6] Open-ggsn, 2019.

[7] Ryu SDN framework, 2019.

[8] M. T. Abbas, T. A. Khan, A. Mahmood, J. J. D. Rivera, and W. Song. Introducing network slice management inside m-cord-based-5g framework. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–2, April 2018.

[9] J. Bannister, P. Mather, and S. Coope. *Convergence Technologies for 3G Networks: IP, UMTS, EGPRS and ATM*. John Wiley & Sons, Ltd, 2004.

[10] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E. Schmidt. A virtual SDN-enabled LTE EPC architecture: A case study for S-/P-gateways functions. In *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, pages 1–7, Nov 2013.

[11] J. Costa-Requena, M. Kimmerlin, J. Manner, and R. Kantola. SDN optimized caching in LTEen mobile networks. In *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 128–132, Oct 2014.

[12] CPqD. ofsoftswitch, 2019.

[13] A. Elakkiya and P. Selvaraj. QoS based IP mobility management scheme for the next generation SDN-LTE network. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 1355–1360, Jan 2018.

[14] GSMA. The mobile economy 2018. Technical report, https://www.gsma.com/mobileeconomy/wp-content/uploads/2018/05/The-Mobile-Economy-2018.pdf, 2019.

[15] GSMA. Mobile economy 2018: Technology migration, 2019.

[16] Juniper Networks. Contrail SDN, 2019.

[17] H. Kaaranen, A. Ahtiainen, L. Laitinen, S. Naghian, and V. Niemi. *UMTS Networks: Architecture, Mobility and Services, Second Edition*. John Wiley & Sons, Ltd, 2005.

[18] J. Kempf, B. Johansson, S. Pettersson, H. LÃijning, and T. Nilsson. Moving the mobile evolved packet core to the cloud. In *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 784–791, Oct 2012.

[19] F. Metzger, C. Schwartz, and T. HoÃ§feld. GTP-based load model and virtualization gain for a mobile network's GGSN. In *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*, pages 206–211, July 2014.

[20] K. Nohl and C. Paget. GSM: SRSLY?, 2009.

[21] osmocom. Osmo-sgsn, 2019.

[22] K. Pentikousis, Y. Wang, and W. Hu. Mobileflow: Toward software-defined mobile networks. *IEEE Communications Magazine*, 51(7):44–53, July 2013.

[23] Sysmocom. SysmoBTS 1002, 2019.

[24] J.-P. Vasseur and J.-M. L. Roux. RFC 5440 – path computation element communication protocol, 2009.

Selected Papers by the Author

M. Nagy and M. Kotočová. An IP based security threat in mobile networks. In *2012 Proceedings of the 35th International Convention MIPRO*, pages 667–670, May 2012.

M. Nagy and I. Kotuliak. Enhancing security in mobile data networks through end user and core network cooperation. In *Proc. of Int. Conf. on Advances in Mobile Computing & Multimedia*, MoMM '13, pages 253:253–253:259, New York, NY, USA, 2013. ACM.

M. Nagy and I. Kotuliak. Utilizing openflow, SDN and NFV in GPRS core network. In *Testbeds and Research Infrastructure: Development of Networks and Communities*, pages 184–193, Cham, 2014. Springer International Publishing.

K. Burda, M. Nagy, and I. Kotuliak. Reducing keepalive traffic in software-defined mobile networks with port control protocol. In *Information and Communication Technology*, pages 3–12, Cham, 2015. Springer International Publishing.

M. Nagy, I. Kotuliak, J. Skalný, M. Kalčok, and T. Hirjak. Integrating mobile openflow based network architecture with legacy infrastructure. In *Information and Communication Technology*, pages 40–49, Cham, 2015. Springer International Publishing.

R. Grežo and M. Nagy. Network traffic measurement and management in software defined networks. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pages 541–546, Dec 2017.

Instructions to the authors

Publishing procedure

All contributions are web-published. A contribution is published without unnecessary delay right after it has been accepted. Contributions are published on the fly in the current issue. It is at the discretion of the Editor-in-chief to determine, when the current issue is closed and a subsequent new one is open. There will be at least two issues in a year but it is left up to the Editor-in-chief to adjust periodicity of the Bulletin to actual needs.

Extended abstracts of theses is the primary type of article in the Bulletin. Each extended abstract will by annotated by identifying the thesis supervisor, who must recommend it for publication and stands for the Editorial Board in a role similar to a reviewer. We offer publishing extended abstracts on the Bulletin's web before the thesis is defended. This preliminary publishing is a specific service to the academic community. As soon as we learn about successful defence, the extended abstract gains the status of accepted paper and will be included in the forthcoming issue. The accepted paper will be annotated with the date of successful defence and name of the insitution where the defence took place.

It is the policy of the Bulletin to offer a free access to all its articles on the web. Moreover, the publisher will seek opportunities to promote as wide as possible access and/or indexing of the articles. All the past issues remain accessible on the web as part of the web portal of the Bulletin. Closed issues will be made available also in a printable form, free for downloading and printing by anyone interested.

Policy on Originality

It is the policy of the Bulletin that Slovak University of Technology be the sole, original publisher of articles. Manuscripts that have been submitted simultaneously to other magazines, journals or to conferences, symposia, or workshops without the prior written consent of the Editor-in-Chief will be rejected outright and will not be reconsidered. Publication of expanded versions of papers that have been disseminated via proceedings or newsletters is permitted only if the Editor-in-Chief judges that there is significant additional benefit to be gained from journal publication. A conference chairperson can arrange with the Editor-in-Chief to publish selected papers from conferences, symposia, and workshops, after suitable reviewing. The papers must meet the editorial requirements for research articles. Acknowledgement of the originating conference will appear as a credit when the paper is published in the Bulletin.

Manuscript information for extended abstracts of doctoral dissertations

All contributions are submitted electronically. Send your manuscript as LaTeXsources and .pdf files by e-mail to `editor.acm@fiit.stuba.sk`. Paper's length should be 6-12 pages. Please, use LaTeXstyle, which is available to download at bulletin web-page `http://slovakia.acm.org/bulletin/`.

Some remarks to the provided style:

- **Headings and Abstract**
  The heading must contain the title, full name, and address of the author(s), thesis supervisor, abstract of about 100-200 words.
- **Categories and Subject Descriptors**
  Define category and subject descriptors according to ACM Computing Classification System (see `http://www.acm.org/about/class/1998/`).
- **Keywords**
  Please specify 5 to 10 keywords.
- **Equations**
  You may want to display math equations in three distinct styles: inline, numbered or non-numbered display (we recommend the numbered style). Please make sure that your equations are clearly formulated and described.
- **Figures and tables**
  Figures and tables cannot be split across pages, the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures/table, use the environment figure/table to enclose the figure and its caption.
- **References**
  Please use BibTeX to automatically produce the bibliography. If possible use abbreviations like: Proceedings – Proc., International – Int., Conference – Conf., Journal – J.
- **Selected papers by the author**
  This section is used for thesis extended abstracts. Please write down all publications which are related to your thesis.

Published by Slovak University of Technology Press,
Vazovova 5, 812 43 Bratislava, IČO: 00397687
on behalf of the ACM Slovakia Chapter
ISSN 1338-1237 (printed edition)
ISSN 1338-6654 (online)
Registration number: MK SR EV 3929/09