Queue Mining:
Queueing Theory meets Process Mining

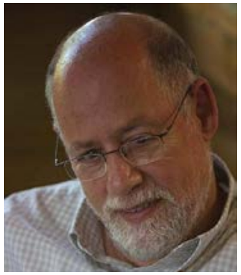
AIS Group Meeting

Arik Senderovich

10/11/2016

Short Bio

- Postdoc @ Technion Israel Institute of Technology

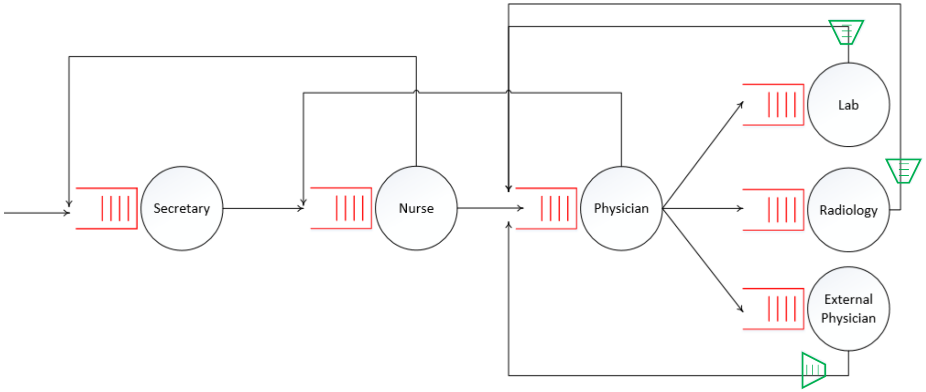


- BSc in IE&M (Knowledge Systems)
- MSc in STAT (Service Engineering)
- PhD in Data Science (Queue Mining)

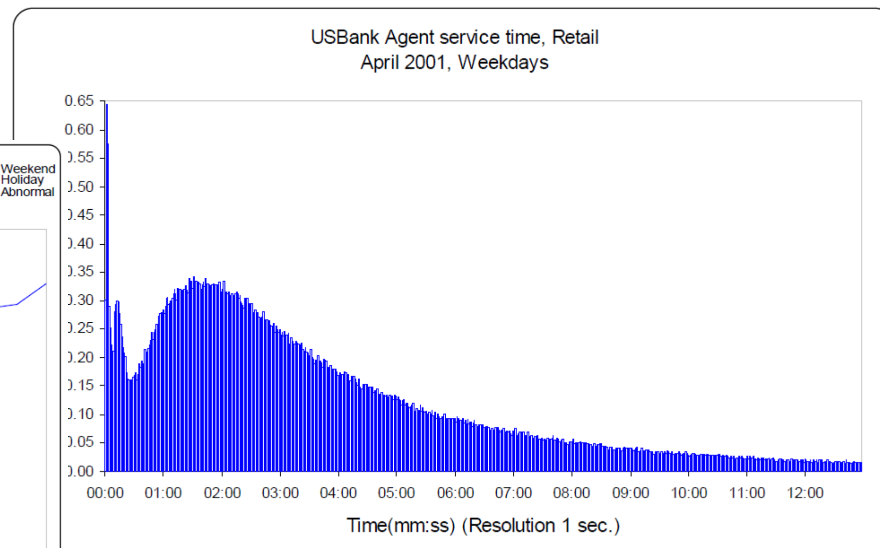
Favorites: Service Processes

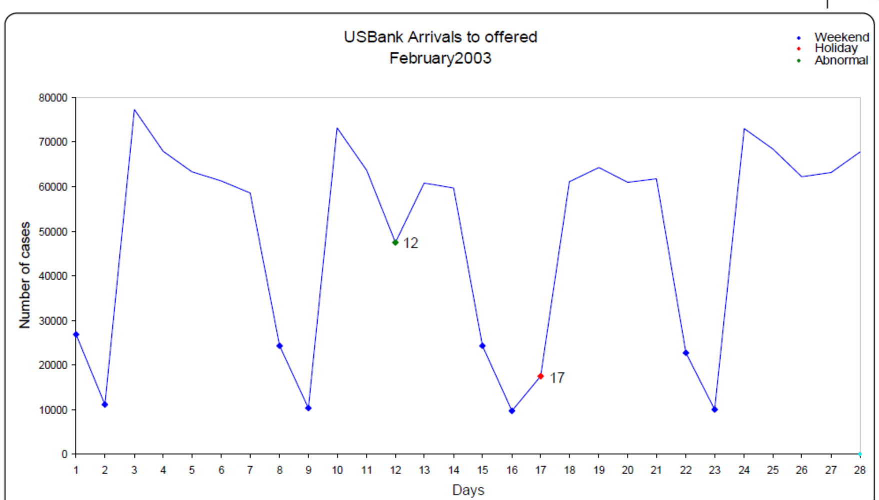
Processes where (efficient and effective) service is the desired business outcome:

- Call centers
- Hospitals
- Transportation

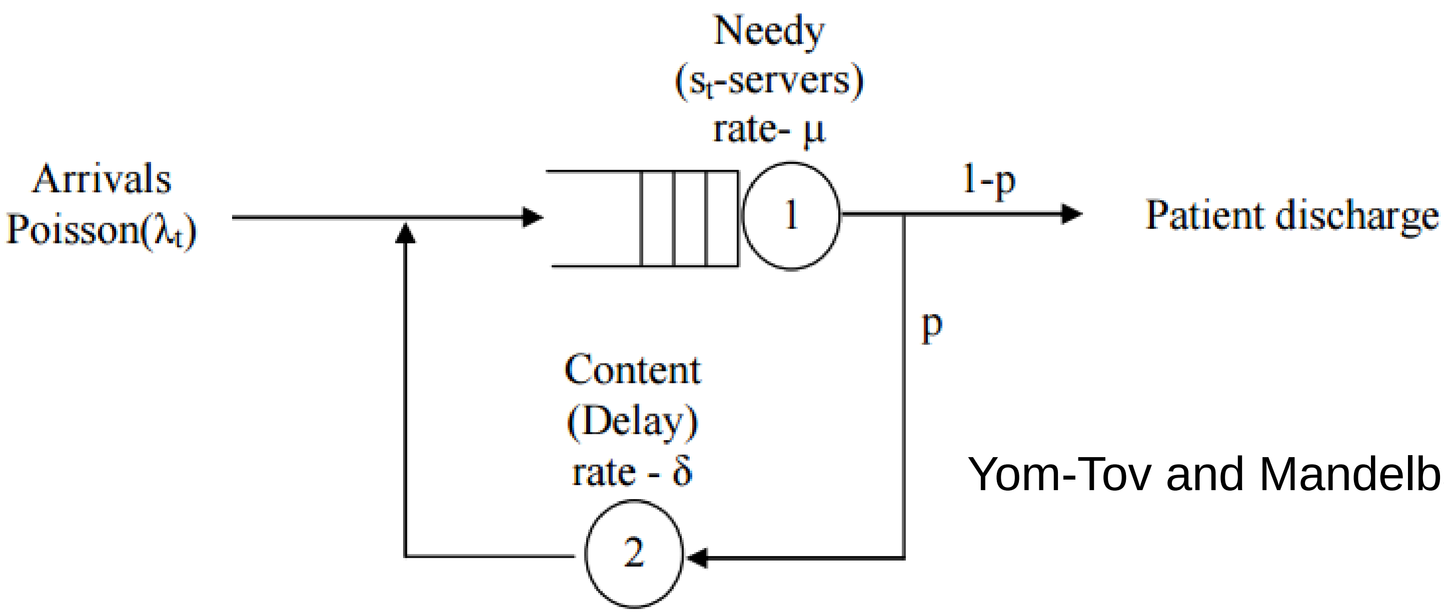


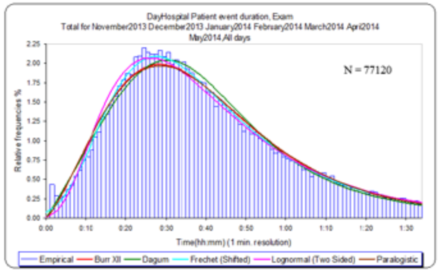
SEELab and SEEData



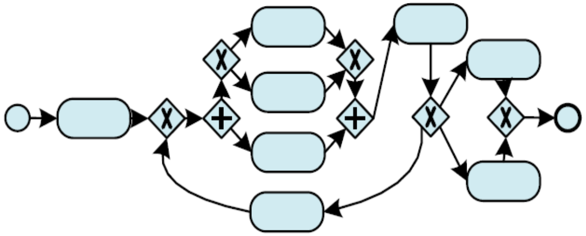


Hand-made Performance Modeling



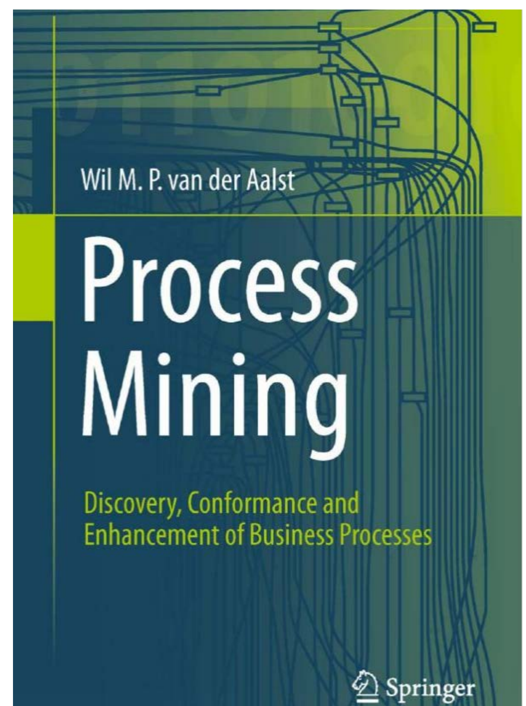


Yom-Tov and Mandelbaum [2014]



Automated process modeling based on data =
Process mining

Logging



Focus: Performance Analysis

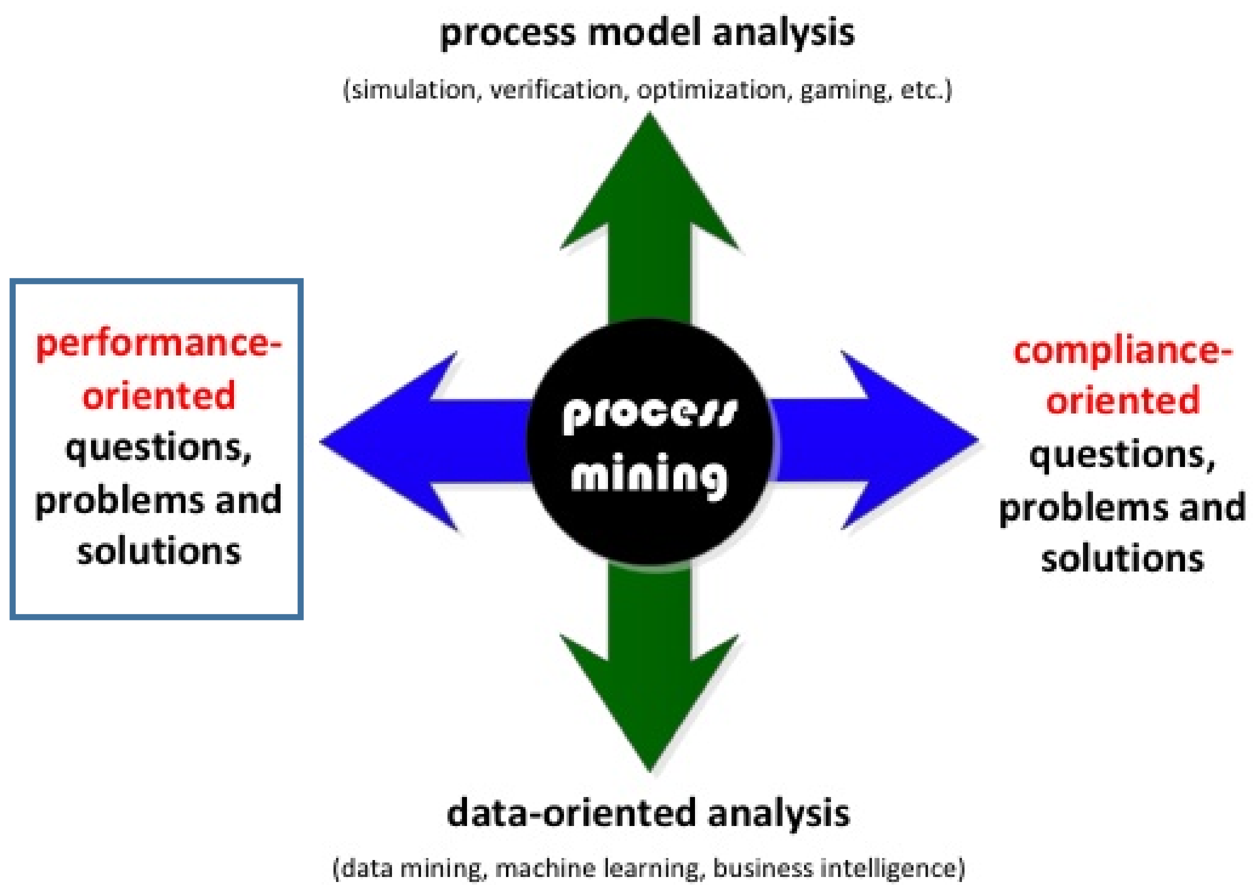
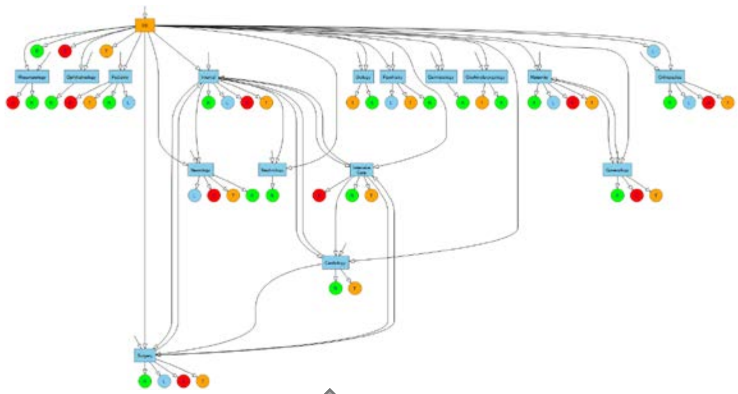


Illustration by [Wil van der Aalst]()



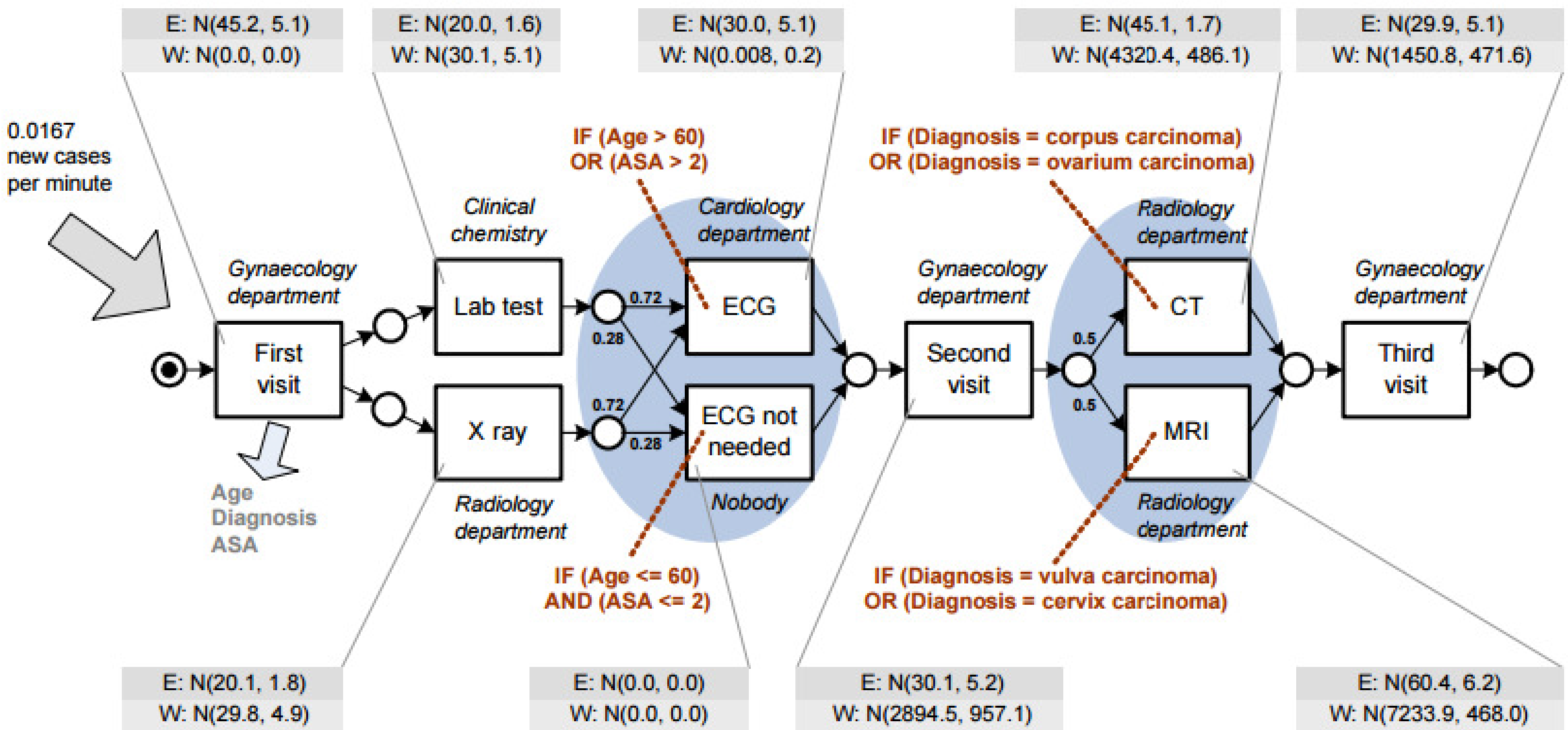
Queue mining =

Process mining with Queueing Theory

Logging

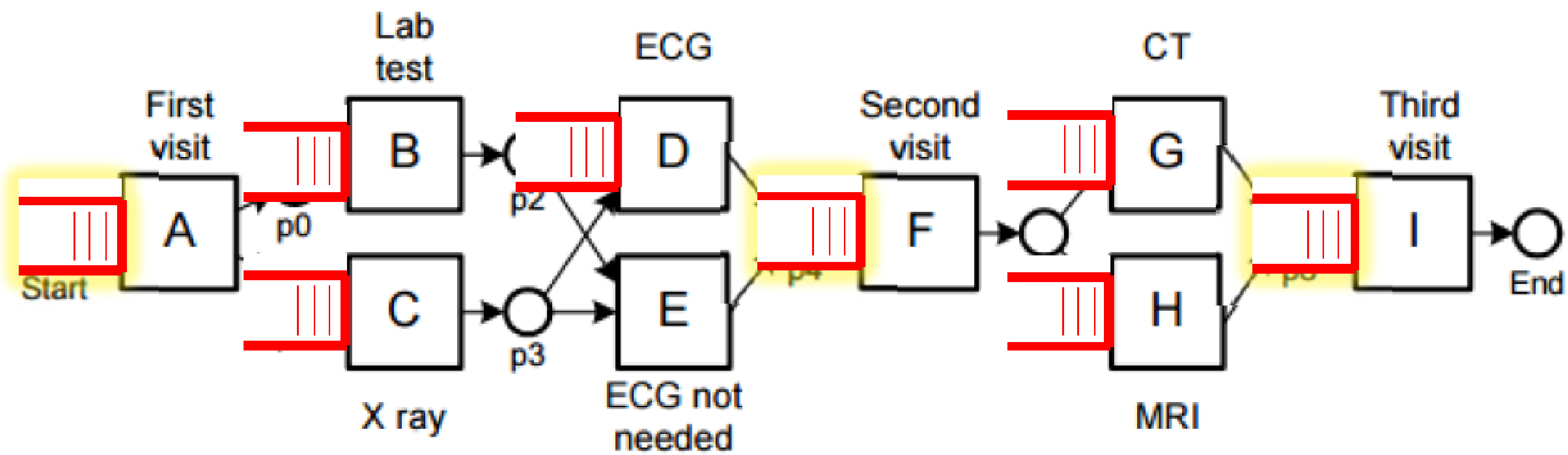
S. A., Weidlich M., Gal A., Mandelbaum A., in Information Systems 2014

State-of-the-art: Approach I



From Rozinat et al. [2009]; Rogge-Solti et al. [2013]

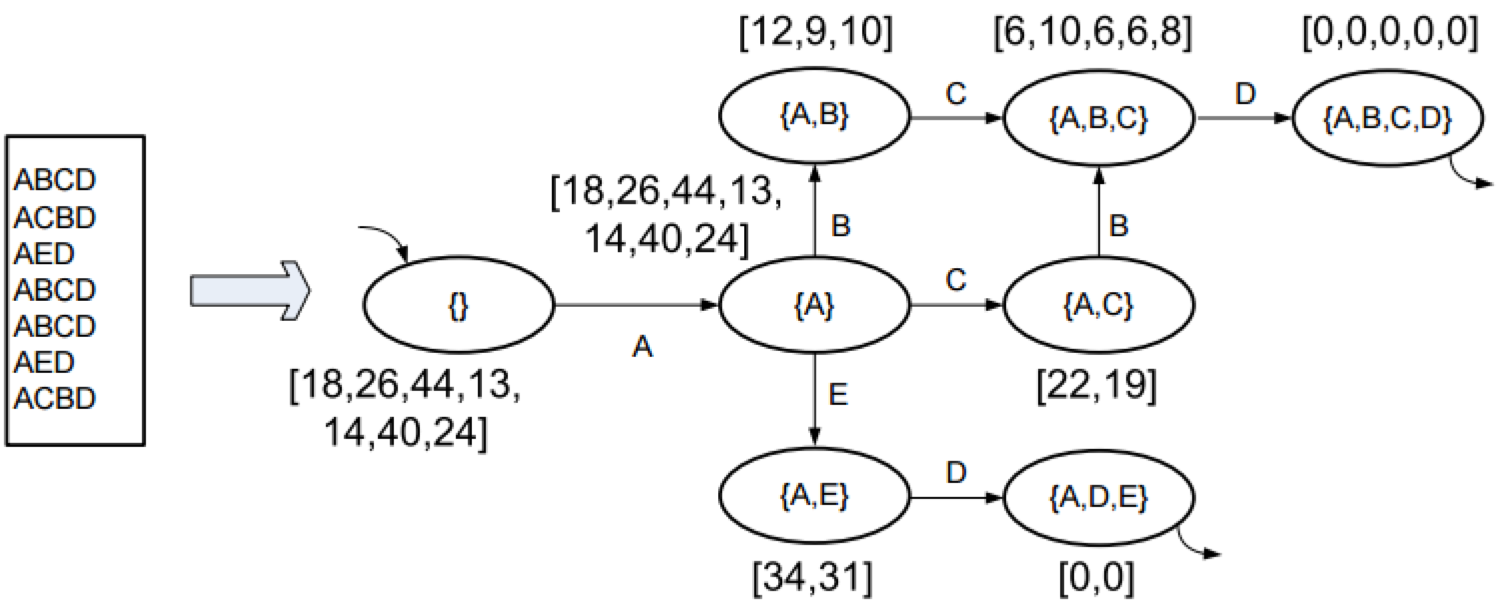
Queue Mining: Approach I



Queueing models:

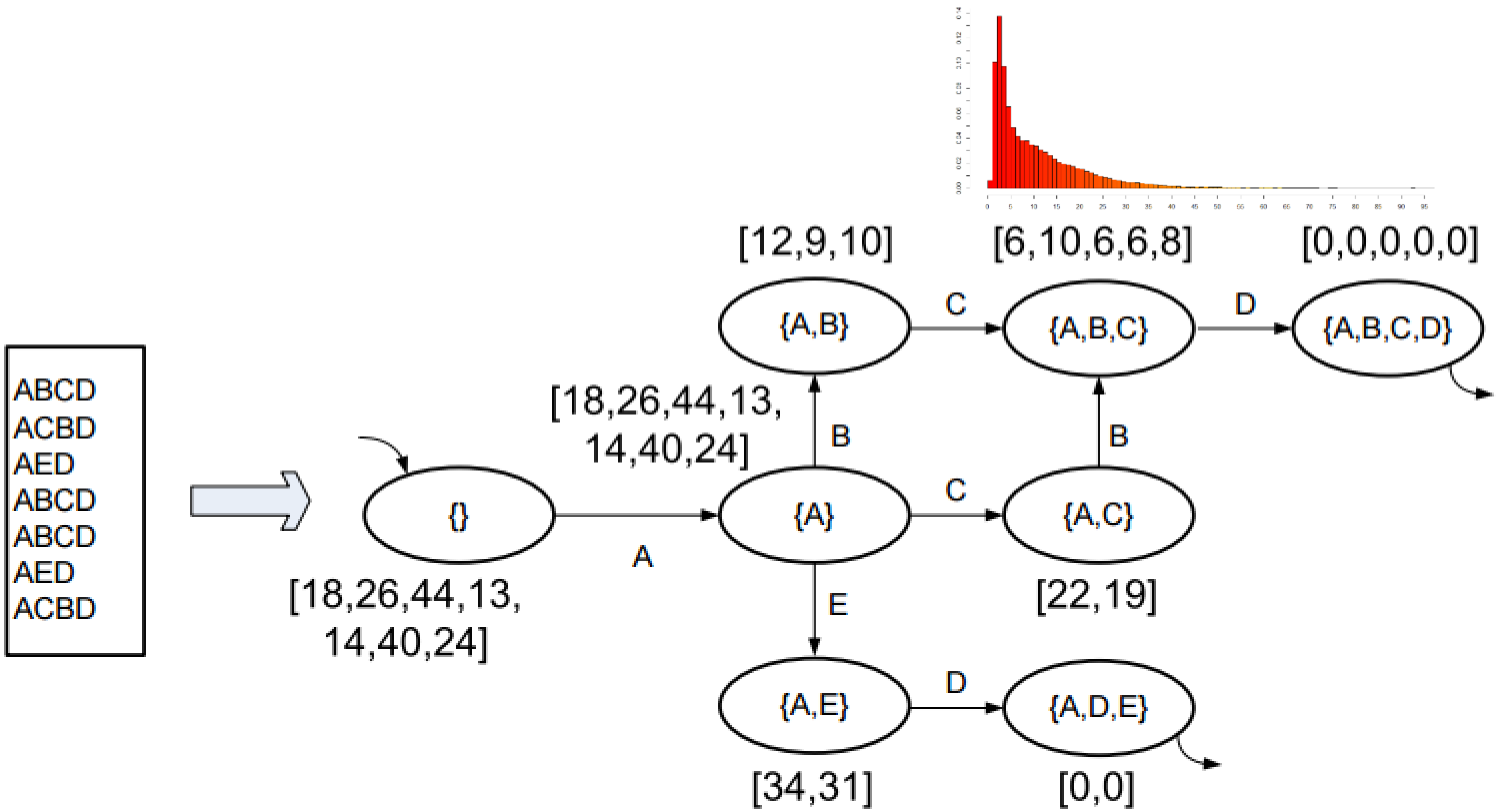
- Analytically simple models (efficiency) – no need for simulation
- (Often) accurate performance analysis w.r.t. data (robust/generalize well)

State-of-the-art: Approach II



From van der Aalst et al. [2011]

Queue Mining: Approach II



Queueing features added to state:

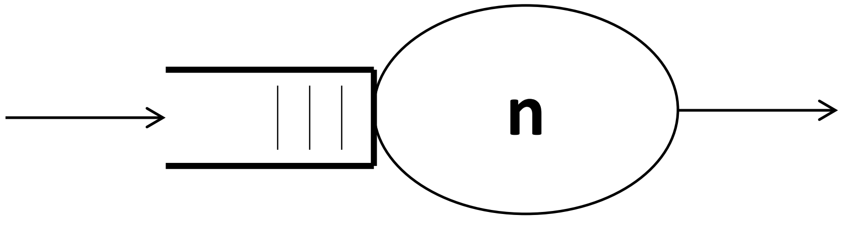
- Examples: queue-lengths, delays, classes
- Input for machine learning techniques

Outline

- Background
- Single-station queueing models
  - Single-class
  - Multi-class
- Queueing networks
  - Pre-defined routing
  - Random routing
- Conformance checking with queueing networks

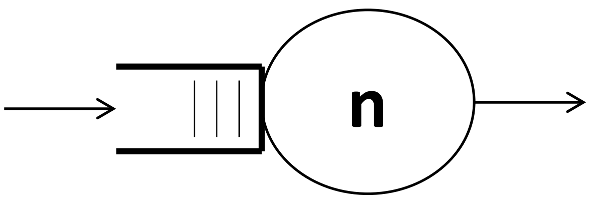
Feel free to ask questions

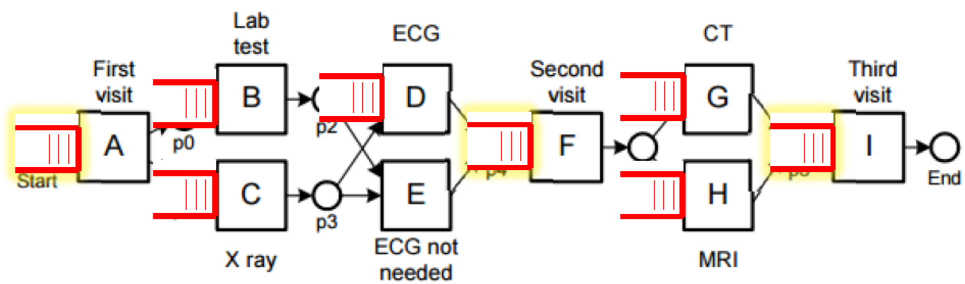
Single-Station Single-Class Queues



Are these useful models?

Single-Station Queues

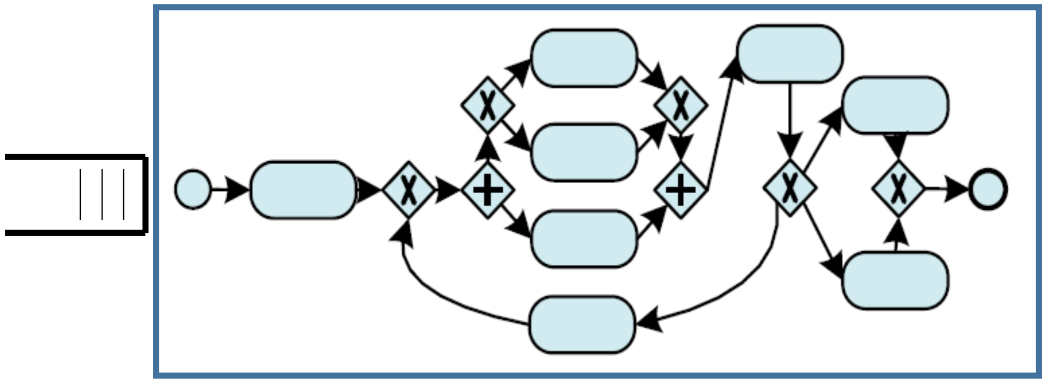




Are these useful models?

- Building block of networks

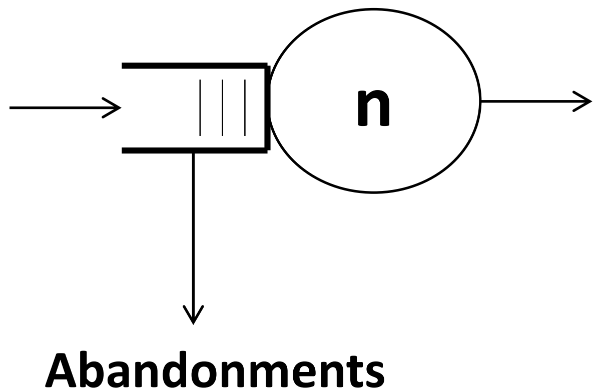
Single-Station Queues



Are these useful models?

- ➢ Building block of networks
- ➢ Single-resource type processes
  - Total time is **delay** (queueing) and process time

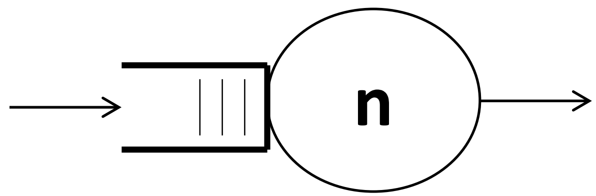
Queueing Model: Building Blocks



Kendall's notation – A/B/C/Y/Z+X:

- A – arrivals, B – service times
- C – static server capacity (n servers); Y – queue size
- Z – service policy (FCFS, LCFS, Processor Sharing…)
- X – (Im)patience

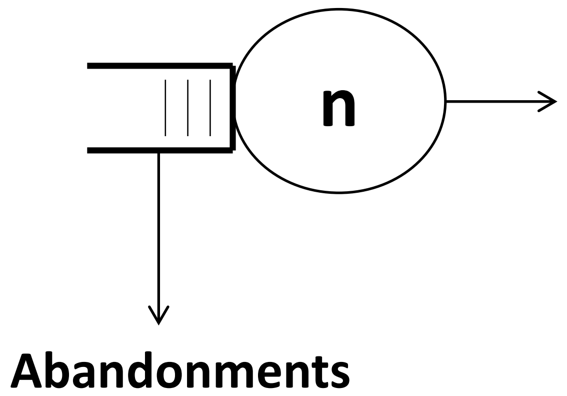
Example: M/M/n



Assumptions (A/B/C/Y/Z+X):

- Dropped notation Y,Z,X (defaults are taken): infinite queue size, FCFS policy, no abandonments
- M - Poisson arrivals (completely random, one at a time, constant rate)
- M - Exponentially distributed service times
- Easy to analyze when parameters are known (data)

Problem: Delay Prediction



CAiSE2014 paper
with Weidlich, Gal, Mandelbaum

How long will the target customer wait?

- Online prediction problem
- Approach I – fit q-model (&parameters) from the log
- Approach II – transition system + learning

Notation and Accuracy Measure

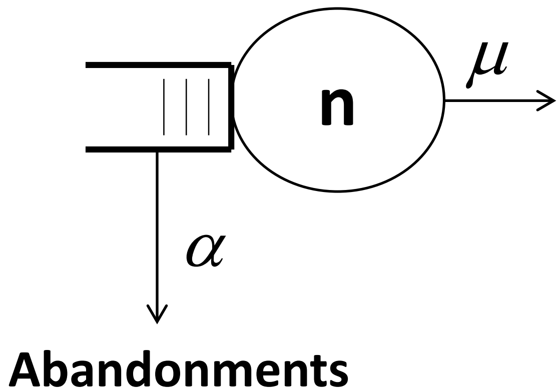
- The actual waiting time of a customer: W_i
- Delay predictor from a certain method: θ_{method}
- Accuracy via the root of average squared-error (RASE):

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(W_i - \theta^{i}{}_{method})^2}$$

- Systemic errors in assumptions- avg. absolute bias:

$$\left|\frac{1}{n}\sum_{i=1}^{n}(W_i - \theta^{i}{}_{method})\right|$$

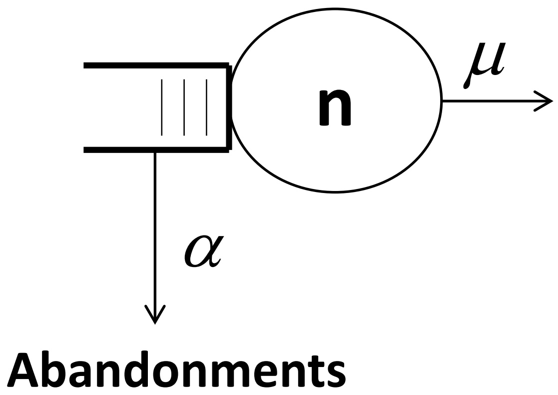
Approach I: Queueing Model is Fitted



G/M/n+M model:

- Exponential service times and (im)patience
- General arrival rates, FCFS policy, unlimited queue

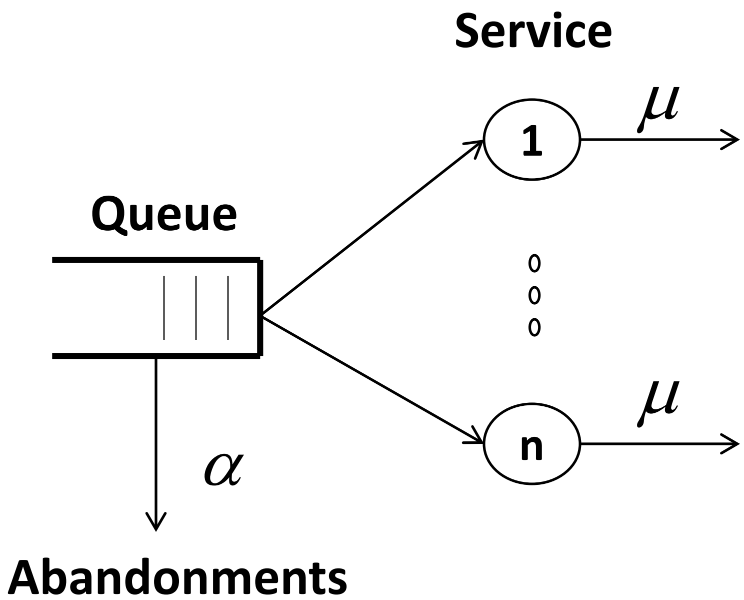
Approach I: Analysis



Two families of delay predictors:

1. Queue-length (state based)
2. Snapshot principle (history based)

Queue-Length Predictors



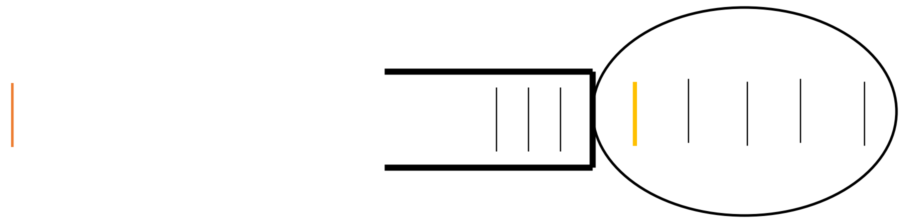
$$\theta_{QLM-NA} = \frac{QL+1}{n\mu}$$

$$\theta_{QLM} = \sum_{i=0}^{QL} \frac{1}{n\mu + i\alpha}$$

Whitt [1999]

Snapshot Prediction: Last-to-Enter-Service

(Armony et al., 2009; Ibrahim and Whitt, 2009)

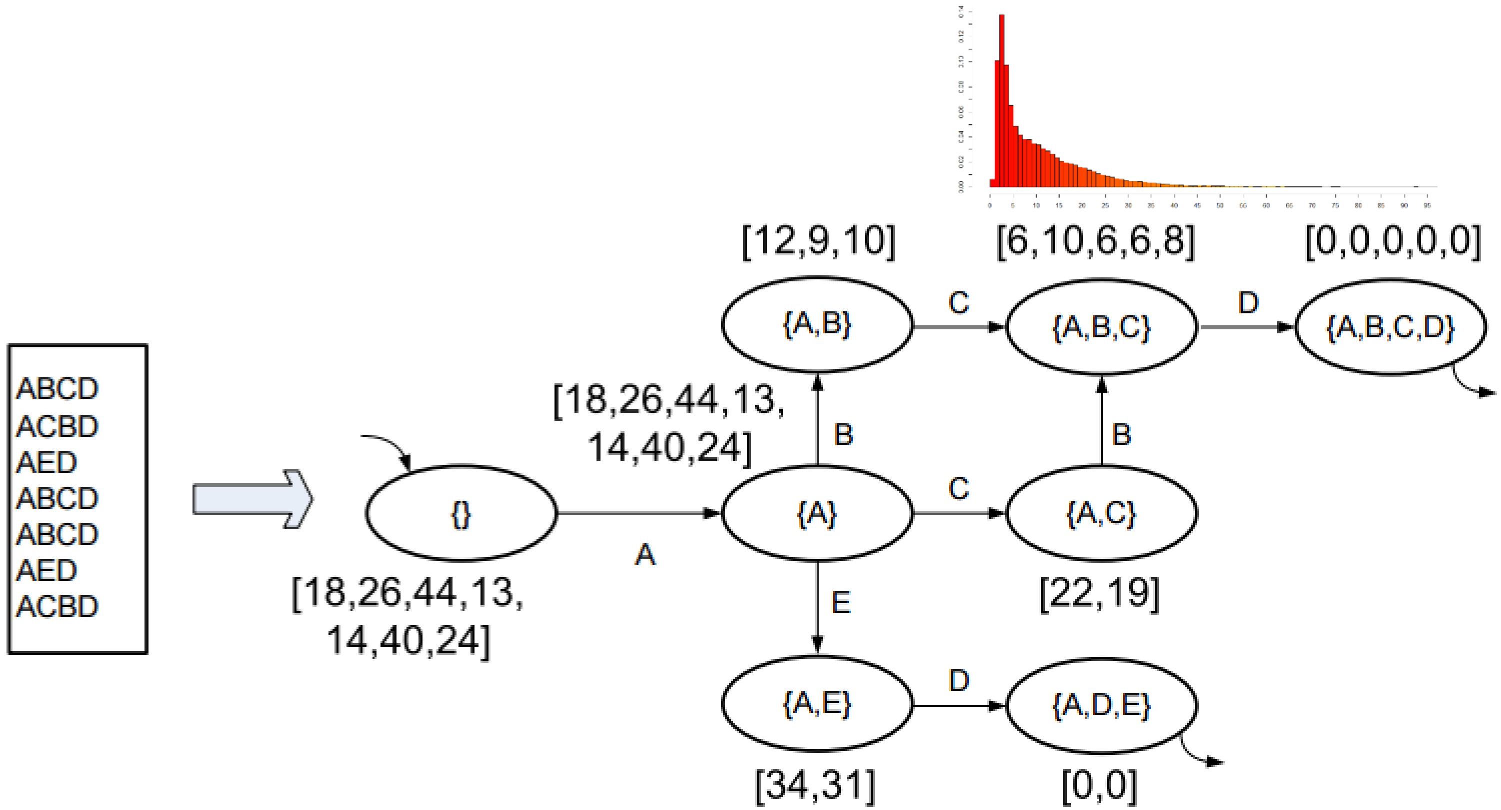


Prediction:

The last customer to enter service waited ***w*** in queue

$$\theta_{LES} = w$$

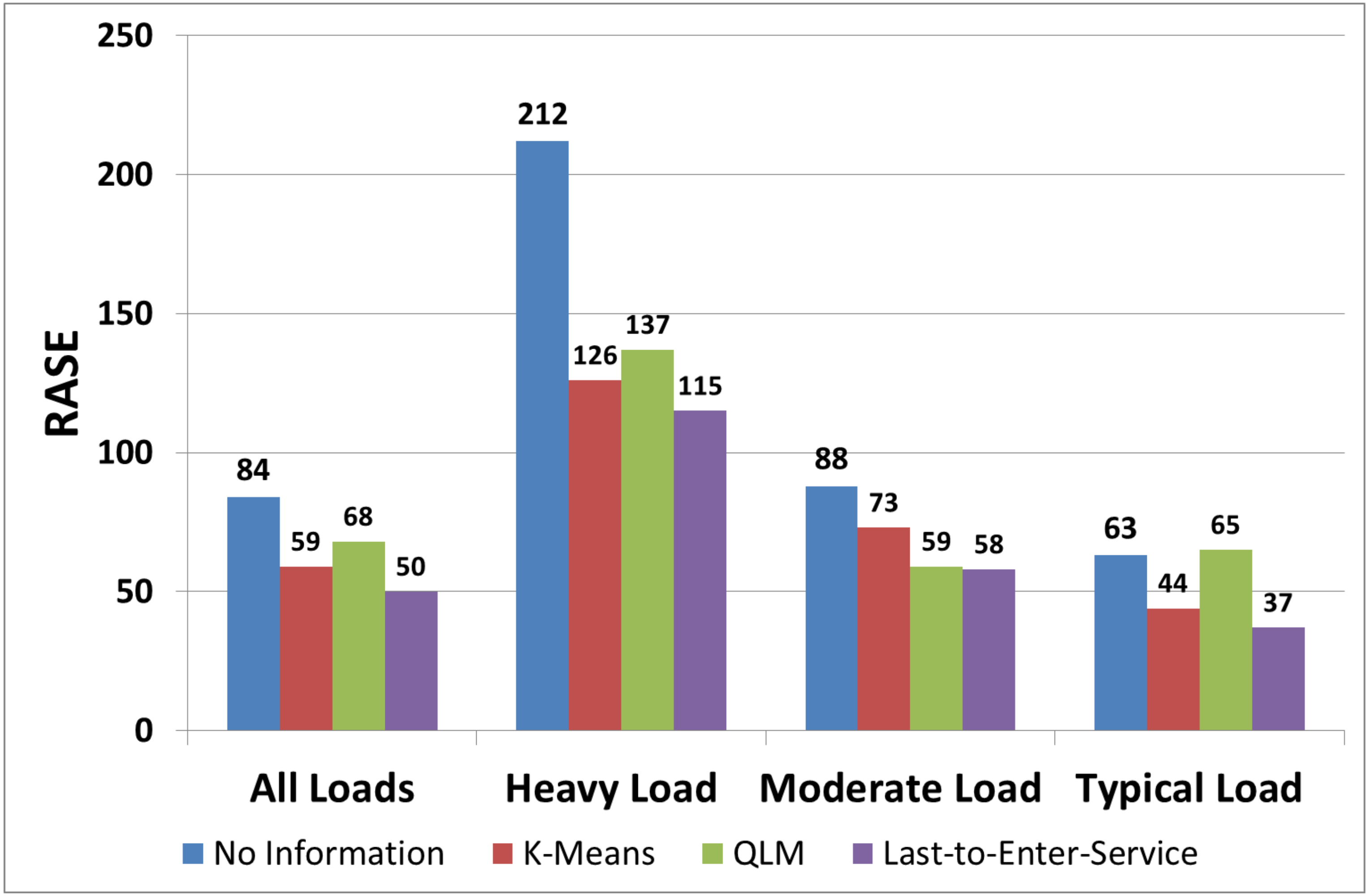
Approach II: Transition System Based



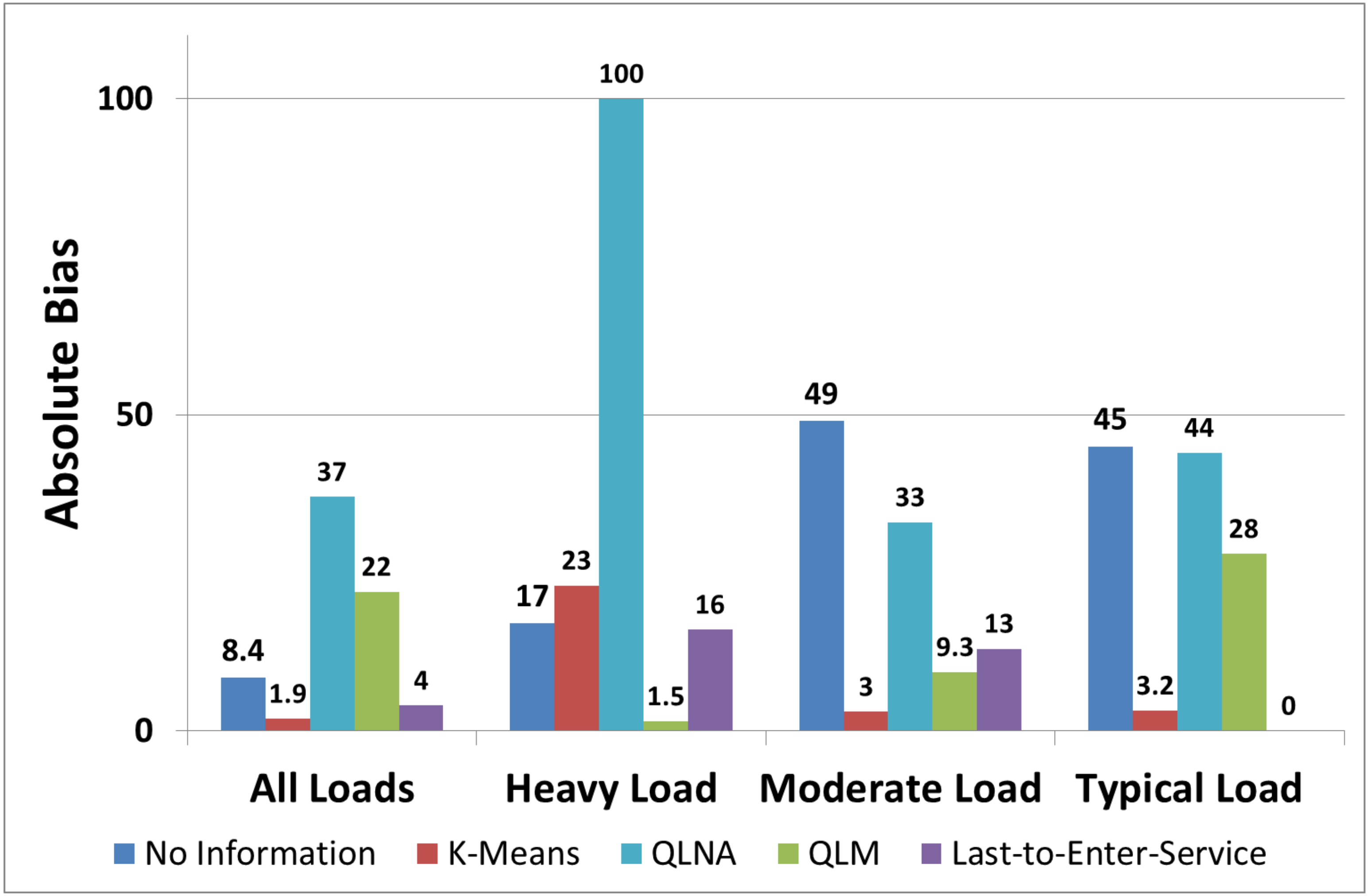
Transition system with queueing features:

- Queue lengths are clustered (heavy, moderate, typ.)
- Prediction is based on QL cluster + progress

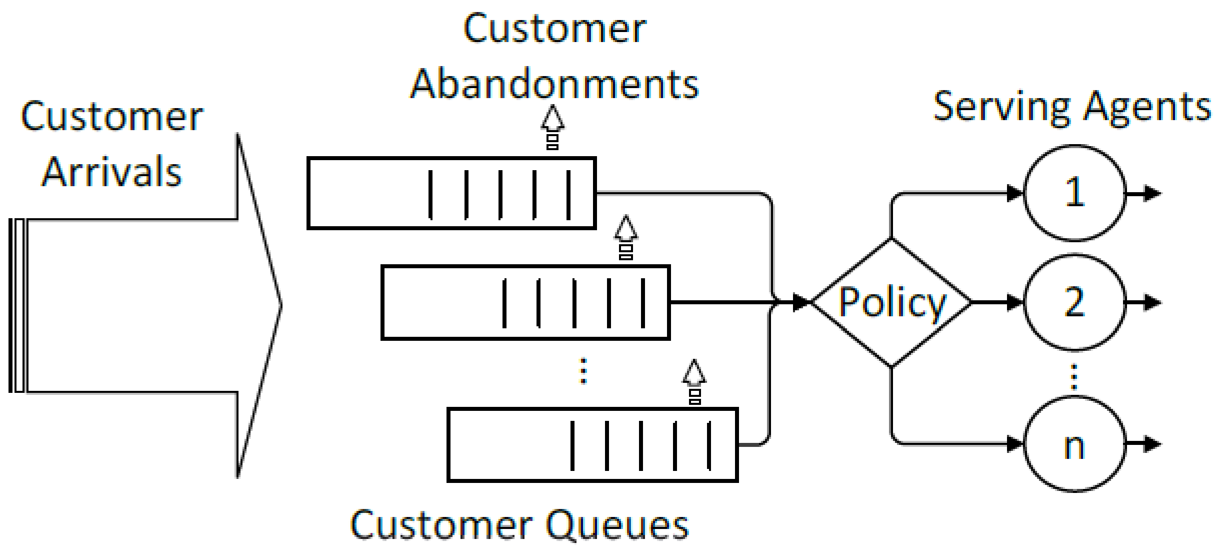
Results I: Bank's Call Center Data



Results II: Bank's Call Center Data



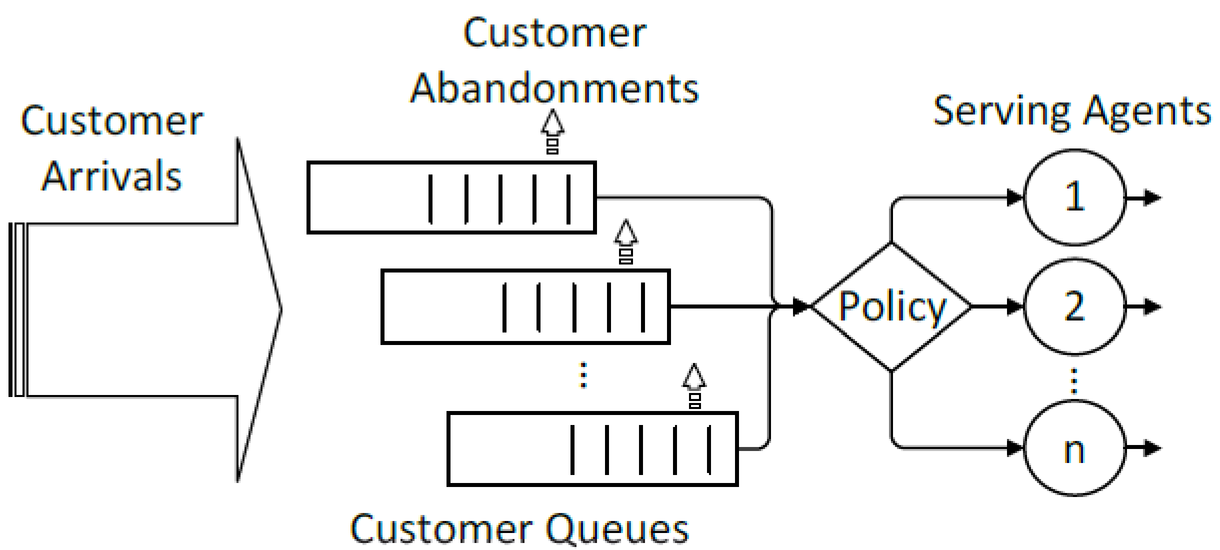
Single-Station Multi-Class Queues



(b) Queueing perspective

Useful?

Single-Station Multi-Class Queues

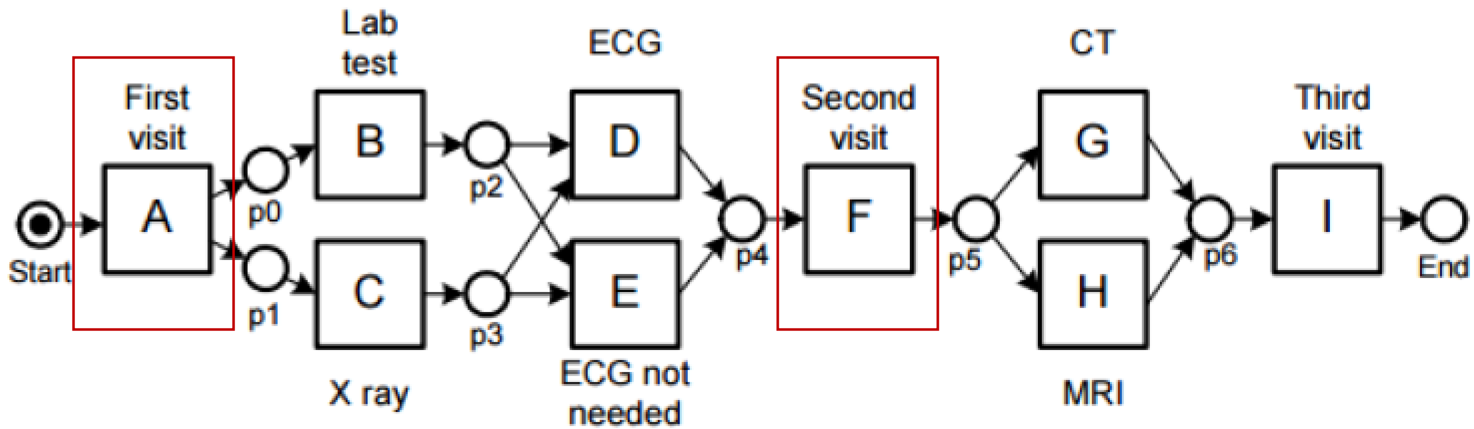


(b) Queueing perspective

Useful?

- Different types of customers (VIP vs. Regular; Urgent vs. Ambulatory)

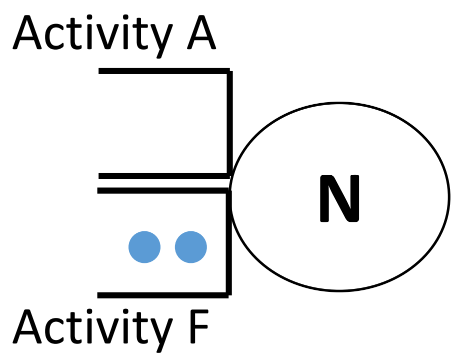
Single-Station Multi-Class Queues



Useful?

- Different types of customers (VIP vs. Regular; Urgent vs. Ambulatory)
- Classes = activities (A vs. F – A gets priority)

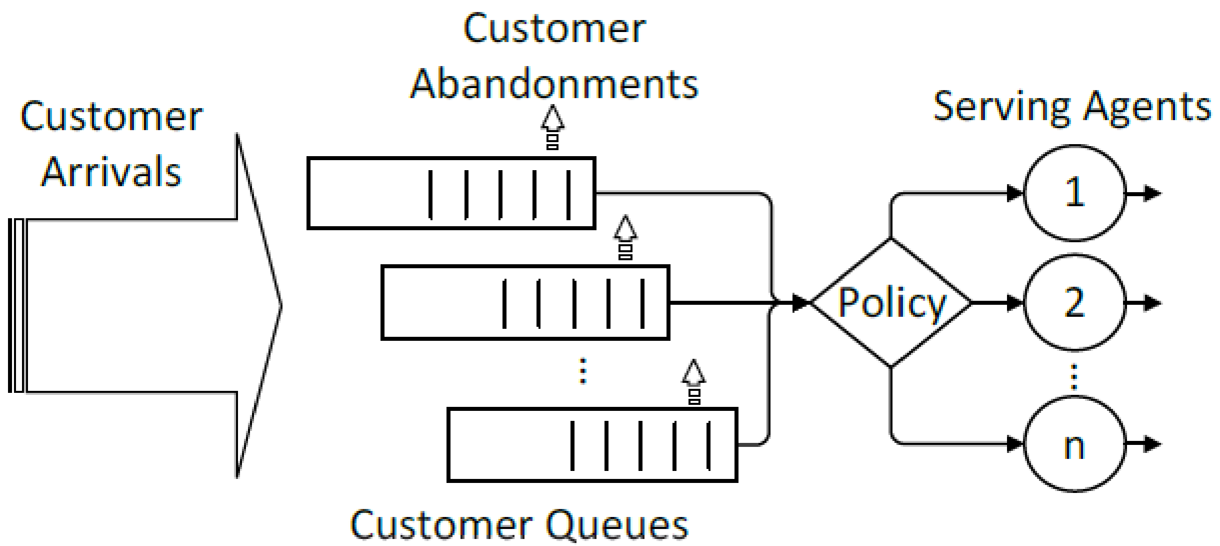
Single-Station Multi-Class Queues



Useful?

- Different classes/types of customers (VIP vs. Regular; Urgent vs. Ambulatory)
- Classes = activities (A vs. F – A gets priority)

Approach I for Multi-Class Queues

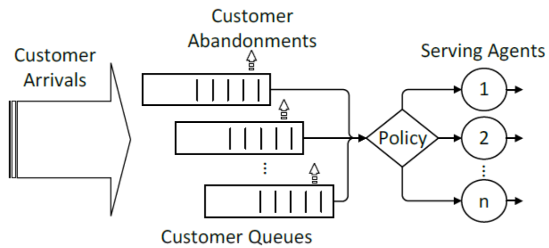


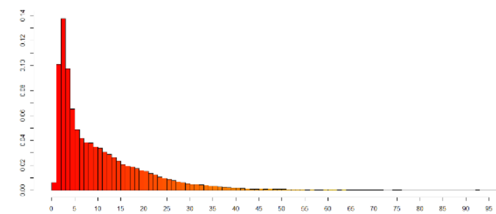
Information Systems [2014]
with Weidlich, Gal, Mandelbaum

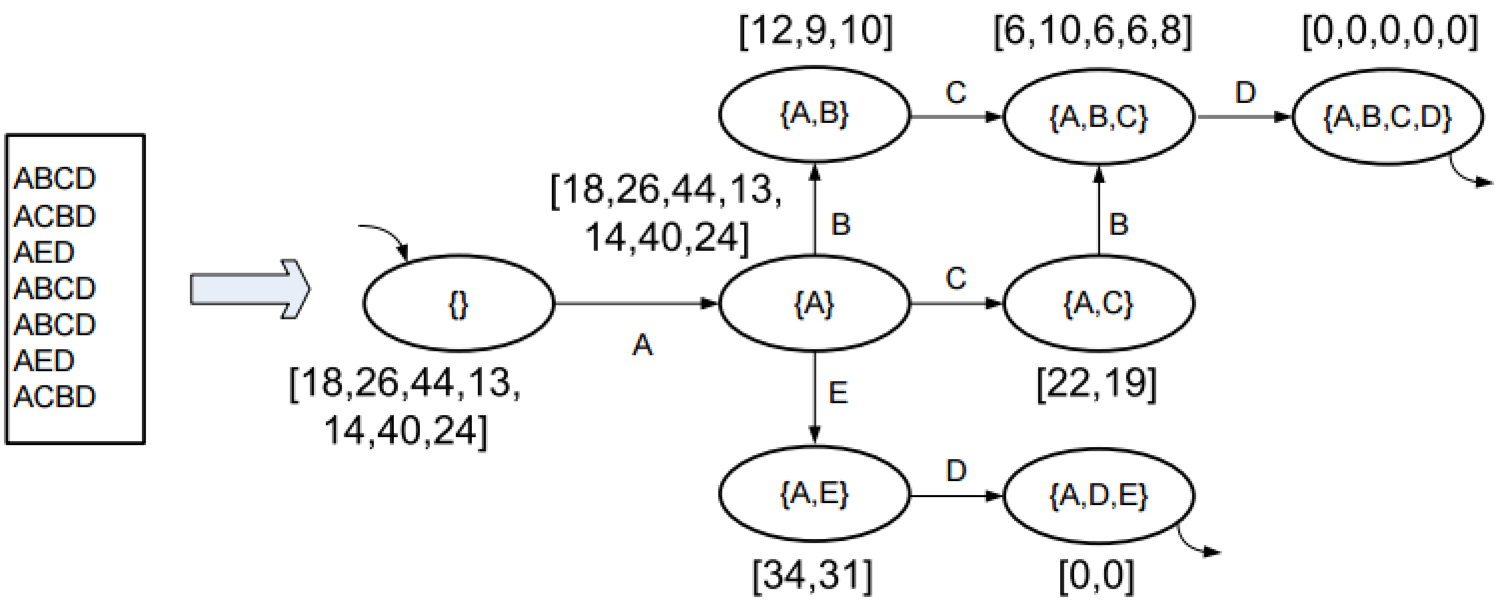
Assuming priority queues model:

- Queue length predictors – derived upper and lower bounds
- Snapshot principle (based on Reiman and Simon [1990])

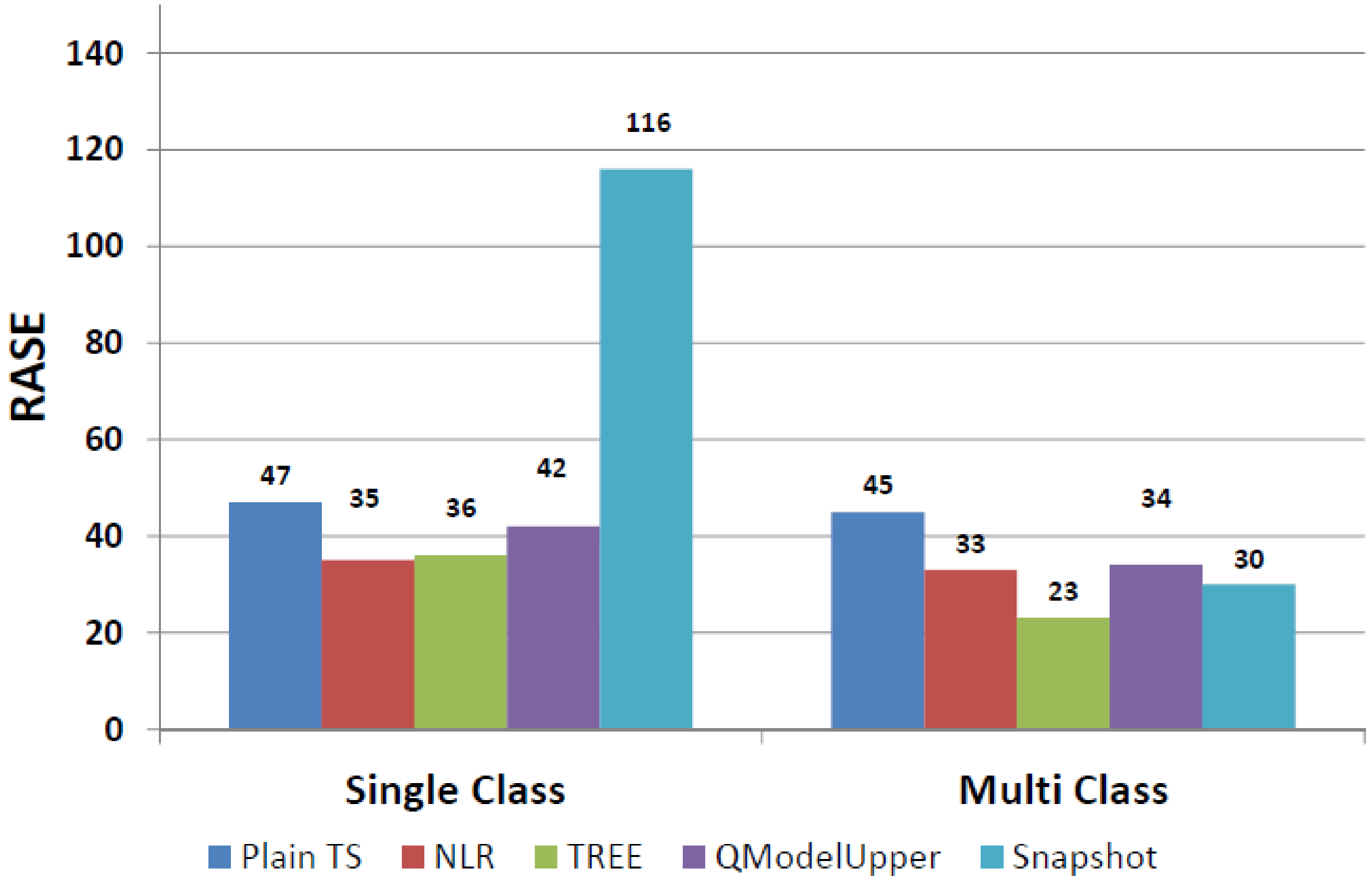
Approach II for Multi-Class Queues





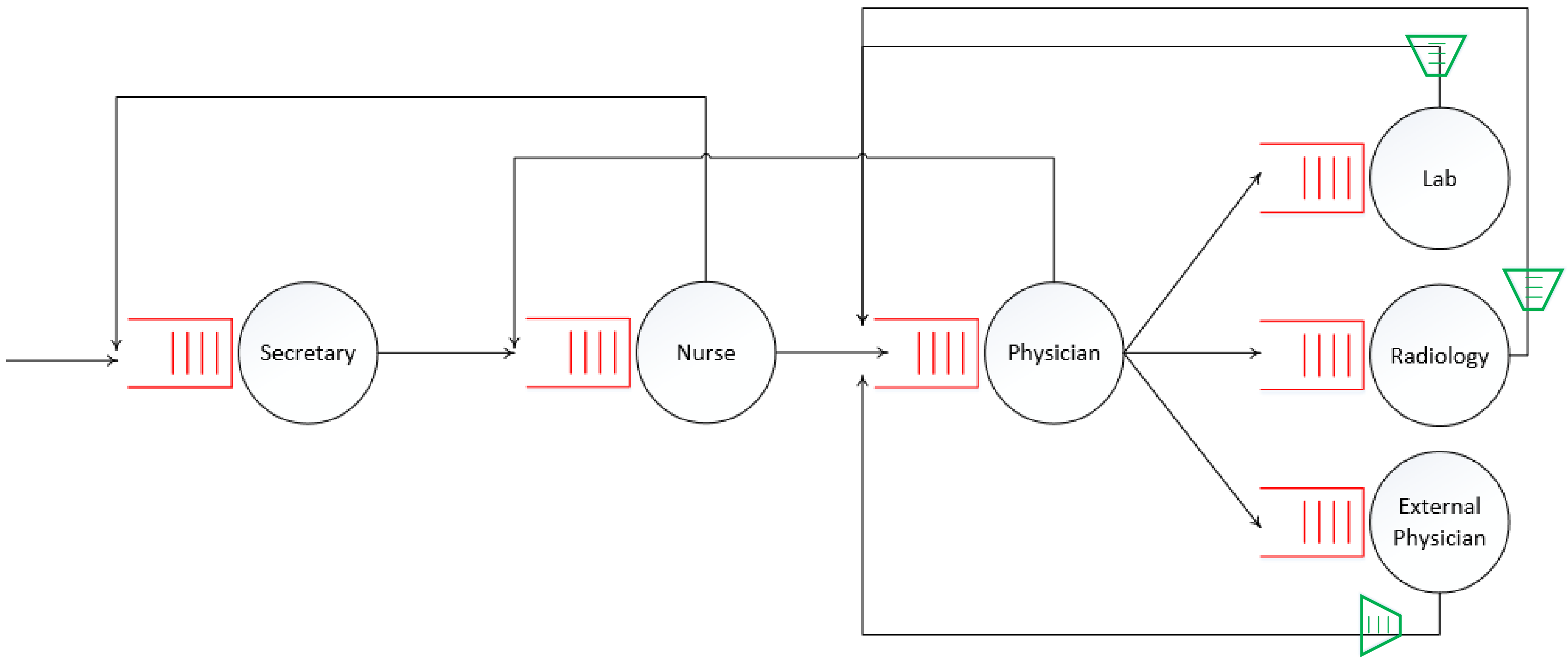


Results: Telecom Call Center Data



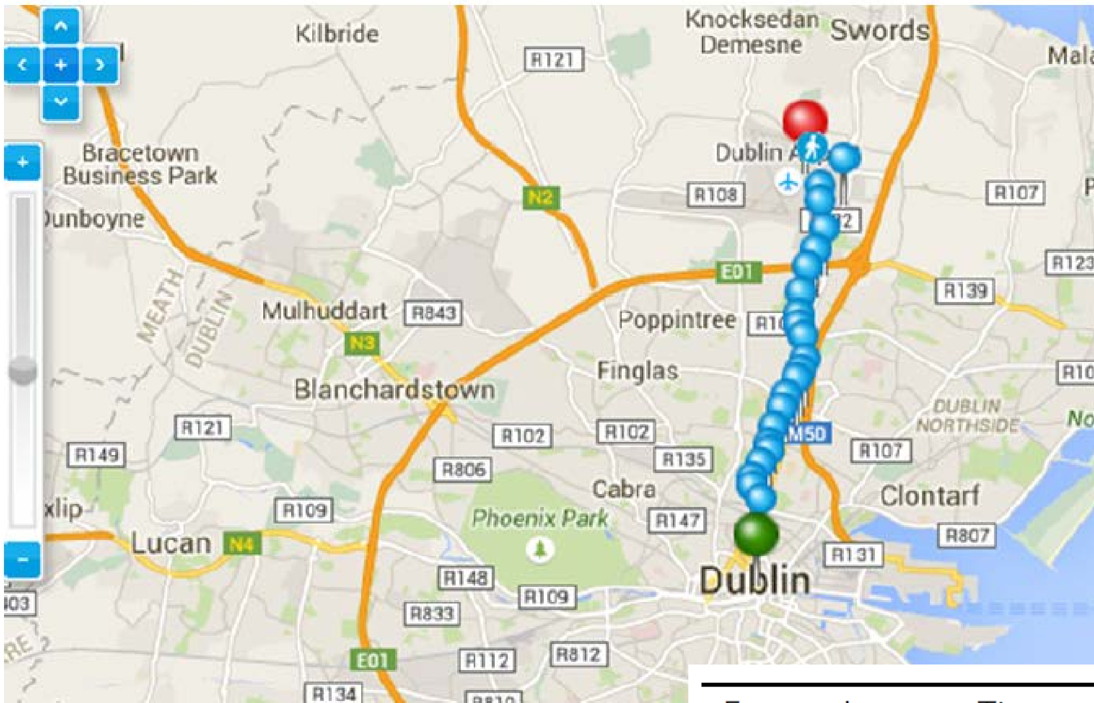
NLR, Tree – similar to De Leoni et al. [2014] (BPM14' best paper)

What about networks of queues?



Snapshot principle holds in q-networks with **pre-defined routing:** public transport, outpatient clinics,...

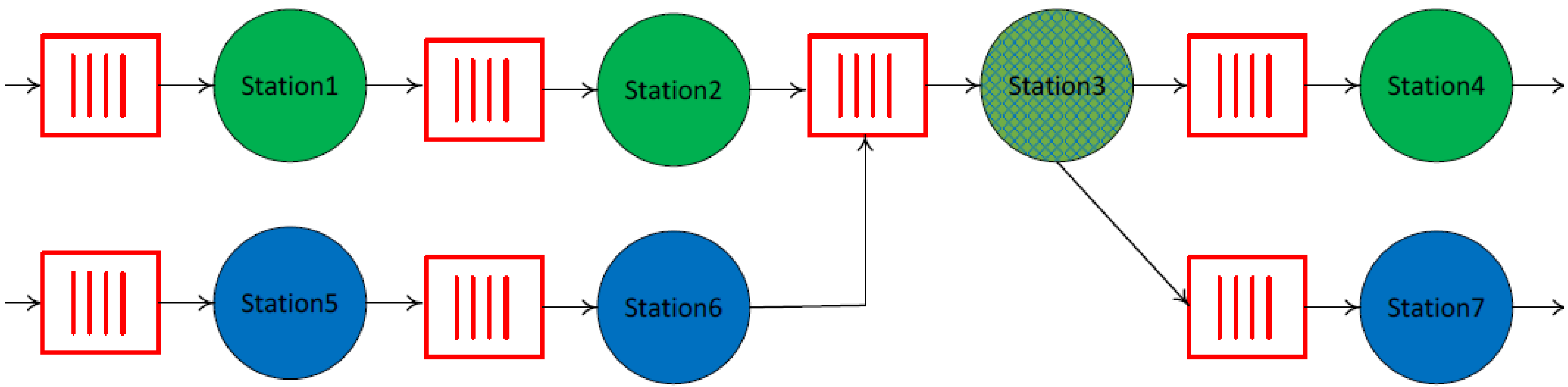
Bus Traveling Time Prediction



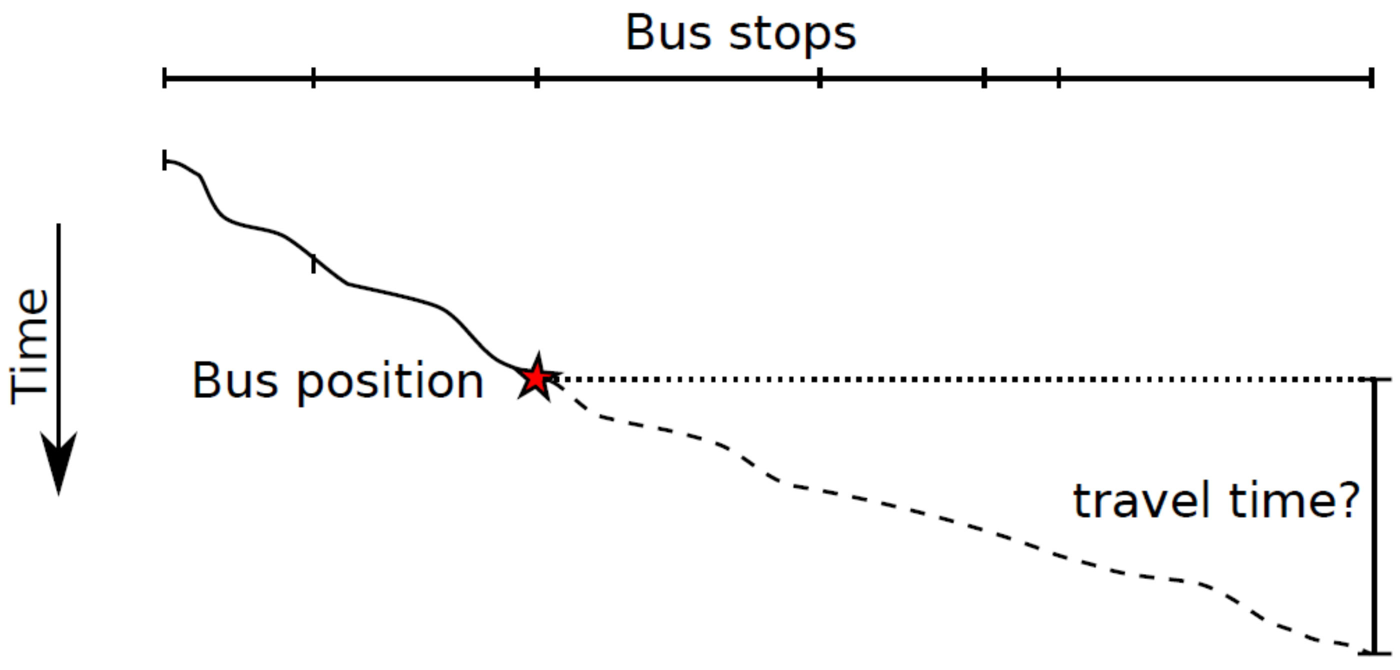
Event Id	Journey Id	Timestamp	Bus Stop	Journey Pattern
1	36006	1415687360	Leeson Street Lower (846)	046A0001
2	36012	1415687365	North Circular Road (813)	046A0001
3	36009	1415687366	Parnell Square (264)	046A0001
4	36006	1415687381	Leeson Street Lower (846)	046A0001
5	36009	1415687386	O'Connell St (6059)	046A0001
6	36012	1415687386	North Circular Road (814)	046A0001
7	36006	1415687401	Leeson Street Upper (847)	046A0001
8	36009	1415687406	O'Connell St (6059)	046A0001

Information Systems [2015]
with Weidlich, Schnitzler,
Gal, Mandelbaum

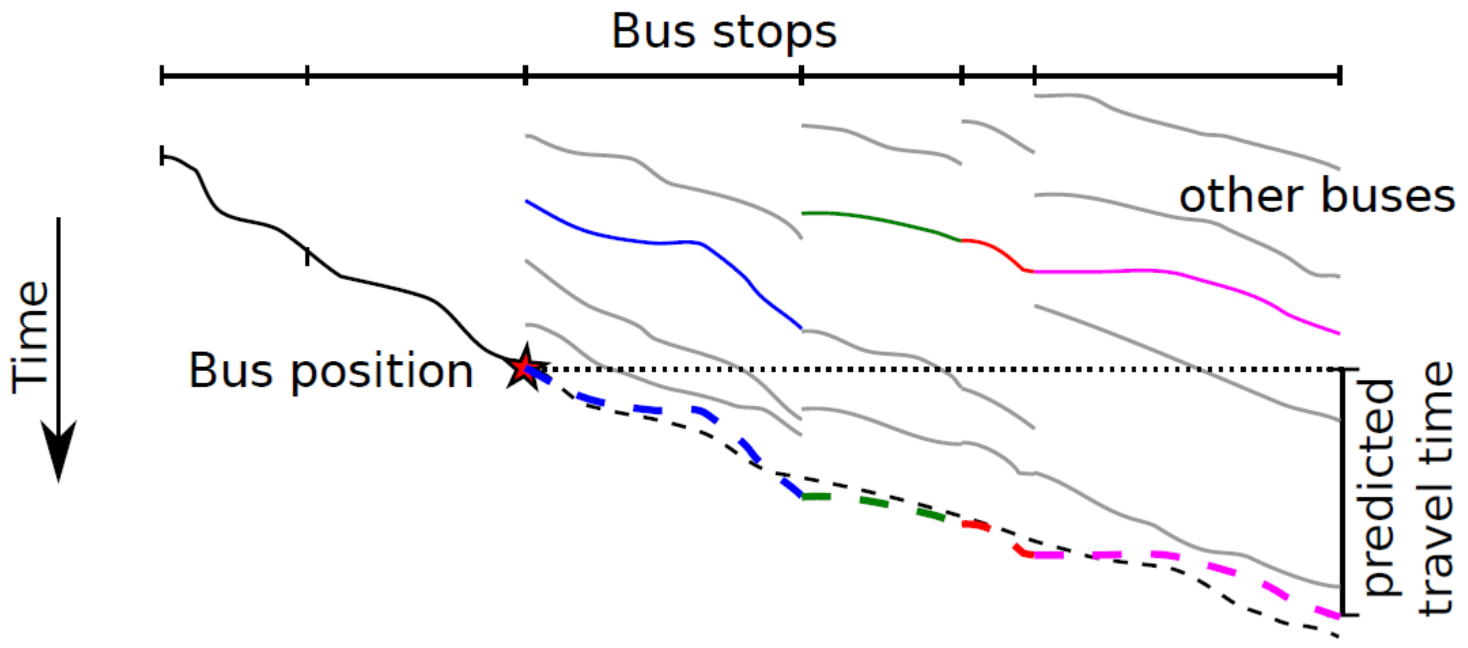
Bus Routes as Q-Networks



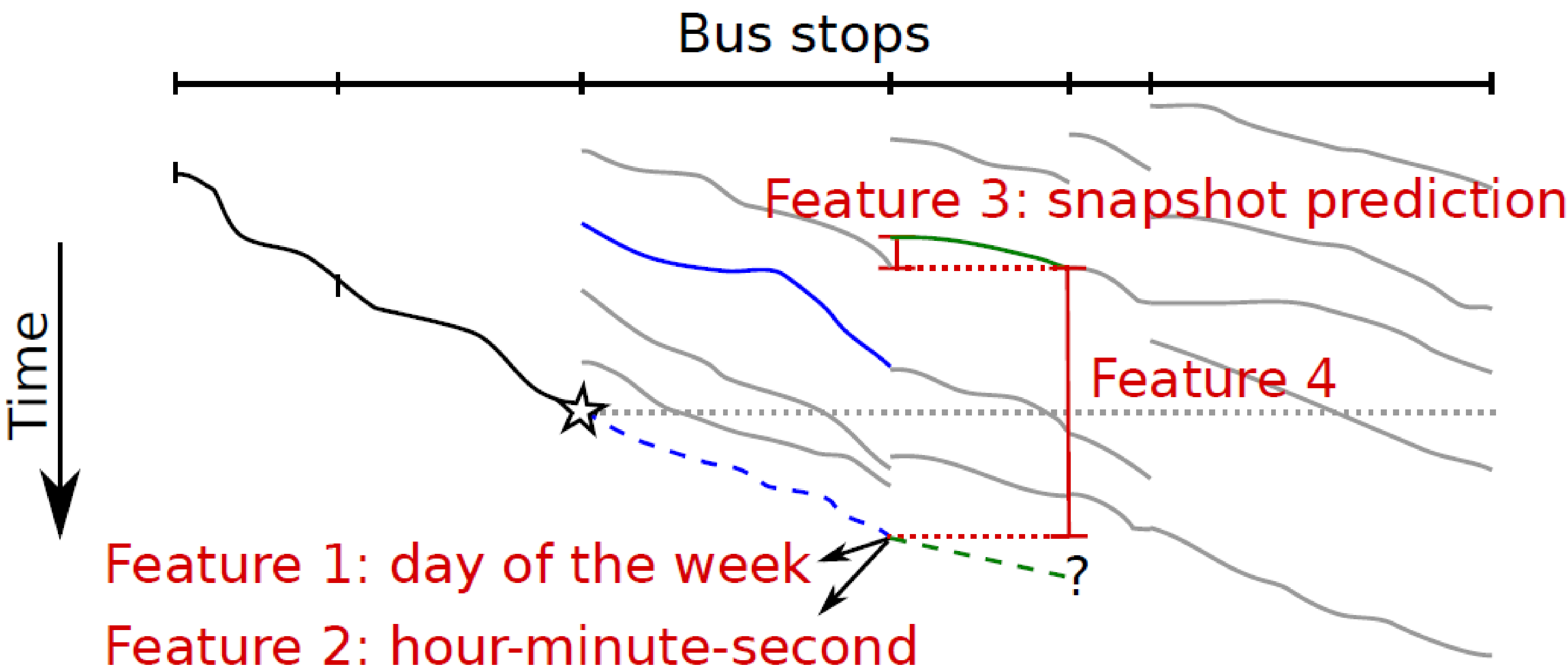
Prediction Problem



Approach I: Snapshot Principle



Approach II: Load-related + Snapshot Features



Ensemble of Regression Trees

RF random forests (bagging)

ET extremely randomized trees

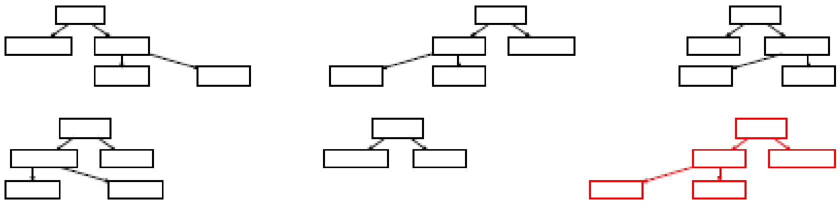
AB adaboost

GB gradient tree boosting

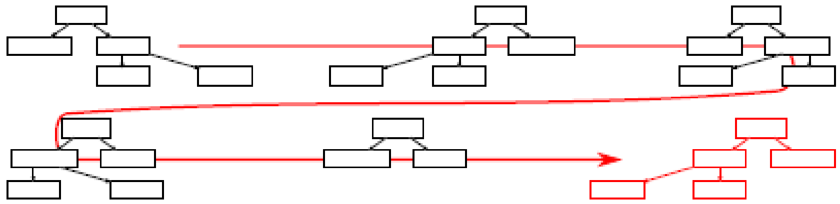
GBLAD robust gradient tree boosting

Intuition:

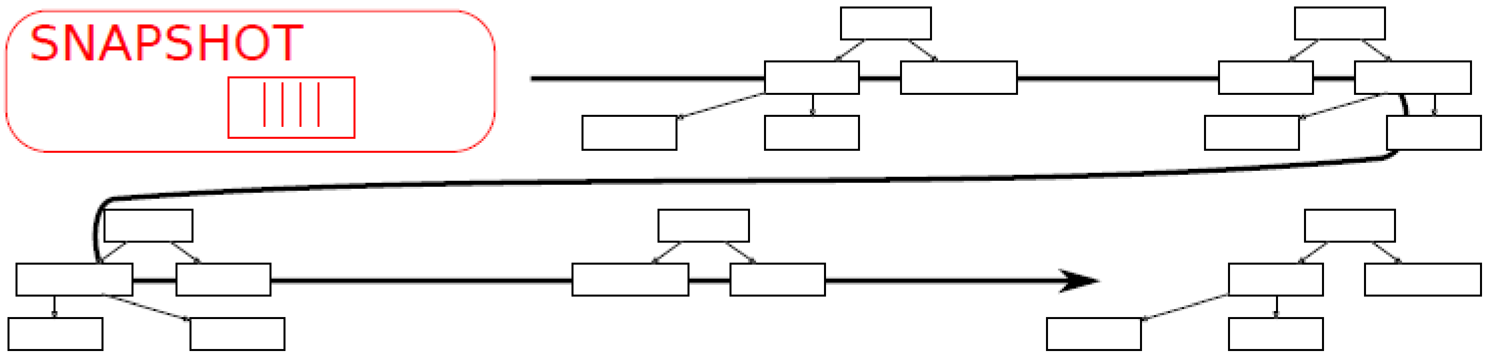
build each tree non-optimally and independently from the others



build trees sequentially, trying to add a tree that correct the flaws of the current ensemble



Boosting over the Snapshot Predictor

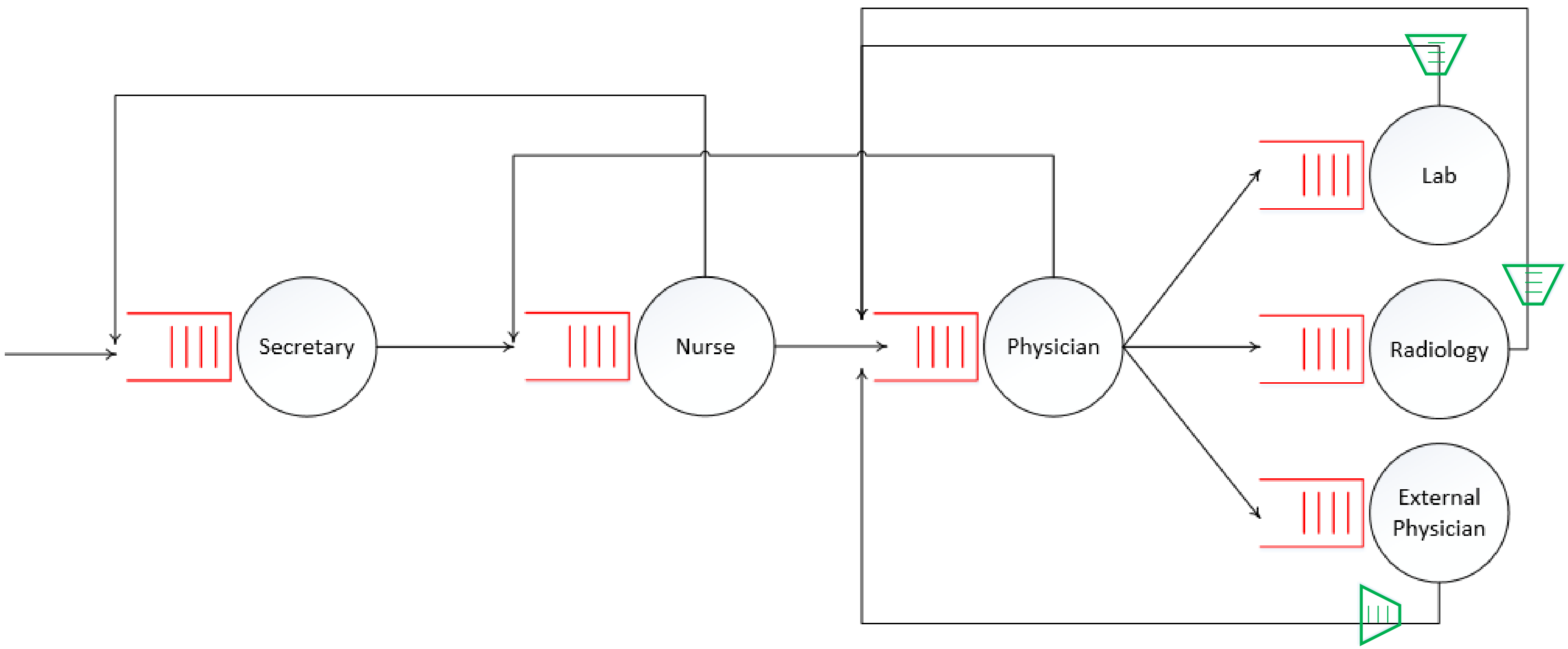


Results: Dublin Buses (GPS data)

Accuracy of the prediction over all trips. Worse, Best, **Best of S+xx and xx**

	RMSE	MARE (%)	MdARE (%)
S	539	23.37	16.15
RF	539	24.11	16.37
ET	519	22.05	15.23
AB	512	27.08	18.05
S+AB	**504**	**26.32**	**16.84**
GB	508	20.46	13.84
S+GB	494	**19.95**	13.53
GBLAD	520	19.38	13.86
S+GBLAD	**514**	19.06	**13.65**

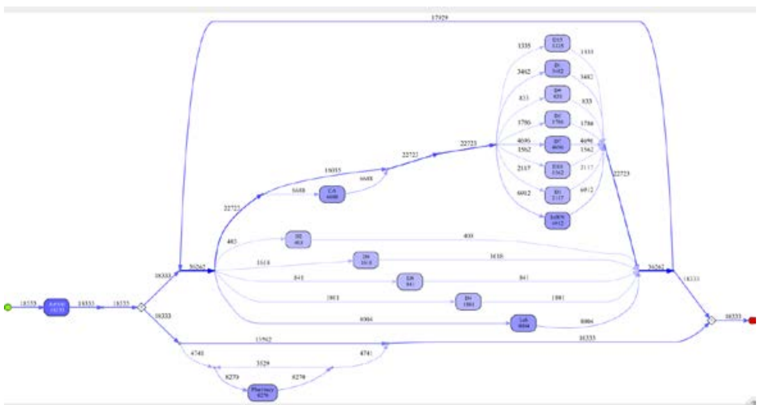
What if routing is not pre-defined?

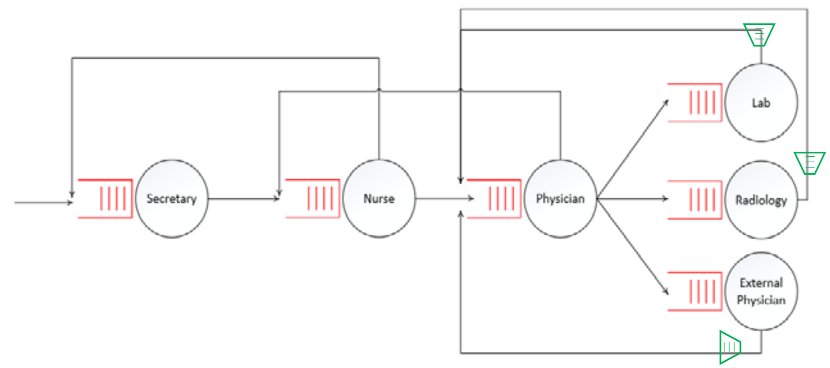


Approximation techniques, e.g. Queueing Network Analyzer (Whitt [1983]):

- Allows concurrency and non-exponential times
- Steady-state approx. (model per hour…)

Idea: PN->GSPN->QN Transformation





Four step approach:

1. Control-flow discovery (e.g., IM)
2. Enrichment (firing times, arrivals, resources,…)
3. Simplification (helps to avoid over-fitting)
4. Translation to QN for analysis (QNA)

BPM [2016] with Shleyfman, Weidlich, Gal, Mandelbaum

Conformance checking:
A Queueing Network Perspective

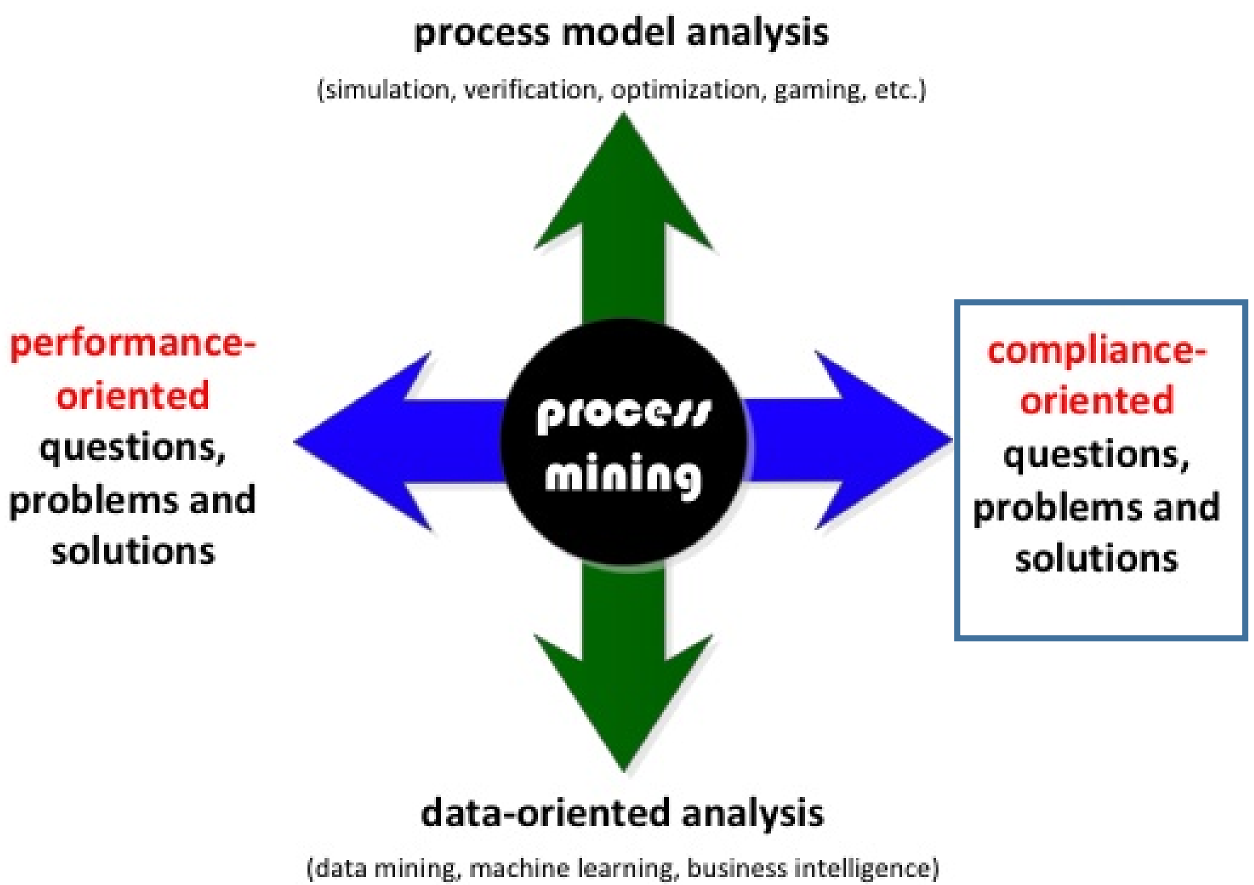
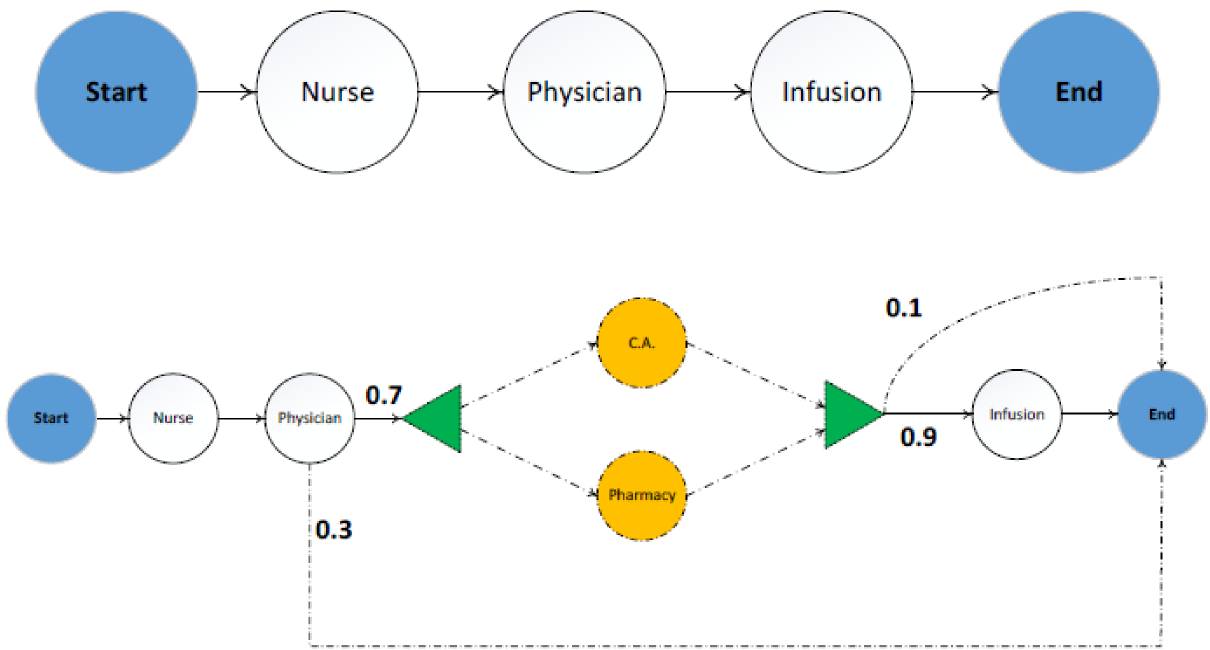


Illustration by [Wil van der Aalst]

Conformance checking: A Queueing Network Perspective



Information systems [2015] with Yedidsion, Weidlich, Gal, Mandelbaum, Kadish, Bunnel

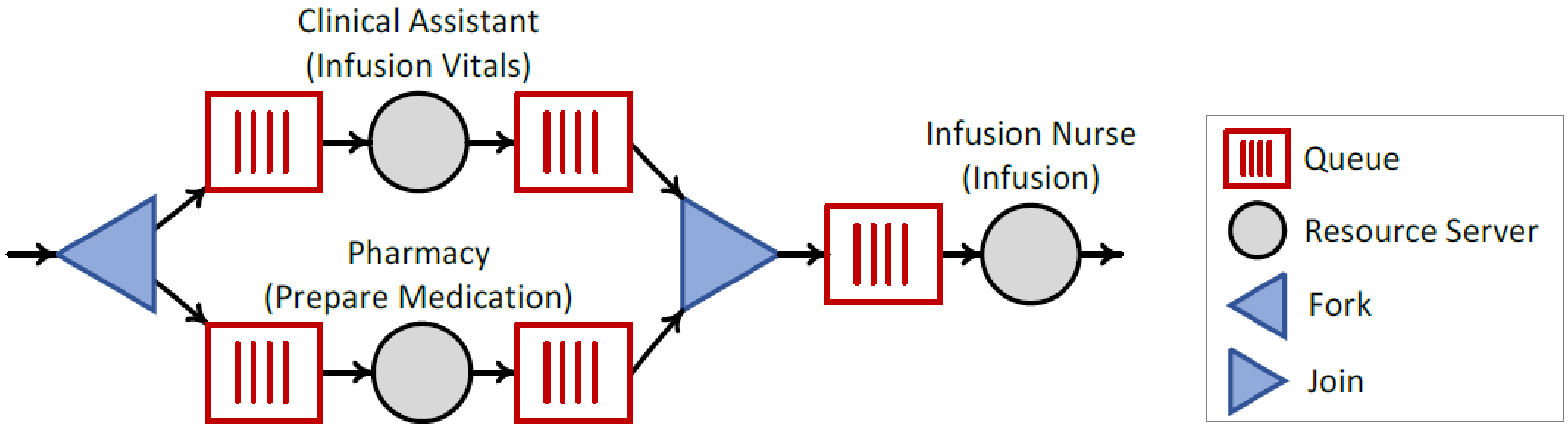
Conformance checking:
A Queueing Network Perspective

The two queueing networks are compared:

1. Detect deviations between planned and actual performance measures
2. Root-cause analysis:
   - Compare structures (unscheduled activities)
   - Building blocks (arrivals, service times,…)

Root-cause of deviations can lead to performance improvement (example is coming up)

Example: Fork-Join Construct



Step I: Unexpected Queueing

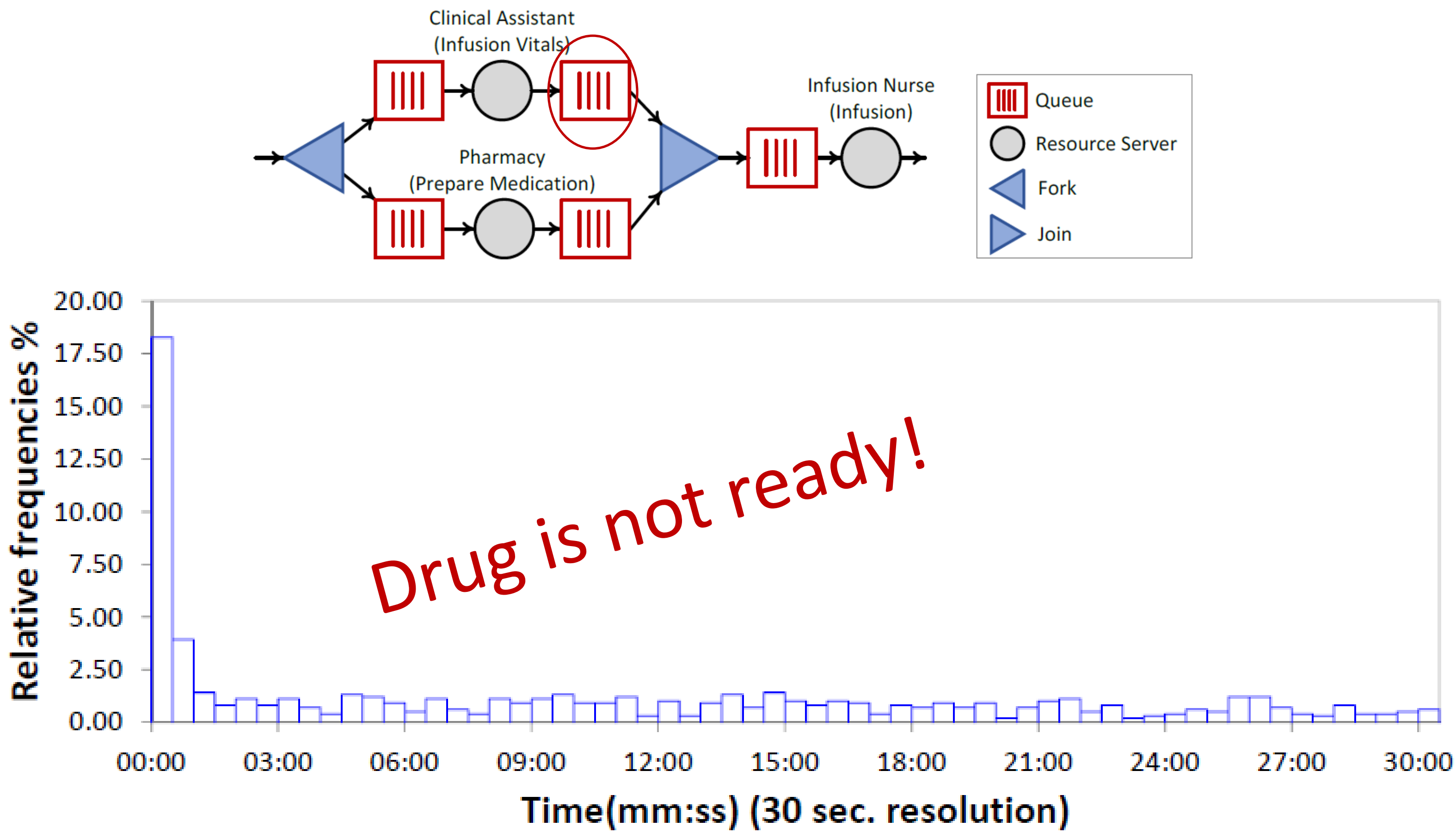
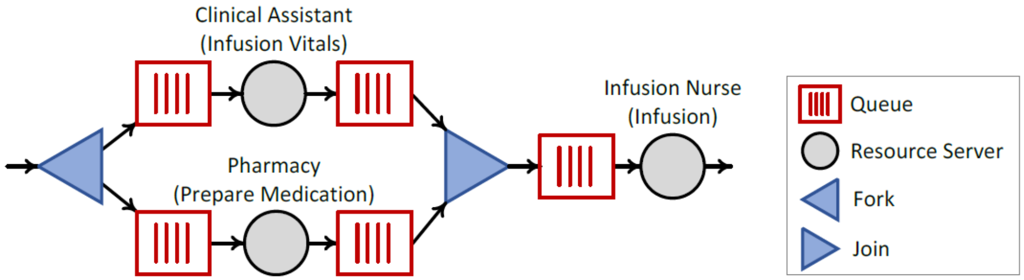


Figure 5: Waiting time for Infusion (after vitals); Sample size = 996, Mean = 25min, Stdev = 29min

Step II:
Production time is not the cause!



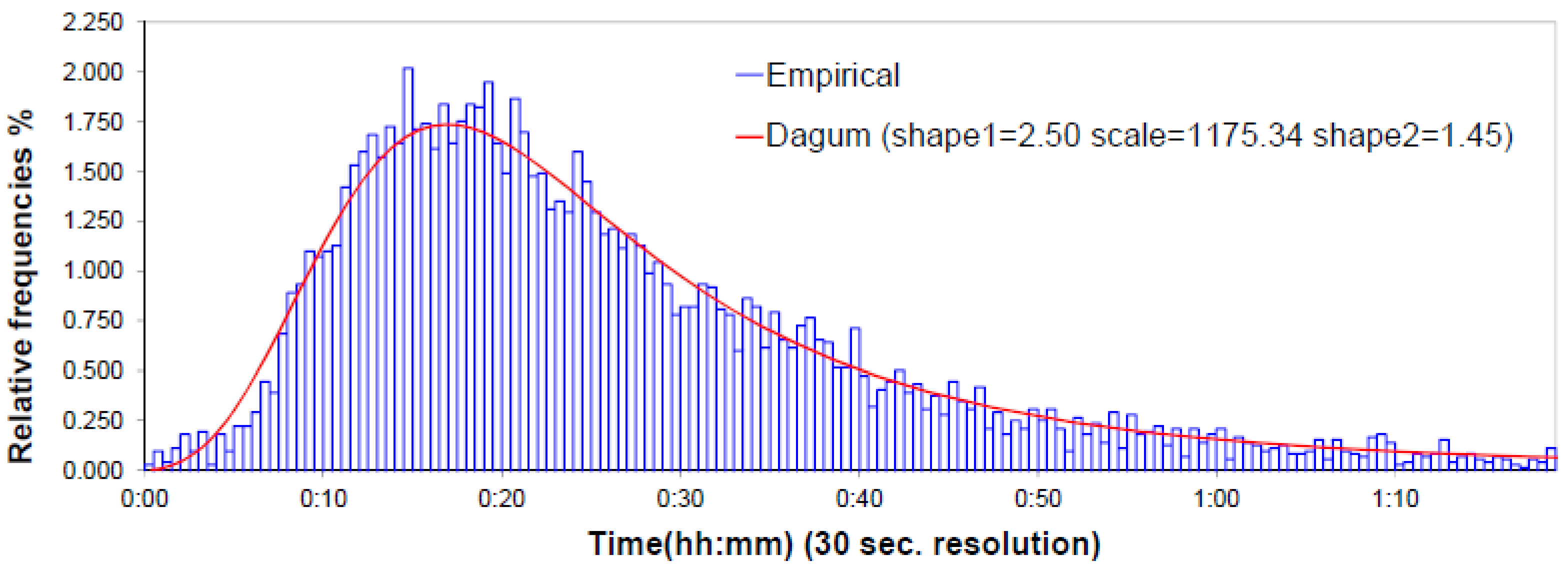
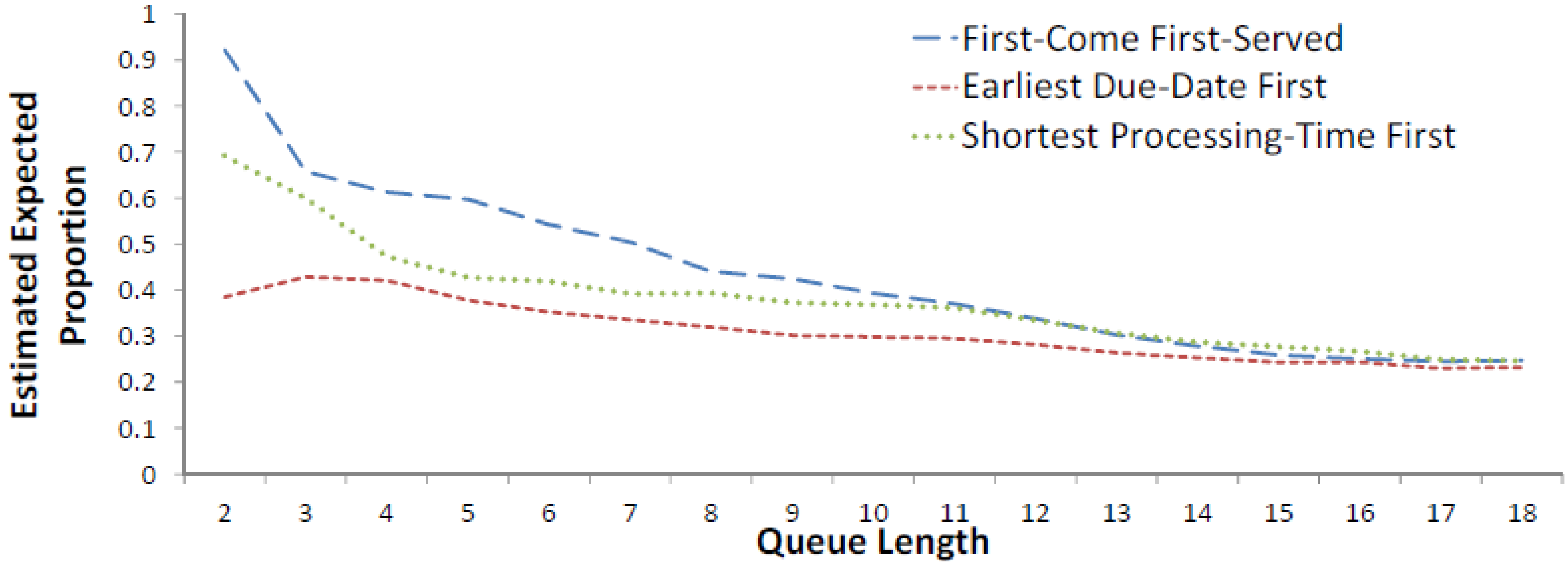
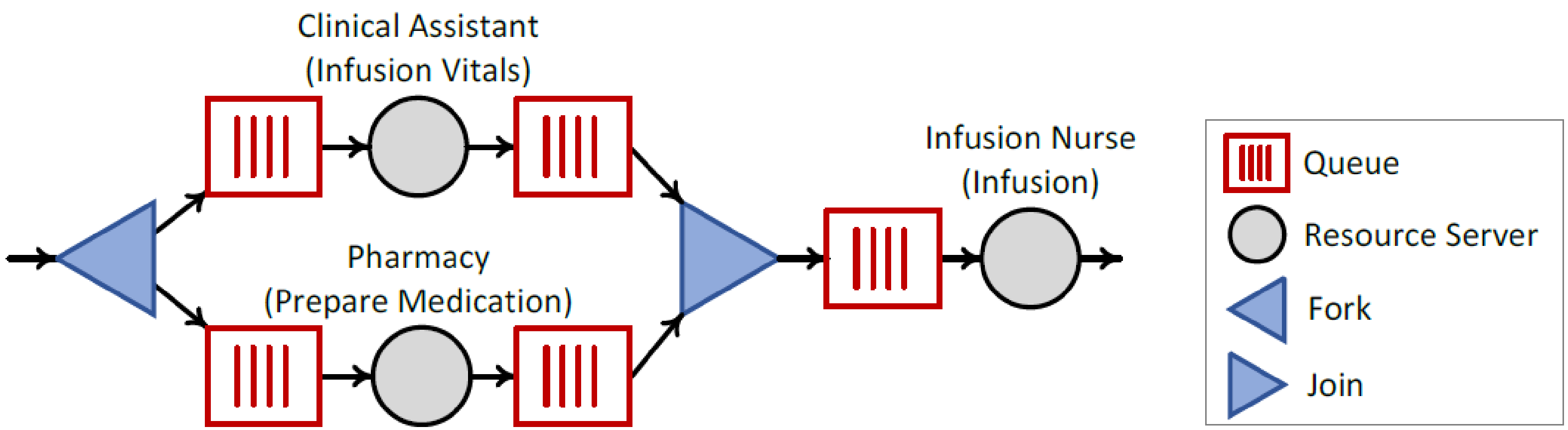


Figure 6: Medication production time; Sample size = 7187, Mean = 30min, Stdev = 24min.

Step II: Production policy is…!



Process Improvement: Idea



- New policy for sequencing “vitals” patients to reduce waiting and increase throughput
- Dominates the EDD policy – proofs and experiments in the paper

Conclusion

- Queueing models are useful for process mining
- Especially: in service processes with scarce resources
- Happy to collaborate on further integration of queueing theory into process mining

Thank you!

sariks@tx.technion.ac.il