# Traffic Intersection Control System

## Reference Diagram

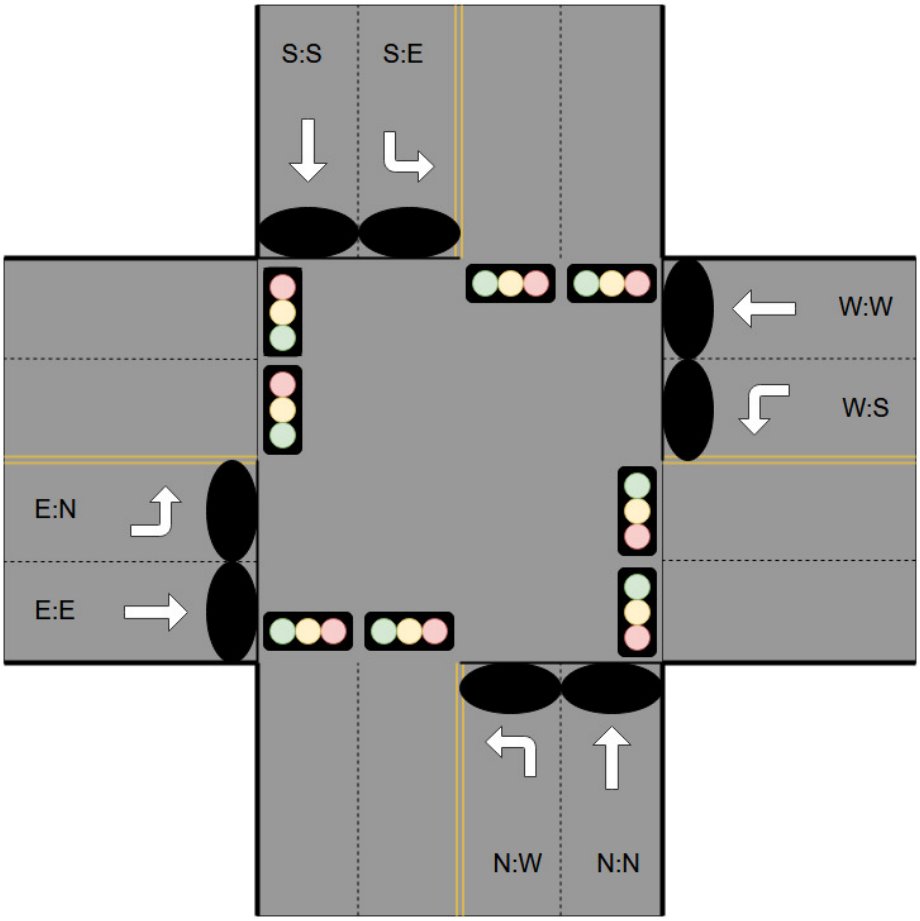


Figure 1: Intersection Diagram

## Specification

Your task is to design the control system for a 4-way traffic intersection with the following configuration.

- Four roads (North, South, East, and West) intersecting at right angles
- Each road has two lanes: a through lane and a left turn lane
- Each lane has a dedicated traffic light and vehicle sensor

The intersection should follow this cycle order.

1. N and S turning traffic
2. N and S through traffic
3. E and W turning traffic
4. E and W through traffic

The intersection is characterized by min and max active times for each traffic pattern, but take note these time limits are only enforced when there are vehicles queued in opposing lanes.

| Traffic Pattern | Min Active | Max Active |
|---|---|---|
| N and S turning | 10s | 60s |
| N and S through | 30s | 120s |
| E and W turning | 10s | 30s |
| E and W through | 30s | 60s |

As a general rule, the controller should optimize for throughput over wait-time, but there should be upper bound on the wait time for any car traveling through the intersection.

As an optimization, you may disregard yellow as a traffic light state, ie. transition directly between red and green.

## Implementation Scope

The controller's responsibility is to monitor traffic conditions for the intersection, and set the traffic signals for each lane accordingly. Your task is to design and implement the traffic control algorithm to the criteria above. You may mock any dependencies your algorithm needs, for example reading sensors or updating traffic signals.

We do not expect you to build a fully validated system, but we will ask you to prove that your design addresses the contraints of this specification. We will be satisfied if you can walk us through simple test cases and present a strategy for validating the design.

## Build and Run Environment

Use whatever build and run environment you are most comfortable with, but we like to see solutions in C, C++, or Python. For embedded developers, you should stick to using C or C++. Keep in mind that if we can't compile or run your code, we'll ask to see you do it.