This is closely analogous to the previous problem, with input and output wires playing the roles of ships and ports.

A *switching* consists precisely of a perfect matching between input wires and output wires — we simply need to choose which input stream will be switched onto which output wire.

From the point of view of an input wire, it wants its data stream to be switched as early (close to the source) as possible: this minimizes the risk of running into another data stream, that has already been switched, at a junction box. From the point of view of an output wire, it wants a data stream to be switched onto it as late (far from the source) as possible: this minimizes the risk of running into another data stream, that has not yet been switched, at a junction box.

Motivated by this intuition, we set up a stable marriage problem involving the input wires and output wires. Each input wire ranks the output wires in the order it encounters them from source to terminus; each output wire ranks the input wires in the reverse of the order it encounters them from source to terminus. Now we show:

(1) *A stable matching between input and output wires defines a valid switching.*

Proof. To prove this, suppose that this switching causes two data streams to cross at a junction box. Suppose that the junction box is at the meeting of Input i and Output j. Then one stream must be the one that originates on Input i; the other stream must have switched from a different input wire, say Input k, onto Output j. But in this case, Output j prefers Input i to Input k (since j meets i downstream from k); and Input i prefers Output j to the output wire, say Output ℓ, onto which it is actually switched — since it meets Output j upstream from Output ℓ. This contradicts the assumption that we chose a stable matching between input wires and output wires. ∎

Assuming the meetings of inputs and outputs are represented by lists containing the orders in which each wire meets the other wires, we can set up the preference lists for the stable matching problem in time $O(n^2)$. Computing the stable matching then takes an additional $O(n^2)$ time.

---

[1]ex852.589.348