Neighborhood Watch: On Network Coding Throughput and Key Sharing

Martin Strohmeier*, Ivan Martinovic*, Utz Roedig+, Karim El Defrawy#, Jens Schmitt>
{martin.strohmeier, ivan.martinovic}@cs.ox.ac.uk, u.roedig@lancaster.ac.uk, keldefra@uci.edu, jschmitt@cs.uni-kl.de

*University of Oxford, UK +InfoLab21, Lancaster University, UK #University of California Irvine, USA
>Distributed Computer Systems Lab, University of Kaiserslautern, Germany

Abstract—**Network coding (NC) has frequently been promoted as an approach for improving throughput in wireless networks. Existing work has mostly focused on the fundamental aspects of NC, while constraints arising in real-world network deployments have not received much attention. In particular, NC requires network nodes to overhear each other's packets, which oftentimes contradicts many security standards that attempt to provide link-layer confidentiality, e.g., by utilizing pairwise encryption keys as is the case IEEE 802.11i and ZigBee. There is an inherent trade-off between gains from NC and link-layer security: if many nodes share the secret link-layer key, NC will improve throughput, yet a leakage of the key will affect many nodes. On the other hand, having distinct secret keys will increase resilience against key compromise, but will also minimize the coding gain. We formulate this security vs. performance trade-off as an optimization problem and evaluate the effectiveness of NC under different sizes of key-sharing groups and network topologies. Our results show that increasing the key-sharing group by a single node can result in a maximum coding gain between 1.3% and 13.7%.**

Index Terms—**Network Coding, Neighborhood Size, Wireless Networks, Key Sharing**

I. Introduction

Network coding (NC) has emerged in the past decade as a popular topic in networking research. It has attracted attention because it facilitates increasing the network throughput under certain conditions while keeping bandwidth usage in the network constant. Nodes using NC overhear multiple messages and create linear combinations thereof before re-broadcasting them in a combined form. Receivers can subsequently use such linear combinations to restore the original individual messages. This effectively increases the information flow with the same number of transmissions, resulting in the so-called *coding gain*. Since its introduction in 2000 [2], NC has been successfully used to improve applications such as peer-to-peer content distribution systems [11] and plain wireless networks [14]. NC's use in wireless networks seems particularly intriguing since overhearing transmissions is an inherent feature of the broadcast medium. While this work focuses on NC in wireless networks, the presented results may apply to other types of networks, too.

From a security point of view one should not equip all nodes in a network with a single encryption key. A lost key could lead to a complete loss of protection within the entire network. Consequently, single key solutions are typically considered to provide only minimal, lightweight security. To increase protection, it is a common procedure to form smaller groups of nodes that share a key. One traditional example is the broken Wireless Equivalent Privacy (WEP) standard which uses only a single key in contrast to the new IEEE 802.11i security standard which implements separate keys for each link. Similarly, in wireless sensor networks (WSNs), the IEEE 802.15.4/ZigBee specification offers a lightweight security relying on a single network key shared by all nodes [19]. Yet, within the ZigBee-PRO standard this approach should be avoided in favor of different unique link-layer and group keys to protect WSNs with stronger security requirements. Finally, the weakness of single network keys is also demonstrated by the recently available KillerBee security framework[1] which offers easy network key recovery once physical access to a sensor is available (low-cost sensor devices usually do not include any tamper-resistant hardware).

Only nodes that share a common key can overhear each others' transmissions and use them to compute linear combinations. In the extreme case where every two nodes share a separate, unique link key, it is obvious that any gain from coding is lost since nothing can be overheard. As a consequence, we argue that in many practical settings, it is unreasonable to assume that all neighbors in a node's range can overhear its transmissions. Furthermore, there is a lot of leeway between the two extremes of either having one single key or pairwise separate link keys; choosing an appropriate group size is an objective directly conflicting with the assumptions of NC.

We believe that many practical deployments will consider the size of groups sharing the same key in a network limited. In particular, we define two different types of such limitations:

- **Local Key-Sharing Limitation**: Each node has a limited number of adjacent nodes which it shares a key with.
- **Global Key-Sharing Limitation**: A fixed number of keys is available for the entire network. Each node has to be member of at least one key group.

It is possible for a network to be subject to both local **and** global limitations; we will consider both types of grouping restrictions throughout.

[1] http://code.google.com/p/killerbee/

While we have argued that in practical deployments key sharing is often a limiting factor, it is also frequently possible to influence the size and composition of groups sharing a key. From an NC perspective, it is desirable to cluster nodes such to maximize network throughput (i.e., to achieve the maximum coding gain) and to evaluate different group sizes according to the trade-off between security needs and throughput requirements. In this paper, we analyze and quantify the impact of key-sharing neighborhoods on NC throughput. Specifically, we make the following contributions:

- We formulate an integer linear program that is able to provide an exact solution of the key distribution problem with respect to maximizing NC throughput.
- We systematically validate the impact of key-sharing neighborhood size and composition on network coding gain using randomly generated as well as real-world wireless network topologies.
- We identify the impact of different types of key-sharing limitations on the coding gain and quantify the trade-off. We additionally analyze the influence of several real-world QoS constraints.

The remainder of the paper is organized as follows. Section II describes related work. Section III briefly explains NC in practice and the experimental setup while Section IV formalizes our system model. Section V illustrates our results. Section VI summarizes and concludes this work.

II. Related Work

The theoretical foundations on which NC is based have been laid by Ahlswede et al. [2] They showed that it is possible - and necessary - to employ coding to reach multicast capacity in a network. This seminal result which is analogous to the max-flow min-cut theorem for routing resulted in multiple research efforts examining NC and investigating different types of networks and scenarios where it could be applied. Li et al. [16] showed that it is even sufficient to use linear codes in multicast networks to reach full capacity. Many subsequent works concerned themselves with the construction of such codes in practice (e.g. [13], [12]) and examined the theoretical bounds of NC in more general networks [15], [6], [17].

One specific result that has been used to conduct further research and is also the basic assumption for this paper is the fact that a throughput gain can generally be achieved with NC in static environments. An overview over this and other important results in NC research can be found in [8], a thorough and comprehensive background in [9].

Despite the popularity of the idea, few works have looked at the effect of security and neighborhood compositions on the effectiveness of NC. Cai and Yeung [4] present an information-theoretic approach, explaining how to modify a code so that a wiretapper with access to only one wire will have no or at least fewer chances of successfully obtaining any information. Various extensions to that work followed trying to make it a viable alternative to end-to-end encryption [3], [7].

Some concrete analysis of the effect of group sizes and NC in previous research is contained in work measuring the performance of algorithms while modifying the number of children/receivers in constructed and usually balanced trees. An example of this can be found in [10] where the authors compare the reliability properties of NC with traditional error control techniques in such trees. Sagduyu and Ephremides [18] integrate NC by design into a TDMA medium access control protocol in wireless ad hoc networks. They use information flow decomposition to develop various decentralized and conflict-free scheduling approaches. In their work, they briefly consider a simulation with multicast groups randomly chosen from a small network but do not look at effects of group size or composition. Finally, Castellucia et al. [5] analyze the information-theoretic foundations of link-layer encryption with NC, showing that it decreases the achievable capacity.

III. Network Coding in Practice

In this section, we briefly explain the operations of NC, identify relevant application scenarios that benefit from a coding approach, and introduce our experimental setup.

A. Network Coding

NC is effective in scenarios where data from multiple sources has to be delivered to multiple sinks. The basic idea can be explained best using Ahlswede's [2] canonical butterfly network as shown in Fig. 1 on the left. Two sources (nodes n_1, n_2) with messages A and B send both messages to two sinks (nodes n_5, n_6). Without NC, node n_1 can transmit A to n_3 and n_5 and n_2 can transmit B to n_3 and n_6. At this point, node n_3 has to decide which message to transmit. Let n_3 forward A to node n_4, now n_4 can forward A to n_6, and n_6 has received A and B. However, node n_5 has only received A and must wait for an additional transmission of B by n_3 and n_4.

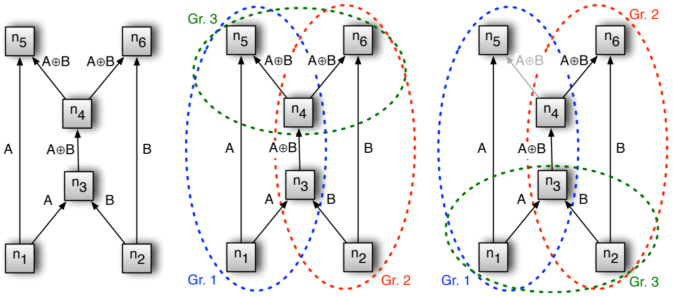

Figure 1. *Example of NC and the effects of groups.* The plain butterfly network (left) sends two messages A and B from n_1 and n_2, respectively, to **both** sinks n_5 and n_6 (exhausting all link capacities). The wrong distribution of size-restricted key groups (right) impacts the information flow: n_4 can send only to one of the sinks at a time, effectively taking away the advantage over traditional routing. The better choice (middle) preserves the coding gain.

With NC, node n_3 can forward a combination of message A and B (e.g. $A \oplus B$). Now, n_4 can transmit this combination to n_5 and n_6. As n_5 also has A, it can recover B from the combined message. Similarly, n_6 can recover A. Thus, NC increased the available network bandwidth, resulting in the so-called coding gain. If we now take security considerations into account and decrease the number of nodes that are allowed to have the same encryption key, this can have severe negative impacts on the achieved gain. In the worst case, depending on group size and composition, NC can even be rendered completely ineffective as illustrated in Fig. 1

B. Application Scenarios

Many application scenarios benefit from NC. In most networks, multiple sources generate data that must be delivered to multiple sinks. NC is particularly useful in wireless networks where the required overhearing of neighbor's transmissions is an inherent feature of the wireless channel.

One example are sensor networks which consist of many wireless nodes able to sense physical properties and to forward the measurements over multiple hops towards a sink where the arriving data is analyzed. Often several sensor nodes detect an event simultaneously and a lot of data has to be forwarded at the same time to the sink for analysis. Furthermore, to add redundancy and to optimize data extraction, many scenarios use multiple sinks to collect data. It is important to deliver sensor data to the sinks quickly to ensure fast event detection. Depending on the application and associated data analysis algorithms, different transport requirements may exist. For example, it might be sufficient to ensure that all messages are received by at least one sink. Alternatively, it may be necessary that all sinks receive all messages or a certain percentage of messages. NC can help deliver data from multiple sources to multiple sinks faster, resulting in an improvement of the quality of service/information.

C. Experimental Setup

To conduct large-scale experiments, we tested on randomized feed-forward graphs with unidirectional links. In this type of network topologies, relay nodes are on various layers between the source and the sink layer, with outgoing links always connecting to a layer closer to the sinks.

A graph's parameters are given as follows:

- The number of $layers$ (at least 3).
- The maximum number of $nodesperlayer$.
- The target layer for each node, taken from a geometric distribution created with a success probability of $p_{g,s}$ over the amount of remaining layers.
- The outgoing connections, generated for each node from a geometric distribution with a success probability $p_{g,c}$ over the amount of viable nodes on the target layer.
- The precise target node to connect to on the target layer is chosen as a uniform random variable.

In addition, our analysis also includes a practical real-world setting of a sensor network application in which a number of wireless sensors are used to deliver the sensed data to a number of sinks. For this part of the experimental setup, we consider the publicly available MoteLab WSN [1] and create a feed-forward topology based on its connectivity map (cf. Fig. 2). This network, deployed in the university building, comprises 30 nodes and offers a concrete instance of an indoor multi-hop sensor network.

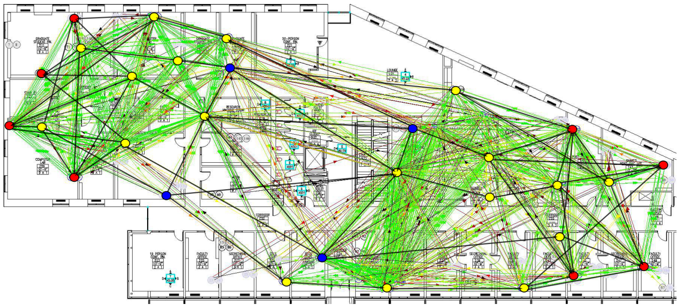

Figure 2. *Real-world feed-forward instance.* The network has 4 sensors used as sources (blue) in the middle which forward the collected data over different layers containing the 18 relay nodes (yellow) to the 8 sinks (red), 4 on the left-hand side of the map and 4 on the right.

IV. System Model

In this section, we formulate an integer linear programming (ILP) model to study the coding gain in networks with different types of key-sharing limitations.

A. Network Model

We assume a multi-hop wireless network modeled as a directed graph $G = (V, E)$. The set of vertices V represents the stations and the set of edges E denotes the connectivity.

The problem we examine can be modeled as a single source multicast problem in a wireless ad hoc network. The set of vertices V consists of $n + k + 1$ nodes (AN). Node 1 is the source node SRC. Nodes $2, 3, ..., n + 1$ are the n relay nodes (RN). Nodes $n + 2, n + 3, ..., n + k + 1$ are the k sink nodes (SN). We then consider that each link is a unit capacity link, and that the source is sending M independent single streams $P_1, P_2, ..., P_M$ through the relay nodes in the core of the network to the sinks. In our scenario, this equals multiple sources S_1,S_2, ..., S_M each multicasting one of the respective streams only, as it can be directly transformed [2].

The objective is to maximize the flow in the network (see Table I for the notation)

$$Max : \sum_{j=n+2}^{n+k+1} \sum_{i=1}^{P(j)} Flow_{i,j} \tag{1}$$

under the following constraints:

$$\forall j, k \in SN, \, \forall i \in P(j), \, j \neq k : \, Flow_{i,j,k} = 0 \tag{2}$$

$$\forall k \in SN, \, \forall i \in RN : \sum_{j=1}^{S(i)} Flow_{i,j,k} = \sum_{h=1}^{P(i)} Flow_{h,i,k} \tag{3}$$

$$\forall i, j \in AN : \sum_{k=1}^{SN} Flow_{i,j,k} {\geq} Flow_{i,j} \tag{4}$$

$$\forall k \in SN, \, \forall i, j \in AN : \, Flow_{i,j,k} \leq Flow_{i,j} \tag{5}$$

The goal of the ILP is to compute the maximum flow possible in a given graph/network, as stated by the Multicast

Table I
USED VARIABLES/NOTATION FOR THE ILP.

Variable	Description
$Flow_{i,j}$	A binary decision variable indicating if node j receives from node i (or i sends to j).
$Flow_{i,j,k}$	A binary decision variable that indexes all edges i, j for each sink k, creating a unique subgraph of the whole network for this sink.
$g_{i,j}$	A binary decision variable indicating if i is logically allowed to send to j (i.e., knows j's key).
$S(i)$	Successors of node i, i.e., all nodes that i can transmit to.
$P(i)$	Predecessors of node i, i.e., all nodes that can transmit to i.
T	Threshold for the allowed number of members in a group.
R	Required number of streams arriving at each source.

Theorem[2]. More precisely, for all sinks the full graph is divided into a distinct subgraph for each sink and the maximum flow of every subgraph is calculated separately. As the links are modeled as binary flows, this corresponds to the number of unique ways between the source and the subgraph's sink[3].

The optimization function maximizes the number of active incoming links over all sinks. Constraint (2) ensures that only the sink of the according subgraph is considered for flow maximization. Constraint (3) guarantees the flow balance, so that the number of incoming and outgoing flows must be equal for all relay nodes. Constraint (4) translates the links used to achieve the maximum flow for each sink into the full network. Constraint (5) ensures that only an edge used for sending in any sink's subgraph is also used in the full network.

We used SCIP[4] to resolve this to the maximum flow network for any given sink. The subgraphs are combined into the original network under the general Constraints (2) and (3) and all possible combinations are tested and solved to optimality. This provides the upper bound for linear NC in a network, where all relay nodes are in the same group, i.e., all successors of a node can overhear all transmissions.

B. Local Key-Sharing Limitations

Local key-sharing limitations are modeled as follows:

$$\forall i \in RN, \forall j \in AN : g_{i,j} \geq Flow_{i,j} \tag{6}$$

$$\forall i \in RN : \sum_{j=1}^{S(i)} g_{i,j} \leq T \tag{7}$$

Constraints (6) and (7) specify the number of nodes a relay node is allowed to send to. In this model, if there are n physical links to the next layers of the feed forward network, only T of these are actually being used in the multicast, constituting a group with a single common key. The special case where all successors are in the same group with the sender models the traditional assumption considered in the NC literature.

We are also interested in looking at the effects of quality of service constraints in this context:

$$\forall j \in SN : \sum_{i=1}^{P(j)} Flow_{i,j} \geq R \tag{8}$$

$$\forall i \in SRC, \forall j \in S(i) : Flow_{i,j} \geq 1 \tag{9}$$

Constraint (8) makes sure that every sink receives a given number of streams while (9) guarantees that every stream sent by the source will be received by at least one sink.

C. Global Key-Sharing Limitations

In the original problem, every node sending in the network had a unique key which potential receivers had to know. To increase security, we now reduce and specify the number of keys in the network. This requires new constraints which model the centralized planning of the key distribution to achieve the optimal throughput. The computational complexity of this added set covering problem, however, makes finding an optimal solution only feasible for smaller networks.

The changes we make to the ILP are as follows: There is a fixed set of keys C available which can freely be distributed among nodes, as long as there are not more nodes using the same key than allowed by the group size limit. Compared to above, with a sufficiently large group size, two or more nodes can use the same key for sending which was not possible before. Clearly, this is a more general problem as it comprises the previous model as a special case.

The following constraints in addition to (1)-(5) are required (see also Table II for the notation):

$$\forall c \in C : \sum_{i=1}^{RN} Key_{i,c} \leq groupsize \tag{10}$$

$$\forall i \in RN, \forall c \in C : SendKey_{i,c} \leq Key_{i,c} \tag{11}$$

$$\forall i \in RN, \forall c \in C : RcvKey_{i,c} \leq Key_{i,c} \tag{12}$$

$$\forall i \in RN : \sum_{c=1}^{C} SendKey_{i,c} \leq 1 \tag{13}$$

$$\forall i, j \in AN : Flow_{i,j} \leq \sum_{c=1}^{C} ShareKey_{i,j,c} \tag{14}$$

$$\forall i, j \in AN, \forall c \in C : ShareKey_{i,j,c} \leq SendKey_{i,c} \tag{15}$$

$$\forall i, j \in AN, \forall c \in C : ShareKey_{i,j,c} \leq RcvKey_{j,c} \tag{16}$$

Equation (10) provides the limit on the group size for each available key. (11), (12) make sure a key is only counted once if used for both sending and receiving. (13) limits each sender to one key at a time. Constraint (14) allows flows between nodes only if they share the same key, while (15) and (16) ensure for each transmission where the key is utilized that the sender uses the key for sending and the receivers for receiving.

V. EVALUATION

In this section, we show that NC is effective in the examined scenarios and evaluate local and global key-sharing limitations. In a best-case scenario in the MoteLab network, all 4 sensor messages are received by all 8 sinks with each node in the

[2] "Linear NC achieves the min-cut/max-flow bound for any multicast network with a single source and multiple destinations." [16]

[3] Can be changed to non-binary variables to model arbitrary bandwidths.

[4] "Solving Constraint Integer Programs", http://scip.zib.de

Table II
ADDITIONAL VARIABLES TO PERFORM KEY DISTRIBUTION.

Variable	Description
$Key_{i,c}$	A binary decision variable indicating if a node i knows key c.
$SendKey_{i,c}$	A binary decision variable indicating if a node i knows and uses key c for sending.
$RcvKey_{j,c}$	A binary decision variable indicating if a node j knows key c and uses it for receiving.
$ShareKey_{i,j,c}$	A binary decision variable indicating if nodes i and j both know key c.

network requiring at most one transmission, achieving the maximum throughput at 32 messages. Our analysis reveals that with NC this best-case throughput can actually be achieved while with normal routing under the same circumstances only 26 messages are delivered. Thus, network throughput with NC is 21.4% higher than without.

The experiments on randomly generated graphs were run with 4 layers, a maximum of 20 nodes per layer and settings for $p_{g,c}$ of 0.9 and $p_{g,s}$ of 0.6. The results have been averaged over 100 different networks of the same type; we identified an average coding gain of 7.1%.

A. Impact of Local Key-Sharing Limitations

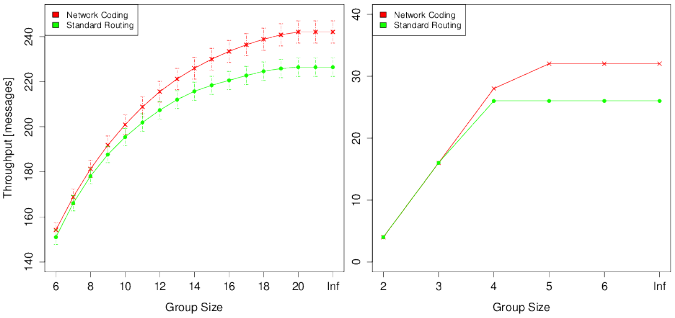

Figure 3. *Optimal multicast throughput of NC and routing in 100 random feed-forward networks with 95% confidence intervals (left) and the MoteLab WSN (right), restricted to certain group sizes.* With increased group size there is a widening throughput gap between both methods, identified as the coding gain, until the effect is saturated.

We introduce the local key-sharing limitations into the ILP to measure their effects on NC and how these compare with the same constraints applied to ordinary routing. Fig. 3 (left) depicts the average achievable upper bound for the examined random networks. Unsurprisingly, there is barely any NC gain with lower group sizes, e.g. around 1.7% for $n = 7$. The coding gain raises steadily (at most 1.3% when group size is increased from 10 to 11), until for $n = 20$ it peaks at an average edge of 7.1 and further relaxation shows no effect.

In the results for the adapted MoteLab network, NC and routing do not differ for $n = 2$ and $n = 3$ and it takes a group size of at least $n = 4$ until NC gains an advantage of 7.7%. At this size, routing already reached its upper bound, while NC throughput improves an additional 13.7% (accounting for almost two thirds of the overall 21.4% coding gain possible in this setting) with groups of 5. This analysis shows that “one-key-for-all” is not required to leverage the advantages of NC and compromises between such lightweight security and pairwise separate link keys can indeed be found.

B. Additional Practical Considerations

Source Constraints: In some network topologies, it can be impossible with multicast routing to transmit every stream to at least one sink in a single round of transmissions due to bottlenecks. While in such a case it also cannot be guaranteed with (linear) NC that a particular sink can be served with one specific stream [6], one can make sure that every stream arrives at least at one arbitrary sink. With routing, this can only be fulfilled if there exists a path for every source to a sink where the edge sets of all these paths are each mutually exclusive. In contrast, it is sufficient when using NC that a path from each source to a sink merely exists as links can be used simultaneously.

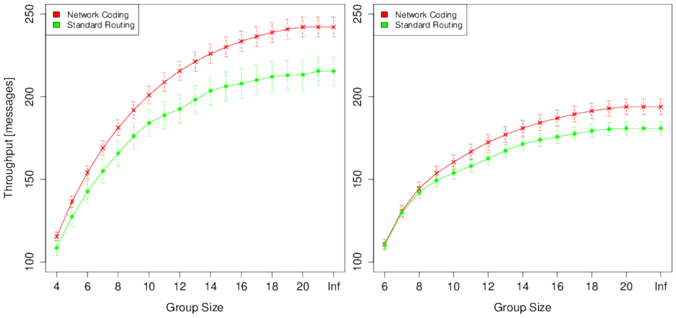

Figure 4. *Random networks with source constraints (left) and with 35% sink constraints (right).* With source constraints, the average throughput drops for routing compared to Fig. 3. With sink constraints, throughput falls sharply for both routing and NC.

With local key-sharing limitations, the difficulty of finding such distinct paths significantly increases. While there could exist a solution in the unrestricted network, it might be impossible to use it with smaller groups. To achieve the results shown in Fig. 4 (left), Constraint (8) was added. The significantly larger coding gain (peaking at 12.5%) stems from networks which cannot fulfill the source constraint with routing under a given group size. These additional 5% of networks, which fail the constraints and are consequently treated as having zero throughput, influence the outcome considerably. Taking these out of the equation, the NC edge roughly shrinks back to the gap of just above 7% which we could observe before. Thus, an advantage of NC also lies in the ability to fulfill constraints that are otherwise impossible to achieve with routing.

Sink Constraints: In some applications (e.g. signal processing), every sink is desired to receive a certain percentage of all available streams to enable sufficient processing, redundancy and reliability. It can also be advantageous if the required percentage is met as quickly as possible. NC can help with these requirements which might be infeasible or slower with routing or require a larger group size for the same results.

Introducing Constraint (9) with a minimum of 7 streams per sink, yields a hard QoS requirement of 35% in our 20 source/sink setting. Fig. 4 (right) shows the results where a group size below 5 renders all networks unsolvable with both coding and routing. As the networks generally become solvable with larger group sizes, roughly the same advantage of NC over routing than in the setting without any QoS can be observed, peaking at 7.2%. Just as before, a number of graphs are not able to serve the QoS requirement (causing the drop in average throughput) but in this case they are generally not satisfiable by both coding or routing. Although there is no additional coding gain observable, one could principally imagine graphs where this might be the case, a simple example being the butterfly network with a 100% sink constraint.

C. Global Key-Sharing Limitations

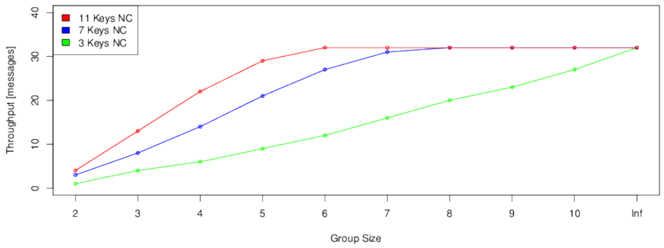

Figure 5. *Differences in throughput between NC in three scenarios with different numbers of available keys in the MoteLab network.* To achieve full NC throughput with 11 keys, groups of 6 are sufficient, while with 7 keys we need groups of 8 and with 3 keys groups of 11 or more.

Given a fixed limited number of keys available to be distributed in a network, NC can reduce the group size needed to achieve the same throughput as with routing. Inversely, with a given maximum group size and throughput, fewer keys are needed in a network with NC. Fig. 5 shows how the limitation influences NC throughput depending on the group size in the MoteLab network. The lower the number of distributed keys, the longer it takes to reach the saturation point where larger key-sharing groups show no further throughput effect.

VI. Discussion & Conclusion

While the main focus of this work was to evaluate security considerations, similar constraints can occur due to other non-security related reasons, including but not limited to energy, protocol, and bandwidth constraints. Energy constraints play a role in WSNs where it is generally not advisable that nodes listen to potential transmissions of all neighbors. Protocol constraints are a factor in networks with cluster-based medium access where only nodes in the same cluster may overhear transmissions. Bandwidth constraints happen when WSN nodes operate on different frequencies to increase available bandwidth, thus globally constraining neighborhood size. We believe that our model and results are useful and transferable to other scenarios apart from security concerns.

In this paper, we analyzed the impact of increased security and smaller sizes of key-sharing groups on the effectiveness of NC. Furthermore, we investigated how neighborhoods should be composed in order to optimize NC throughput. It is important to answer these questions in order to design efficient and secure communication mechanisms for networks utilizing NC. We conclude that the traditional assumption that NC only works with one single network-wide key does not hold. While in networks which are limited to a very small group size the usefulness of NC is indeed questionable, even small increases can have a positive effect on throughput and its related metrics. The trade-off between group size and coding gain can be quantified with the use of our integer linear program. By choosing an appropriate group size, a compromise between the considered conflicting objectives can be found. Finally, as justified in this paper, if networks are created with NC and group size already determined, negative throughput effects can be mitigated through an integrated treatment of security and network planning.

References

[1] http://motelab.eecs.harvard.edu, motelab: Harvard sensor network testbed (retrieved 01/02/2013).

[2] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4), July 2000.

[3] K. Bhattad and K. R. Narayanan. Weakly secure network coding. *NetCod*, April 2005.

[4] N. Cai and R.W. Yeung. Secure network coding. In *Information Theory, 2002. IEEE International Symposium on*, page 323, 2002.

[5] Claude Castellucia, Karim El Defrawy, and Gene Tsudik. Link-layer encryption effect on achievable capacity in wireless network coding. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–5. IEEE, 2010.

[6] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *Information Theory, IEEE Transactions on*, 51(8):2745 – 2759, 2005.

[7] J. Feldman, Malkin T., R. A. Servedio, and C. Stein. On the capacity of secure network coding. In *42nd Annual Allerton Conference on Communication, Control, and Computing*. Camb. University Press, 2004.

[8] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network coding: An instant primer. 2005.

[9] C. Fragouli and E. Soljanin. Network coding fundamentals. *Found. Trends Netw.*, 2:1–133, January 2007.

[10] M. Ghaderi, D. Towsley, and J. Kurose. Network coding performance for reliable multicast. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1 –7, 2007.

[11] C. Gkantsidis and P.R. Rodriguez. Network coding for large scale content distribution. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2235 – 2245 vol. 4, march 2005.

[12] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, Shi. J., and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413 –4430, 2006.

[13] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M.G.M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *Information Theory, IEEE Transactions on*, 51(6):1973 – 1982, 2005.

[14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *SIGCOMM Comput. Commun. Rev.*, 36:243–254, August 2006.

[15] R. Koetter and M. Medard. An algebraic approach to network coding. *Networking, IEEE/ACM Transactions on*, 11(5):782 – 795, 2003.

[16] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *Information Theory, IEEE Transactions on*, 49(2):371 –381, 2003.

[17] A. Ramamoorthy, J. Shi, and R.D. Wesel. On the capacity of network coding for random networks. *Information Theory, IEEE Transactions on*, 51(8):2878–2885, 2005.

[18] Y.E. Sagduyu and A. Ephremides. Crosslayer design for distributed mac and network coding in wireless ad hoc networks. In *Information Theory, 2005. Proceedings. International Symposium on*, 2005.

[19] Ender Yüksel, Hanne Riis Nielson, and Flemming Nielson. Zigbee-2007 security essentials. In *Proc. 13th Nordic Workshop on Secure IT-systems*, pages 65–82, 2008.