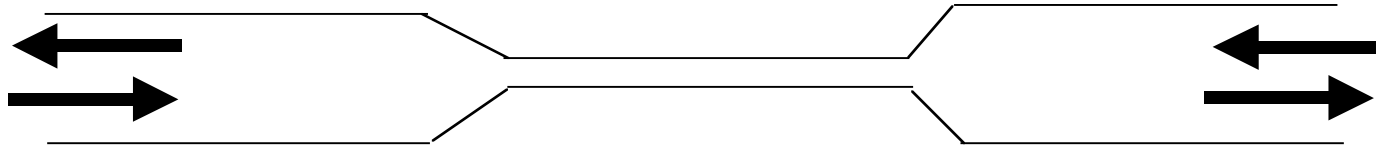
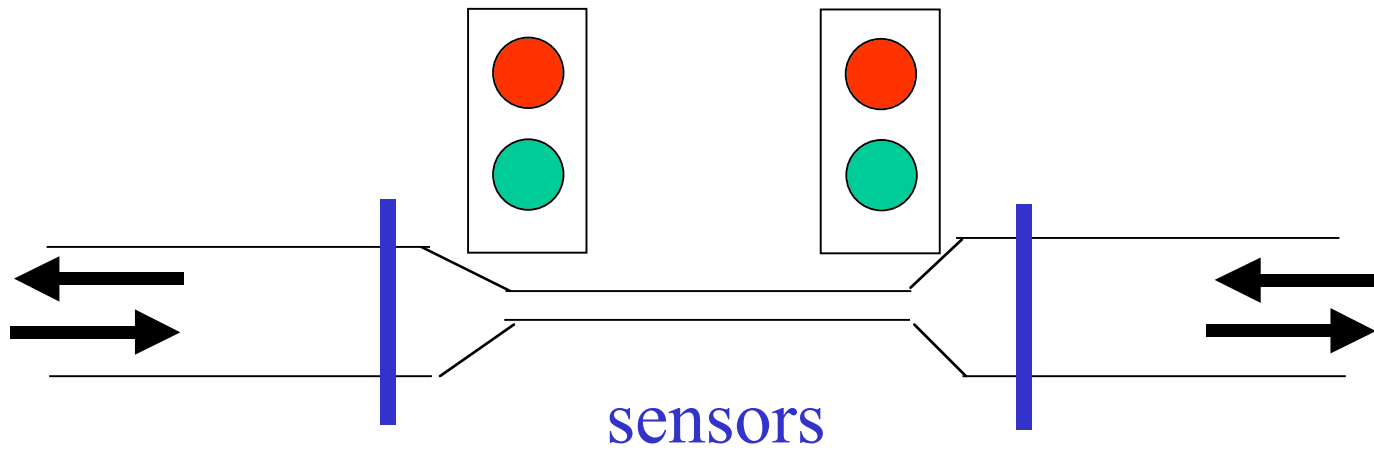


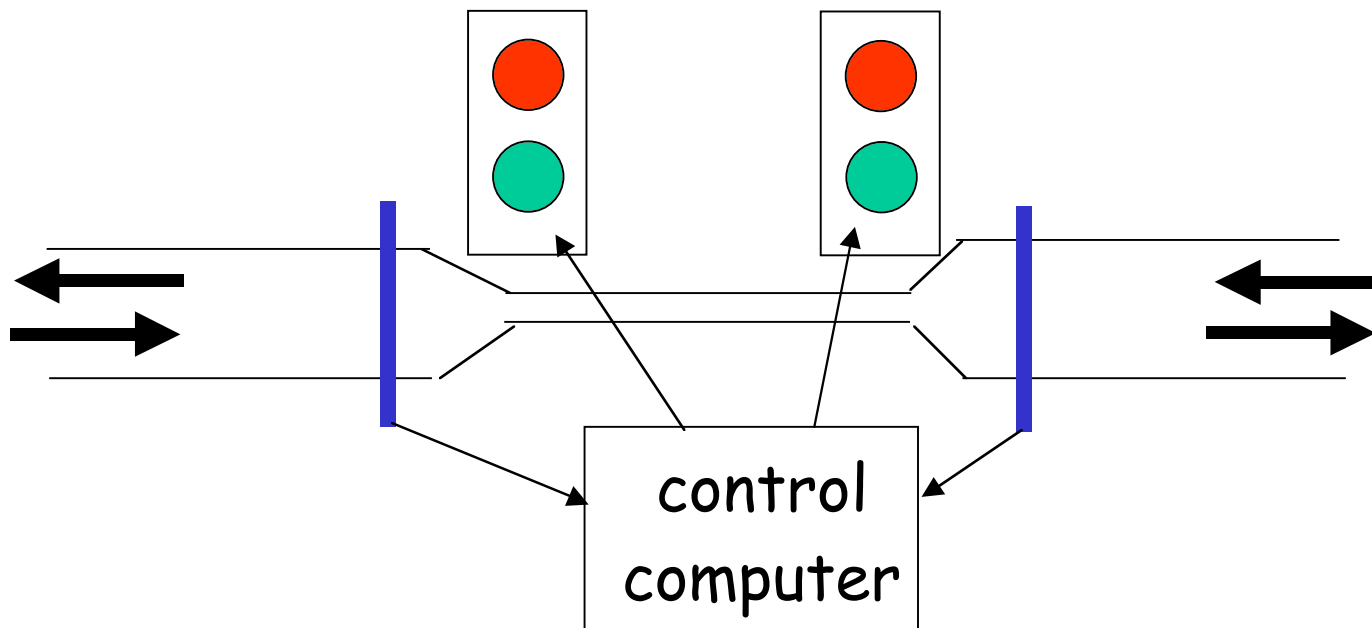
Automatic Traffic Lights for a one-lane bridge ...



Install lights, sensors, and controller so that crashes cannot occur on bridge between cars going in opposite directions.



More Advanced Problem ...



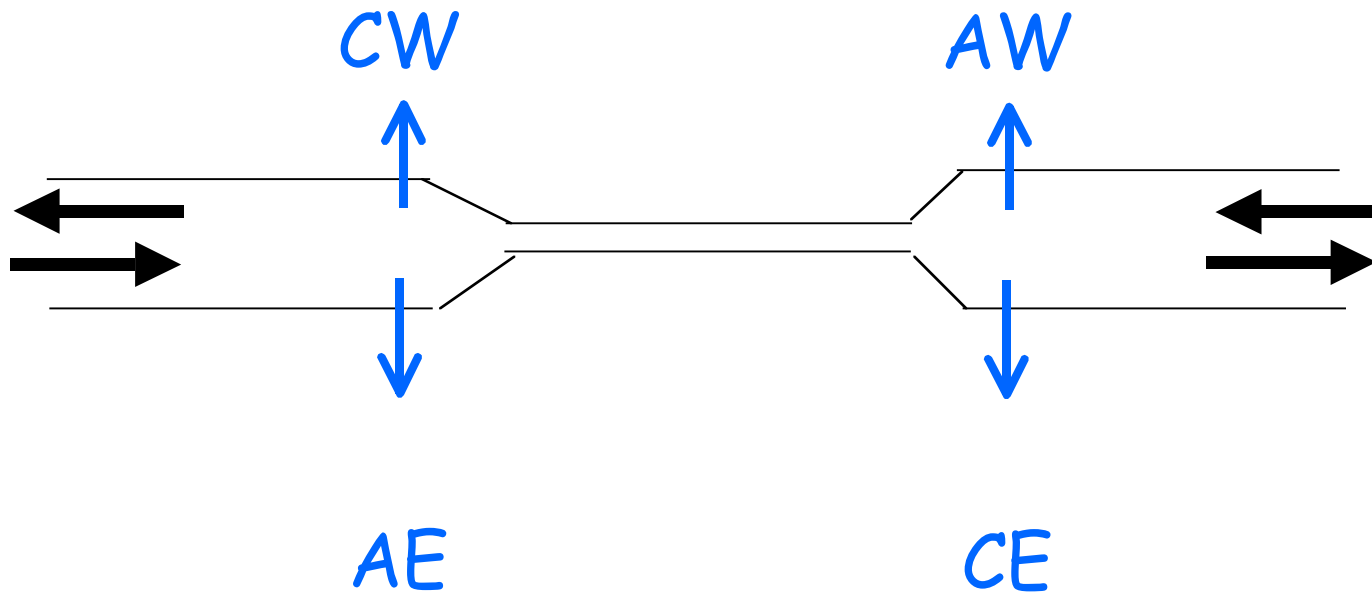
Program control computer to avoid collisions
on the bridge, and "starvation" too.

Problem ...

Find a control program that makes bridge crossings safe and alive.

Safe means cars can cross only in one direction at a time.

Alive means the light will never be red in both directions if cars wait.



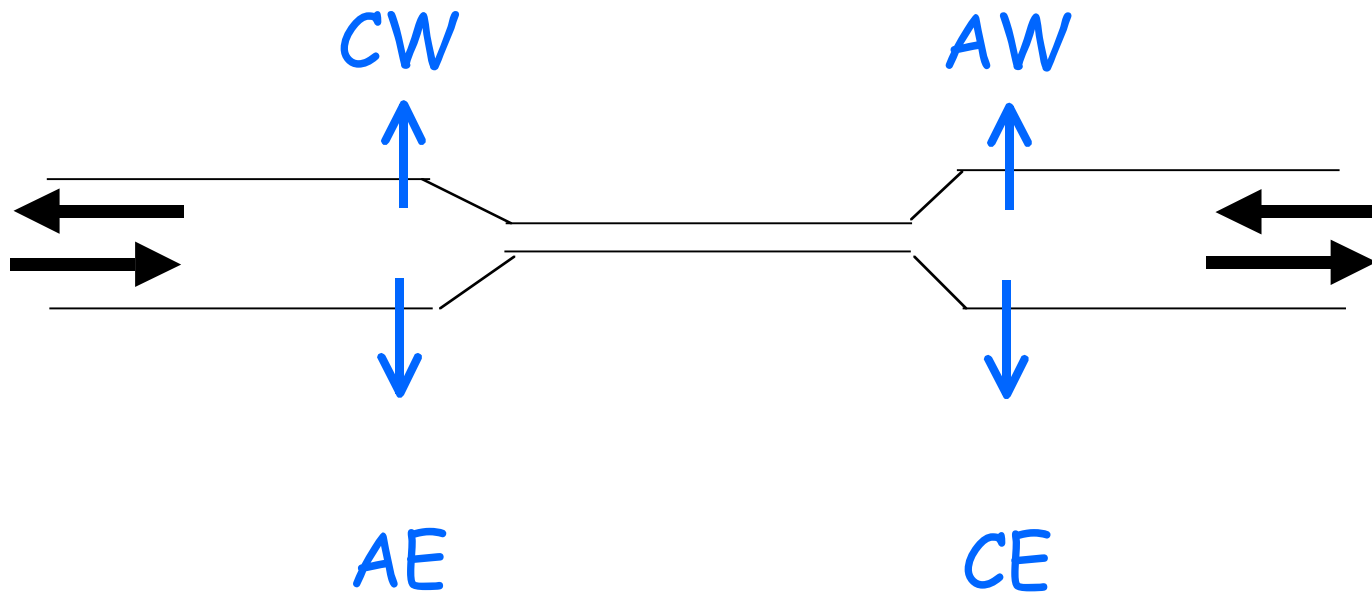
SENSOR EVENTS:

AE = arrival, eastbound

CE = completion, eastbound

AW = arrival, westbound

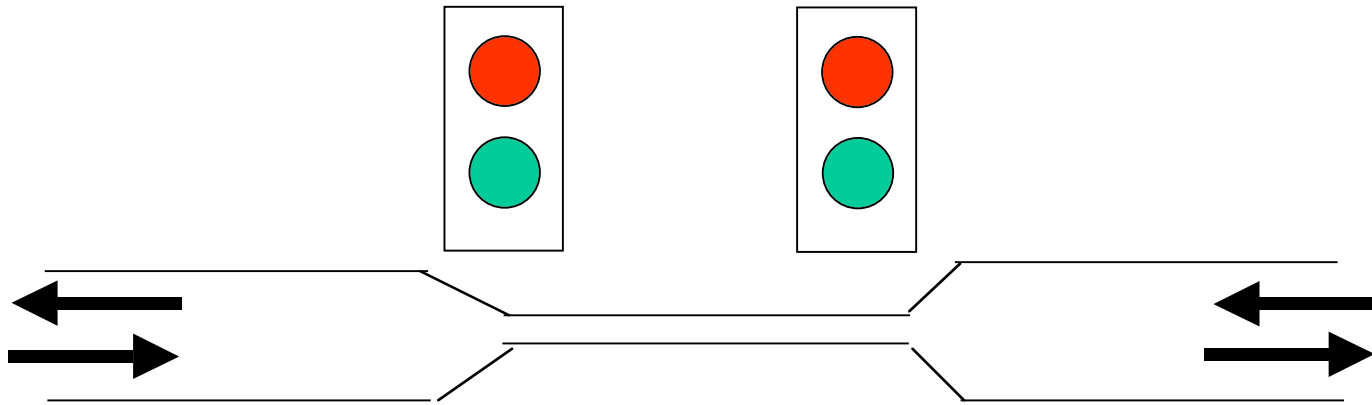
CW = completion, westbound



VARIABLES:

nw = no. westbound cars between AW & CW

ne = no. eastbound cars between AE & CE



$(G,R) \longleftrightarrow (R,R) \longleftrightarrow (R,G)$

states: **E**

"allow eastbound"

I

"all stop"

W

"allow westbound"

condition variables:

WL = "light on west end authorizes eastbound"

EL = "light on east end authorizes westbound"

setstate(state)

set the lights corresponding to "state" (E,I,W)

perform WL.signal if state=E

perform EL.signal if state=W

triggers

when a sensor event occurs, an interrupt procedure calls the corresponding event handler: AE, CE, AW, CW.


```
AW: nw=nw+1;
    if state=I then setstate(W)
CW: nw=nw-1;
    if nw=0 then
        if ne>0 then setstate(E)
        else setstate(I)
```

```
AE: ne=ne+1;
    if state=I then setstate(E)
CE: ne=ne-1;
    if ne=0 then
        if nw>0 then setstate(W)
        else setstate(I)
```

car_enter:

if eastbound then

{if WL is not green then WL.wait, trigger AE}

else if westbound then

{if EL is not green then EL.wait, trigger AW}

proceed on to bridge

car_exit:

if eastbound then trigger CE else trigger CW

Monitor:

Control program expressed as monitor with entries AE, CE, AW, CW, car_enter, and car_exit.

Environment:

External routine generates N threads, each representing a car that arrives at random and takes a fixed time to cross the bridge (e.g., 30 seconds, representing the speed of the car and length of the bridge). Thread can recycle after a random delay and present another car to the bridge.