East Car In

BridgeSegment3

BridgeSegment2

Bridgesegment 1

Eastbound q2 q2 q2

West car In q1 q1 q1 WestBound

East car Generator and West Car Generator are two car Generators .

There are two queues q1 and q2.

All cars from West are queued up in q1 and all cars from East are queued up in q2.

When Signal is Green for East to West : And if respective queues are not empty. An car is removed from queue and processed.(i.e Travel through Segment) And State of segment is handled as per signals time.

BridgSegment States/Phase:

“West\_to\_East “ - When Green Signal is for west to east direction and there is no car on bridge segment.

“West\_to\_East “ - When Green Signal is for west to east direction and there is no car on bridge segment.

“WtoE “ - When Green Signal is for west to east direction and there is a car on bridge segment.

“EtoW “ - When Green Signal is for East to West direction and there is a car on bridge segment.

**public** **void** deltext(**double** e,message x)

{

Continue(e);

// SignalRemainingTime = SignalRemainingTime - e;

**for** (**int** i=0; i< x.getLength();i++)

{

**if** (messageOnPort(x, "EastcarIn", i) )

{

**if** (phaseIs("East\_To\_West") || phaseIs("EtoW"))

{

SignalRemainingTime = SignalRemainingTime - e;

**if**(q2.isEmpty() & currentcar==**null** )

{

**if**(SignalRemainingTime >= carTravelTime)

{

SignalRemainingTime = USignaltime-carTravelTime - e ;

// USignaltime = SignalRemainingTime;

car = x.getValOnPort("EastcarIn", i);

currentcar = car;

holdIn("EtoW", carTravelTime);

//q2.remove();

}

}

**else**

{

car = x.getValOnPort("EastcarIn", i);

q2.add(car);

}

}

**else** **if** (!phaseIs("East\_To\_West") || !q2.isEmpty())

{

//SignalRemainingTime = SignalRemainingTime - carTravelTime;

car = x.getValOnPort("EastcarIn", i);

q2.add(car);

// holdIn("EtoW", carTravelTime);

}

}

}

// ----------------------------West Car Handled below ----------------------//

**for** (**int** i=0; i< x.getLength();i++)

{

**if** (messageOnPort(x, "WestcarIn", i) )

{

**if** (phaseIs("West\_To\_East") || phaseIs("WtoE"))

{

**if**(phaseIs("West\_To\_East") && q1.isEmpty())

{

**if**(SignalRemainingTime >= carTravelTime)

{

SignalRemainingTime = SignalRemainingTime-carTravelTime ;

car = x.getValOnPort("WestcarIn", i);

currentcar = car;

holdIn("WtoE", carTravelTime);

//q2.remove();

}

**else**

{

car = x.getValOnPort("WestcarIn", i);

q1.add(car);

//sigma=400;

}

}

}

**else** **if** (!phaseIs("West\_To\_East") || !q1.isEmpty())

{

//SignalRemainingTime = SignalRemainingTime - carTravelTime;

car = x.getValOnPort("WestcarIn", i);

q1.add(car);

//holdIn("EtoWfaltu", carTravelTime);

}

}

}

}

**public** **void** deltint( )

{

//currentcar = null;

**if** (phaseIs("WtoE"))

{

**if**((SignalRemainingTime >= carTravelTime) & (!q1.isEmpty()))

{

q1.remove();

SignalRemainingTime = SignalRemainingTime - carTravelTime;

car = (entity)q1.first();

currentcar = car;

holdIn("WtoE", carTravelTime);

}

**else** **if** (SignalRemainingTime < carTravelTime)

{

holdIn("West\_To\_East", SignalRemainingTime);

}

**else** **if**(SignalRemainingTime <= 0)

{

SignalRemainingTime = trafficlightdurationtime;

holdIn("East\_To\_West", SignalRemainingTime);

}

**else** **if** ((SignalRemainingTime >= carTravelTime) & (q1.isEmpty()))

holdIn("WestToEast", SignalRemainingTime);

}

**else** **if** (phaseIs("EtoW"))

{

**if**((SignalRemainingTime >= carTravelTime) & (!q2.isEmpty()) & currentcar==**null**)

{

q2.remove();

SignalRemainingTime = SignalRemainingTime - carTravelTime;

car = (entity)q2.first();

currentcar = car;

holdIn("EtoW", carTravelTime);

}

**else** **if** ((SignalRemainingTime >= carTravelTime) & (q2.isEmpty()) & currentcar==**null**)

holdIn("East\_To\_West", SignalRemainingTime);

**else** **if** (SignalRemainingTime < carTravelTime)

{

holdIn("East\_To\_West", SignalRemainingTime);

}

**else** **if** (currentcar!=**null**)

{

holdIn("East\_to\_West", SignalRemainingTime);

}

**else** **if**(SignalRemainingTime <= 0)

{

SignalRemainingTime = trafficlightdurationtime;

holdIn("West\_To\_East", SignalRemainingTime);

}

}

**else** **if**(phaseIs("East\_To\_West"))

{

**if**((SignalRemainingTime >= carTravelTime) & (!q2.isEmpty()))

{

q2.remove();

SignalRemainingTime = SignalRemainingTime - carTravelTime;

car = (entity)q2.first();

currentcar = car;

holdIn("EtoW", carTravelTime);

}

**else** **if** (SignalRemainingTime < carTravelTime)

{

holdIn("East\_To\_West", SignalRemainingTime);

}

**else** **if**(SignalRemainingTime == 0)

{

SignalRemainingTime = trafficlightdurationtime;

holdIn("West\_To\_East", SignalRemainingTime);

}

**else**

holdIn("East\_To\_West", SignalRemainingTime);

}

**else** **if**(phaseIs("West\_To\_East"))

{

**if**((SignalRemainingTime >= carTravelTime) & (!q1.isEmpty()))

{

q1.remove();

SignalRemainingTime = SignalRemainingTime - carTravelTime;

car = (entity)q1.first();

currentcar=car;

holdIn("WtoE", carTravelTime);

}

**else** **if** (SignalRemainingTime < carTravelTime)

{

holdIn("West\_To\_East", SignalRemainingTime);

}

**else** **if**(SignalRemainingTime <= 0)

{

SignalRemainingTime = trafficlightdurationtime;

holdIn("East\_To\_West", SignalRemainingTime);

}

}

**if** (SignalRemainingTime < 0)

{

SignalRemainingTime = trafficlightdurationtime;

**if** (phaseIs("West\_To\_East"))

{

holdIn("East\_To\_West", trafficlightdurationtime);

}

**else** **if** (phaseIs("East\_To\_West"))

{

holdIn("West\_To\_East", trafficlightdurationtime);

}

}

}