# ScalaNC - Scalable heterogeneous link aggregation enabled by Network Coding

Daniel Behnke, Matthias Priebe, Sebastian Rohde, Karsten Heimann, and Christian Wietfeld TU Dortmund University, Communication Networks Institute (CNI)

Otto-Hahn-Str. 6, 44227 Dortmund, Germany

{daniel.behnke, matthias.priebe, sebastian.rohde, karsten.heimann, christian.wietfeld}@tu-dortmund.de

Abstract—The aggregation of multiple carriers and even heterogeneous links is a well established method to boost data rates in wireless networks. At the same time, Network Coding has been proven as an efficient and robust mechanism to distribute content in mesh networks or cloud storages. In this paper, we introduce with ScalaNC a new method of heterogeneous link aggregation, which requires no changes in the network infrastructure and incorporates Network Coding to allow for a scalable trade-off between speed, latency and security in terms of confidentiality (information distribution) of the aggregated end-to-end connection. The proposed method is particularly useful in emergency response scenarios, in which first responders need to access cloud storages to retrieve data in harsh communication environments. Using the scalable parameterization the users can increase real-time capability, speed or security depending on the actual needs. ScalaNC is validated with an experimental setup, in which the increase in real-time capability, goodput and security with high packet error rates was demonstrated: while the goodput of an aggregated link operated with Multipath TCP is reduced to around 50% with occurring packet errors, the ScalaNC-enabled aggregated link maintains stable goodput.

#### I. INTRODUCTION

Starting from a first description of the idea in the year 2000 [1], Network Coding has gained attention in research to optimize the coding procedures and adapt the idea to different use cases. Starting point for the research of Network Coding was the increase of packet throughput in a simple wired network with butterfly topology [2]. With the use of methodologies like opportunistic listening Network Coding is sensible in wireless networks as well [3]. A slightly different application are mobile storage clouds [4], Network Coding is used to offer a decentralized approach to store data maintaining high availability.

The research project SecInCoRe aims to enable collaboration and data-exchange between emergency services and stakeholders [5]. Fig. 1 exemplifies the use-case as emergency situations like forest fires do not stop at national borders. Specific events like pandemics have impacts on societies in many countries. Contingency measures can be supported by an enhanced collaboration between the different emergency services. Using a <u>C</u>loudbased <u>E</u>mergency <u>Information System</u> (CEIS) allows for knowledge sharing and communication between different organizations.



Fig. 1: Use-Case: Scalable Multi-Link Cloud Access for Emergency Response

One important aspect of such a system is the secure and reliable access to the CEIS on the incident scene. Depending on the availability, different communication technologies like WiFi, 4G networks or even LPWAN are used. The aim is to utilize each communication link in an appropriate manner, enhancing overall user experience regarding speed, real-time data and security of sensitive information. To fulfill these aims we developed a scalable heterogeneous over the top multi path communication concept in order to manage several communication links in parallel and employ *Network Coding* to enable a smooth transition from maximum performance to maximum reliability.

The remainder of the paper is organized as follows: First, we analyze existing approaches of *Network Coding* and multi-link usage in Section II. In Section III, we introduce our approach for *ScalaNC*, the *Network Coding*enabled scalable link aggregation. Consequently, we depict our laboratory setup in Section IV and present our results in Section V. We conclude the paper by summarizing our results and providing an outlook to ongoing research in Section VI.

# II. Related work to ScalaNC in research and Applications

As described previously, *Network Coding* has a wide field of application and has been analyzed in different network environments. In [2], a theoretical analysis for different network states are presented. Zhao et al. compared the analytical and measured performance in IEEE 802.11 networks, showing that the performance match in basic scenarios, but side effects as heterogeneous links reduces the gain in high-load scenarios [6]. For these scenarios an analytical model is presented in [7].

Many myths about the usage of *Network Coding* have been addressed in [8], showing that no excessive overhead is needed and potential applications for *Network Coding* exceed the basic example of *butterfly* networks. Therefore it is stated that *Network Coding* can be a useful addition in an end-to-end approach and with our proposed technique we want to go further in that direction. Hence, it is important to underline that *Network Coding* can be used in practice as analyzed in [9].

Our approach works on layers above TCP and UDP, but it is possible to adapt the TCP implementations to *Network Coding* as demonstrated in [10]. First steps have been done to outline the usage of *Network Coding* in multilink networks with *Multipath TCP (MPTCP)* [11][12]. Results regarding the performance gain of *Network Coding* in multi-link scenarios are presented in [13] as well as an architecture approach to combine *Network Coding* and *Software-Defined Networking* for future 5G networks.

In contrast to these approaches where every relay node has to be adapted to use *Network Coding* and allowing *MPTCP* packets, we want to provide an end-to-end approach considering the network in between as transparent. We thus make use of the existing transport layer protocol UDP and implement *ScalaNC* as an over the top protocol in user space. Subsection III-B elaborates in more detail on related multipath solutions.

# III. SCALANC - NC ENABLED SCALABLE LINK AGGREGATION

In this section, the multipath communication architecture and the *Network Coding* as superstructure is introduced.

## A. Key Performance Indicators

The background for the development of a novel communication methodology is the need for an efficient and robust communication capability in the mentioned mobile use cases. Based on the needs, the protocols are evaluated considering the following key performance indicators.

• Speed

The goodput indicates the applicability of the different protocols for continuous data streams like video streams as well as file transfer time needed.

• Real-time data

For real-time data, latency is critical. Hence, the

latency of packet transmissions using *MPTCP* is compared to the latency with *ScalaNC*. The *Network Coding* principle of coding a generation of 10 symbols (cf. Section III-C) before transmission might increase the latency of data transmission.

• Security (data confidentiality)

In [14] the authors analyze the security of distributed cloud storage. Referring to that deliberations, we consider the information distribution as a measure for the parts of the whole information that are transmitted on each link. The aim is to reduce the information density on each link in order to increase the security in terms of confidentiality regarding attacks on the communication. To measure the information distribution, we are using the Theil-Index as basis. The Theil-Index [15] describes the measure for economic inequality, in our case the overall income is the number of packets sent on all links with a mean income  $\mu$ 

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{1}$$

with number of links N and the number of packets x sent on link i. The Theil-Index is therefore defined as

$$T = \frac{1}{N} \sum_{i=1}^{N} \frac{x_i}{\mu} \ln \frac{x_i}{\mu}$$
(2)

In order to ease the interpretation of the index we scale it to take values from  $\left[\frac{1}{N} \ 1\right]$ . An *Information Distribution Index (IDI)* of 1 indicates evenly distributed information for all links, an *IDI* of  $\frac{1}{N}$  indicates that all information has been transferred using one link.

$$IDI = e^{-T} \tag{3}$$

### B. UDP driven multipath connection

The increasing need for high data rates and resilient communication capabilities in many emergency situations and 5G development motivates the usage of multiple networks at the same time. This includes heterogeneous technologies like high capacity cellular networks (e.g. LTE) and WiFi ad-hoc networks, but also low capacity LPWAN links such as LoRa.

ScalaNC aggregates multiple single path communications via UDP (through fixed sender/receiver relationships) to an aggregated multipath. This aggregated multipath, makes use of different physical paths, and enables dynamic selection of the best path for every single packet. In particular, contradicting performance guarantees such as maximum performance by using all links by means of their maximum achievable throughput or maximum distribution of packets by even spreading on a per packet basis. In order to enable this type of communication an additional scheduling layer must be introduced between the application and the UDP relationships that deals with

	ScalaNC	Multipath TCP	тои	QUIC
Protocol	UDP	TCP	UDP	UDP
Link aggregation	+	+	-	In Future
Reliability	FEC/ Retransmissions	Retransmissions	-	FEC/ Retransmissions
Interface to application	+	-	+	For HTTP/2
Standardized	-	+	IETF Draft	IETF Draft
Cross- platform	+	-	+	+
Security	Information Distribution	-	Encryption	TLS 1.3

Fig. 2: ScalaNC in comparison to other solutions

the distribution of packets on the different paths.

Fig. 2 classifies ScalaNC in comparison to other well-known solutions. Google also recently announced QUIC as an UDP based drop-in replacement of TCP/TLS/HTTP/2 [16]. Facebook introduced *Transport Over UDP (TOU)* as an Internet draft [17]. In addition, there have been other abandoned approaches focusing on TCP like MCTCP [18] or [19]. *ScalaNC* stands out from these approaches by enabling the use of UDP in a reliable fashion, forward error correction in combination with full cross-platform capability and a quality of service interface to the application. Hence, the main benefits of *ScalaNC* are:

• End-to-end connection

No need to adapt all routing nodes on the path

- UDP usage
- Fast connection establishment
- Path scheduling algorithms

Optimize usage of available paths by flexible and adaptive scheduling algorithms

- Application-driven optimized resource allocation Context sensitive scheduling based on application information
- Security (data confidentiality) The used scheduling algorithm pr

The used scheduling algorithm provides a sufficient information distribution in order to avoid transmitting all information on one link.

Network Coding guarantees in order delivery of all data packets to the application. Additionally, in a first step, a scalable scheduler based on link capacities, in regards to distribution and performance is implemented.

# C. ScalaNC - Network Coding and scalable Link Aggregation

The <u>Sca</u>lable <u>Link Aggregation</u> enabled by <u>Network</u> <u>Coding</u> (*ScalaNC*) overview as depicted in Fig. 3 visualizes the two key integrations for the multipath communication:

1) Scalable Network Coding

Using *Network Coding*, it is possible to create additional redundant packets (Forward Error Correction, FEC), those packets allow the recovery of the origin data even if some packets are lost during the transmission. The user can parameterize the coding specifications between two poles. On the one hand goodput and eavesdropping protection can be optimized by using a low amount of redundancy packets, on the other hand reliability and robustness can be optimized by using a high amount of redundancy packets.

2) Link Distribution Management

In order to address the increase data confidentiality, the information should be distributed on all available links. If one link is attacked, the origin data cannot be restored completely.

The influence of these parameterizations on the performance criteria will be evaluated in Section V. The next two subsections give an in depth look on the layers of *ScalaNC* and describe the transmission and scheduling principles.



Fig. 3: *ScalaNC* Overview

1) Scalable Network Coding and transmission management: The uncoded data created by an application is processed first by Network Coding. Received data from upper layers is divided into predefined chunks, so called generations which consist of a fixed number of network coded symbols. Hence, if random linear Network Coding algorithms are used, nearly infinite numbers of coded packets can be derived from the uncoded source data, each of them holding a part of information from the whole generation, this is called pure coding. For further information on Network Coding principles itself refer to [2]. The coded symbols are passed to the Link Distribution Management in the following step. According to the available links and enabled scheduler, this layer decides on the used link(s) depending on the pre-requirements of the data.

For the data transmission on the link UDP is used. In contrast to the approach shown in [13], *ScalaNC* uses this way of transmission to give every application an easy way to get link aggregation capability without requiring changes to the system architecture itself as it is needed when using MPTCP.

Furthermore it overcomes the MPTCP problem, that even if standardized, middleboxes or firewalls can reject MPTCP headers because of their rules, destroying link aggregation capability. Nevertheless UDP based transmissions suffer the reliability of TCP transmissions due to the lack of retransmissions. Therefore, *ScalaNC* introduces a feedback system to ensure correction of lost data.



Fig. 4: ScalaNC Sequence Diagram

Fig. 4 presents the procedure of *ScalaNC* transmission. The transmission of data starts by sending one complete generation. If the predefined generation size of linear independent symbols, in this case 10 symbols, is completely received the generation can be decoded. The receiver acknowledges the reception of a generation with an ACK message. If the transmission of a symbol is unsuccessful as with symbol 10 of generation 2, a new symbol is encoded and transmitted to substitute the lost symbol. As every symbol has a part of the whole information, it is not important to retransmit exactly symbol 10. Decoding can start when 10 linear independent packets are received. Hence, Network Coding makes retransmission management easier in comparison to standard ARQ techniques. Retransmissions are initiated after a fixed time out period which has been determined experimentally. In future research the determination of the retransmission time out will be adapted to a measured Round-Trip-Time value. In order to cope with the delay in networks multiple generations can be encoded at the same time.

Besides retransmission *ScalaNC* also supports the creation of forward error correction packets to correct occurring errors in advance. Another infrequent type of error that can occur in the transmission of network coded symbols is receiving a linear dependent symbol. A linear dependent symbol is a successful received packet which carries no new information for the decoding process and therefore needs a retransmission.

Our proposed technique is aware of individual link capacities and therefore avoids congestion of the network.

The sending and receiving nodes use the same structure. As we use multiple encoders and decoders at the same time, received symbols are matched to the corresponding decoder. If a decoder has received a complete generation, it is decoded and acknowledged. The decoded data is forwarded to the application. In Table I the *Network Coding* parameters are presented. In our scenarios, we differentiate two cases: First, transmitting certain files, e.g. case studies, from the cloud to the rescue services at the incident scene with a fixed and known amount of packets to be transmitted as fast as possible. Second, video streams are transmitted to the cloud and therefore to connected head quarters.

Network Coding is parameterized with a generation size of 10 symbols. This means a generation is completely received after 10 symbols. This provides fast encoding and decoding speeds and is a common generation size used in other research, e.g. [13], as well. The used coding scheme is pure coding, that means in contrast to systematic coding, that every symbol is coded. The advantage of that procedure is that assuming the usage of multiple paths, in case of one eavesdropped link, the attacker cannot decode the whole information, because only a part of the whole data is received.

TABLE I: Network Coding parameters

Parameter	Value	
Coding algorithm	Pure Coding	
Generation size	10 symbols	
Symbol size	1427  Byte	
FEC	0% to $100%$	

2) Link Distribution Management: ScalaNC gives the opportunity to increase the confidentiality of data by transmitting as less information as possible on one link. In case of eavesdropping the information density on one link makes it very difficult to decode even parts of the data. The information density on one link is the least if every link transmits the same amount of data, that means when the packets are evenly spread on all links. We are introducing a scalable packet distribution algorithm in order to spread the traffic on all available links n in the most evenly manner and being able to influence the achievable throughput. The maximum achievable data rate of each single link cap(i) has to be known in advance to be able to calculate the distribution.

The crucial factor for an even distribution of packets is the throughput of the slowest link in the setup. When the packets shall be distributed in an even manner the link capacity of the slowest link cannot be exceeded, so all links have to adapt to the throughput of the slowest link. This results in the minimum achievable throughput of the aggregated multilink which is called minTP and can be calculated by multiplying the slowest link throughput with the number of links n. If a higher throughput is needed by an application, the distribution cannot be fully equal but can be scaled in order to keep the information density on each link as low as possible. Using the following equation the intended throughput aimTP for a parameter  $x, x \in$ [0, 1] can be computed.

$$aimTP(x) = minTP + (maxTP - minTP) \times x \quad (4)$$

The lower bound is the mentioned minTP, the upper bound is the cumulated link throughput maxTP of all available links cap(i):

$$maxTP = \sum_{i=1}^{n} cap(i) \tag{5}$$

The throughput range can be quantized by the user by parameterizing x, in this work 10 intermediate steps are possible.

After the intended throughput is determined, the link distribution LD for all links has to be calculated. This is done using an iterative process, starting with the "slowest" link. This link will be fully loaded using his full throughput cap(i). The remaining part of the load in order to achieve the aimTP has to be distributed on the other links, in the best case each remaining link can be loaded with the same traffic otherwise the second slowest link will be fully loaded. Afterwards, the computation process continues with the next link. Using the following equation the part of the aimTP LD can be calculated for each link *i*.

$$LD(i) = \begin{cases} \frac{cap(1)}{aimTP}, & i=1\\ \frac{1-\sum_{x=1} LD(x)}{n-(i-1)}, & i>1 \land \frac{1-\sum_{x=1} LD(x)}{n-(i-1)} \le \frac{cap(i)}{aimTP} & (6)\\ \frac{cap(i)}{aimTP}, & i>1 \land \frac{1-\sum_{x=1} LD(x)}{n-(i-1)} > \frac{cap(i)}{aimTP} \end{cases}$$

For each packet, a link will be selected in order to establish this distribution. By respecting individual link capacities our proposed technique avoids congestion, while keeping information density on each link as low as possible for data rate values between minTP and maxTP of the aggregated multi link.

#### IV. EXPERIMENTAL SETUP

Considering the use case definition in Section I, we are using two nodes in our setup. One represents the cloud system, while the other one represents a mobile node at the incident scene. Both nodes are connected via multiple heterogeneous links to a virtual network. The network itself is transparent for the nodes besides the maximum link capacities which are known.

Fig. 5 depicts the experimental setup. Three links are established between the two nodes. A bare metal switch is used to parameterize each link in terms of capacity and packet error rates. The network emulation tool *netem* [20] is used for the parameterization.

Table II provides the scenario parameters for the different runs. The link capacity differs in three configurations, using two heterogeneous configurations (A and C) and a homogeneous configuration (B). In all three configurations several packet error rate parameters are used in order to



Fig. 5: Experimental setup for multiple paths

TABLE II: Measurement series

Parameters	Link 1	Link 2	Link 3			
Configuration A						
Capacity	$60\mathrm{Mbit/s}$	$20\mathrm{Mbit/s}$	$10\mathrm{Mbit/s}$			
PER-SET 1	0%	0%	0%			
PER-SET 2	5%	5%	5%			
PER-SET 3	15%	10%	5%			
Configuration B						
Capacity	$15\mathrm{Mbit/s}$	$15\mathrm{Mbit/s}$	$15\mathrm{Mbit/s}$			
PER-SET 1	0%	0%	0%			
PER-SET 2	5%	5%	5%			
PER-SET 3	15%	10%	5%			
Configuration C						
Capacity	$75\mathrm{Mbit/s}$	$10\mathrm{Mbit/s}$	$5\mathrm{Mbit/s}$			
PER-SET 1	0%	0%	0%			
PER-SET 2	5%	5%	5%			
PER-SET 3	15%	10%	5%			

show results for no packet errors, minor amount of packet errors and a significant amount of packet errors.

Network Coding is implemented using the kodo library [21]. In this setup, UDP connections are used for Network Coding and compared to MPTCP provided by the Linux Kernel Multipath TCP project [22]. The used congestion control algorithm is cubic for MPTCP.

#### V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of *ScalaNC*. In a first step, the influence of Forward Error Correction and Link Distribution Management is analyzed to find optimal parameters. Next, *ScalaNC* is compared to the well-known MPTCP in common setups. Concluding, we present the performance and the parameterization for typical requirements in emergency communication.

#### A. Influence of Forward Error Correction

In a setup with three heterogeneous links, the influence of Forward Error Correction for *Network Coding* on goodput and latency is analyzed. The capacity of the links is parameterized following Config A with 60 Mbit/s, 20 Mbit/s and 10 Mbit/s with PER distribution of 15%, 10% and 5%.



Fig. 6: Goodput and latency for different FEC parameters

Fig. 6 depicts the results for different FEC parameterization starting at 0% up to 100%. As expected, the goodput is continuously decreasing with increasing FEC, this is caused by the rising number of redundant packets. Remarkable is that these redundant packets help to repair occurring packet errors. The naive expectation is a bisection of the goodput for a FEC of 100% in comparison to the FEC of 0%, hence a goodput of 39 Mbit/s is expected, the results demonstrate the effect of redundant packets by increasing the goodput up to 44 Mbit/s.

The other graph presents the latency of the data transmission. This is the average time for a successful transmission of 10 coded symbols. The latency is reduced by higher FEC due to the redundant packets. A saturation can be recognized for FEC values higher than 50 %.

### B. Influence of Link Distribution Management

The used setup differentiates in terms of link capacity. The first link provides a capacity of 75 Mbit/s, second link provides 10 Mbit/s and third link provides 5 Mbit/s (Configuration C). Using this setup, the influence of Link Distribution Management (cf. Fig. 7) is analyzed, therefore FEC 0% is used to minimize Forward Error Correction influence. The goodput is rising linearly as intended. The *Information Distribution Index (IDI)* is an indicator for the link distribution of the three links and provides information about the eavesdropping protection of the transmission. The uniform link distribution provides an *IDI* of 1, the information density transmitted on one link is minimal, which provides the highest security against

eavesdropping, but the goodput is reduced to 15 Mbit/s (minTP). The drawback of the rising goodput is the decrease of *IDI*, down to 0.6 for the balanced link distribution using the capacity of all links to reach maximum goodput.



Fig. 7: Goodput and IDI for different link distribution parameters

#### C. Comparison against MPTCP

Analyzing the results, we are using a parameterization to provide the best performance in terms of goodput. Hence, we are not using FEC and the balanced link distribution. In two capacity configurations three different packet error rate sets are used to compare the performance of *ScalaNC* and *MPTCP*.

In Configuration A, the capacity of the links provide a theoretical maximum of 90 Mbit/s (maxTP). In Fig. 8 the results are presented. Without any packet errors, the goodput is the same for both protocols (87 Mbit/s), the latency is lower for MPTCP. With occurring packet errors the goodput for MPTCP is decreasing to 46 Mbit/s respectively 44 Mbit/s, the latency is rising to approx. 200 ms. On the contrary, *ScalaNC* provides a relatively stable goodput of 78 Mbit/s respectively 76 Mbit/s, the latency is slightly increasing to 50 ms.

To conclude, ScalaNC is more stable to packet errors than MPTCP. In order to assess the results, a different configuration is analyzed as well. After the heterogeneous link capacity in Configuration A, we are using the same capacity of 15 Mbit/s on all three links in Configuration B (cf. Fig. 9).

Again, in the basis setup without any packet errors, the goodput for both protocols is the same with 43 Mbit/s by



Fig. 8: Performance evaluation of ScalaNC (solid lines) and MPTCP (dashed lines) in Configuration A



Fig. 9: Performance evaluation of ScalaNC (solid lines) and MPTCP (dashed lines) in Configuration B

a maxTP of 45 Mbit/s. In this configuration, the influence of packet errors on MPTCP is even more remarkable. The goodput is reduced to 23 Mbit/s respectively 13 Mbit/s. Using *ScalaNC* the goodput is very stable, this protocol is able to compensate almost all packet errors. The goodput is reduced to 41 Mbit/s and 39 Mbit/s, latency is rising to 40 ms. For MPTCP the latency is rising up to 280 ms.

As a first conclusion, the results have shown that ScalaNC is able to handle occurring packet errors very well and is providing very stable performance in those situations.

#### D. Parameters for real world situations

In this section, we aim to present results for different usual communication scenarios. Basis is once again Configuration A with packet error rates of 15%, 10%and 5%. The following scenarios provide characteristic requirements regarding the communication system:

1) Real-time multimedia transmission: In this scenario, a multimedia stream, e.g. a live feed from a camera surveying the incident scene, has to be transmitted. Hence, the latency of the transmission should be very low and goodput has to be sufficient. Fig. 10(a) provides the results for such a scenario. With a FEC of 50 % it is possible to reduce the latency significantly and a balanced link distribution provides best goodput.

2) Maximum security: If the data is very sensible, it is the aim to avoid transmitting all information on only one link. With a FEC of 0% and uniform distributed load, it

is possible to achieve maximum security. Even if two links are attacked, only a maximum of 66.6% of the whole data can be retrieved. Here, a goodput of 30 Mbit/s is possible.

3) Large file transfer: In this scenario, a large amount of data has to be transmitted as fast as possible from sender to receiver. Hence, the aim is to achieve maximum goodput. Therefore, the number of redundant packets (FEC) is 0 and a balanced link distribution is used. With this parameterization, the goodput rises up to 79 Mbit/s with the drawback of a higher latency and reduced security.

4) Mixture of different data types: Finally, we provide a common scenario, where different communication activities take place. Hence, it is the aim to find an equilibrated parameterization providing good performance in all three characteristics. With a FEC of 50 % and a link distribution value 5, the latency is very low (13 ms), the goodput achieves 40 Mbit/s and the *IDI* is at 0.92.

### VI. CONCLUSIONS AND FURTHER WORK

In this work, we have introduced *ScalaNC*, a novel scalable link aggregation protocol based on *Network Coding* using multiple available communication technologies at the same time. Our performance evaluations demonstrate the advantage of this approach in scenarios with network access over three different links. *ScalaNC* is very robust against PER. It is suitable in scenarios with harsh channel conditions due to mobility or rural areas with limited network coverage. Depending on the knowledge or



Fig. 10: ScalaNC in real world communication scenarios

variety of the communication environment *ScalaNC* can be parameterized to meet the current requirements in a optimal manner.

In future experiments, *ScalaNC* will adapt to changing communication conditions automatically. Additionally, different *Network Coding* varieties such as on-the-fly coding will be considered in order to reduce the latency even more.

#### Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no<sup>o</sup>607832 (project SecInCoRe) and the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no°644657 (project AutoMat - Automotive Big Data Marketplace for Innovative Crosssectorial Vehicle Data Services). The text reflects the authors' views. The European Commission is not liable for any use that may be made of the information contained therein. For further information see www.secincore.eu and www.automat-project.eu. Part of the work on this paper has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", project A4 and the German Federal Ministry of Education and Research (BMBF) for the project LARUS (Supporting Maritime Search and Rescue Missions with Unmanned Aircraft Systems, 13N14133).

#### References

- R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.
- [2] T. Ho and D. Lun, Network Coding: An Introduction. New York, NY, USA: Cambridge University Press, 2008.
- [3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network

Coding," IEEE/ACM Transactions on Networking, vol. 16, no. 3, pp. 497–510, June 2008.

- [4] M. Sipos, F. H. P. Fitzek, and D. E. Lucani, "On the feasibility of a network coded mobile storage cloud," in 2015 IEEE International Conference on Communications (ICC), June 2015, pp. 466–471.
- [5] M. Kuhnert, O. Paterour, A. Georgiev, K. Petersen, M. Buescher, J. Pottebaum, and C. Wietfeld, "Next generation, secure cloud-based pan-european information system for enhanced disaster awareness," in *Proceedings of the ISCRAM* 2015 Conference, Kristiansand, Norway, May 2015.
- [6] F. Zhao, M. Medard, M. Hundeboll, J. Ledet-Pedersen, S. A. Rein, and F. H. P. Fitzek, "Comparison of analytical and measured performance results on network coding in IEEE 802.11 ad-hoc networks," in 2012 International Symposium on Network Coding (NetCod), June 2012, pp. 43–48.
- [7] A. Moreira and D. E. Lucani, "On coding for asymmetric wireless interfaces," in 2012 International Symposium on Network Coding (NetCod), June 2012, pp. 149–154.
- [8] M. Medard, F. H. P. Fitzek, M. J. Montpetit, and C. Rosenberg, "Network coding mythbusting: why it is not about butterflies anymore," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 177–183, July 2014.
- [9] A. Paramanathan, M. V. Pedersen, D. E. Lucani, F. H. P. Fitzek, and M. Katz, "Lean and mean: network coding for commercial devices," *IEEE Wireless Communications*, vol. 20, no. 5, pp. 54–61, October 2013.
- [10] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 490–512, March 2011.
- [11] S. Gheorghiu, A. L. Toledo, and P. Rodriguez, "Multipath TCP with Network Coding for Wireless Mesh Networks," in 2010 IEEE International Conference on Communications, May 2010, pp. 1–5.
- [12] J. Cloud, F. du Pin Calmon, W. Zeng, G. Pau, L. M. Zeger, and M. Medard, "Multi-Path TCP with Network Coding for Mobile Devices in Heterogeneous Networks," in 2013 IEEE 78th Vehicular Technology Conference (VTC Fall), Sept 2013, pp. 1– 5.
- [13] J. Hansen, D. E. Lucani, J. Krigslund, M. Medard, and F. H. P. Fitzek, "Network coded software defined networking: enabling 5G transmission and storage networks," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 100–107, September 2015.
- [14] M. Sipos, F. H. P. Fitzek, D. E. Lucani, and M. V. Pedersen, "Distributed cloud storage using network coding," in 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC), Jan 2014, pp. 127–132.
- [15] H. Theil, "The information approach to demand analysis," *Econometrica*, vol. 33, no. 1, pp. 67–87, 1965. [Online]. Available: http://www.jstor.org/stable/1911889
- [16] (2017) Quick UDP Internet Connections (QUIC). [Online]. Available: https://datatracker.ietf.org/wg/quic/about/
- [17] (2016, 07) Internet-Draft: Transport layer protocols over UDP. [Online]. Available: https://tools.ietf.org/html/ draft-herbert-transports-over-udp-01
- [18] M. Scharf and T. R. Banniza, "MCTCP: A Multipath Transport Shim Layer," in 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, Dec 2011, pp. 1–5.
- [19] B. Wang, W. Wei, J. F. Kurose, D. F. Towsley, K. R. Pattipati, Z. Guo, and J. P. Zheng, "Application-layer multipath data transfer via TCP: Schemes and performance tradeoffs," *Perform. Eval.*, vol. 64, pp. 965–977, 2007.
- [20] (2017) Network Emulation netem. [Online]. Available: https: //wiki.linuxfoundation.org/networking/netem
- [21] M. Pedersen, J. Heide, and F. Fitzek, "Kodo: An open and research oriented network coding library," *Lecture Notes in Computer Science*, vol. 6827, pp. 145–152, 5 2011, in Proceedings of the 2011 Networking Workshops. (eds.) Casares-Giner, Vincente, Manzoni, Pietro & Pont, Ana.
- [22] (2017, 04) Multipath TCP. [Online]. Available: https://www. multipath-tcp.org/