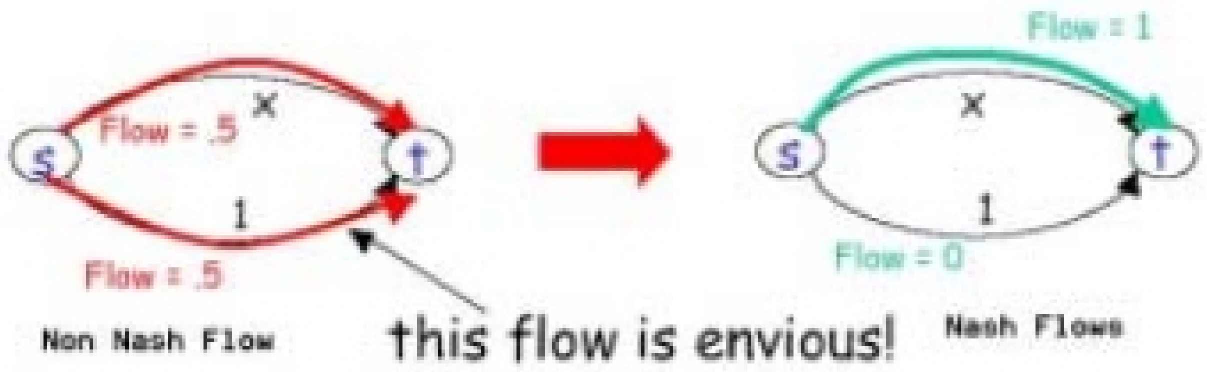
# CPSC690 Project Report
# Selfish Routing Effect on Network Traffic

Zheng MA
Second year PhD student of Yale University

### 3.1 Selfish Routing

**Assumptions:**

- traffic unit are small relative to network scale (like cars in high way system);
- we consider the traffic distribution in the network at steady state.
- A flow is at *Nash Equilibrium* (or is a *Nash Flow*) if **all** the flows in network. are routed on min-latency paths (which means any flow changing path will result in an increase of latency). For example:




- Nash flow(formal specification): for any two paths $p_1$ and $p_2 \in R_i$, (where $R_i$ is the set of potential paths from $s_i$ to $t_i$). We require that $\sum_{p\in R_i} f_p = r_i$, if $f_p > 0$ then $l_{p_1}(f) \leq l_{p_2}(f)$. This means either an path has traffic 0 for this traffic or it has nonzero traffic and has smaller latency than any other paths with zero traffic. And all paths which has traffic will have equal latency, define this latency as $L_i(f)$ ($f$ is flow vector, when we write $f$ we always mean the flow vector in the future) for OD pair $s_i$ to $t_i$.
- So here we can define the *Cost* $C(f)$ of a flow in $G$ as the total latency incurred by $f$, which is $C(f) = \sum_{p\in P} l_p(f) f_p$. By summing over the edges in a path $P$ and reversing the order of summation, we may also write $C(f) = \sum_{e\in E} l_e(f_e) f_e$

According to the result of Beckman et al [2], let $h_e(x) = \int_0^x l_e(t)dt$, the selfish routing behavior can be formalized as the following convex optimization problem:

$$Min \sum_{e\in E} h_e(f_e) \tag{1}$$

subject to:

$$\sum_{p\in R_i} f_p = r_i, \forall i \in \{1,\ldots,k\} \tag{2}$$

$$f_e = \sum_{p\in P: e\in p} f_p, \forall e \in E \tag{3}$$

$$f_p \geq 0, \forall p \in P \tag{4}$$

Because $l_e(x)$ is nondecreasing, the $h_e(x)$ is a convex function. We can use some result of convex programming to solve the above optimization problem.

### 3.2 System Optimal Routing

The system optimal routing should minimizes total latency in the network (not like the selfish user – each OD pair, they only optimize their own latency). Recall the expression of $C(f)$, we can easily see, the system optimal result should be the solution to the following non-linear program:

$$Min \sum_{e \in E} c_e(f_e),\ here\ c_e(f_e) = l_e(f_e) f_e \quad (5)$$

subject to:

$$\sum_{p \in R_i} f_p = r_i, \forall i \in \{1, \ldots, k\} \quad (6)$$

$$f_e = \sum_{p \in P: e \in p} f_p, \forall e \in E \quad (7)$$

$$f_p \geq 0, \forall p \in P \quad (8)$$

## 4 Flow assignment solver

Given above formulation, I implemented an flow assignment solver for system optimal solution and selfish user nash equilibrium solution. This is a driver for a matlab program