Wireless Inter-Session Network Coding - An Approach Using Virtual Multicasts

Michael Heindlmaier[†], Desmond S. Lun[*], Danail Traskov[†], and Muriel Médard[‡]

[†]Institute for Communications Engineering
Technische Universität München
Munich, Germany
{michael.heindlmaier, danail.traskov}@tum.de

[*]CCIB
Rutgers University
Camden, NJ
dslun@camden.rutgers.edu

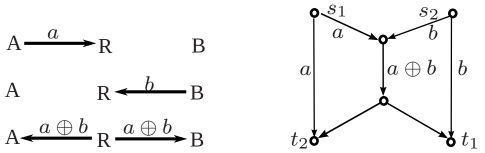
[‡]RLE
MIT
Cambridge, MA
medard@mit.edu

***Abstract*—This paper addresses the problem of inter-session network coding to maximize throughput for multiple communication sessions in wireless networks. We introduce virtual multicast connections which can extract packets from original sessions and code them together. Random linear network codes can be used for these virtual multicasts. The problem can be stated as a flow-based convex optimization problem with side constraints. The proposed formulation provides a rate region which is at least as large as the region without inter-session network coding. We show the benefits of our technique for several scenarios by means of simulation.**

I. Introduction

Network coding [1] has been an active field of research in the past years. Its benefits for unicast and multicast traffic have been well studied [2]–[4]. For single source network coding problems necessary conditions are also sufficient [1] and distributed linear coding schemes are available [5]. Multi-source network coding is a considerably more difficult problem: The optimal solution for inter-session network coding might require nonlinear encoding functions [6] and even deciding a linear inter-session coding problem can be NP-complete [7]. Nonetheless, suboptimal linear coding schemes have been proposed both for wireline and wireless networks. Previous work on inter-session coding for wireline networks has focused on a graph-theoretic characterization of network codes [8], [9]. For example, in [10], the authors present an approach to translate the solution for the well-known butterfly network to larger networks. In [11] inter-session network coding is combined with multiple description coding. Recent work [12] presented an interference alignment approach to perform inter-session coding.

We focus on wireless networks, where the most prominent technique is the COPE protocol [13] which exploits the broadcast nature of the wireless medium. As shown in Fig. 1a, the number of transmissions can be reduced compared to routing if packets from different sessions are combined. This technique is called *reverse carpooling* in [14] due to the fact that the combined information has to travel in opposite directions. The interplay of the COPE protocol with medium access has been analyzed in [15] and approaches to model its performance were proposed in [16], [17]. COPE has been successfully implemented and shows gains especially when the *reverse carpooling* situation occurs.



(a) The COPE example

(b) Butterfly example

Fig. 1: In the COPE example, A and B want to exchange the packets a and b. Node A sends packet a to the relay node R in the first transmission. So does B with packet b in a second time step. In the third time step, the relay broadcasts the XOR-ed packet $a \oplus b$ to both A and B, so b and a can be decoded, respectively. Routing would require four transmission steps. Similarly for the Butterfly network, where s_1 wants to communicate with t_1, and s_2 with t_2.

The technique COPE relies on can also be interpreted as a local multicast to a set of neighbors. We pick up the idea of a virtual multicast, however, in a different way than COPE does. We are motivated by the Butterfly example in Fig. 1b, where we can achieve higher rates for two sessions by creating a virtual multicast. Furthermore, random linear codes [5] can be used as a coding scheme in this setup [1]. The idea of our approach is to create virtual multicast sessions, combine packets from original sessions and deliver them to a virtual terminal set. Our technique is thus a *carpooling* approach, as packets are travelling in the same direction. In the Butterfly example the virtual multicast runs from the original sources to both original terminal nodes. However, our approach allows to establish the virtual multicast at an arbitrary position in the network and we will illustrate in Sec. IV that a multicast to the original terminals is not always optimal. In general, unlike many other techniques for inter-session coding, the difficulty of our approach is not the coding scheme itself but the proper choice of this virtual terminal set. We will define the virtual multicast model precisely in Sec. III-C.

The original sessions and the virtual multicast can be characterized in terms of flows from sources to terminals, similar to [3]. Therefore, the whole problem can be stated as a convex flow optimization problem.

The main contribution of this work is a formulation that enables inter-session network coding by introducing virtual multicast sessions. We evaluate the gains that can be expected with this technique compared to an uncoded approach for various networks.

We will present the system model in Sec. II before we state the problem formulation in Sec. III. The integration of multiple virtual multicast sessions will be discussed in Sec. IV. Simulation results are presented in Sec. V before we conclude the paper in Sec. VI.

II. System Model

A wireless packet network is modeled by a hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, $\mathcal{N}$ denoting the set of nodes and $\mathcal{A}$ denoting the set of hyperarcs. The demands are given by a set of connections, denoted $\mathcal{C}$. The pair $(\mathcal{H}, \mathcal{C})$ defines the *network problem*. A hyperarc $(i, N(i)) \in \mathcal{A}$ models a lossy one-to-many broadcast connection from a node $i \in \mathcal{N}$ to its neighbors $N(i) \subset \mathcal{N}$. Throughout this work, we assume that medium access takes care of scheduling transmissions, so each time node i transmits a packet, all neighbors $N(i)$ will be able to hear it. Accordingly, there is one hyperarc per node in the network, so $|\mathcal{A}| = |\mathcal{N}|$. A connection $c \in \mathcal{C}$ consists of a pair (s_c, T_c), that is a source node $s_c \in \mathcal{N}$ and a set of terminal nodes $T_c \subset \mathcal{N}$. If $|T_c| = 1$, c is a unicast connection, if $|T_c| > 1$, it is a multicast connection. The end-to-end throughput of connection c is denoted by R_c. With each connection we associate a concave utility function $U_c(R_c)$ depending on the throughput of that session.

The rate at which node i injects packets into its outgoing hyperarc is denoted by z_i. Due to erasures the transmitted packet might be received only by a subset $K \subset N(i)$ of the neighbors of i. We denote the arrival rate for packets exactly received by $K \subset N(i)$ by z_{iK}. So, $z_i = \sum_{K \subset N(i)} z_{iK}$. Let $b_{iK} = \frac{\sum_{L \subset N(i) | L \cup K \neq \emptyset} z_{iL}}{z_i}$ be the probability that a packet, sent out by node i, is received by at least one node in the set $K \subset N(i)$. Not every packet transmitted by node i belongs to the same session, so we denote by $y_i^{(c)}$ the packet injection rate at i for packets of connection c, so $z_i = \sum_{c \in \mathcal{C}} y_i^{(c)}$. The region $\mathcal{Z}$ specifies the set of feasible injection rate vectors $\mathbf{z} = (z_i)_{i \in \mathcal{N}}$. $\mathcal{Z}$ is assumed to be given from lower layers. Variables $x_{ij}^{(t,c)}$ represent the information flow of connection c for terminal $t \in T_c$ between nodes $i \in \mathcal{N}$ and $j \in N(i)$.

III. Formulation of the different Approaches

We formulate the problem in three different ways as a utility maximization problem. For simplicity, we consider the problem of maximizing the sum-utility, $\sum_{c \in \mathcal{C}} U(R_c)$.

The first two formulations will lead to our main description in Sec. III-C which builds up on the latter ones. First, we state the network problem if no inter-session coding is allowed in Sec. III-A. In Sec. III-B, we present an approach which combines all the information of all sources and multicasts it to every terminal in the network, like in the Butterfly example. This technique will help us to describe the Virtual Multicast formulation in Sec. III-C, which can do both - combine packets from different sessions and deliver it to a virtual terminal set or stick to the non-coding approach.

A. No Inter-session Coding (NIC)

Each session $c \in \mathcal{C}$ performs intra-session network coding but no inter-session network coding. So, flows of different sessions cannot be combined and have to share the network resources. The problem can be written as follows, extending the formulation in [3] for multiple sessions:

$$\text{maximize} \sum_{c \in \mathcal{C}} U_c(R_c) \qquad \text{subject to}$$

$$z_i \geq \sum_{c \in \mathcal{C}} y_i^{(c)} \geq 0, \qquad \forall\, i \in \mathcal{N}, \tag{1}$$

$$\mathbf{z} = (z_i)_{i \in \mathcal{N}} \in \mathcal{Z}, \tag{2}$$

$$y_i^{(c)} b_{iK} \geq \sum_{j \in K} x_{ij}^{(t,c)}, \quad \forall\, i \in \mathcal{N}, K \subset N(i), t \in T_c, c \in \mathcal{C}, \tag{3}$$

$$x_{ij}^{(t,c)} \geq 0, \qquad \forall\, i \in \mathcal{N}, j \in N(i), t \in T_c, c \in \mathcal{C}, \tag{4}$$

$$\sum_{j \in N(i)} x_{ij}^{(t,c)} - \sum_{\{j | i \in N(j)\}} x_{ji}^{(t,c)} = \begin{cases} R_c & i = s_c, \\ 0 & \text{else}, \end{cases}$$
$$\forall\, i \in \mathcal{N} \setminus \{t\}, t \in T_c, c \in \mathcal{C}. \tag{5}$$

Here, (3) relates the outgoing information flow to the packet injection rates while (5) assures flow conservation for each terminal of the connections.

B. Multisource Multicast (MSM): Completing the Multicast

If we allow all connections to be coded together, as each terminal wants to have all the data, min-cut conditions are necessary and sufficient [1, Theorem 8]. Again, it can be formulated as an optimization problem with linear constraints by introducing a super-source and virtual edges connecting this super-source with the original sources. The outgoing rate at the virtual source is $\sum_{c \in \mathcal{C}} R_c$, the capacity of virtual edges from the super-source to the original source is equal to R_c. In [18] this fact was used in the context of correlated sources. In this setup there is only one virtual session from the super-source to the receiver set T^*, where T^* contains all terminals of all sessions, i.e. $T^* = \bigcup_{c \in \mathcal{C}} T_c$. To simplify the notation, we can omit the virtual source and edges and consider session c originating at s_c again, knowing that there exist coding techniques to capture this setup. The following formulation applies:

$$\text{maximize} \sum_{c \in \mathcal{C}} U_c(R_c) \qquad \text{subject to}$$

$$z_i \geq 0, \qquad \forall\, i \in \mathcal{N}, \tag{6}$$

$$\mathbf{z} = (z_i)_{i \in \mathcal{N}} \in \mathcal{Z}, \tag{7}$$

$$z_i b_{iK} \geq \sum_{j \in K} x_{ij}^{(t)}, \quad \forall\, i \in \mathcal{N}, K \subset N(i), t \in T^*, \tag{8}$$

$$x_{ij}^{(t)} \geq 0, \qquad \forall\, i \in \mathcal{N}, j \in N(i), t \in T^*, \tag{9}$$

$$\sum_{j \in N(i)} x_{ij}^{(t)} - \sum_{\{j | i \in N(j)\}} x_{ji}^{(t)} = \begin{cases} R_c & i = s_c, \\ 0 & \text{else}, \end{cases}$$
$$\forall\, i \in \mathcal{N} \setminus \{t\}, t \in T^*. \tag{10}$$

The Butterfly network in Fig. 1b is an example where this technique works well. By multicasting all source information to every terminal we can do better than NIC here.

C. Virtual Multicasts (VM)

Neither NIC nor MSM is always better than the other strategy for general networks, so it is not clear which technique to choose. The following considerations let us combine both ideas: We introduce a virtual multicast connection $\tilde{c}$ which can extract flow from the other sessions. To distinguish this new session from the previous ones, we call $\mathcal{C}$ the set of original connections. The virtual session has a set of sink nodes $T_{\tilde{c}}$, which receive all the traffic absorbed by the virtual session. At each node i, $\lambda_i^{(c)} \geq 0$ denotes the amount of flow extracted from one of the original connections c and injected into session $\tilde{c}$. Let $\mu_i^{(t,c)} \geq 0$ be the amount of traffic re-injected at node i into session c for sink t from session $\tilde{c}$. For all $c \in \mathcal{C}$ and $t \in T_c$, we have $\mu_i^{(t,c)} = 0$ for $i \notin T_{\tilde{c}}$. That is, traffic can only be re-injected at the virtual terminal nodes. Let $\tilde{\mathcal{C}} = \mathcal{C} \cup \{\tilde{c}\}$ denote the set of original and virtual connections.

$$\text{maximize} \sum_{c\in\mathcal{C}} U_c(R_c) \qquad\qquad \text{subject to}$$

$$z_i \geq \sum_{c\in\tilde{\mathcal{C}}} y_i^{(c)} \geq 0 \qquad \forall\ i \in \mathcal{N}, \tag{11}$$

$$\mathbf{z} = (z_i)_{i\in\mathcal{N}} \in \mathcal{Z}, \tag{12}$$

$$y_i^{(c)} b_{iK} \geq \sum_{j\in K} x_{ij}^{(t,c)}, \quad \forall\ i \in \mathcal{N}, K \subset N(i), t \in T_c, c \in \tilde{\mathcal{C}}, \tag{13}$$

$$x_{ij}^{(t,c)} \geq 0 \qquad \forall\ i \in \mathcal{N}, j \in N(i), t \in T_c, c \in \tilde{\mathcal{C}}, \tag{14}$$

$$\sum_{j\in N(i)} x_{ij}^{(t,c)} - \sum_{\{j|i\in N(j)\}} x_{ji}^{(t,c)} = \begin{cases} R_c + \mu_i^{(t,c)} - \lambda_i^{(c)} & i = s_c, \\ \mu_i^{(t,c)} - \lambda_i^{(c)} & \text{else}, \end{cases}$$
$$\forall\ i \in \mathcal{N} \setminus \{t\}, t \in T_c, c \in \mathcal{C}, \tag{15}$$

$$\mu_i^{(t,c)} = 0 \qquad \forall\ i \notin T_{\tilde{c}}, t \in T_c, c \in \mathcal{C}. \tag{16}$$

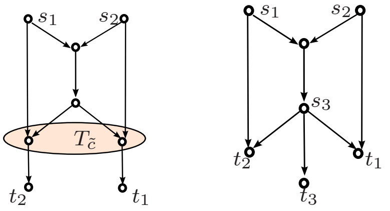
For the virtual session $\tilde{c}$, we have to add the following constraints:

$$\sum_{j\in N(i)} x_{ij}^{(t,\tilde{c})} - \sum_{\{j|i\in N(j)\}} x_{ji}^{(t,\tilde{c})} = \sum_{c\in\mathcal{C}} \lambda_i^{(c)},$$
$$\forall\ i \in \mathcal{N} \setminus \{t\}, t \in T_{\tilde{c}}, \tag{17}$$

$$\sum_{i\in T_{\tilde{c}}} \mu_i^{(t,c)} = \sum_{i\in\mathcal{N}} \lambda_i^{(c)}, \qquad \forall\ t \in T_c, c \in \mathcal{C}. \tag{18}$$

The modified flow conservation constraints in (15) guarantee proper accounting of flow extraction and re-injection for the virtual session. Flow conservation for the virtual session is assured in (17). Those nodes i, where $\sum_{c\in\mathcal{C}} \lambda_i^{(c)} > 0$ can be considered as the virtual sources for the virtual multisource multicast. Note that each node can act as a virtual source, however, the set of virtual terminals $T_{\tilde{c}}$ has to be specified. In (18), we make sure that all the collected flow from one session is re-injected into the original session at the virtual terminals.

The formulation above accepts the virtual terminal set $T_{\tilde{c}}$ as an arbitrary input. We briefly address considerations about this selection in Sec. IV. For now, we choose the virtual terminal set to yield $T_{\tilde{c}} = \bigcup_{c\in\mathcal{C}} T_c$. This choice allows us to perform both the uncoded strategy and the multisource multicast.



(a) Network 1 (b) Network 2

Fig. 2: Wireline example networks

Let $\mathcal{R}_{NIC}$, $\mathcal{R}_{MSM}$ and $\mathcal{R}_{VM}$ denote the achievable rate region of a network problem using communication strategy NIC, MSM and VM, respectively.

Theorem 1: If $T_{\tilde{c}} = \bigcup_{c\in\mathcal{C}} T_c$, every solution that can be obtained by NIC and by MSM can also be obtained by VM. Therefore, $\mathcal{R}_{VM} \supseteq \mathcal{R}_{NIC}$ and $\mathcal{R}_{VM} \supseteq \mathcal{R}_{MSM}$.

Proof:

$\mathcal{R}_{VM} \supseteq \mathcal{R}_{NIC}$:

If we set all variables related to the virtual session to zero, i.e. $\mathbf{y}^{(\tilde{c})} = \{y_i^{(\tilde{c})}\}_{\forall i\in\mathcal{N}} = \mathbf{0}$, $\mathbf{x} = \{x_{ij}^{(t,\tilde{c})}\}_{\forall i\in\mathcal{N}, j\in N(i), t\in T_{\tilde{c}}} = \mathbf{0}$, $\boldsymbol{\lambda} = \{\lambda_i^{(c)}\}_{i\in\mathcal{N}, c\in C} = \mathbf{0}$ and $\boldsymbol{\mu} = \{\mu_i^{(t,c)}\}_{i\in\mathcal{N}, t\in T_c, c\in\mathcal{C}} = \mathbf{0}$, the constraints in VM are exactly the same as for NIC. NIC is thus a special case of VM.

$\mathcal{R}_{VM} \supseteq \mathcal{R}_{MSM}$:

If we set $\lambda_{s_c}^{(c)} = R_c$ and zero elsewhere and $\mu_t^{(t,c)} = \sum_{i\in\mathcal{N}} \lambda_i^{(c)} = \sum_{c\in\mathcal{C}} R_c$, zero elsewhere, the virtual session $\tilde{c}$ carries all the flow of all sessions. The virtual session then corresponds to the single session in MSM. MSM is thus a special case of VM. ■

It follows from Theorem 1 that VM achieves at least the throughput of NIC and MSM. As we will see in Sec. V, VM can often do better than both.

IV. Multiple Virtual Sessions and Choice of $T_{\tilde{c}}$

So far only the special case of multicast of $T_{\tilde{c}} = \bigcup_{c\in\mathcal{C}} T_c$ was considered. We briefly show that this can be suboptimal. For simplicity, we use a wireline network with unit link capacity for the illustration in this section. Consider the network in Fig. 2a, where NIC and MSM can only provide the rate region $R_1 + R_2 \leq 1$. This is due to the fact that the only paths from s_1 to t_1 and from s_2 to t_2 are not edge-disjoint. VM with the choice of $T_{\tilde{c}} = \bigcup_{c\in\mathcal{C}} T_c$ performs better but is not optimal here. We can fully exploit the advantage of VM if we choose the highlighted nodes to be the virtual terminal set, yielding the following rate region: $R_1 \leq 1$; $R_2 \leq 1$.

Both sources multicast information with $R_1 = R_2 = 1$ to the nodes in the virtual terminal set. Then, these nodes forward the demanded information to t_1, t_2, respectively.

The formulation in Sec. III-C can be easily extended to additional virtual sessions. In order to determine the best terminal set, virtual sessions for each subset of nodes in the network would have to be introduced, resulting in an exponential number of virtual sessions. However, the computational complexity already becomes too high for small networks, so we have to restrict us to use only a small number of virtual connections. As mentioned, VM with $T_{\tilde{c}} = \bigcup_{c \in \mathcal{C}} T_c$ can combine NIC and multisource-multicast to all terminals. However, for many connections in the network it might restrict us too much requiring every terminal of all sessions to receive all data. Consider the network in Fig. 2b with three unicast sessions $(s_1, t_1), (s_2, t_2), (s_3, t_3)$. While it is useful to multicast the information from s_1 and s_2 to both t_1 and t_2, we should not include t_3 to the virtual terminal set as it would reduce R_3. To capture this difficulty, we can introduce one virtual session for each subset of terminal nodes. The original sessions can inject flow to virtual sessions if a multicast to the respective terminal set is helpful. So, with this rationale, the number of virtual sessions is exponential in the number of terminal nodes in the network which is still tractable for small networks.

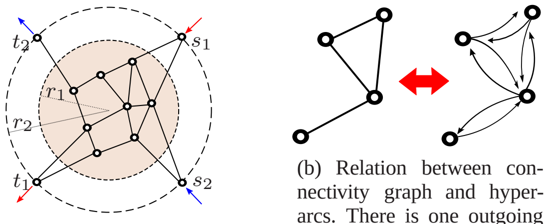
V. Performance Evaluation

We simulate the algorithm for wireless networks to see what gains can be expected. In all the simulations we used proportional fairness as utility function for each session, i.e. $U_c(R_c) = \log(R_c) \quad \forall c \in \mathcal{C}$. As $\sum_c U_c(R_c)$ is a concave function and all constraints are linear, all the presented optimization problems are convex. We use Matlab for simulations and CVX [19] to solve the optimization problems. Note that while the examples in Sec. IV were wireline networks, wireless networks with broadcast transmissions are simulated here. We restrict the representation of wireless networks to a connectivity graph. That is, nodes which are in mutual radio range are connected. The hyperarcs presented in Sec. II have to be considered as a one-to-many connection from each node to all its neighbors and are omitted for the sake of clarity in the following figures. The relation of hyperarcs and connectivity model is briefly shown in a small network in Fig. 3b.

We evaluate the performance of VM compared to NIC with respect to two indicators. First, if VM can improve the optimal sum-utility, the corresponding network problem is called *utility-improved (UI)*. If VM provides a higher rate for all sessions, that is $R_{c,VM} \approx k \cdot R_{c,NIC}, \quad \forall c \in \mathcal{C}$, we call the network problem *rate-improved (RI)*, with $k > 1$ denoting the average gain factor. Of course, every *rate-improved* problem is also *utility-improved*.

A. Mesh Networks

We consider the type of wireless mesh networks shown in Fig. 3a. Sources and terminals of two unicast connections are placed on a circle with radius r_2 whereas the other nodes of the network are randomly placed within a circle of radius r_1. Nodes are in mutual radio range if their distance is below a certain threshold value. The whole information has to travel through the center of the network, so we assume that all center nodes are able to overhear all packets from all sessions anyway. So, we can multicast to the whole center network and let the nodes at the border of the center forward the information demanded by connected sinks, similar to the example in Fig. 2a. However, it turned out that this choice of $T_{\tilde{c}}$ did not lead to observable gains of VM over NIC. However, if the virtual terminal set it chosen to be $T_{\tilde{c}} = \bigcup_{c \in \mathcal{C}} T_c$, we obtain convincing results: For the particular case of 6 nodes in the network, i.e. two nodes in the center, VM returns a higher sum-utility in nearly 75% of the cases. For 282 out of 1000 network problems, VM provides a higher rate for both sessions. The average gain for the rate-improved 6-node networks is 14%. Table I summarizes results for this setup and networks of larger size. The number of UI network problems decreases with more nodes in the center. In that case, the center of the network has enough capacity to forward both sessions, so it is not the bottleneck anymore. In many of these cases the maxflow bounds can achieved for both connections with NIC, which essentially means that both sessions do not interfere.



(a) Connectivity graph of a mesh network.

(b) Relation between connectivity graph and hyperarcs. There is one outgoing hyperarc per node.

Fig. 3: Connectivity graph and hyperarcs.

# nodes	6	8	10
# simulations	1000	1000	1000
# UI network problems	744	406	219
# RI network problems	282	117	32
average gain factor k for rates	1.14	1.10	1.13

TABLE I: Simulation results for mesh networks.

B. Random Geometric Wireless Networks

In this experiment we ask whether the gains observed for the mesh networks considered before also translate to random networks without special structure. Accordingly, nodes are randomly placed on a square with constant node density. Nodes can hear each other if their distance is smaller than a certain threshold. Sources and terminals are randomly chosen. For a network consisting of 10 nodes with two multicast sessions and two terminals for each connection, UI and RI network problems occur less frequently. However, for the average RI problem the rate region is increased by 33%. The network in Fig. 4 with two unicast sessions is an example for such an RI network problem. Here,
$R_{1,NIC} = 0.8 < R_{1,MSM} = 0.91 < R_{1,VM} = 0.92$ and
$R_{2,NIC} = 0.8 < R_{2,MSM} = 0.91 < R_{2,VM} = 0.97$, which corresponds to a gain of 15% and 21%, respectively.

We add further sessions to evaluate the behavior of VM if there is more interference in the network. For a 12-node network with three unicast sessions, VM can improve the utility for 259 out of 1000 network problems. As proposed in Sect. IV, one virtual session is created for each subset of original terminals. We count 37 RI network problems, with an average gain of 38%. As summarized in Table II, we lose 9% on average if only one virtual session is considered. The more sessions are added, the more frequently UI network problems can be observed. For a 12-node network with 4

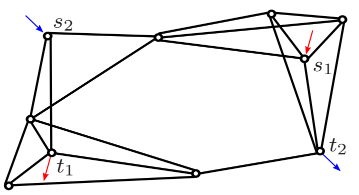


Fig. 4: Connectivity graph of an RI network problem with 10 nodes. Arrows represent sources and terminals of sessions.

# nodes	10	12	12
# simulations	1000	1000	1000
# sessions	2	3	4
# UI network problems	197	259	514
# RI network problems	63	37	46
average gain factor k for rates	1.33	1.38	1.28
average gain without receiver subsets	-	1.29	1.20

TABLE II: Simulation results for random networks.

unicast sessions, VM increases the utility in more than 50% of the cases. The results are shown in Table II.

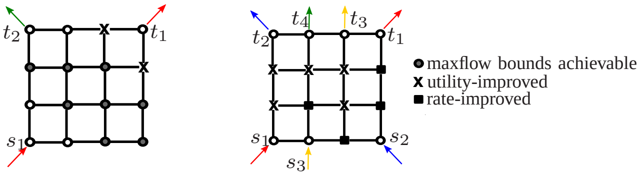
C. Wireless Grid Networks

We are further interested about how the positions of source and terminal have an impact on the performance of VM. Therefore, we analyze its behavior on more structured wireless networks shown in Fig. 5. In this setup, we are not particularly interested in the specific gains of VM but rather at which position it turns out to be useful. Consider the network in Fig. 5a: One session (s_1, t_1) and the position of the terminal for the second session, t_2, are fixed. We vary the position of the source for the second session and investigate if we can achieve higher rates by using VM with $T_{\tilde{c}} = \bigcup_{c \in \mathcal{C}} T_c$. We distinguish four cases: Positions of the source which let all the sessions achieve the maxflow bounds are marked gray. Positions leading to UI network problems are marked with a cross while positions leading to RI network problems are marked by a square. Other positions are unchanged.

In this case, only two positions close to the terminal lead to a UI network problem. For most positions we can achieve the maxflow bounds for each session individually which means that the sessions do not interfere. In the network in Fig. 5b we consider four connections. Three sessions (s_1, t_1), (s_2, t_2), (s_3, t_3) are fixed, we vary the position of the source for the fourth session, s_4. Now, as there is more interference in the network, we can do better with VM for every possible position in the network. We conclude that VM is particularly useful if interference reduces the rates achievable by NIC.

VI. Conclusion and Further Work

We presented an approach for inter-session network coding by creating virtual multicast sessions and leveraging the sufficiency of random linear network coding for this setup. Virtual sessions can be defined for an arbitrary virtual terminal set. The optimal choice of these sets is beyond the scope of this paper and should be considered in future work. The proposed technique turns out to be beneficial for many networks, with a rate increase of up to 30% for each connection, at the cost of computational complexity.



(a) Two sessions.

(b) Four sessions.

Fig. 5: Connectivity graphs of grid networks.

VII. Acknowledgement

This work was supported the Defense Advanced Research Projects Agency (DARPA) under subcontract #18870740-37362-C issued by Stanford University as well as by the European Commission through NEWCOM++.

References

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

[2] D.S. Lun, M. Médard, and R. Kötter. Network coding for efficient wireless unicast. In *2006 Int. Zurich Seminar on Comm.*, pages 74–77.

[3] D.S. Lun, N. Ratnakar, M. Médard, R. Kötter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE Trans. Netw.*, 14(SI):2608–2623, 2006.

[4] Y. Wu, P.A. Chou, and S.Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. In *IEEE Information Theory Workshop, 2004*, pages 304–309, 2004.

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

[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, aug. 2005.

[7] A.R. Lehman and E. Lehman. Complexity classification of network information flow problems. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 142–150. Society for Industrial and Applied Mathematics, 2004.

[8] Y. Wu. On constructive multi-source network coding. *Information Theory, 2006 IEEE International Symp. on*, pages 1349 –1353, 2006.

[9] C.C. Wang and N.B. Shroff. Pairwise intersession network coding on directed networks. *Information Theory, IEEE Transactions on*, 56(8):3879 –3900, aug. 2010.

[10] D. Traskov, N. Ratnakar, D.S. Lun, R. Kötter, and M. Médard. Network coding for multiple unicasts: An approach based on linear optimization. In *2006 IEEE Int. Symp. on Inf. Theory*, pages 1758–1762, 2006.

[11] M. Shao, X. Wu, and N. Sarshar. Rainbow network flow with network coding. *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, pages 1 –6, jan. 2008.

[12] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou. Network coding for multiple unicasts: An interference alignment approach. In *2010 IEEE Int. Symp. on Information Theory*, pages 1878 –1882, 2010.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3):497–510, 2008.

[14] M. Effros, T. Ho, and S. Kim. A tiling approach to network code design for wireless networks. *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, pages 62 –66, mar. 2006.

[15] F. Zhao and M. Médard. On analyzing and improving COPE performance. *Information Theory and Applications Workshop*, 2010.

[16] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1028 –1036, may. 2007.

[17] A. ParandehGheibi, A. Ozdaglar, M. Effros, and M. Medard. Optimal reverse carpooling over wireless networks - a distributed optimization approach. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1 –6, 2010.

[18] A. Lee and M. Médard. Minimum-cost subgraphs for joint distributed source and network coding. In *Workshop on Network Coding*, 2007.

[19] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, January 2011.