HSN Spring Workshop 2006

Virtual Private Network Design: On the Tree Routing Conjecture

By Zoltán Király and Dömötör Pálvölgyi
Dept of Comp. Sci. and CNL, ELTE, Budapest

What is the VPN Design?

- In the VPN (Virtual Private Network) model the communication net is represented by an indirected, connected graph G.
- Each node (user) has a threshold b that determines the maximum rate of communication it intends to make (total of sending to and receiving from the other nodes).
- Each edge (physical link) has a cost c for renting one unit of capacity.
- We have to declare a path between each pair of nodes through which they communicate *before* knowing who will communicate with who.
- We also have to reserve bandwidth w on each edge so that for any communication needs (under the given bound $b(v)$ for each v) if we direct the information via the declared paths, the total traffic through each edge remains under the bandwidth of e.
- Our goal is to minimize the total cost $\sum w(e)c(e)$.

Different Versions

If we have a graph G with communication bounds b on the nodes and edge costs c, let us denote this triple simply by $\mathcal{G}$ and the cost of the optimal solution for the VPN design problem by $S(\mathcal{G})$. Different versions of the problem:

- *Tree-version*:
  The reserved edges have to form a tree.
  $T(\mathcal{G})$ = cost of the optimal solution.
  $S(\mathcal{G}) \leq T(\mathcal{G})$.
  Gupta et al. showed that this can be solved in polinomial time.
  Tree Routing Conjecture: $S(\mathcal{G}) = T(\mathcal{G})$.

- *MultiPath-version*:
  Between two nodes more paths are allowed but the ratio how the communication is split has to be declared in advance.
  $M(\mathcal{G})$ = cost of the optimal solution.
  $M(\mathcal{G}) \leq S(\mathcal{G})$.
  With an LP-formulation $M(\mathcal{G})$ can be computed in polinomial time.
  Conjecture: $M(\mathcal{G}) = T(\mathcal{G})$.

- *Couple-versions*:
  For any of the versions, we can introduce another restriction, that each node is allowed to communicate with only one node at a time.
  The cost of the optimal solution is denoted by (respectively) $S_2(\mathcal{G})$, $T_2(\mathcal{G})$ and $M_2(\mathcal{G})$.
  $S_2(\mathcal{G}) \leq S(\mathcal{G})$, $M_2(\mathcal{G}) \leq M(\mathcal{G})$ and $T_2(\mathcal{G}) \leq T(\mathcal{G})$.

About the Tree Solution

If the reserved edges form a tree, then it is unnecessary to declare the paths because there is only a single path between any pair of nodes, this makes it easy to realize in practice. There is also a very simple algorithm to find the optimal tree. So this is why we would like the Tree Routing Conjecture to be true, this way we could solve any VPN problem fast and in an easy-to-realize way.

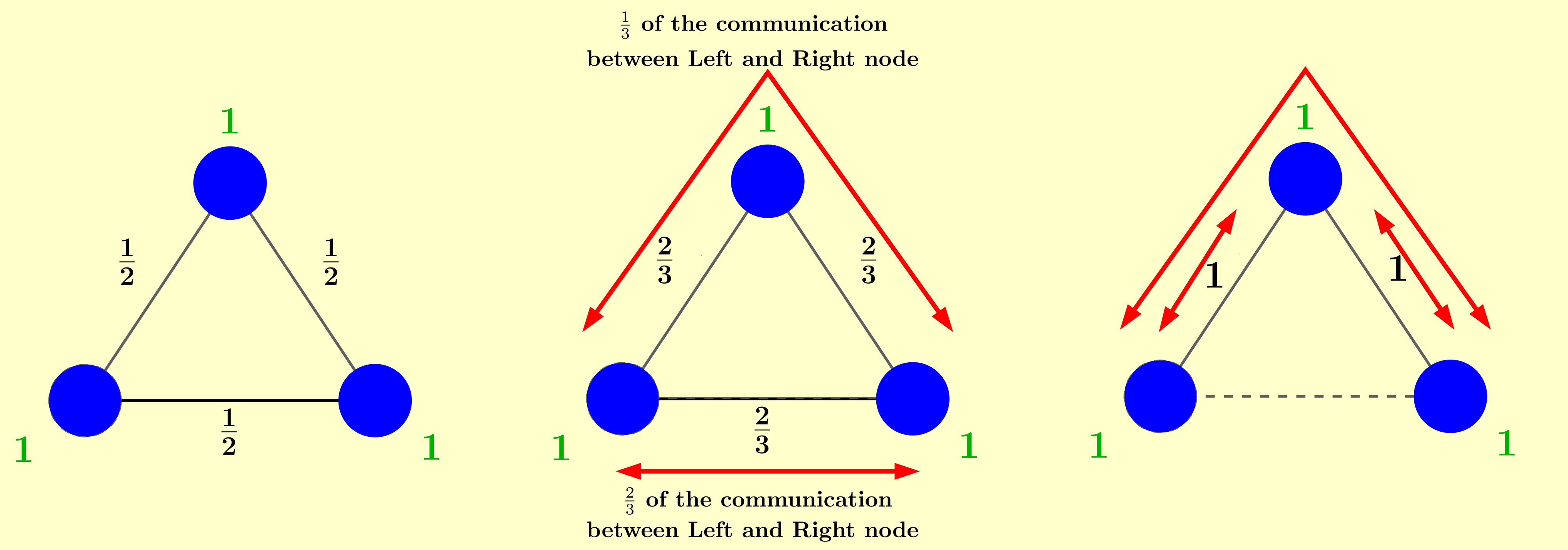
The solution of the Tree-version:
We will prove that the optimal solution is the tree of the cheapest paths from a vertex r with appropriate bandwidths. Let us suppose that we already have a tree solution T for a VPN problem $\mathcal{G}$.

- Def A_e and B_e: components of $T - e$.
- Bandwidth needed: $w(e) = \min\{\sum_{v \in A_e} b(v), \sum_{v \in B_e} b(v)\}$.
- $Cost(T) = \sum_{e \in T} w(e)c(e) = \sum_{e \in T} \min\{\sum_{v \in A_e} b(v), \sum_{v \in B_e} b(v)\}c(e)$.
- Def r: weighted center of the tree T (with weights $b(v)$).
- Def R_e: component *not* containing r in $T - e$.
- Def $d_H(r, v)$: cheapest path between r and v in the graph H.
- $Cost(T) = \sum_{e \in T} \sum_{v \in R_e} b(v)c(e) = \sum_v b(v) \sum_{(e \in T: v \in R_e)} c(e) = \sum_v b(v) d_T(r, v)$.
- Corollary: $Cost(T) \geq \sum_v b(v) d_G(r, v)$
- If T = tree of the cheapest paths from r, then equality holds.

Algorithm: compute the cheapest paths between any two vertices and then find the r for which $\sum_v b(v) d_G(r, v)$ is minimal.

Simple Examples

Here we examine the possible solutions of VPN design for a given graph. The graph G is a three-length circuit with communication bound 1 on each vertex, and edge costs 1. We examine three different designs.



This renting can solve any traffic with the given bounds between the three nodes, but it is *NOT* a VPN design; for different traffic matrices we have to choose different paths between the nodes. (However, it is a good solution for the MultiPath Couple-version, so $M_2(\mathcal{G}) = 3/2$.)

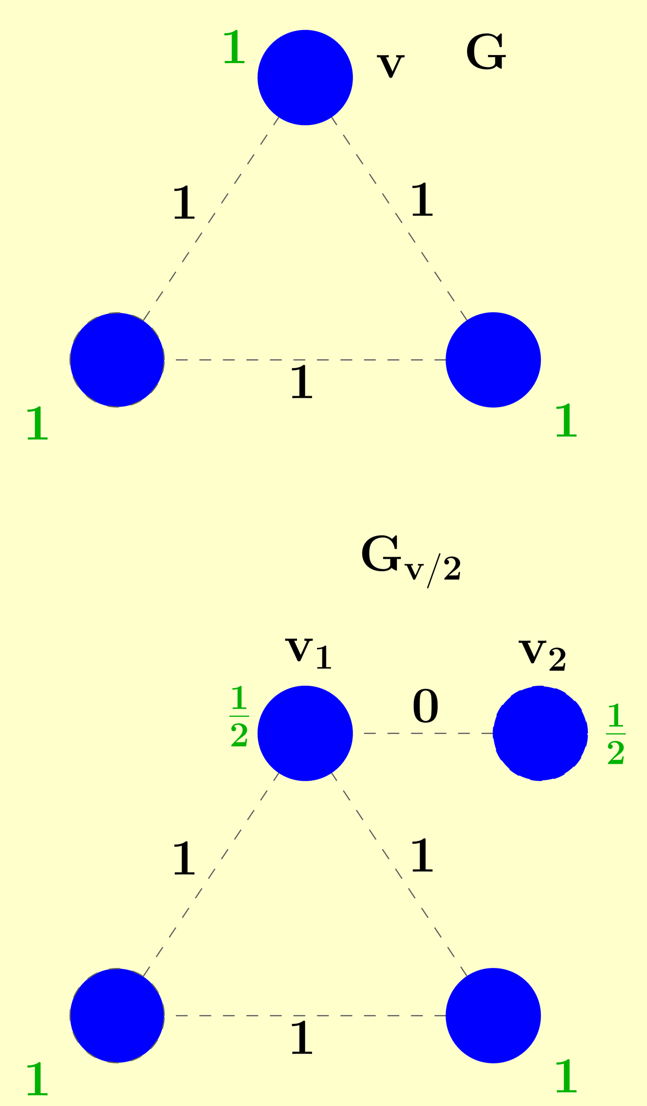
If we rent bandwidth $2/3$ on each edge, and between any two nodes we declare two paths, $2/3$ of the information going on the shorter and $1/3$ going on the longer path, we can satisfy any communication needs. It is an easy computation to prove that this is optimal, hence $M(\mathcal{G}) = 2$.

By renting 1 bandwith on these two edges and none on the lower one, we can achive the same cost as before, therefore $T(\mathcal{G}) = 2$ as well in concordance with the conjecture that $M(\mathcal{G}) = T(\mathcal{G})$.

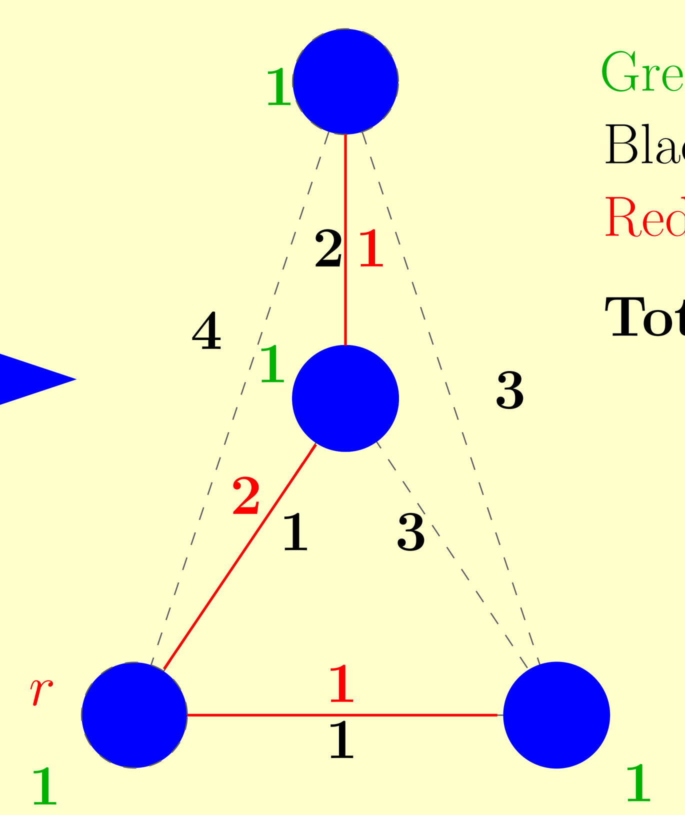
Splitting a Node

Let $\mathcal{G}_{v/k}$ denote the problem that we get from $\mathcal{G}$ by replacing node v with a path $(v_1, \ldots, v_k)$ where the neighbors of v become the neighbors of v_1, $b_{v/k} = b$ on the old vertices and $b_{v/k}(v_i) = b(v)/k$ on the new ones, $c_{v/k} = c$ on the old edges and for the new edges $c_{v/k}(v_i v_{i+1}) = 0$ (see figures).

- $T(\mathcal{G}_{v/k}) = T(\mathcal{G})$.
- $M(\mathcal{G}_{v/k}) \leq M(\mathcal{G})$: paths from v_i = paths from v, other paths and bandwidths remain the same.
- $M(\mathcal{G}_{v/k}) \geq M(\mathcal{G})$: paths from $v = \frac{1}{k}$ times the sum of the paths from v_is, other paths and bandwidths remain the same.
- $M(\mathcal{G}_{v/k}) = M(\mathcal{G})$.
- In MultiPath-version if two vertices are linked with a zero cost edge, then in optimal solution they can have the same routings.
- In Tree-version and MultiPath-version $b(v) = 0$ or $b(v) = 1$ for all v can be supposed.
- $\mathcal{G}_k$:= repeat the above procedure for each vertex of G.
- Using $\mathcal{G}_k$, each vertex has k copies with $b_k(v_i) = \frac{1}{k}$.



An Optimal Tree Solution



Green: $b(v)$
Black: $c(e)$
Red: $w(e)$ if not 0

Total Cost = 5

The Couple-versions

Here we suppose $b(v) = 0$ or $b(v) = 1$ for all v.

- Proposition: $T_2(\mathcal{G}_k) = T(\mathcal{G}_k) = T(\mathcal{G})$.
- Example: $M_2(\mathcal{G}) < M(\mathcal{G})$ for graph on top of page.
- Unknown: $M_2(\mathcal{G}_2) \stackrel{?}{=} M(\mathcal{G}_2) = M(\mathcal{G})$.
- Theorem: $M_2(\mathcal{G}_k) \to M(\mathcal{G})$ if $k \to \infty$.
- Corollary: $M_2(\mathcal{G}_k) - T_2(\mathcal{G}_k) \to 0$ if $k \to \infty$ would imply the stronger conjecture.
- Unknown: $S_2(\mathcal{G}) \stackrel{?}{=} S(\mathcal{G})$.