

```

In[899]:= Clear[Si, s, LT, avgManualToll, avgElectronicTollRate]
          Si = 3; (*Input Lanes*)
          s = 9; (*Booths*)
          LT = 3; (*total # of lanes in a partition*)
          (*Solve[s==(Si)*(LT),(*input variable to solve for*)]*)
          avgManualToll = 1 /  $\left(\frac{350}{60^2}\right)$  (*rate of human collected tolls*);
          avgElectronicTollRate = 1 /  $\left(\frac{1200}{60^2}\right)$  (*rate of electronically collected tolls*);

          Clear[ETCRate]
          ETCRate[ETC_] := 1 / avgElectronicTollRate
          Clear[HRate]
          HRate[H_] := 1 / avgManualToll

In[909]:=  $\lambda$  = ErlangDistribution[5.6 / 30, 10 / 30];

In[910]:= Clear[ModelFunctionElectronic]
          (*Returns mean system time for electronic systems*)
          ModelFunctionElectronic[ $\lambda$ _, ETC_, H_] :=
            QueueProperties[
              QueueingProcess[
                 $\lambda$  / (ETC + H),
                ETCRate[ETC],
                ETC
              ],
              "MeanSystemTime"
            ]

In[912]:= Clear[ModelFunctionElectronicQueueTable]
          (*Returns mean system time for manual systems*)
          ModelFunctionElectronicQueueTable[ $\lambda$ _, ETC_, H_] :=
            QueueProperties[
              QueueingProcess[
                 $\lambda$  / (ETC + H),
                ETCRate[ETC],
                ETC], "ServiceRate"]

```

```

In[914]:= Clear[ModelFunctionManual]
(*Finds MTC Mean System Time*)
ModelFunctionManual[λ_, ETC_, H_] :=
  QueueProperties[
    QueueingProcess[
      λ / (ETC + H),
      HRate[H],
      H],
    "MeanSystemTime"]

In[916]:= Clear[ModelFunctionManualQueueTable]
(*Finds MTC Service Rate*)
ModelFunctionManualQueueTable[λ_, ETC_, H_] :=
  QueueProperties[
    QueueingProcess[
      λ / (ETC + H),
      HRate[H],
      H],
    "ServiceRate"]

In[918]:= Clear[currentSystemFunction]
(*Finds the Mean System Time*)
currentSystemFunction[λ_, ETC_, H_] :=
{
  Table[
    QueueProperties[
      QueueingProcess[(λ) / (ETC + H), ETCRate[ETC]
    ], "MeanSystemTime"],
    {i, 1, ETC}
  ], Table[
    QueueProperties[
      QueueingProcess[(λ) / (ETC + H), HRate[H]
    ], "MeanSystemTime"], {i, 1, H}
  ]
}

In[920]:= Clear[avgFunction]
avgFunction[λ_, ETC_, H_] := 
$$\frac{\text{Total}[\text{Flatten}[\text{currentSystemFunction}[\text{ETC}, \text{H}, \lambda]]]}{\text{ETC} + \text{H}} // \text{N}$$


```

■ Functions

```

In[922]:= Clear[EdgeFunction]
EdgeFunction[Si_, s_] := Table[DirectedEdge["InitialVertex", pi], {i, 1,  $\frac{s}{S_i}$ }]

In[924]:= Clear[PartitionEdgeFunction]
PartitionEdgeFunction[Si_, s_, LT_] := Flatten[Table[
  Map[DirectedEdge[pi, #] &, Partition[Map[B# &, Range[s]], {LT}][[i]]], {i, 1, Si}]

In[926]:= Clear[MergeLaneFunction]
MergeLaneFunction[Si_, s_, LT_] := Flatten[Table[
  Map[DirectedEdge[#, mi] &, Partition[Map[B# &, Range[s]], {LT}][[i]]], {i, 1, Si}]

In[928]:= Clear[EdgeFunction2]
EdgeFunction2[Si_, s_] := Table[DirectedEdge[mi, "EndVertex"], {i, 1,  $\frac{s}{S_i}$ }]

In[930]:= Clear[GraphFunction]
GraphFunction[Si_, s_, LT_, λ_, ETC_, H_] :=
  Graph[
    Union[
      EdgeFunction[Si, s],
      PartitionEdgeFunction[Si, s, LT],
      MergeLaneFunction[Si, s, LT],
      EdgeFunction2[Si, s]],
    GraphLayout → {"LayeredDigraphEmbedding", "Orientation" → Bottom}, VertexStyle →
    Flatten[
      {Table[
        PartitionEdgeFunction[
          Si, s, LT][[i, 2]] -> Red,
        {i, s, s - Si + 1, -1}
      ], Table[
        PartitionEdgeFunction[
          Si, s, LT][[i, 2]] -> Red,
        {i, 2, Ceiling[ $\frac{s}{2}$ ]}
      ]
    }
  ], EdgeWeight → Flatten[
    {
      (*Merge Lane Edge Weights*)
      Table[
        Partition[

```

```

MergeLaneFunction[
  Si, s, LT],
{3}
]
[[j, i]] →
i - 1
, {i, LT, 1, -1},
{j, 2, 3}
],
Table[
  Partition[
    MergeLaneFunction[Si, s, LT],
    {3}
  ]
  [[1, i]] →
  LT - i
  , {i, 1, LT}],

(*Queue Lane Edge Weights*)
Map[# → ModelFunctionManual[λ, ETC, H] &,
  Complement[
    PartitionEdgeFunction[Si, s, LT],
    Flatten[
      {Table[
        PartitionEdgeFunction[Si, s, LT][[i]], {i, s, s - Si + 1, -1}},
        Table[
          PartitionEdgeFunction[Si, s, LT][[i]], {i, 2, Ceiling[ $\frac{s}{2}$ ]}
        ]
      }
    ]
  ],
Table[
  PartitionEdgeFunction[Si, s, LT][[i]] →
  ModelFunctionElectronic[λ, ETC, H],
  {i, s, s - Si + 1, -1}
],
Table[
  PartitionEdgeFunction[Si, s, LT][[i]] →
  ModelFunctionElectronic[λ, ETC, H],
  {i, 2, Ceiling[ $\frac{s}{2}$ ]}
]

```

```

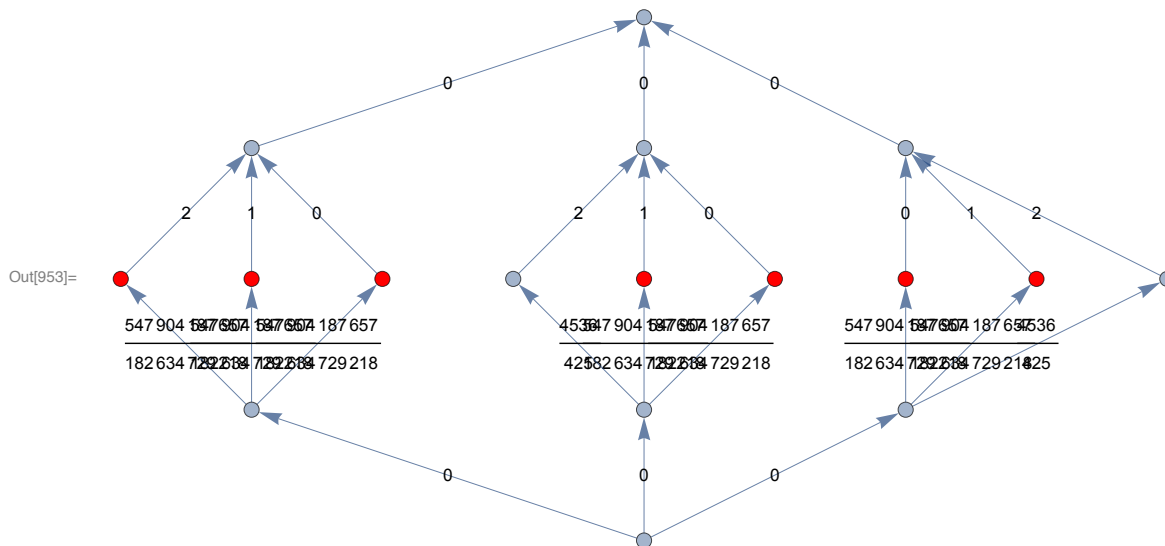
]

(*Initial and Ending edges*)
,
Map[
  # → 0 &,
  Union[EdgeFunction[Si, s],
    EdgeFunction2[Si, s]
  ]
]
}
],
EdgeLabels → "EdgeWeight",
VertexWeight → Flatten[
  {Table[mi → 0, {i, 1, Si}
  ]
}
],
ImageSize → Large
]

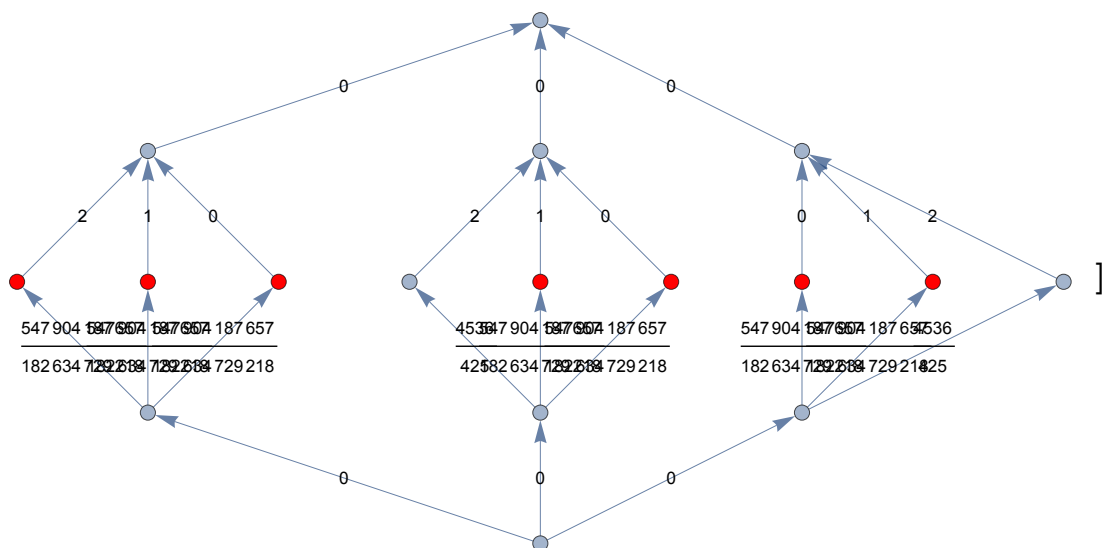
```

In[932]:= Clear[λ]

In[953]:= GraphFunction[Si, s, LT, 10/30, 7, 2]

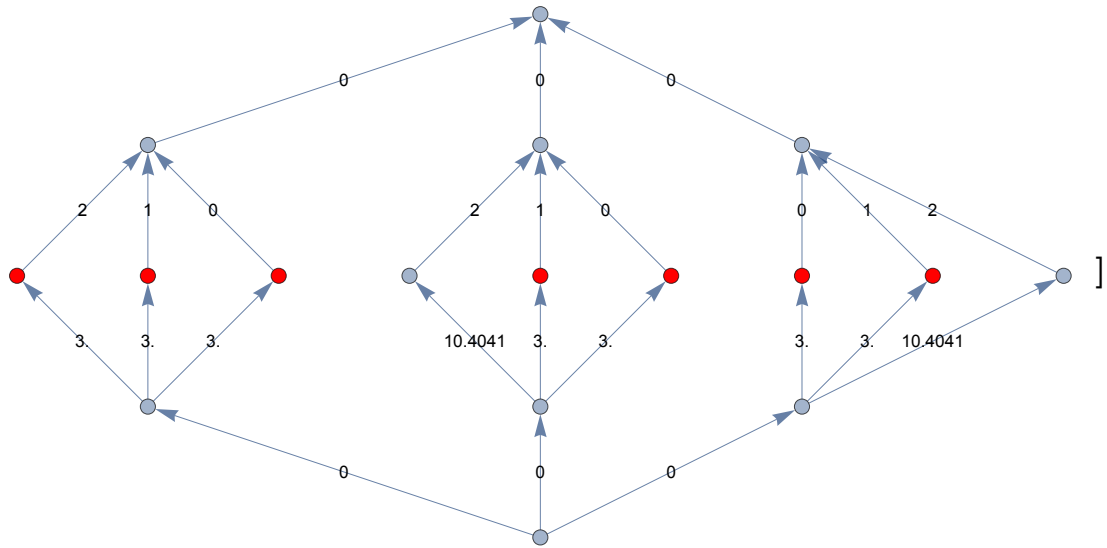


In[955]:= averageSystemTime[

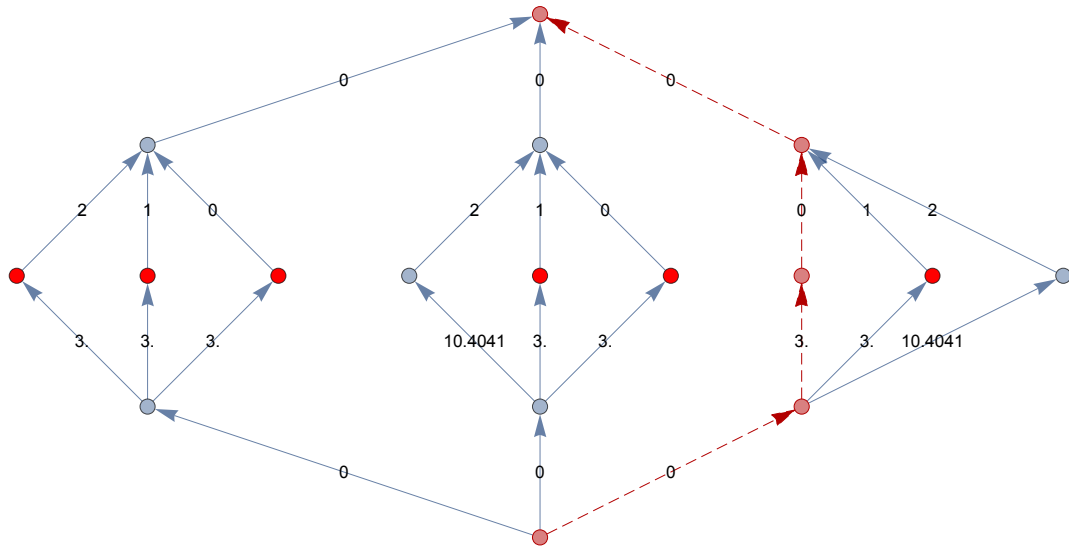


Out[955]=
$$\frac{442\,828\,340\,111\,569}{155\,239\,519\,835\,300}$$

In[935]:= GraphHighlighter[



Out[935]=



```
In[936]:= Clear[currentModelEdges]
currentModelEdges[servers_] :=
  Flatten[
    {
      Table[
        DirectedEdge[
          "InitialVertex", mi],
        {i, servers}
      ],
      Table[
        DirectedEdge[
          mi, "EndVertex"],
        {i, servers}
      ]
    }
  ]
```



```

In[938]:= Clear[currentModelGraph]
currentModelGraph[servers_, ETC_, H_, λ_] :=
  Graph[
    Flatten[
      {
        "InitialVertex",
        "EndVertex",
        Table[mi, {i, 1, servers}]
      }
    ],
    currentModelEdges[servers],
    ImageSize → Large,
    GraphLayout → {"LayeredDigraphEmbedding", "Orientation" → Bottom},
    VertexStyle → Table[mi → Red, {i, servers, servers - ETC + 1, -1}],
    EdgeWeight → Flatten[
      {Table[
        DirectedEdge[mi, "EndVertex"] →
          Ceiling[ $\frac{\text{servers}}{2}$ ] - i + 1,
        {i, 1, Floor[ $\frac{\text{servers}}{2}$ ]}
      ],
      Table[
        DirectedEdge[mi, "EndVertex"] →
          i - Floor[ $\frac{\text{servers}}{2}$ ],
        {i, servers, Ceiling[ $\frac{\text{servers}}{2}$ ], -1}
      ],
      Table[
        DirectedEdge["InitialVertex", mi] →
          ModelFunctionElectronic[λ, ETC, H],
        {i, servers, servers - ETC + 1, -1}
      ],
      Table[
        DirectedEdge["InitialVertex", mi] →
          ModelFunctionManual[λ, ETC, H],
        {i, 1, servers - ETC + 1}
      ]
    ]
  ],
  VertexLabels → "Name",
  EdgeLabels → "EdgeWeight", ImageSize → 1000
]

```

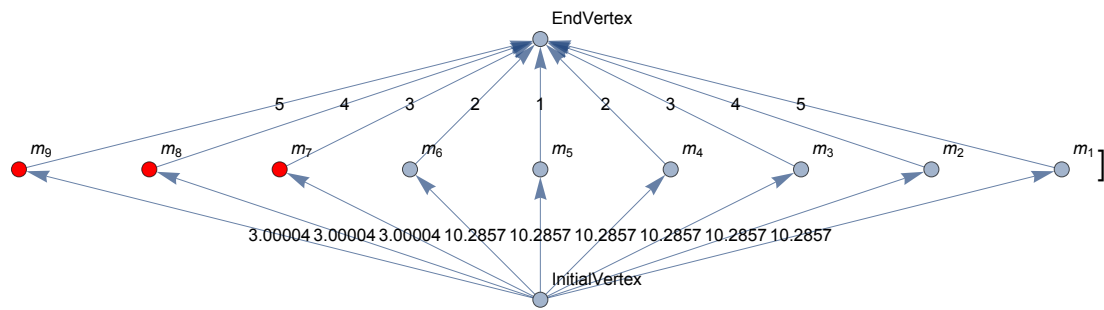
```
In[940]:= Clear[GraphHighlighter]
GraphHighlighter[graph_] :=
  HighlightGraph[
    graph,
    DirectedGraph[
      PathGraph[
        FindShortestPath[
          graph,
          "InitialVertex",
          "EndVertex"
        ]
      ]
    ],
    GraphHighlightStyle →
      {"Dashed"}
  ]
```

```

In[942]:= Clear[averageSystemTime]
averageSystemTime[graph_] :=
  Mean[
    Map[
      Total[#] &,
      Partition[
        DeleteCases[
          Flatten[
            Map[
              PropertyValue[
                {graph, ##},
                EdgeWeight] &,
            Map[
              EdgeList[
                DirectedGraph[
                  PathGraph[#]
                ]
              ] &,
            FindPath[
              graph,
              "InitialVertex",
              "EndVertex",
              Infinity,
              All
            ]
          ],
          {2}
        ]
      ],
      $Failed
    ],
    {2}
  ]
]
]

```

```
In[944]:= averageSystemTime[
```



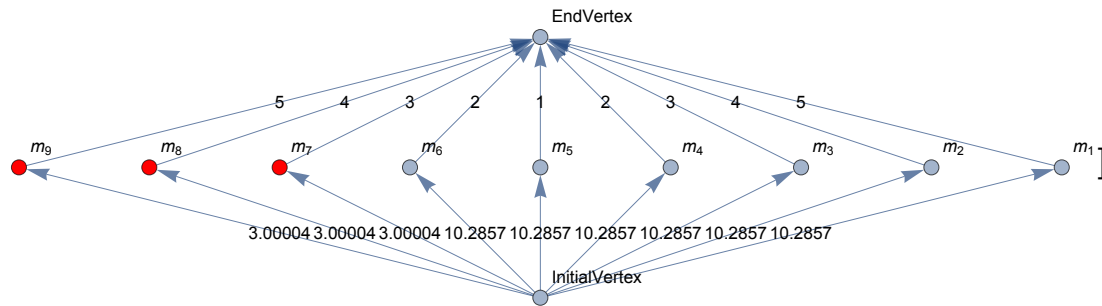
```
Out[944]= 11.0794
```

```

In[945]:= Clear[standardDeviationModel]
standardDeviationModel[graph_] :=
  StandardDeviation[
    Map[
      Total[#] &,
      Partition[
        DeleteCases[
          Flatten[
            Map[
              PropertyValue[
                {graph, ##},
                EdgeWeight] &,
            Map[
              EdgeList[
                DirectedGraph[
                  PathGraph[#]
                ]
              ] &,
            FindPath[
              graph,
              "InitialVertex",
              "EndVertex",
              Infinity,
              All
            ]
          ],
          {2}
        ]
      ],
      $Failed
    ]
  ], {2}]
]
]

```

```
In[947]:= standardDeviationModel[
```



```
Out[947]= 3.3113
```

```
In[948]:=
```

```
In[949]:= Clear[CostFunction]
```

```
CostFunction[t_] :=
```

```
135 000 (LP - LC) t - 65 000 (LP - LC) - (88 000) (LP - LC) /. {LP -> 7, LC -> 3}
```

```
Plot[CostFunction[t], {t, 0, 2}, AxesLabel -> {"t", "Dollar Amount Saved"},
```

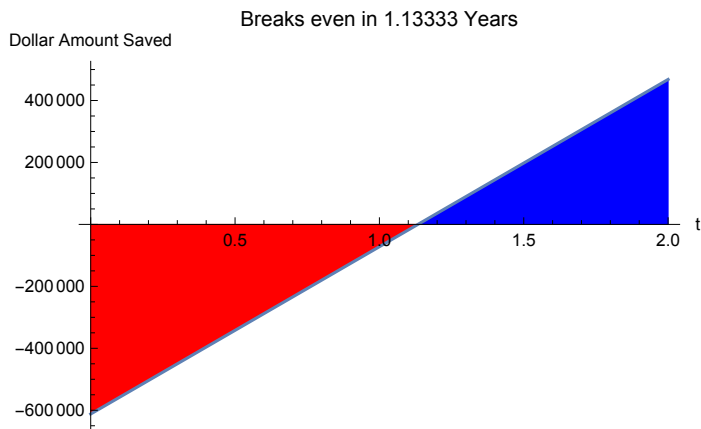
```
Filling -> Axis, FillingStyle -> {Red, Blue},
```

```
PlotLabel -> "Breaks even in ""1.13333"" Years"]
```

```
(*Print["Initial Cost = $",CostFunction[0]]*)
```

```
(*Print["Breaks even in ",
```

```
Tostring[Flatten[t/.Solve[CostFunction[t]==0,t]][[1]]//N], " Years"]*)
```



```
Out[951]=
```

```
In[952]:=
```