Minimum-Cost Multicast over Coded Packet Networks

Jan Schalkamp
Advisor: Stephan M. Günther
Seminar: Innovative internet technologies and mobile communications WS11/12
Chair for Network Architecture and Services
Department of Computer Science, Technische Universität München
Email: jan.schalkamp@in.tum.de

ABSTRACT

Streaming data to different receivers from a single source is a very common scenario. In general this would be achieved by using single source multicast, for which finding the most cost-efficient paths is very hard as it has to be computed in a centralized manner and is only affordable for little networks since it is not solvable in polynomial time. Although today's networks use e.g. heuristics to quickly find sub-optimal solutions to transmit data from one source to multiple receivers, it would be nice to find even better solutions in less time. However, changing the problem a little bit might help. Although routers are currently only able to forward and copy packets, there is no reason to limit them to only those functionalities. Having a network of routers capable of network coding – meaning being able to apply an arbitrary causal function on multiple incoming packets, resulting in one outgoing packet – can change this. This work gives an introduction to finding optimal cost solutions for multicast in a coded packet network, and shows that finding these solutions is possible in a decentralized manner resulting in only polynomial effort to calculate.

Keywords

Coded Packet Network, Multicast, Network Coding, Lagrangian Dual Problem, Optimization Problem, Minimum Cost Flow Problem

1. INTRODUCTION

Sending data from a single source to multiple sinks is an important task of today's networks. This, for example, is the case for video streaming services like YouTube, as the amount of traffic from video streaming is a crucial part of their expenses. Therefore it is vitally to reduce these cost as good as possible. There is a multitude of ways to achieve the minimal cost, which depend mostly on the characteristics of the network being used. If routers were only able to forward packets, the weighted shortest paths to each sink are the most cost-efficient way of delivering the data. These paths can easily be found by using the widely known Dijsktra algorithm. But as today's routers are also capable of copying packets and forwarding them to different receivers, the task of optimizing cost becomes more difficult. In general multicast would be used to transmit data from a single source to multiple sinks. However, finding the minimal cost in this kind of network becomes very difficult as calculating a minimum spanning tree is $\mathcal{NP}$-complete, which leads to a not affordable cost of computation. As there is no reason to limit routers to only being able to copy and forward packets, coded packet networks were introduced by Ahlswede et al [3]. In these networks, routers are able to combine multiple incoming packets by an arbitrary function to only one outgoing packet. This work, based on Desmund Lun's paper *Minimum-Cost Multicast Over Coded Packet Networks*, shows that this new characteristic of a network yields a way of minimizing the cost for multicast in a decentralized manner, which is also solvable in polynomial time. Therefore, we first take a look at the ways to deliver data from a single source to multiple sinks in different networks. Secondly a general optimization problem for minimizing costs in routed and coded packet networks are formulated. As solving these optimization problems would be out of the scope of this work, it is only shown that a specific kind of cost function leads to a way of solving the optimization problem in a decentralized manner.

2. COMPARING WAYS TO DELIVER INFORMATION TO MULTIPLE SINKS

As this work focuses on optimizing the cost for multicast in coded packet networks it is of great help to compare these networks with routed networks first. Therefore we represent a network by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, that is directed, weighted. $\mathcal{N}$ determines the set of Nodes, which would for example be routers in a real world application. The arcs between nodes are depicted by the set $\mathcal{A}$. Furthermore, we write an arc of set $\mathcal{A}$ as tuple (i, j) with $i, j \in \mathcal{N}$ as the arc's start, respectively end nodes. To model a network, we also need some kind of capacity c_{ij} which is the maximum rate at which packets can be transmitted over an arc (i, j) and cost for that transmission, which will be denoted by the arc's positive weight a_{ij}.

2.1 Dedicated unicast in routed networks

As we wish to transmit data from a single source $s \in \mathcal{N}$ to multiple sink nodes $t \in \mathcal{T}$ with $\mathcal{T} \subset \mathcal{N}\backslash\{s\}$, it is the easiest approach to send the data to the sinks one by one. Therefore, one unicast connection per sink node is established from the source to the sink node and the same information is transmitted $|\mathcal{T}|$ times. To transmit with the lowest cost the shortest path to each node is evaluated, for example by using the Dijsktra algorithm [6]. This results in a shortest path tree, giving the source the necessary information to send it's data to each sink via the lowest cost path per sink node. This would be the minimum-cost way to distribute data in a network only able to establish unicast connections. But as we will see in the forthcoming section 2.2, distributing the same information to multiple sink nodes by unicast is

doi: 10.2313/NET-2012-04-1_7

not the best way to do it. An example of the unicast approach can be seen in Figure 1. Here the shortest paths from the source Q to the sinks S_1 and S_2 are $(Q, 1), (1, S_1)$ and $(Q, 2), (2, S_2)$. Each path has costs of $2 \cdot (5+2)$, resulting in a total cost of 28.

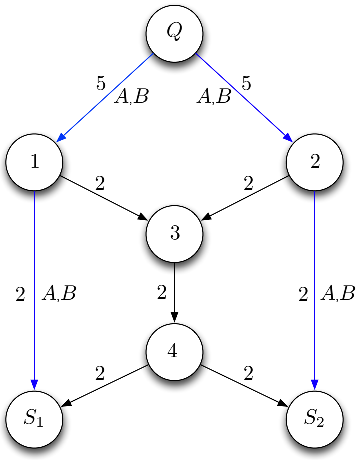

Figure 1: Unicast example

2.2 Multicast in routed networks

In networks with nodes being capable of not only forwarding packets but also copying and directing these packets to multiple nodes, multicast is a way to distribute information to multiple sinks. Due to nodes being able to copy packets it is no longer necessary to send each packet $|\mathcal{T}|$ times as can be seen in Figure 2. Here, the information A and B is only sent once to node 4 at costs of $2 \cdot (5+2+2) = 18$ and afterwards node 4 copies the received information and sends it to each sink at costs of $2 \cdot (2 \cdot 2) = 8$, resulting in total costs of 26. So, as one can see in comparison to the unicast's cost of 28, multicast is a better solution than unicast. As the optimal distribution path in routed networks is a Steiner tree, the total costs of transmitting information can only be as high as using the shortest path to each sink – just like the unicast solution does. This would be the case, if the costs for arcs $(1, 3)$ and $(2, 3)$ in our example were increased to 4. Calculating such a Steiner tree in polynomial time is not possible, which leads to a problem for big networks [4]. Nevertheless multicast without the use of Steiner trees is used in todays networks, as even sub-optimal solutions, calculated with the help of heuristics and specialized multicast protocols, are better than using unicast. In section 2.3 we see, that it is possible to find even lower cost solutions in polynomial time in a special kind of networks.

2.3 Multicast in coded packet networks

Just like in section 2.2 and real world networks, our nodes are able to forward, copy and direct copies to multiple nodes. But there is no reason to limit nodes to only that functionality. In coded packet networks, nodes are additionally able to use an arbitrary causal function on multiple incoming packets. This can be used to combine two packets , e.g. by using bitwise XOR $(A \oplus B)$, in order to reduce the total amount of packets needed to be transmitted through the network. As in the example of Figure 3, node 3 combines the incoming

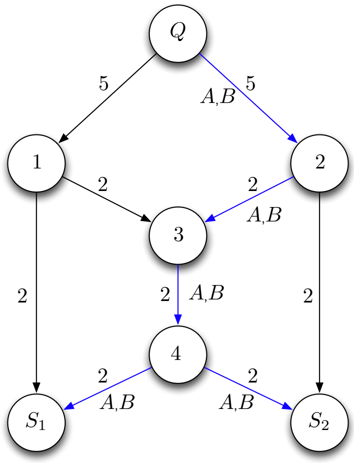

Figure 2: Multicast example

packets A and B to the new packet $A \oplus B$ and sends it via 4 to both sink nodes. Now, the packet $A \oplus B$ is useless for each sink, as they are not able to derive both original packets A and B from it. Therefore sink S_1 receives packet A and sink S_2 receives packet B, as this enables both sinks to derive the other packet by using XOR on $A \oplus B$ and A, respectively B. Now each arc is only used once and the total cost are 24. Just like in routed networks, the multicast depends greatly on the network's structure. Increasing the costs of arcs $(1, 3)$ and $(2, 3)$ again to 4, the optimal solution would be once more a unicast connection per sink. The advantage of using multicast in coded packet networks is not only the possibly lower cost of sending packets, but also the time to calculate those solutions. As finding the Steiner tree in a routed network was not solvable in polynomial time, network coding enables us to find optimal solutions for multicast in polynomial time, making it interesting for real world networks.

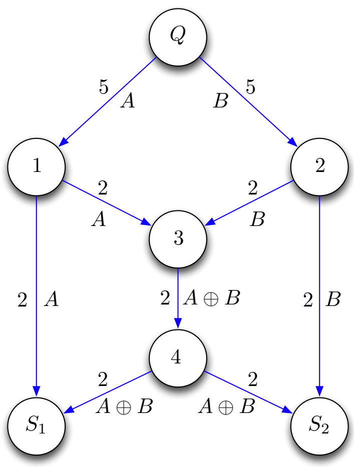

Figure 3: Multicast in coded packet networks example

doi: 10.2313/NET-2012-04-1_7

3. MULTICAST AS OPTIMIZATION PROBLEM

On the way of finding a solution to the problem of minimizing cost in a coded packet network, we first have to state our goal as an optimization problem in order to provide suitable algorithms capable of solving it. As we already have declared the capacity of an arc as c_{ij} with $(i,j) \in \mathcal{A}$, we denote z_{ij} as actual rate at which packets are injected into the arc. As the costs of a transmission in a network emerge from even those rates z_{ij} – because only used arcs lead to costs – we can state the monotonous increasing cost function depending on the vector $\vec{z}$:

$$f(\vec{z}) : \mathbb{R}^{|\mathcal{A}|} \rightarrow \mathbb{R}^{+} := [0, \infty). \quad (1)$$

Minimizing this function as it stands would lead to a not usable solution, as we haven't modeled our network correctly. Therefore, we develop some constraints, which help us to state our conditions for the network. As above stated, z_{ij} is the rate at which packets are injected into arc (i,j) and c_{ij} is the capacity of arc (i,j). Therefore we state our first constraint, the *capacity constraint*:

$$c_{ij} \geq z_{ij}, \forall (i,j) \in \mathcal{A}. \quad (2)$$

Sending some information over an arc $(i,j) \in \mathcal{A}$ does not necessarily mean that this information is new and also does not show which sink it is provided for. Therefore, we introduce $x_{ij}^{(t)}$ as the rate of new information intended for sink $t \in \mathcal{T}$. As the rate of new information for a single sink can not be greater than the overall rate of information at arc (i,j) and the rate of information must not be negative, as it would mean that information could be destroyed or lost, we state the *coupling constraint*:

$$z_{ij} \geq x_{ij}^{(t)} \geq 0, \forall (i,j) \in \mathcal{A}, \forall t \in \mathcal{T}. \quad (3)$$

Our third constraint describes the fact, that our network only has one source node $s \in \mathcal{N}$ that is providing information and multiple sink nodes $t \in \mathcal{T}$ that are consuming even that information. No inner node is allowed to either produce or consume information. This constraint is the *flow conservation constraint*:

$$\sum_{\{j|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)}, \forall i \in \mathcal{N}, \forall t \in \mathcal{T} \quad (4)$$

where

$$\sigma_i^{(t)} := \begin{cases} R, & \text{if } i = s \\ -R, & \text{if } i \in \mathcal{T} \\ 0, & \text{otherwise.} \end{cases}$$

R means the real, positive rate at which packets are transmitted to the sink nodes. For dedicated unicast this is easy to see, as the source node is producing information, the inner nodes forward every received information without tampering with it and sink nodes consume the information. In routed networks using multicast it is not that easy to see, as inner nodes are permitted to copy information and therefore a single packet may be of use for more than one sink. Therefore, the amount of incoming and outgoing packets does not have to be the same. In Figure 2, node 4 is an example for that. It receives packets A and B once but transmits them twice. To see that our third constraint (4) is still valid, we have to look at the actual information being hold by the packets received by node 4. As both A and B hold information for sinks S_1 and S_2, the received and the transmitted amount of information is equal. To make it easier to see, we take a closer look at the example of Figure 2 in numbers (only a look at node 4): Each packet's information rate from node 3 to 4 is $x_{3,4}^{(S_1)} = x_{3,4}^{(S_2)} = \frac{1}{2}$ since both sinks S_1 and S_2 are equally interested in packet A's as well as in packet B's information. Each packet provides half of the overall information each sink is interested in. Therefore, the incoming information is $2 \cdot (\frac{1}{2} + \frac{1}{2}) = 2$. The outgoing information to both sinks S_i is $x_{4,S_i}^{(S_i)} = \frac{1}{2}$ for packet A as well as B, which results in an outgoing amount of information of $2 \cdot (2 \cdot \frac{1}{2}) = 2$ as well. This holds also for the case of a coded packet network.

These three constraints are enough to tackle the problem of minimizing the cost in a multicast network. However, there are some differences between the optimization problem in routed and coded packet networks. This is dealt with in the two forthcoming sub chapters.

3.1 Optimizing multicast in a routed network

As we are now able to formulate the optimization problem, we first have a look at minimizing the cost for multicast in routed networks with one source node and multiple sink nodes. Of course, we still assume that the rates of new information $x_{ij}^{(t)}$ for sink t are greater than or equal to 0. Also, constraint (4), introduced as *flow conservation constraint*, and the *capacity constraint* (2), shall be valid in this network. Since we want to minimize a given cost function like (1) our optimization problem reads as follows:

$$\begin{aligned} &\text{minimize } f(\vec{z}) \\ &\text{subject to } c_{ij} \geq z_{ij} && \forall (i,j) \in \mathcal{A}, \\ &z_{ij} \geq \sum_{t\in\mathcal{T}} x_{ij}^{(t)}, x_{ij}^{(t)} \geq 0 && \forall (i,j) \in \mathcal{A}\ , \forall t \in \mathcal{T}, \\ &\sum_{\{i|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{i|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)} && \forall i \in \mathcal{N}\ , \forall t \in \mathcal{T}. \end{aligned}$$

Note, that the second constraint has changed a little bit in comparison to (3), as we now explicitly have a routed network in which packets cannot be coded. Now, as we have upper and lower bounds for the optimal vector $\vec{z}$, namely the capacity (2) and coupling constraint (3), we see that our optimal vector $\vec{z}$ lies within the positive orthant[1] of a bounded polyhedron. Being monotonically increasing, our cost function $f(\vec{z})$ can only be minimized by minimizing each of it's components z_{ij}. This, on the other hand, is reached when the coupling constraint $z_{ij} \geq \sum_{t\in\mathcal{T}} x_{ij}^{(t)}$, $x_{ij}^{(t)} \geq 0$, $\forall (i,j) \in \mathcal{A}$, $\forall t \in \mathcal{T}$ holds with equality.

3.2 Optimizing multicast in a coded packet network

We now consider a coded packet network. Nodes are able to apply arbitrary causal functions on multiple incoming packets like $A \oplus B$. The problem looks nearly the same as in a routed network, but since nodes are able to use network

[1] An n-dimensional vector lies within the positive orthant, iff all it's components are positive.

doi: 10.2313/NET-2012-04-1_7

coding, the coupling constraint changes accordingly:

$$\begin{aligned}
&\text{minimize } f(\vec{z}) && (5)\\
&\text{subject to } c_{ij} \geq z_{ij} && \forall (i,j) \in \mathcal{A},\\
&z_{ij} \geq x_{ij}^{(t)} \geq 0 && \forall (i,j) \in \mathcal{A}\ , \forall t \in \mathcal{T},\\
&\sum_{\{i|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{i|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)} && \forall i \in \mathcal{N}\ , \forall t \in \mathcal{T}.
\end{aligned}$$

As nodes are now able to code packets, they also become able to send the same amount of information with less packets. As Figure 3 shows, arc (3,4) only has to be used once but still carries the same amount of information for sinks S_1 and S_2 (assuming S_1 received A and S_2 received B already). The coupling constraint is less restrictive. $\vec{z}$ is now optimal iff the following equality holds for every component of $\vec{z}$ [2]:

$$z_{ij} = \max_{t\in\mathcal{T}}\{x_{ij}^{(t)}\} = ||x_{ij}^{(t)}||_{\infty,(t)},\ \forall (i,j) \in \mathcal{A},\ \forall t \in \mathcal{T}. \quad (6)$$

As we have not specified the cost function $f(\vec{z})$ yet, the next section deals with solving the optimization problem for a specific class of cost functions.

4. SOLVING THE OPTIMIZATION PROBLEM IN CODED PACKET NETWORKS

This section deals with a specific class of cost functions. We now only consider linear, separable cost functions, and separable constraints for each arc $(i,j) \in \mathcal{A}$. Separable constraints mean in this case that each arc's capacity is subject to a separate constraint, bounding it to a positive value independent from the other arcs' bounded capacities. Linear means, that the cost of transmitting data over arc $(i,j) \in \mathcal{A}$ grows linear with the amount of data transmitted. For instance, this is the case if the cost represent monetary cost, such as a fixed value per kilobyte. We also demand that the cost of transmitting data is non-negative. These constraints lead to a cost function $f(\vec{z})$ looking as follows:

$$\begin{aligned}
f(\vec{z}) &:= \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij} \qquad (7)\\
&\text{with } a_{ij} \geq 0\ \forall (i,j) \in \mathcal{A}.
\end{aligned}$$

In the forthcoming sub sections, we formulate the optimization problem in two different ways, as a different formulation of the problem may lead to a different way of solving the optimization.

4.1 A first approach to stating the optimization problem

In this approach, the optimization problem looks nearly the same as the previous stated general optimization problem (5). We just rewrote the coupling constraint, so that z_{ij} is no longer bounded explicitly above. However, it is still bounded above implicitly by the characteristics of a coded packet network (6), that limits z_{ij} by the maximum of the arc's flows $x_{ij}^{(t)}$.

$$\begin{aligned}
&\text{minimize } \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij} && (8)\\
&\text{subject to } c_{ij} \geq z_{ij} \geq 0 && \forall (i,j) \in \mathcal{A},\\
&z_{ij} \geq x_{ij}^{(t)} && \forall (i,j) \in \mathcal{A}\ , \forall t \in \mathcal{T},\\
&\sum_{\{i|(i,j)\in\mathcal{A}\}} x_{ij}^{(t)} - \sum_{\{i|(j,i)\in\mathcal{A}\}} x_{ji}^{(t)} = \sigma_i^{(t)} && \forall i \in \mathcal{N}\ , \forall t \in \mathcal{T}.
\end{aligned}$$

Although solving this problem would lead to the optimal solution, we are not satisfied with this form of the problem, as it cannot be solved in a distributed manner. This might seem a little strange as it does look a lot like a normal minimum-cost flow problem for which a variety of algorithms exist [8]. Unfortunately our flow conservation constraint (4) is referring to the flows $x_{ij}^{(t)}$ and not to the vector $\vec{z}$. This is a result of network coding, as flows can now share bandwidth instead of competing for it. Hence, as it stands, this problem has a major disadvantage: it requires full knowledge of the network and is only solvable in a centralized manner.

4.2 The optimization as dual problem

Dualizing a problem has the benefit of reducing the amount of constraints. Those constraints are not be lost, but are dualized into the problem's function. We therefore have to formulate the Lagrangian of the primal problem (8) to obtain the dual function, which will be maximized in order to obtain the dual problem. However, it is possible that optimizing the dual problem may lead to a different solution than the primal problem, wherefore the equality of this case will be shown.

Dualizing the coupling constraint $z_{ij} \geq x_{ij}^{(t)}$ forms the following Lagrangian:

$$\begin{aligned}
\mathcal{L}(\vec{x},\vec{z},\vec{\lambda}) &= f(\vec{z}) + \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} (x_{ij}^{(t)} - z_{ij})\\
&= \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij} + \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} (x_{ij}^{(t)} - z_{ij})\\
&= \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij} + \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} (\lambda_{ij}^{(t)} x_{ij}^{(t)} - \lambda_{ij}^{(t)} z_{ij})\\
&= \sum_{(i,j)\in\mathcal{A}} a_{ij} z_{ij} - \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} z_{ij} + \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} x_{ij}^{(t)}\\
&= \sum_{(i,j)\in\mathcal{A}} z_{ij} \cdot \left(a_{ij} - \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} \right) + \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} x_{ij}^{(t)}.
\end{aligned}$$

Since we dualized the coupling constraint, it is no longer an explicit constraint of our optimization problem. The remaining constraints, the capacity and flow conservation constraint are transformed into a set $\mathcal{X}$ that contains all vectors $\vec{x}$ fulfilling even these constraints. Towards stating the dual problem, the dual function $\Theta(\vec{\lambda})$ has to be found first by minimizing the Lagrangian $\mathcal{L}$ with respect to $\lambda_{ij}^{(t)} \geq 0\ \forall (i,j) \in \mathcal{A}, \forall t \in \mathcal{T}$. As z_{ij} is no longer bounded above, it is allowed to take any value in order to minimize the Lagrangian $\mathcal{L}$, wherefore

$$\min_{\vec{x}\in\mathcal{X},\vec{z}} \sum_{(i,j)\in\mathcal{A}} z_{ij} \cdot \left(a_{ij} - \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} \right) + \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} x_{ij}^{(t)} \quad (9)$$

doi: 10.2313/NET-2012-04-1_7

evaluates to $-\infty$, if $a_{ij} - \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)}$ is either greater or less than 0. In case $\sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} = a_{ij}$ it is exactly zero and the minimized Lagrangian (9) evaluates to $\min_{\vec{x}\in\mathcal{X}} \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} x_{ij}^{(t)}$. Therefore the dual function $\Theta(\vec{\lambda})$ looks as follows:

$$\Theta(\vec{\lambda}) := \inf_{\vec{x}\in\mathcal{X},\vec{z}} \mathcal{L}(\vec{x},\vec{z},\vec{\lambda})$$

$$\inf_{\vec{x}\in\mathcal{X},\vec{z}} \mathcal{L}(\vec{x},\vec{z},\vec{\lambda}) = \begin{cases} \min_{\vec{x}\in\mathcal{X}} \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} x_{ij}^{(t)}, & \text{if } \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} = a_{ij}, \\ -\infty, & \text{otherwise.} \end{cases}$$

Although $-\infty$ would be a correct mathematical solution, it holds no meaning in reality, wherefore we eliminate that possible solution by adding a constraint concerning the Lagrangian multipliers λ, leading to the dual problem:

$$\begin{aligned} &\text{maximize} && \Theta(\vec{\lambda}) && (10)\\ &\text{with} && \Theta(\vec{\lambda}) := \min_{\vec{x}\in\mathcal{X}} \sum_{(i,j)\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} x_{ij}^{(t)}, \\ &\text{subject to} && \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} = a_{ij}, \lambda_{ij}^{(t)} \geq 0 \ \forall (i,j)\in\mathcal{A}, \forall t\in\mathcal{T}. \end{aligned}$$

To see that the optimal solution to the dual problem is equal to the primal problem's (8) optimal solution, we have to show that strong duality[2] holds. This can be done by rewriting the primal problem's dualized constraint (the coupling constraint) as constraint functions:

$$h_{ij}^{(t)}(z_{ij}) := x_{ij}^{(t)} - z_{ij} \leq 0.$$

It is obvious that each $h_{ij}^{(t)}$ is less than or equal to zero, because z_{ij} is equal to or greater than $x_{ij}^{(t)}$ (6). Slater's condition[3] holds in case all constraint functions are less than or equal to zero and at least one $h_{ij}^{(t)} < 0$ exists. Since each sink needs at least one raw packet to retrieve the information of the coded packet, it is guaranteed that such an $h_{ij}^{(t)}$ exists and strong duality is proven [7].

We rewrite our problem to make differences to the primal problem (8) more obvious:

$$\begin{aligned} &\text{maximize} && \Theta(\vec{\lambda}) \\ &\text{with} && \Theta(\vec{\lambda}) := \sum_{t\in\mathcal{T}} \zeta^{(t)}, && (11)\\ & && \zeta^{(t)} := \min_{\vec{x}^{(t)}\in\mathcal{X}^{(t)}} \sum_{(i,j)\in\mathcal{A}} \lambda_{ij}^{(t)} x_{ij}^{(t)} && (12)\\ &\text{subject to} && \sum_{t\in\mathcal{T}} \lambda_{ij}^{(t)} = a_{ij}, \lambda_{ij}^{(t)} \geq 0 \ \forall (i,j)\in\mathcal{A}, \forall t\in\mathcal{T}. \end{aligned}$$

Note, that the set $\mathcal{X}$ has now been separated by flows into multiple sets $\mathcal{X}^{(t)}$. Since we broke the coupling between the flows by dualizing the coupling constraint, we can now use distributed methods to solve the optimization problem. As mentioned in section 4.1, the primal problem looked a lot like a normal minimum-cost flow problem. By reformulating the dual problem (10), we defined a $\zeta^{(t)}$(12), which is in fact a standard minimum-cost flow problem and can therefore be solved e.g. by using the ϵ-relaxation method. The overall optimization (11) can then be done by using subgradient-optimization. Both techniques will not be covered at this point. Subgradient-optimization of even this optimization problem is to be found in [1], whereas the ϵ-relaxation is discussed in [5].

[2]Strong duality holds, iff the difference between the primal problem's and the dual problem's solution is 0. Here, (minimize $f(\vec{z})$) − (maximize $\Theta(\vec{\lambda})$) = 0.

[3]Slater's condition is a constraint qualification guaranteeing equality of primal and dual optimal solutions.

5. CONCLUSION

As we have seen by comparing the different methods for delivering data from a single source to multiple sinks, solving the optimization problem strongly depends on the network's capabilities. In case of a routed network with unicast capabilities, the optimization problem is even $\mathcal{NP}$-complete. Hence, we concentrated on minimizing the cost in coded packet networks wherefore we first had a look at how to state the optimization problem with respect to the three constraints (capacity, coupling and flow conservation) in general. Afterwards we specified a linear cost function in order to see, that the first approach of solving the problem would not have been solvable in polynomial time. However, by dualizing the coupling constraint, and abbreviating the capacity and flow conservation constraint by the set $\mathcal{X}$, we were able to obtain the dual problem by maximizing the Lagrangian and adding an additional constraint in order to exclude a not meaningful solution. It was also crucial to show that the dual problem's optimal solution was the same as the optimal primal problem's solution by showing that strong duality holds with help of Slater's condition. In the end, we rewrote the dual problem to see that a part of it is a standard minimum-cost flow problem for which a variety of algorithms exist. Having solved that particular subproblem, the whole problem became solvable by employing subgradient-optimization.

6. REFERENCES

[1] Desmond S. Lun, Niranjan Ratnakar, Muriel Medard, Ralf Koetter, David R. Karger, Tracey Ho, Ebad Ahmed and Fang Zhao: *Minimum-Cost Multicast Over Coded Packet Networks*, IEEE Transactions on information theory, Vol. 52 No.6, pp. 2608 - 2623, 2006.

[2] Stephan M. Günther: *Optimal Cost Multicast over Coded Packet Networks*, Optimization in Communication and Signal Processing, Ferienakademie Sarntal, 2009.

[3] Rudolf Ahlswede, Ning Cai, Shuo-Yen Rober Li and Raymond W. Yeung: *Network Information Flow*, IEEE Transactions on information theory, Vol 46., no. 4, pp. 1204 - 1216, 2000.

[4] F. Hwang, D.S. Richards and P. Winter: *The Steiner tree problem*, North Holland, 1992.

[5] D.P. Bertsekas and J.N. Tsitsiklis: *Parallel and Distributed Computation: Numerical Methods.*, Englewood Cliffs, NJ: Prentice Hall, 1989

[6] T.H. Cormen and C.E. Leiserson, R.L Rivest and C. Stein: *Introduction to algorithms*, MIT Press, Cambridge, Ma, 2001.

[7] Stephen Boyd and Lieven Vandenberghe: *Convex Optimization*, 6th edition, Cambridge University Press, 2008.

[8] Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin: *Network Flows: Theory, Algorithms, and Applications.*, Prentice-Hall, Inc., 1993.