IEOR 4615: Service Engineering
Lecture 7, February 12, 2015

Precedence Constraints

and

Randomness

OUTLINE

1. Classical Deterministic Model: PERT/CPM
   - Program Evaluation and Review Technique (US Navy c.1950)
   - Critical Path Method
2. From PERT to Stochastic PERT
3. Dynamic Stochastic PERT (DS-PERT, use simulation)
4. Processing Networks
   - Arrest to Arraignment (Larson 1993)
   - Hospital Emergency Room
   - Group Play on a Golf Course (Whitt 2014)

Program Evaluation and Review Technique
PERT

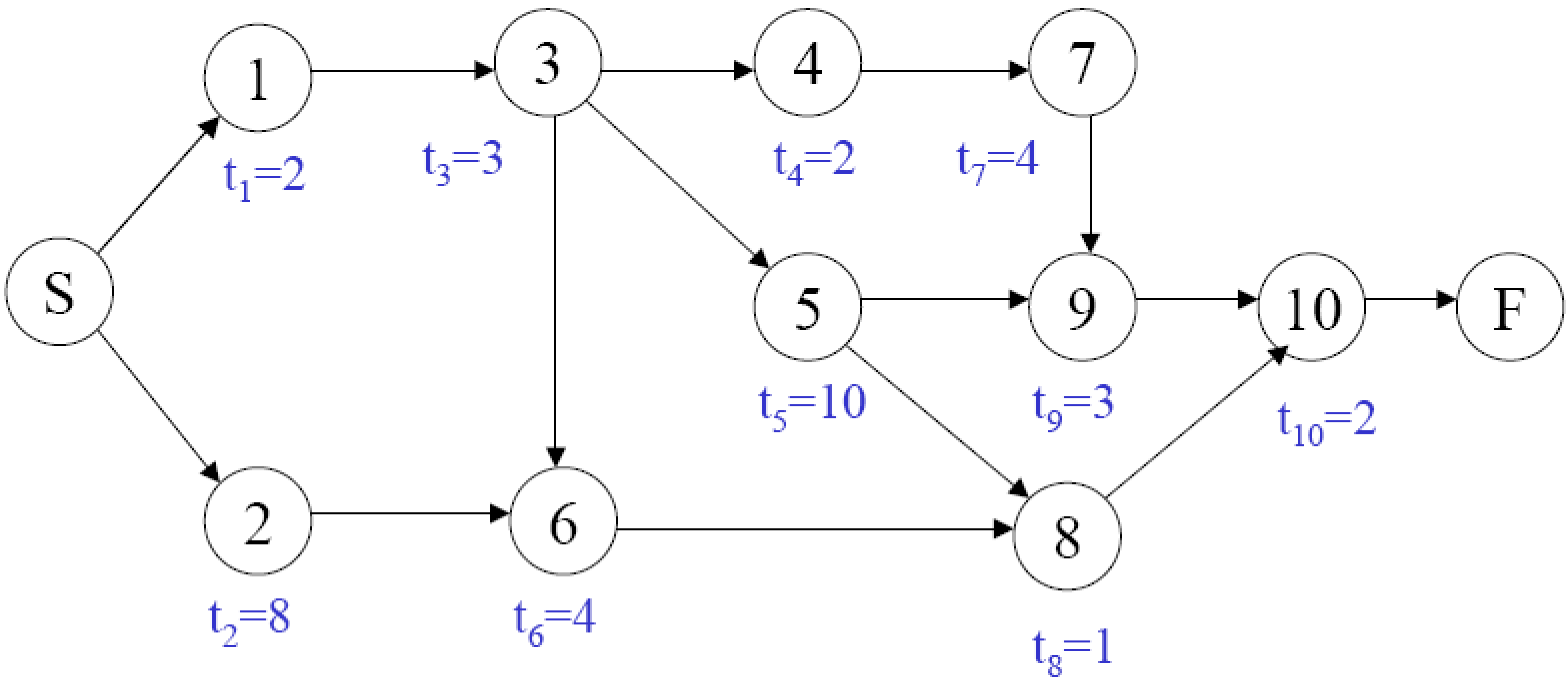
Critical Path Method
CPM

For Managing Projects

Tennis Tournament Activities (Fitzsimmons, pp 391–392)

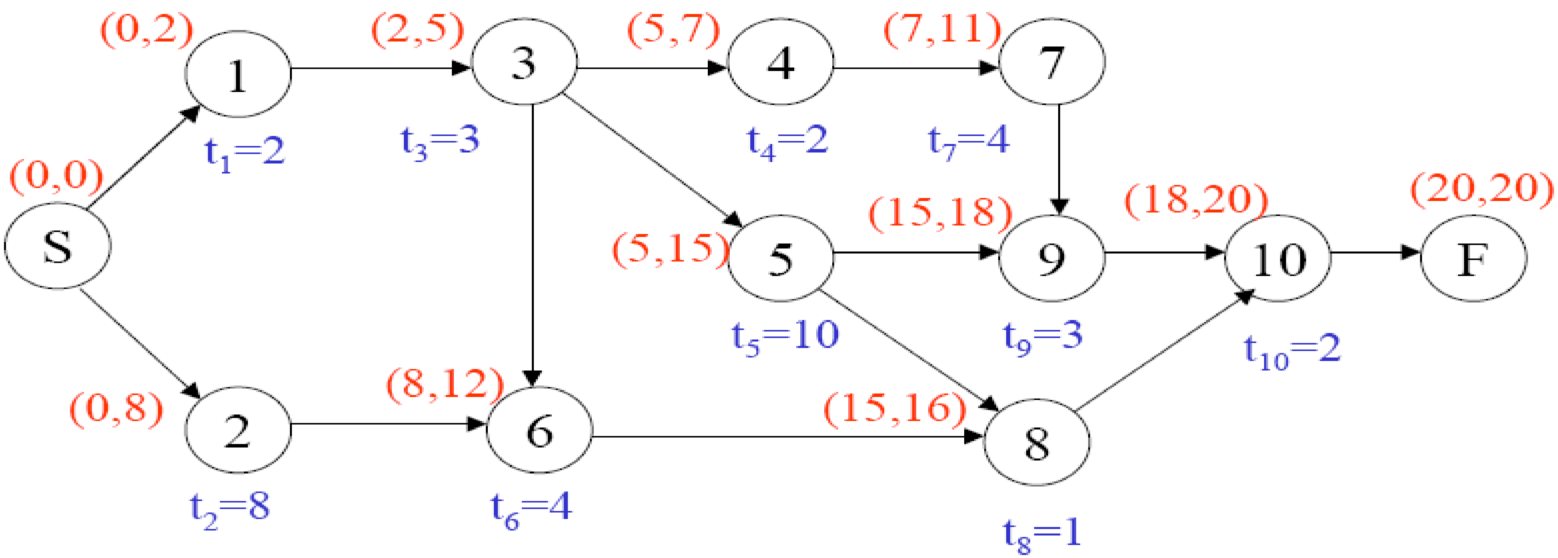
Task Description	Code	Immediate Predecessors
Negotiate for location	1	—
Contact seeded players	2	—
Plan promotion	3	1
Locate officials	4	3
Send invitations	5	3
Sign player contracts	6	2,3
Purchase balls and trophies	7	4
Negotiate catering	8	5,6
Prepare location	9	5,7
Tournament	10	8,9

PERT Chart



PERT = **P**rogram **E**valuation and **R**eview **T**echnique.

Critical Path Method: Forward Pass

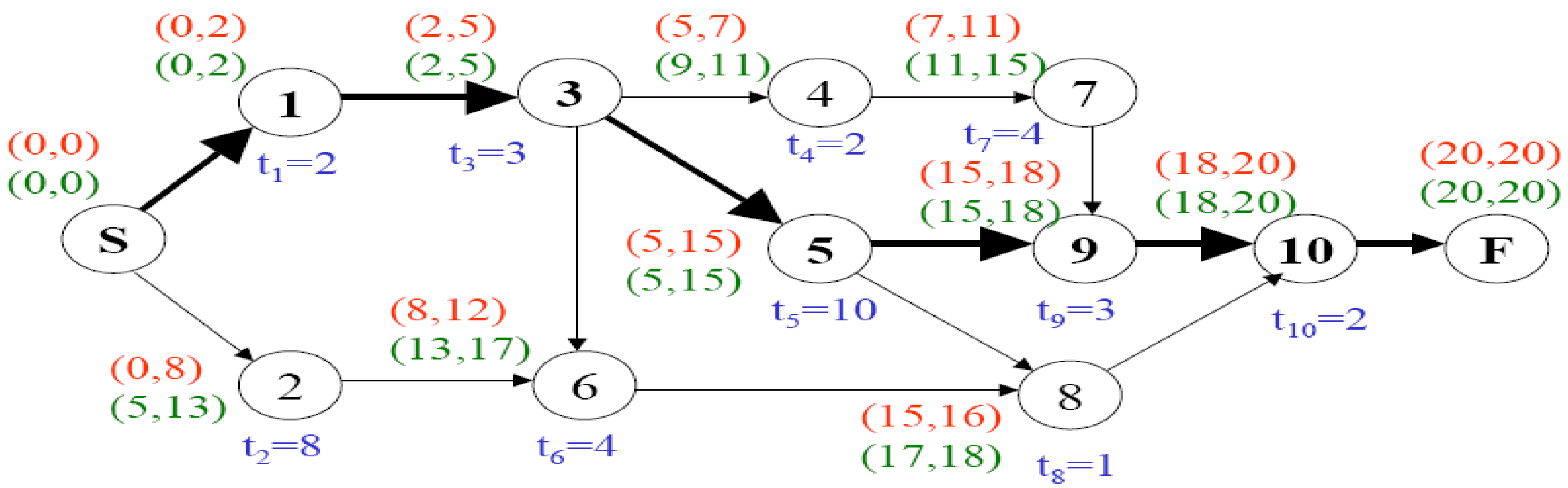


Initialization: $(\mathrm{ES})_i = (\mathrm{EF})_i = 0$ for Start node.

Early Start: $(\mathrm{ES})_i = \max\{\mathrm{EF}$ of all predecessors$\}$.

Early Finish: $(\mathrm{EF})_i = (\mathrm{ES})_i + t_i$.

Critical Path Method: Backward Pass



Initialization: $(\mathrm{LS})_i = (\mathrm{ES})_i$ for Finish node.

Late Finish: $(\mathrm{LF})_i = \min\{\mathrm{LS}$ of all successors$\}$.

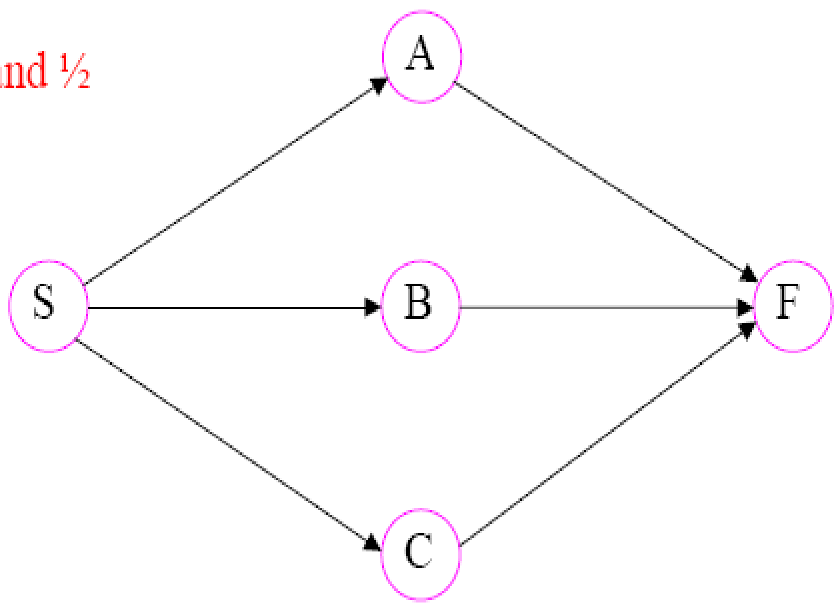
Late Start: $(\mathrm{LS})_i = (\mathrm{LF})_i - t_i$.

Critical Path(s): $(\mathrm{ES})_i = (\mathrm{LS})_i$ and $(\mathrm{EF})_i = (\mathrm{LF})_i$.

Slack: $(\mathrm{TS})_i = (\mathrm{LS})_i - (\mathrm{ES})_i = (\mathrm{LF})_i - (\mathrm{EF})_i$.

Static *Stochastic* PERT

	Activities	A	B	C
Avg.	Duration	3	4	5
	Distribution	3	4	1 or 9
				wp ½ and ½



Project time = max {3, **4**, 1} = 4 wp ½
max {3, 4, **9**} = 9 wp ½

Randomness Complicates Even Simple Projects

$$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow$$

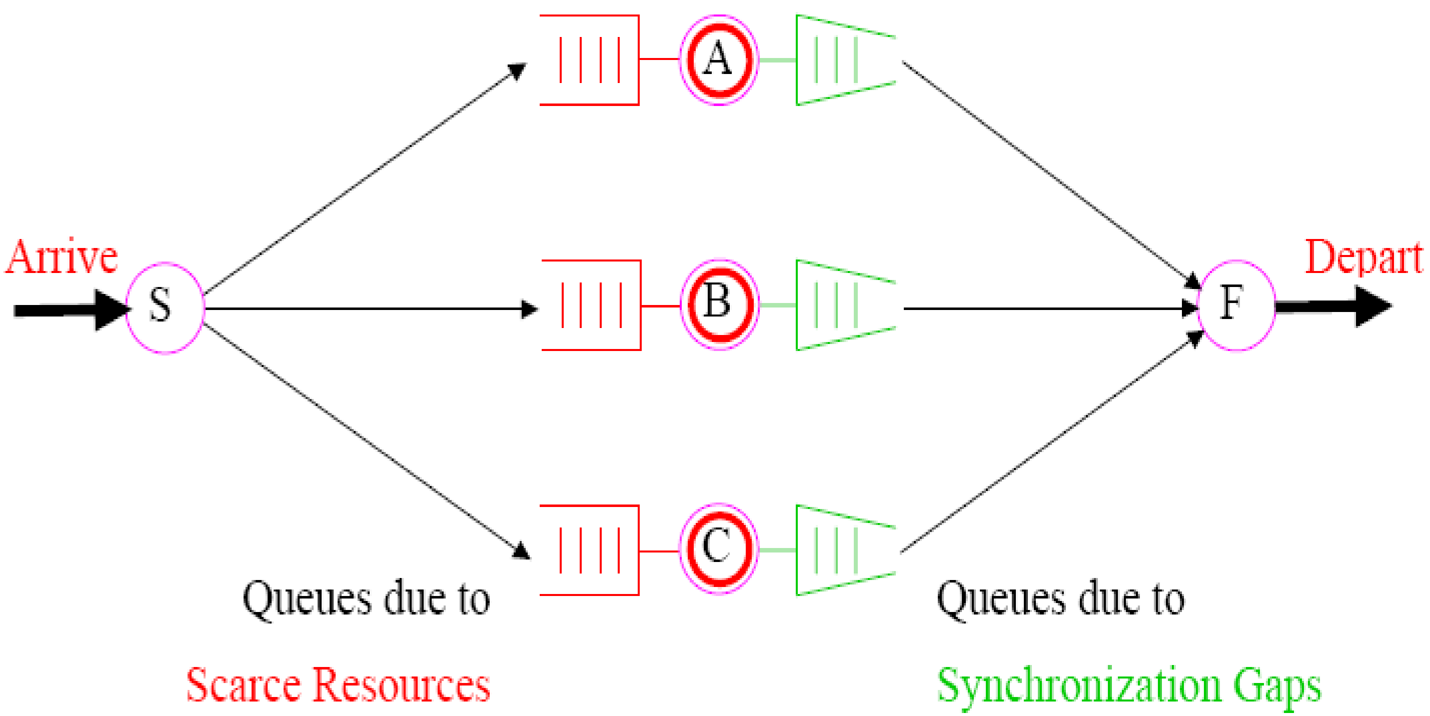
Simple Stochastic PERT

- Project Completion Time = $T = T_1 + T_2 + T_2 + T_4$
  - Assume 4 independent random variables.
  - Mean easy: $ET = ET_1 + ET_2 + ET_3 + ET_4$
  - Variance easy: $VarT = VarT_1 + VarT_2 + VarT_3 + VarT_4$
  - Distribution easy if all normal, but not otherwise.
  - Otherwise can use ***Laplace transforms***
    - $\boldsymbol{L}(T_j) = \boldsymbol{E}[\exp(-sT_j)]$
    - $\boldsymbol{L}(T) = \mathbf{L}(T_1)\,\mathbf{L}(T_2)\,\mathbf{L}(T_3)\,\mathbf{L}(T_4)$ simple product
    - **Numerical inversion**, Ex. 1.1.1. of posted paper.

Dynamic Stochastic PERT
Jobs Arriving Randomly Over Time

Activities, Resources, Random durations

Multiple projects



DS-PERT: Four Guiding Questions

- **Can we do it?**
  - Capacity analysis
- **How long will it take?**
  - Response time analysis
- **Can we do better?**
  - Sensitivity analysis
- **How much better can we do?**
  - optimization

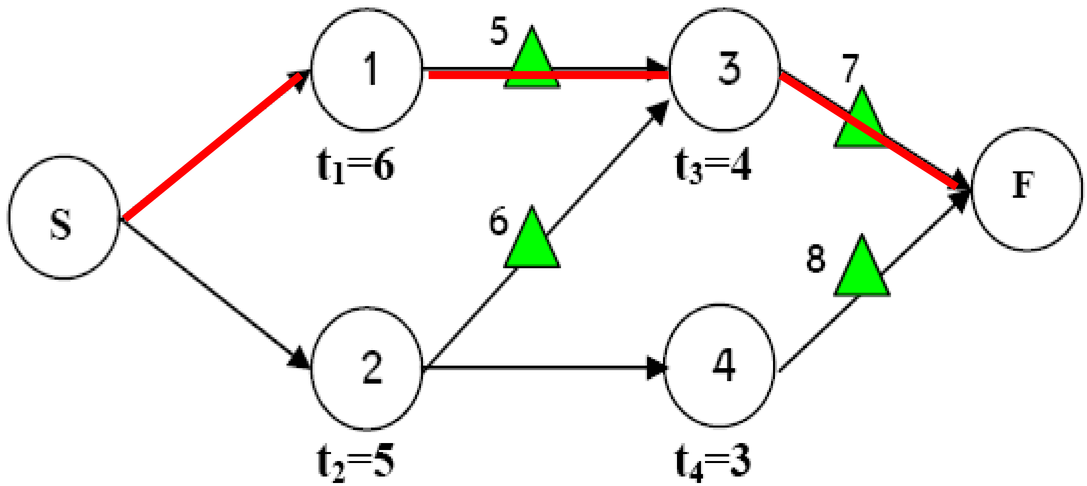
A Comparison of Alternative Models and Controls

Use ***Stochastic Simulation***

Advertisement for **IEOR 4404**

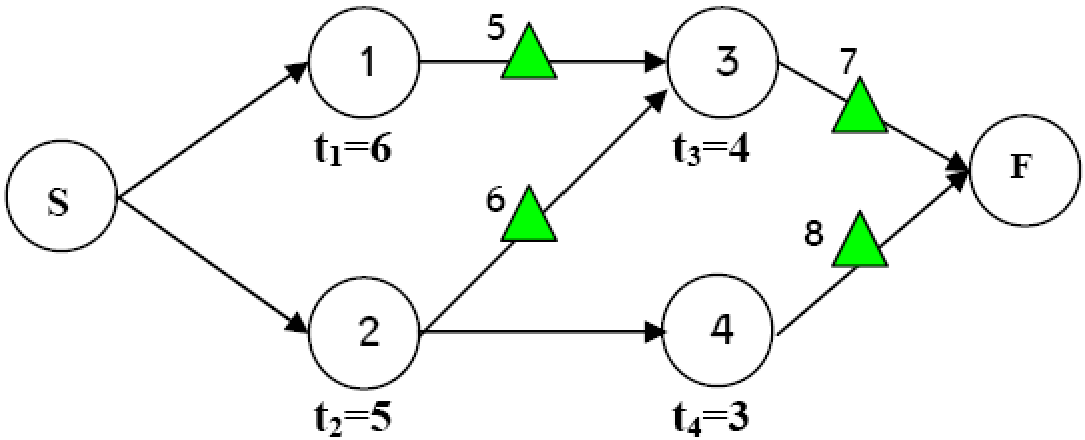
Model 1. Deterministic PERT/CPM

Synchronization queue



Critical path is S-1-3-F.
Project Completion Time is 10 days.

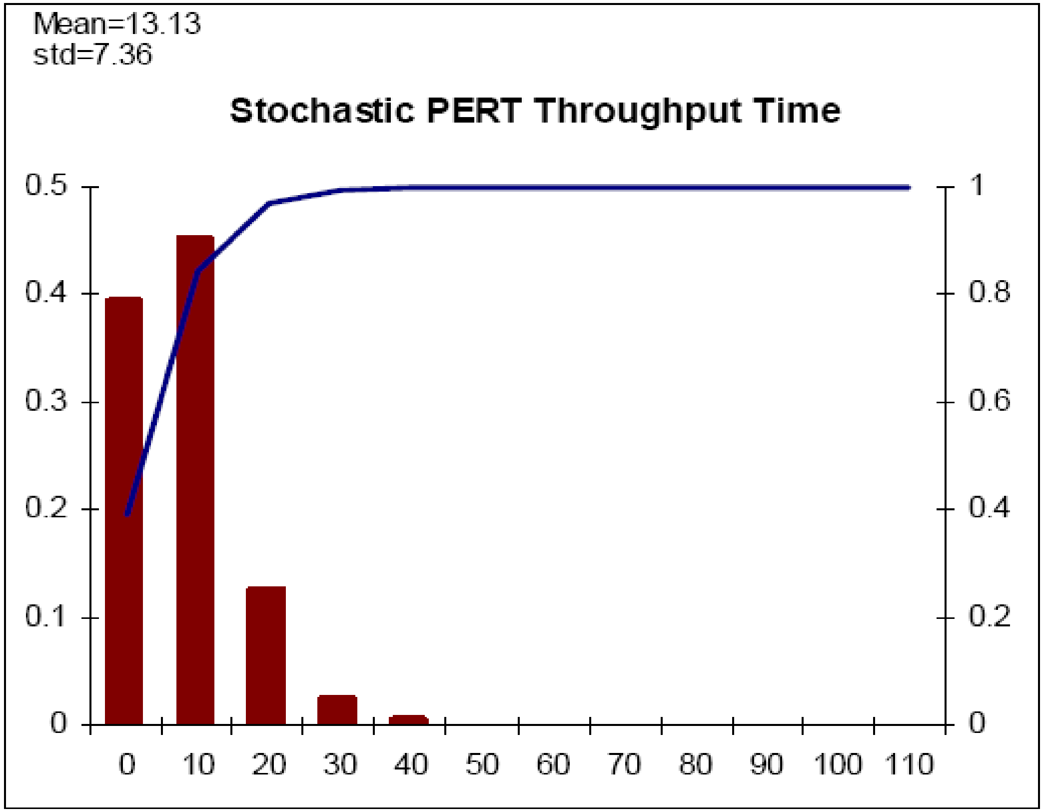
Model 2. Stochastic PERT/CPM



- Task times are now ***exponential*** with those means.
- Expected completion time now **13.13 days**, while standard deviation 7.4; compared to **10 days**

Stochastic Static PERT Throughput Time

Mean: 13.13 days.
Std: 7.36. Half C.I: 0.095



Critical Paths

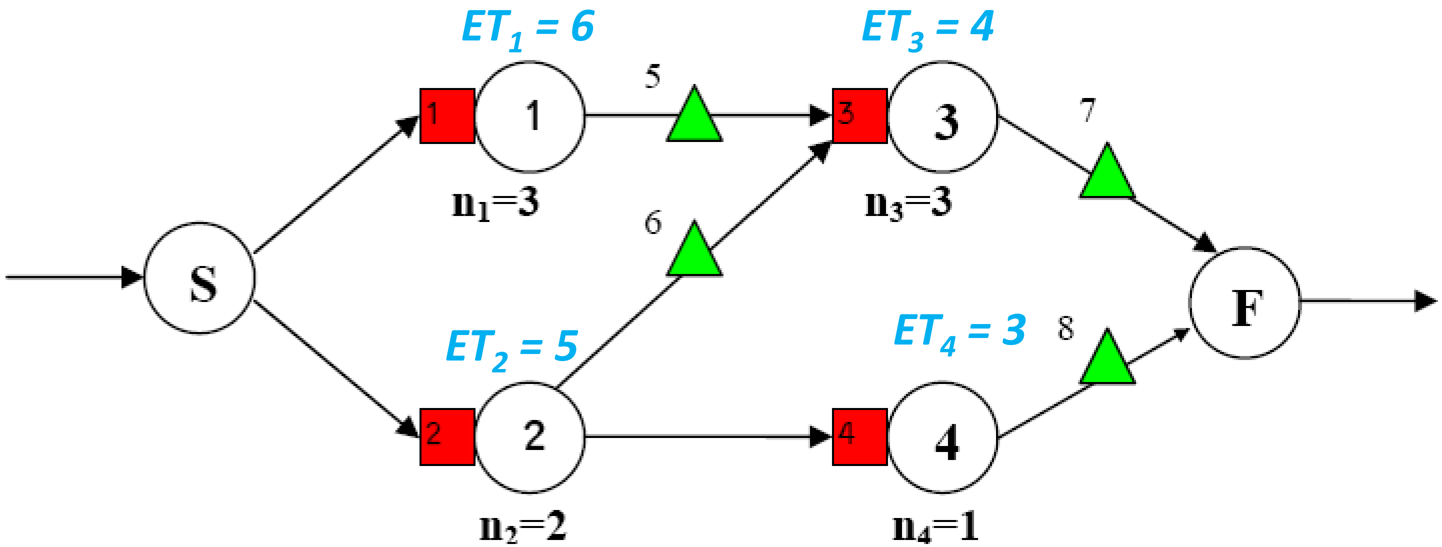
Path	Frequency	half C.I.
s-1-3-f	0.47	0.0074
s-2-3-f	0.26	0.006
s-2-4-f	0.27	0.0058
	1.00	

Critical Activities

Task	Criticality index
1	0.47
2	0.53
3	0.73
4	0.27

Criticality Index = Probability that the task is on a critical path.

Model 3. Dynamic Stochastic PERT



- ***Poisson Arrivals***, rate $\lambda=0.286$ (1 per 3.5 days)
- n_j homogeneous ***servers*** at station j
- ***FCFS service discipline***
- Task times still ***exponential*** with those means.
- Expected completion time now **32.2 days**, while standard deviation 21.2; compared to **10 & 13 days**

Capacity Analysis: Can We Do It?

Yes, if traffic intensity (resource utilization) is less than 1 at each resource

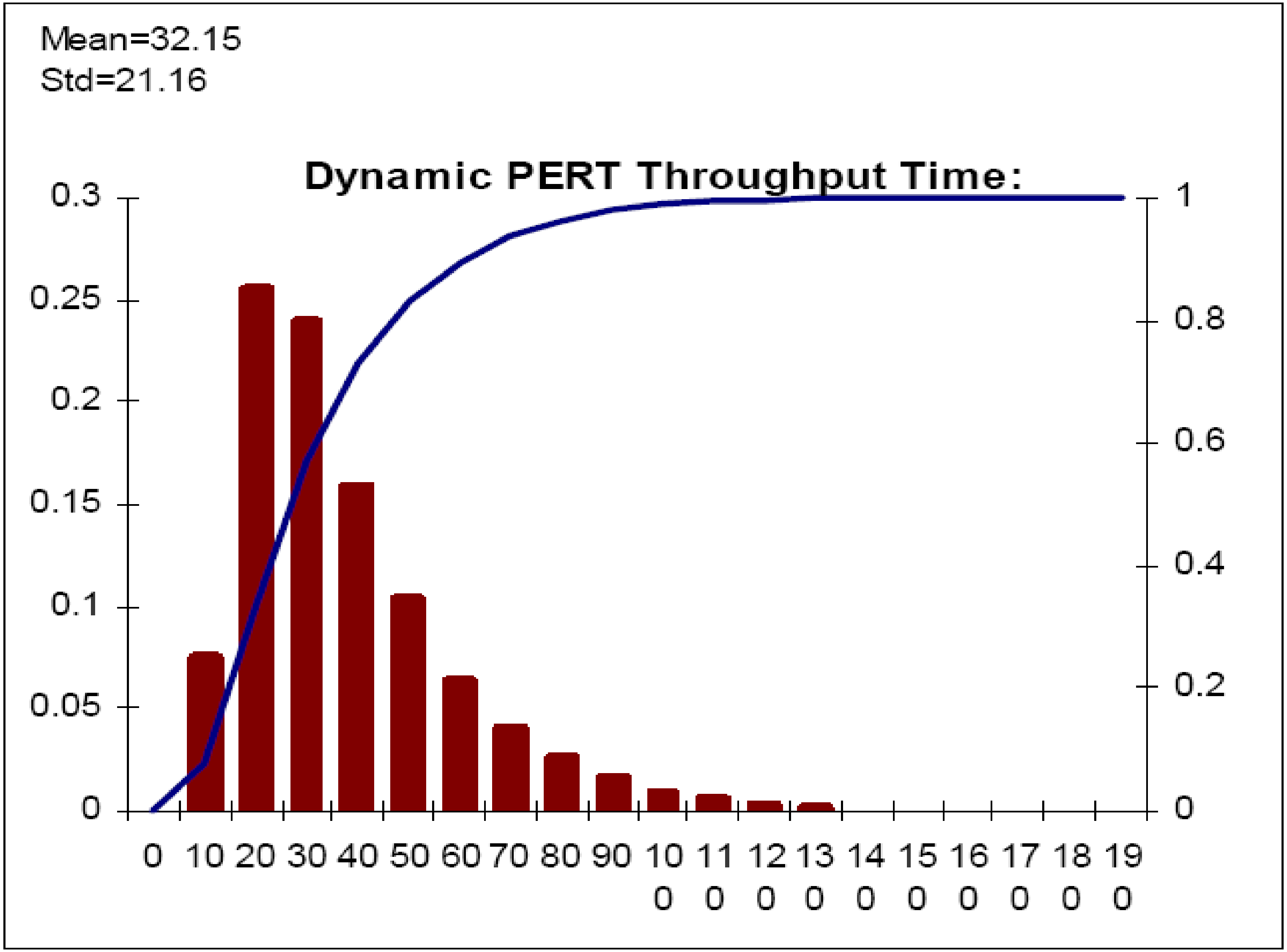
Resource Utilizations: 0.57, 0.71, 0.38 and 0.86

$\rho_1 = \lambda_1 ET_1/n_1$ = (0.286)x6/3 = 2/3.5 = 0.57

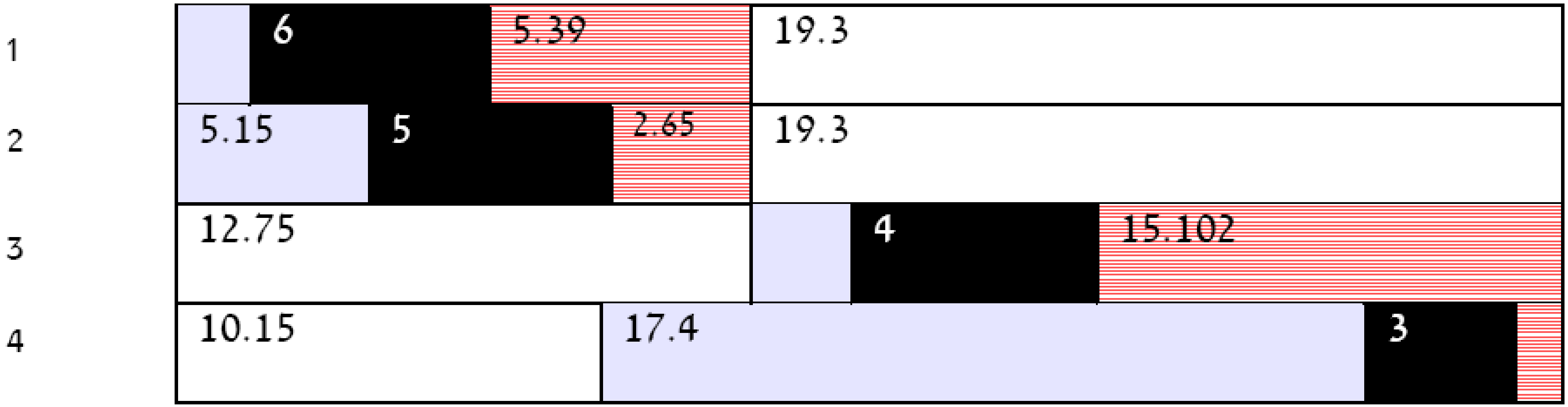
Resource 4 is the bottleneck: $\rho_4 = \lambda_4 ET_4/n_4 = 0.86$

Response Time Analysis: How Long Will It take?

Stochastic Simulation



What is happening at the four resources?



- waiting time.
- processing.
- synchronization.
- Internal (Idle)

Waiting Times at the Queues

Queues 1-4: Resource Queues

Queues 5-8: Synchronization Queues

Queue	mean	half C.I.	% from mean
1	1.42	0.063	4.43
2	5.15	0.318	6.17
3	0.234	0.009	3.84
4	17.4	1.49	8.5
5	5.39	0.298	5.5
6	2.65	0.076	2.86
7	15.102	1.42	9.5
8	1.589	0.071	4.46

Critical Paths

Path	Frequency	half C.I.
s-1-3-f	0.146	0.0067
s-2-3-f	0.104	0.0052
s-2-4-f	0.750	0.0110
	1.000	

Critical Tasks

Task	Criticality index
1	0.146
2	0.854
3	0.250
4	0.750

Task 2 has highest criticality index, but **task 4** was the **bottleneck**,

$\rho_2 = \lambda_2 ET_2/n_2 = 0.71$ while $\rho_1 = \lambda_1 ET_1/n_1 = 0.86$

Reason: Task 2 participates in more paths in the network.

What-if Analysis

1. Mean Service time at Station 2: 5 → 4
    1. Mean *ET* decrease from 32.1 to ***23.7*** days
2. Mean Service time at Station 4: 3 → 2
    1. Mean *ET* decrease from 32.1 to ***18.9***
3. Make Arrivals Deterministic at same rate
    1. Mean *ET* decrease from 32.1 to ***22.5***
4. No. 3 above + move server from 3 to 4
    1. Mean *ET* decrease from 32.1 to ***15.7***
5. Change from exponential to uniform dists
    1. [0,7],[3,9],[3,5],[2,4]
    2. Mean *ET* decrease from 32.1 to ***12.8***
6. No. 5 above + move server from 3 to 4
    1. Mean *ET* decrease from 12.8 to ***11.3*** (compare to 10)
7. No. 3 + No. 6
    1. Mean *ET* decrease from 11.3 to ***10.5*** (compare to 10)

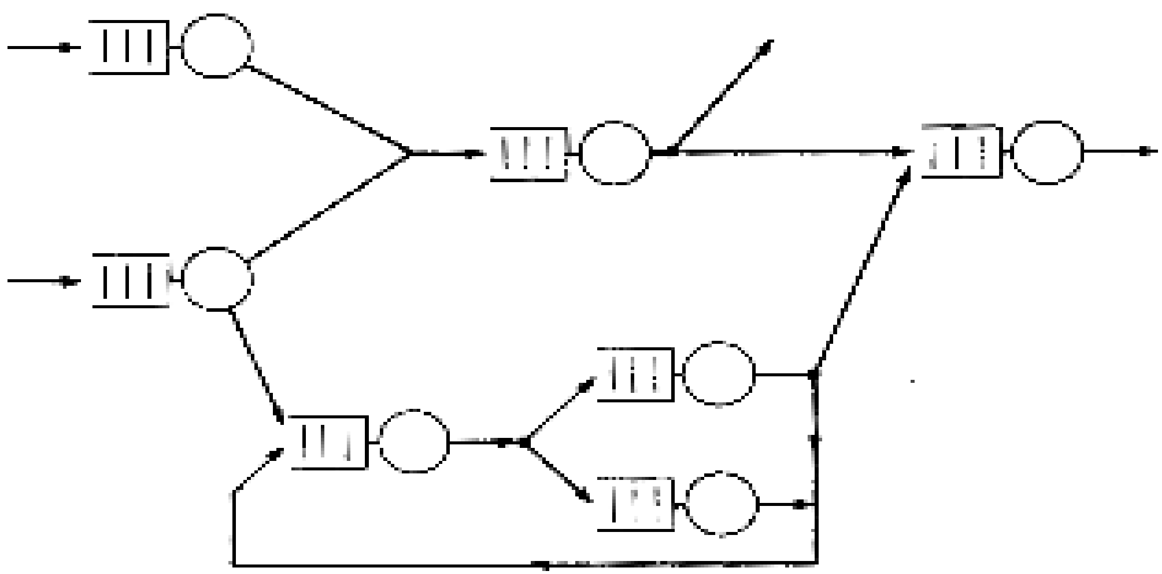
Dynamic Stochastic Control

- **No control** (above)
  - ***ET = 32.1*** days
- **MinSLK:** highest priority in queue to a minimum slack activity, with slack times updated
  - ***ET = 21.6*** days
- **QSC** (Queue Size Control): Do not admit new job when bottleneck queue exceeds limit ***6***
  - ***ET = 18.6*** days
- Many others: Stochastic Scheduling

Processing Networks

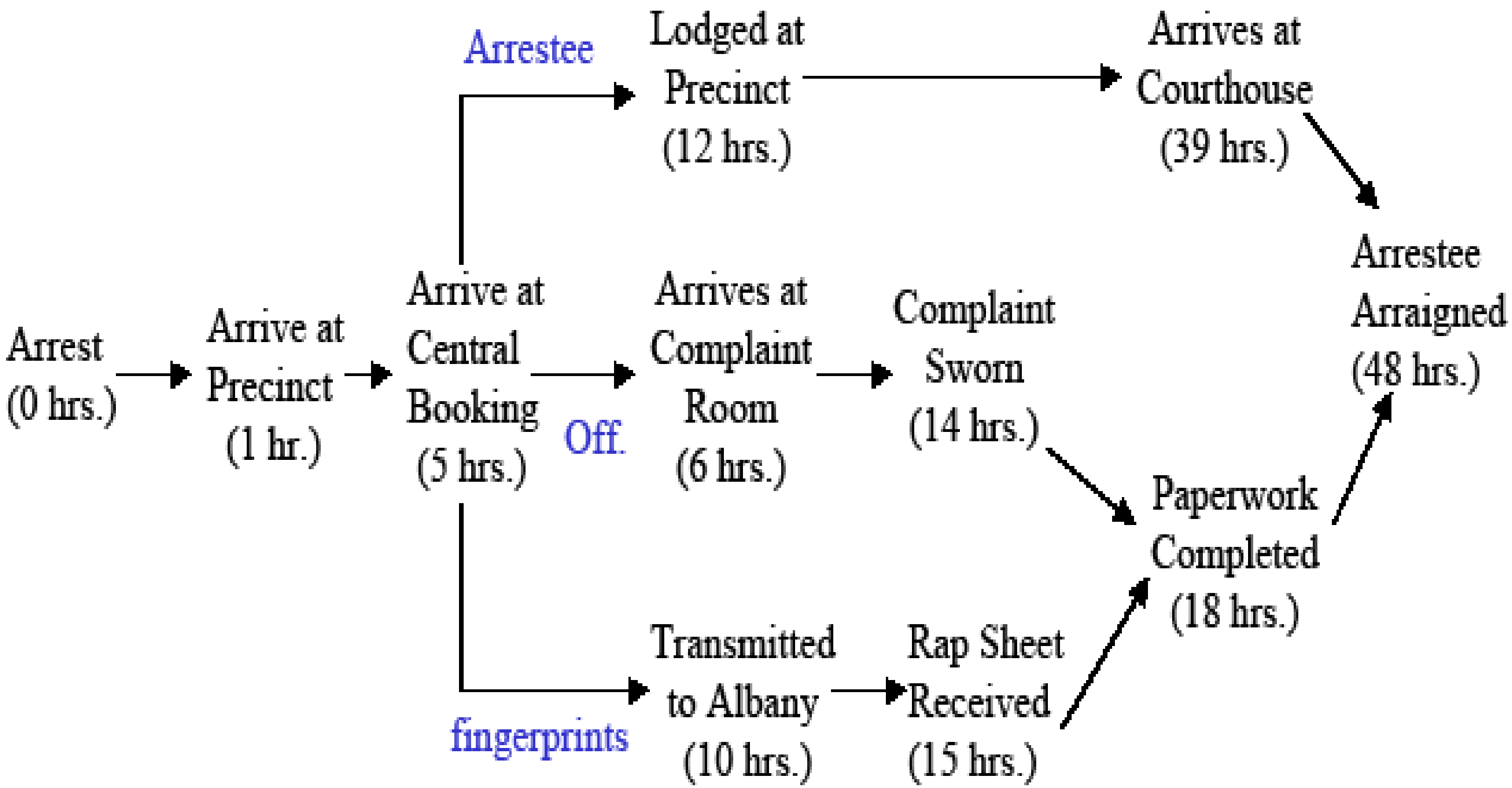
(Includes DS-PERT above)

Contrast with
An Open Network of Queues

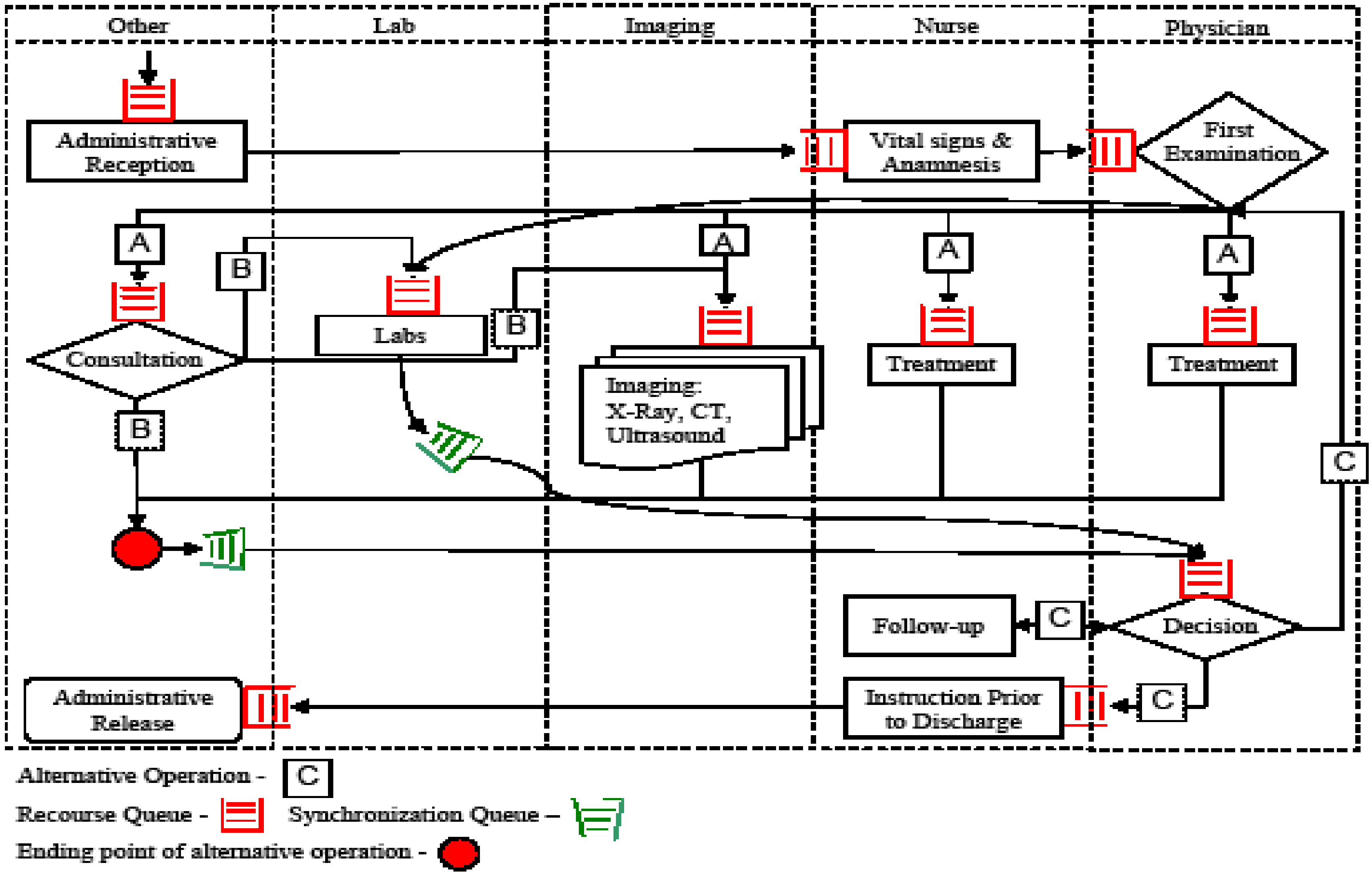


…but now also ***add*** **precedence constraints**
and **synchronization queues**

Arrest-to-Arraignment Process
Larson et al. (1993)



Patient Flow in an Emergency Department



Processing Networks: Building Blocks

1. **Customers** (jobs) are Served, **Flow**, Processed;
   Attributes: Arrivals, Services, Routes, Patience,...

2. **Activities** (tasks, services) are what the "jobs" are made of;
   Attributes: **Partially ordered** via Precedence-Constraints, summarized in an **Activity (Precedence) Graph** (nodes = activities, arcs = precedences).

3. **Resources** serve the Customers (perform the Activities);
   Attributes: **Scarce**, limited by **Processing (Dynamic) Capacity** (maximal sustainable service rate; in discrete events, capacity also equals the reciprocal of average service-time); Customers' Constituency, Pools, ..., summarized in a **Resource-Graph** (nodes = queues + resource-pools, arcs = flows).

4. **Queues** (Buffers) are where activities (customers) wait for their service-process to continue; **Human** (vs. Inventories)
   Attributes: Storage (Static) Capacity, which could be infinity;
   Operational queues are either **Resource-Queues** (waiting for a resource to become available) or **Synchronization-Queues** (waiting for a precedence-constraint to be fulfilled).

5. **Protocols** embody **information** for admission, routing, scheduling, data-archival and retrieval, quality-monitoring, performance measures (definition, monitoring),...

References (and sources of references) Processing Networks

1. M. Armony, S. Israelit, A. Mandelbaum, Y. N. Marmor, Y. Tseytlin and G. B. Yom-Tov. 2011. Patient flow in hospitals: a data-based queueing-science perspective. The Technion.
2. Larson, R. C., M. F. Cahn, M. C. Shell (1993) Improving the New York City Arrest-to-Arraignment System. *Interfaces*, vol. 23, 76-96. (See Lecture 2.)
3. F. Baccelli, W. A. Massey and D. Towsley. 1989. Acyclic fork-join queueing networks. *Journal of the Association for Computing Machinery*. 36 615-642.
4. Hongyuan Lu and Guodong Pang. 2014. Gaussian Limits for A Fork-Join Network with Non-Exchangeable Synchronization in Heavy Traffic. Penn. State University.

More References
(and sources of references)
Processing Networks

1. W. Whitt, The Maximum Throughput on a Golf Course. *Production and Operations Management*, published online November 20, 2014.

2. C. J. Willits and D. C. Dietz. 2001. Nested fork-join queueing network model for analysis of airfield operations. *Journal of Aircraft*. 38 848-855.