Tutorial 8 - Traffic engineering

Gidon Rosalki

2025-12-18

Notice: If you find any mistakes, please open an issue at `https://github.com/robomarvin1501/notes_networking`

1 Introduction to Traffic Engineering

Last tutorial we discussed ways to find the fastest route to a destination. However, we do not necessarily always want to send all the data through there, since that will increase the traffic, limiting how much information can be sent that way, resulting in slower transmission. We want to share the transmission across more possible routes in order to reduce network traffic.
We do not normally touch this, networks usually mostly manage themselves. Routing protocols adapt to changes, in order to manage traffic. TCP (to be discussed in the future) can identify congestion, and adapt to reduce the traffic on a network. However, they do not do that as efficiently as we might have originally desired, TCP cannot send data over a less congested path if one exists.

We want to monitor our resources (like bandwidth) in order to distribute load in ways that will reduce congestion, delay, and comply with application specific requirements. This can be achieved by fine tuning the routing protocol parameters, which are for the most part, the weights of the graph. This is not necessarily an easy problem.

2 Modelling networks as linear programs

There are many scenarios where we can model our world using functions. This modelling can help us find solutions to problems in (hopefully) polynomial, or even linear time. We will return to linear programming, as discussed in algorithms. The standard form is for a given set of constraints $\{c_i\}_{i=1}^{n}$, and constraints

$$\{a_{i,j}\}_{i,j=1}^{m,n}$$
$$\{b_i\}_{i=1}^{m}$$

Then we can maximise:

$$\max_x \left\{ \sum_{j=1}^{n} c_j x_j \right\}$$
$$\text{Subject to } \sum_{j=1}^{n} a_{ij} x_j \leq b_j \ i \in [m]$$
$$x_j \geq 0 \ j \in [n]$$

We know from algorithms that every LP problem has another representation, which provides an upper bound for the original LP, and the solution for one defines the solution for the other (duality theorem):

$$\min_y \left\{ \sum_{j=1}^{n} c_j y_j \right\}$$
$$\text{Subject to } \sum_{j=1}^{n} a_{ij} y_i \geq c_j \ j \in [n]$$
$$y_i \geq 0 \ i \in [m]$$

We also saw in algorithms the problem of MAX-FLOW, where for a given network, with a source node, and a destination (sink) node, find the maximum flow between the source and the sink.

We will not be discussing MAX-FLOW here, but rather *Multi-Commodity Flow.* This is because in a network, there are normally more than one set of active source, and destination pairs. We call the set of demands the "Demand Matrix".
So now we have a set of commodities K where $K = \{K_i = (s_i, t_i, d_i)\}_{i=1}^{n}$, and $f_i(u, v)$ which is the flow of the

commodity i on the edge (u,v). Note that we allow every commodity to have its own demand (compared to just maximising the flow). The constraints become

$$\forall i, \forall v \in V \setminus \{s_i, t_i\} : \sum_{u:(u,v)\in E} f_i(u,v) = \sum_{w:(v,w)\in E} f_i(v,w)$$
$$\forall (u,v) \in E : \sum_{u:(u,v)\in E} f_i(u,v)$$
$$\forall i, \forall (u,v) \in E : f_i(u,v) \geq 0$$

We have many things that we can optimise. For example, we can maximise the total amount of sent traffic (i.e., maximise $\sum_v |f_v|$, where $|f_v|$ is the total amount of traffic sent by v). We could also minimise congestion, where we minimise the load on the most congested edge (So minimise $\max_e \left\{ \frac{f_e}{c_e} \right\}$ where f_e is the flow along edge e). We could try and fairly allocate the resources, and even more, but those are less interesting to us right now.

So, let us consider the case of maximising the total amount of traffic sent, maximum multi commodity flow, which we will call Max-MCF. We need to satisfy the capacity constraint, in that no edge can send more than its capacity. When solving the problem, we can decide how much traffic to send from each of the commodities (nodes).

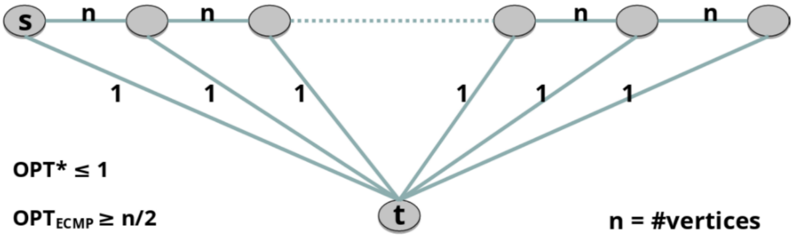
In MinCong-MCF we aim to minimise the load on the most congested edge. We need to send all the commodities, but this may possibly exceed capacities. By using MCF we optimised routes from sources to destinations (according to some optimisation goal). Most times we want to use multiple (shortest) paths between hosts. So far, we assumed that we route on a **single** shortest path. This means that our routing mechanisms need to split traffic, which brings us on to ECMP.

3 ECMP

ECMP is Equal Cost Multi Path. We want routers to route on all shortest paths between a source/destination pair. Each router keeps a set of next hops along shortest paths to a destination (i.e., the first node from the current node, on the shortest path to the destination node). Will "split" traffic evenly along those next hops (routers).

3.1 Optimising

We want to set the graphs weights so that ECMP would be the optimal solution, with respect to some optimization goal. For example: minimising the most congested link but allowing total flow to exceed the link capacity. There are not always link weights such that ECMP are optimal, but we can create link weights such that ECMP will hopefully approach the optimal solution. Consider the following graph:



Here we want to send n units of flow from s to t. This can be achieved by sending 1 along their direct link, and sending $n-1$ to the next node. This node can then send 1 along its direct link, and $n-2$ to the next node. This can keep happening until n units have been sent from s to t

Finding the optimal weights under the constraints of using ECMP and weights is a problem that is NP-hard. Even finding an approximation for the min-congestion flow with *any* constant factor is NP-hard. This is even true for a single source-destination pair. So instead of solutions, we are left with heuristics. We are not going to learn the heuristics properly, but will learn the simplest which is equally splitting among the available paths.

So, we need to establish how we know how to split. One method is to simply go over all the next hops from this node, in a "round robin" fashion. This has the problem that it could lead to a reordering of the packets, due to each of the hops having a different round trip time. So, we will add the constraint that we want all the packets of the same *flow* to have the same *path*.

3.2 ECMP Hashing

We would like all packets of the same flow to traverse the same path. To do this, we will use modulo N hash, where N is the number of next hops for the destination. The hash table maps between a flow and the next hop (router). There exists many hash functions one can use, and in practice, we don't know which hash functions vendors use.

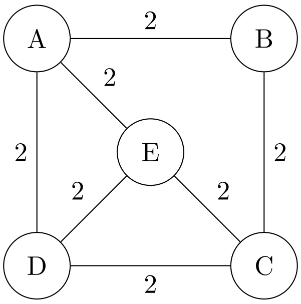
So, in order to hash, we use information from the packet header. This includes the source IP address, and port number, destination IP address and port number, and the protocol in use. These comprise the ECMP 5-tuple. This way, each packet of the same flow will have these same 5 values.

This does have some problems. Multiple "heavy" flows in the network, which share the same link, can result in poor performance. One could say that one needs to distribute the "heavy" flows differently, but this is difficult, since it is unclear to define heavy, and to identify it.

4 Questions

4.1 Definitions

Consider the following network



The capacity of each edge is 2. The weight on each edge in OSPF (uses link state) is positive/infinite On each edge, every sub-flow can be in either direction, and the total flow is the sum of all the sub-flows.

We will create the following notations:

- $Max - MCF_{OSPF/ECMP}$ is the maximisation problem of the total network flow, when using OSPF and ECMP, without restricting each flow to a single path
- $Max - MCF_{OPT}$ is the maximisation problem of the total network flow without restricting each flow to a single path, and without restricting us to equal division, any division is possible between nodes that are part of the shortest path

Similarly:

- $MinCong - MCF_{OSPF/ECMP}$ is the minimisation problem of the maximal congestion (over an edge), when using OSPF and ECMP, without restricting each flow to a single path
- $MinCong - MCF_{OPT}$ is the minimisation problem of the maximal congestion without restricting each flow to a single path, and without restricting us to equal division, any division is possible between nodes that are part of the shortest path

Recall that in both MinCong-MCF problems, **all** commodities must be sent, even if one surpasses the edge capacities, and in Max-MCF, the solution is restricted such that the edges capacities are not surpassed.

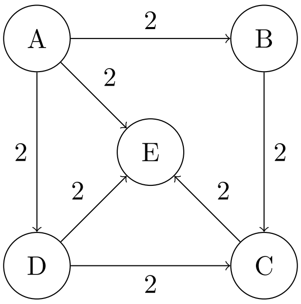
4.2 Question 1

Assume that there are two commodities (source, destination, demand):

$$(A, C, 5)$$
$$(C, E, 3)$$

What is the optimal solution of $Max - MCF_{OPT}$?

4.2.1 Solution



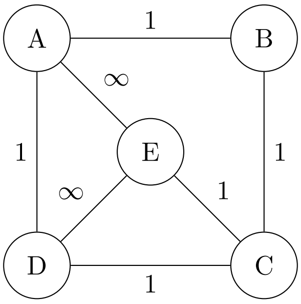
The optimal solution is 6, A sends 4 units, and C sends 2. Let us assume by contradiction that there is a better solution $s' > 6$. Let us now consider node C, for whom all the edges are saturated. Since C is in both commodities, then in s', the flow on one of the edges connected to C will increase, which violates the capacity constraints.

4.3 Question 2

Is there a feasible assignment of edges weights s.t. the solution to $Max - MCF_{OSPF/ECMP}$ yields the optimal solution from the previous question?

4.3.1 Solution

Such an assignment exists, and is shown below:



So, the packets from A to C will be equally routed through B and D, and packets from C to E will be routed directly. Solving this optimisation problem will (in this case) then imply the same flows as in the solution to $Max - MCF_{OPT}$

4.4 Question 3

Consider the following commodities:

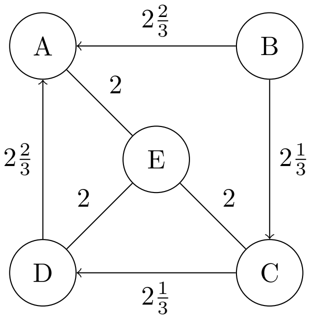
$$(B, D, 5) \tag{1}$$
$$(C, E, 3) \tag{2}$$
$$\tag{3}$$

What's the optimal solution for the $MinCong - MCF_{OPT}$ problem in this scenario? Show a flow that achieves that solution, and explain your answer.

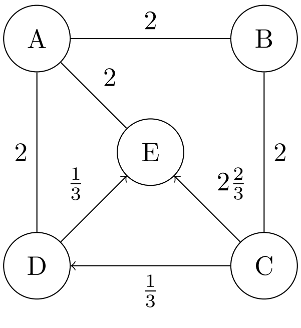
4.4.1 Solution

Let us analyse the maximum load by looking at the cut $(V, V \setminus \{B, C\})$. We will note that **all** the commodities must flow through this cut. Therefore, the minimal maximum load is lower bounded by the minimal load on this cut. The total commodities sun to 8 units, and the cut has 3 edges, with a total capacity of 6 units. Therefore, the minimal maximum load is $\frac{8}{6} = \frac{4}{3}$. So, we want a maximum of $\frac{f}{c} = \frac{4}{3} \implies f = \frac{8}{3}$ for each edge.

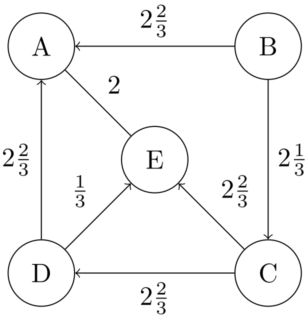
Here is a possible solution. Firstly, the flow for $(B, D, 5)$:



Now the flow for $(C, E, 3)$:



Resulting in an overall flow of:



4.5 Question 4

Suggest the weights assignment such that the solution to $Min-Cong_{OSPF/ECMP}$ (the maximum load) is 1.5. Explain your answer. This is for the following commodities:

$$(B, D, 5) \tag{4}$$
$$(C, E, 3) \tag{5}$$
$$\tag{6}$$

4.5.1 Solution

The maximum must be 3, since

$$\frac{f}{c} = \frac{f}{2} = 1.5 \implies f = 3$$

There, the edges will have the loads:

$$(A, B), (A, D), (B, C), (C, D) : \frac{2.5}{2} \tag{7}$$
$$(C, E) : \frac{3}{2} \tag{8}$$

4.6 Question 5

Is there a feasible assignment of edge weights such that the solution for $MinCong - MCF_{OSPF/ECMP}$ yields the optimal solution from question 3, which had the maximum load of $\frac{4}{3}$? If there is such an assignment, show it and explain why it yields the result, and if there is not such an assignment, then explain why.

4.6.1 Solution

There is no such assignment. We will assume by contradiction that such an assignment exists. In ECMP, if a node routes a flow to multiple neighbours, then the flow is split equally between them. Consider the node B:

- If B routes all 5 units over one edge, then the load on said edge will be $\frac{5}{2} > \frac{4}{3}$ - obviously worse than the optimal solution in question 3
- If B routes all 5 units over its two edges, then the total outgoing flows from C will be $3 + 2.5$. Considering C, the minimal load on its edges will be where it routes over both (C, E) and (C, D), and the load on the edges will be $\frac{5.4}{4} > \frac{4}{3}$, so once again, worse than the optimal solution from question 3

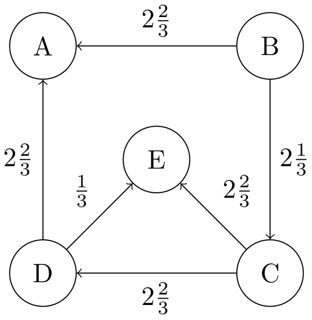
We may perform a similar analysis starting on node C which shows that C will not send on (C, B), and conclude the answer.

4.7 Question 6

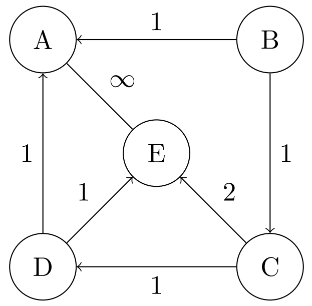
Now consider a modified version of OSPF/ECMP, in which flows can be unequally divided between neighbours with shortest paths. Is there now a feasible assignment of edges weights such that the solution for $Min-Cong_{OSPF/ECMP}$ yields the optimal solution from question 3 (maximal load= $\frac{4}{3}$)?

4.7.1 Solution

Such an assignment exists. The idea is to assign weights that will enable such a solution. For example, consider a possible solution (that yields the optimal maximal load) from question 3:



We can use the following weights assignment to yield this solution:


A
1
B
∞
1
E
1
1
2
D
C
1