

Layered Queueing Model Collection

Murray Woodside, Feb 2018

This note describes a collection of models made and reported over the years. They provide examples of layered system features and different kinds of analysis.

- EX1. Simple Web Service Model (layered service, software bottleneck)
- EX2. Bookstore2002 (service class interaction)
- EX3. E-Commerce2011 (moderately complex layered service system)
- EX4. Business Reporting System (components)
- EX5. Replicated Business System (replication supporting every scale of system)
- EX6. Building Security System (logical resource, bottlenecks)
- EX7. Small Distributed Database System (peer-to-peer system)
- EX8. Large Telephony Switch (bottleneck detection and movement)
- EX9. SIP Telephony System (scalability study)
- EX10Voice Packet Switch2000 (hardware layers)
- EX11 Optical Ring Failover Software (deadlines, determinism and priorities)

EX1. Simple Web Service Model (layered service, software bottleneck)

Reference: Rolia web server paper tools 97

This example was used to demonstrate software bottlenecks in a simple way.

See the [powerpoint slides](#), [files](#)

EX2. Bookstore2002 (service class interaction)

Ref WOSP 2002 paper, Dorin Petriu MASc thesis

This model was used for many purposes around 2002; Dorin explained deriving a LQN model from requirements scenarios, and examined some issues for contention between two very different classes.

See the powerpoint slides

EX3. E-Commerce2011 (moderately complex layered service system)

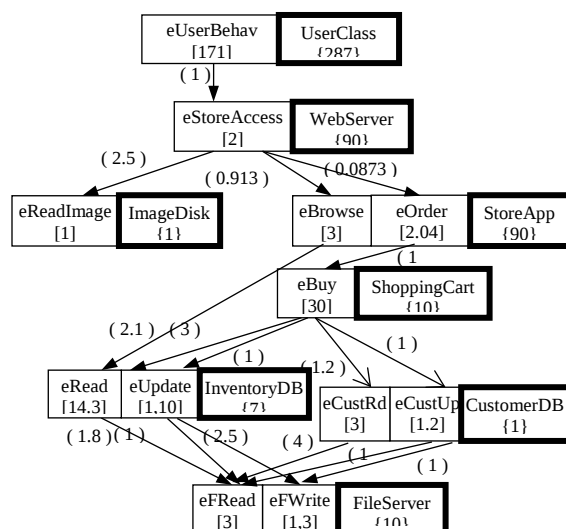
Ref Jim Li PhD thesis 2011, paper?

E-Commerce example used in the paper:

“CloudOpt: Multi-Goal Optimization of Application Deployments across a Cloud” by Jim ZW Li, Murray Woodside, John Chinneck, Marin Litoiu, Proc. 7th Int. Conf on Network and Service Management, Paris, Oct 2011,

which deals with optimal deployment with replicated (scaled-out) servers in a cloud. Jim Li used this model architecture as a template to create different applications (same structure, different parameters) to study co-locating many applications in a cloud.

Diagram not showing the processors:



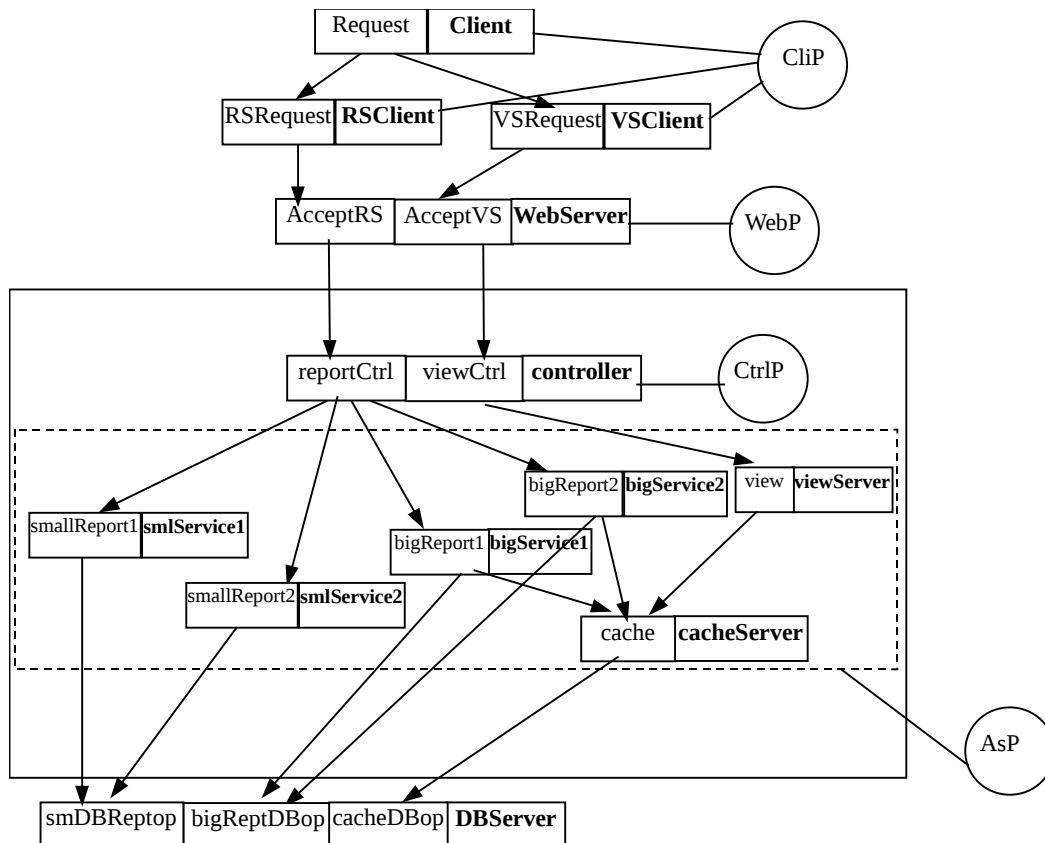
The model file ex3eCommerce2011.xlqn uses SPEX commands to generate models with random parameters, and solves them.

EX4 Business Reporting System

This example was published in:

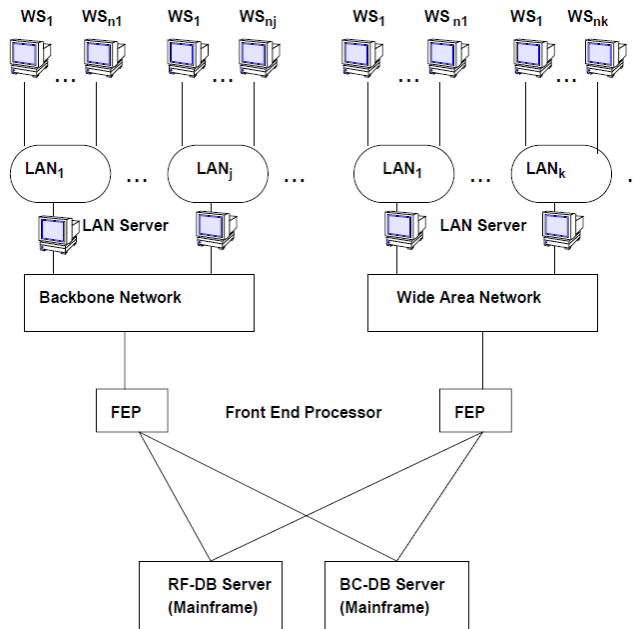
Xiuping Wu and Murray Woodside, "Performance Modeling from Software Components," in Proc. 4th Int. Workshop on Software and Performance (WOSP 04), Redwood Shores, Calif., Jan 2004, pp. 290-301.

It was used as an example of component-based model-building. It has been used and adapted by other authors, e.g. Anne Koziol et al.



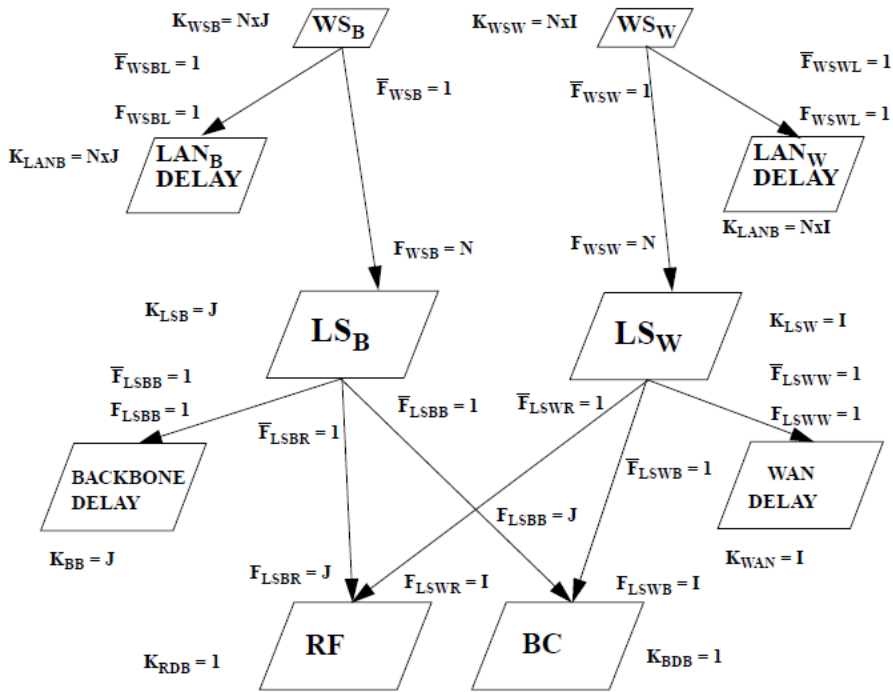
EX5 Replicated Models: Business Client-Server System

This model represents a data-entry system of a large company, feeding two databases. It was used as a key example in Amy Pan's thesis Solving Stochastic Rendezvous Networks of Large Client-Server Systems with Symmetric Replication, by Amy M. Pan, 2001, and in the thesis by Al-Omari ref below.



System layout.

Replication is fully symmetric, so each replicated task has exactly the same environment and has caller/called relationships either to the same task, or to a replica of the same task. For any call, fan-out is the number of replicas of the called task, that are called; fan-in is the number of replicated callers, that make the call.



WS - Workstation
 LS - LAN Server
 I - Number of LANs on WAN
 J - Number of LANs on Backbone
 N - Number of Workstations per LAN

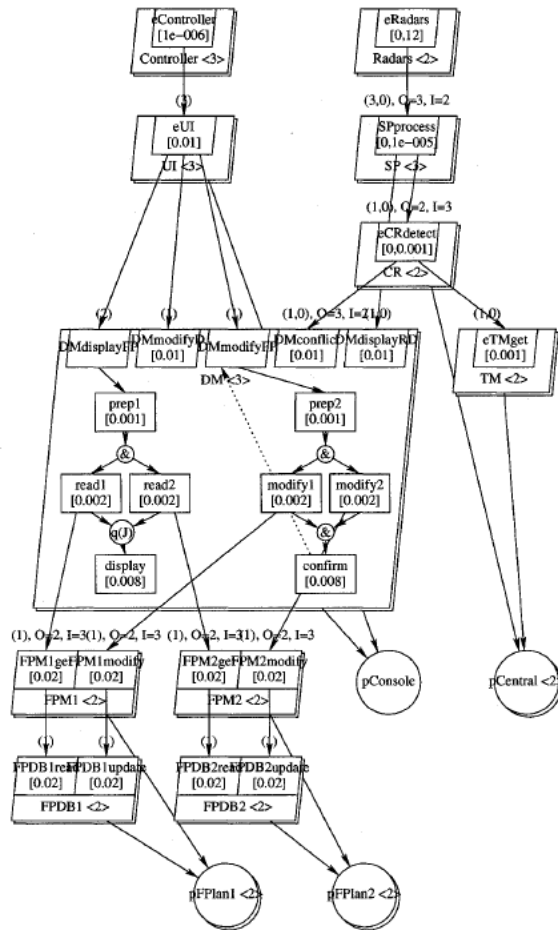
LQN Diagram: K = replication level, $\bar{F}$ = fanout, F = fanin of calls

Service Time of LAN Server for Local Service	= 0.03 seconds
Service Time of LAN Server for Database Service	= 0.01 seconds
Service Time of RF Database Light Transactions	= 0.04 seconds
Service Time of RF Database Heavy Transactions	= 0.08 seconds
Service Time of BC Database Light Transactions	= 0.08 seconds
Service Time of BC Database Heavy Transactions	= 0.16 seconds
Service Time of LAN delay	= 0.00039 seconds
Service Time of WAN delay	= 0.04 seconds
Service Time of Backbone delay	= 0.02 seconds

Demand parameters of the model

Second example of replication: Air Traffic Control System

This model describes an air traffic control system with replication and quorum consensus for reliability, taken from Performance Modeling of Replication Techniques in Parallel and Distributed Layered Service Architectures by Tariq Al-Omari, PhD thesis, 2007, Chapter 8.



EX6 Building Security System (logical resource, bottlenecks)

REf Jing Xu Tools 2003 paper

See the powerpoint for a discussion.

EX7 Small Distributed Database System (peer-to-peer system)

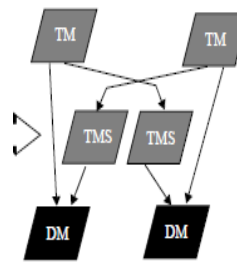
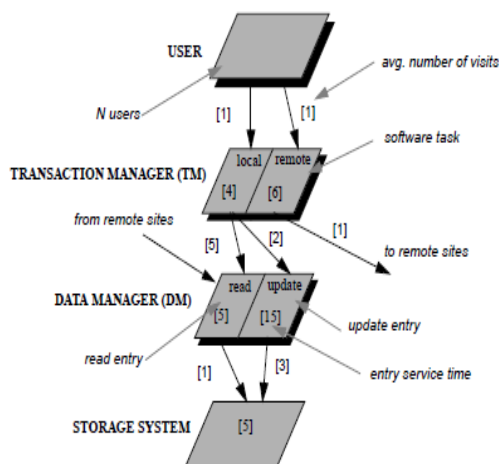
The value of this model is it shows how to model symmetric peer-to-peer systems. Basically each party runs an agent for the remote parties, to accept and manage their operations. Here the agent is called TMS and does data manager transactions as a proxy for remote parties.

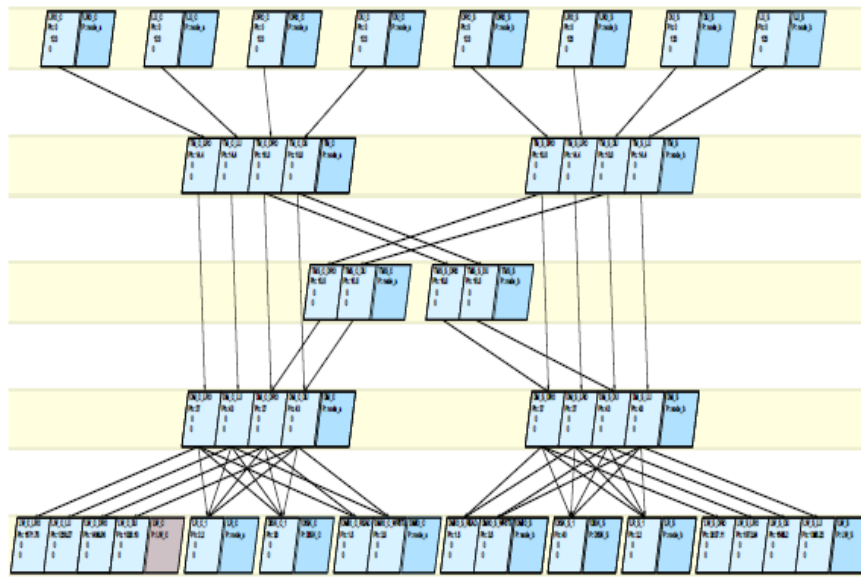
The actual system is from long ago, the work describing it is referenced in Sheikh's work.

REf Fahim Sheikh MEng thesis, and:

F. Sheikh and C.M. Woodside, "Layered Analytic Performance Modelling of a Distributed Database System", Proc. 1997 International Conf. on Distributed Computing Systems, May 1997, pp. 482-490.

Here is the single-node version:





User classes generate transactions
(read local/remote, r/w local/remote)

Transaction managers for 4 classes

TMS handles remote transactions

Data managers for 4 classes

Storage managers for 4 classes

LQN structure Site A on left, B on right

EX8 Large Telephony Switch (bottleneck detection and movement)

Ref Peter Tregunno MASc thesis, and:

G. Franks, D. Petriu, M. Woodside, J. Xu, P. Tregunno, "Layered bottlenecks and their mitigation", Proc of 3rd Int. Conference on Quantitative Evaluation of Systems QEST'2006, pp. 103-114, Riverside, CA, USA, Sept. 2006

Peter studied simple ways to identify bottlenecks, and did a case study of an Alcatel switch architecture. The paper describes a series of improvements that shifted and relaxed the bottleneck.

The system architecture:

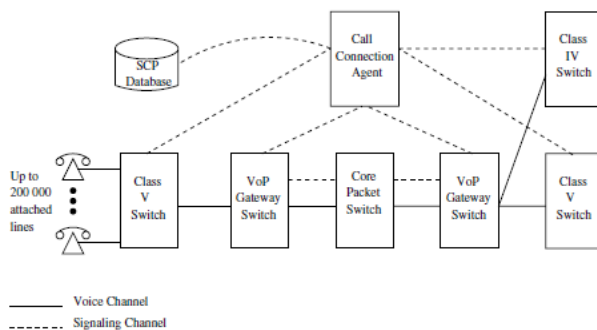


Figure 7. Abstract view of a telephone network

The LQN in schematic form

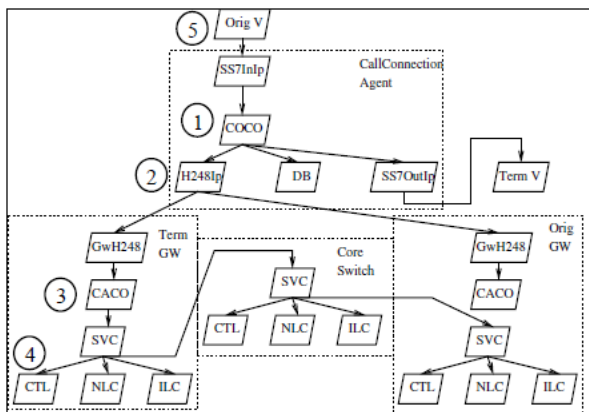


Figure 8. Resource Graph for the Switch. Each node shown has an associated processor node

EX9 SIP Telephony System (scalability study)

Ref Jogalekar PhD thesis, paper: Prasad Jogalekar, Murray Woodside, "Evaluating the Scalability of Distributed Systems", IEEE Trans. on Parallel and Distributed Systems, v 11 n 6 pp 589-603, June 2000

This model represents an experimental Nortel design for a telephony system based on PCs and an ATM switch. The model is basically just a queueing network, as all the tasks are infinite multiplicity. The model was used to experiment with a scalability metric based on cost-effective scaling out. A scaling path or plan was defined and then systems along that path were evaluated for cost (based on number of components) and performance. The scale of 120 model had 75000 users, indicating the power of mean value analysis in large systems.

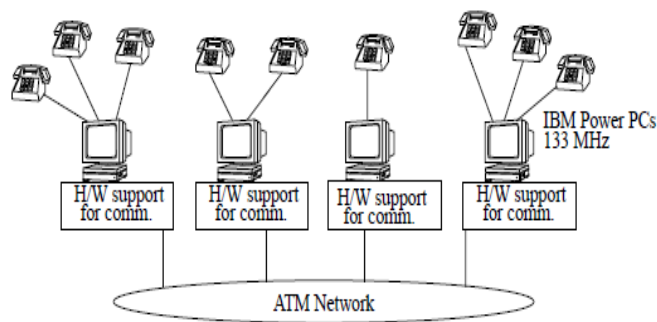
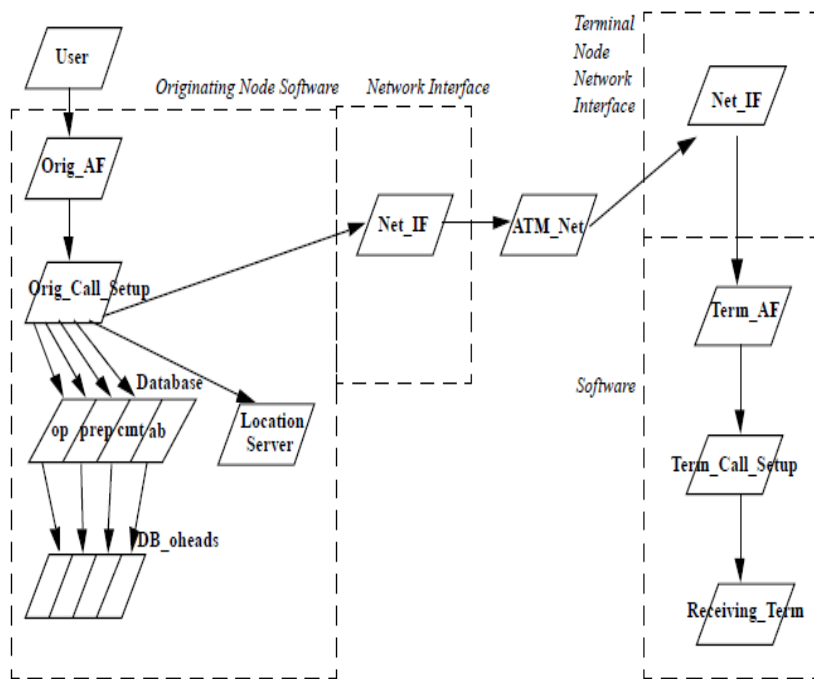


Figure 6: Overview of the call-processing system's hardware infrastructure



Layered model showing one replica per task.

E10 Voice and Data Packet Switch2000 (hardware layers)

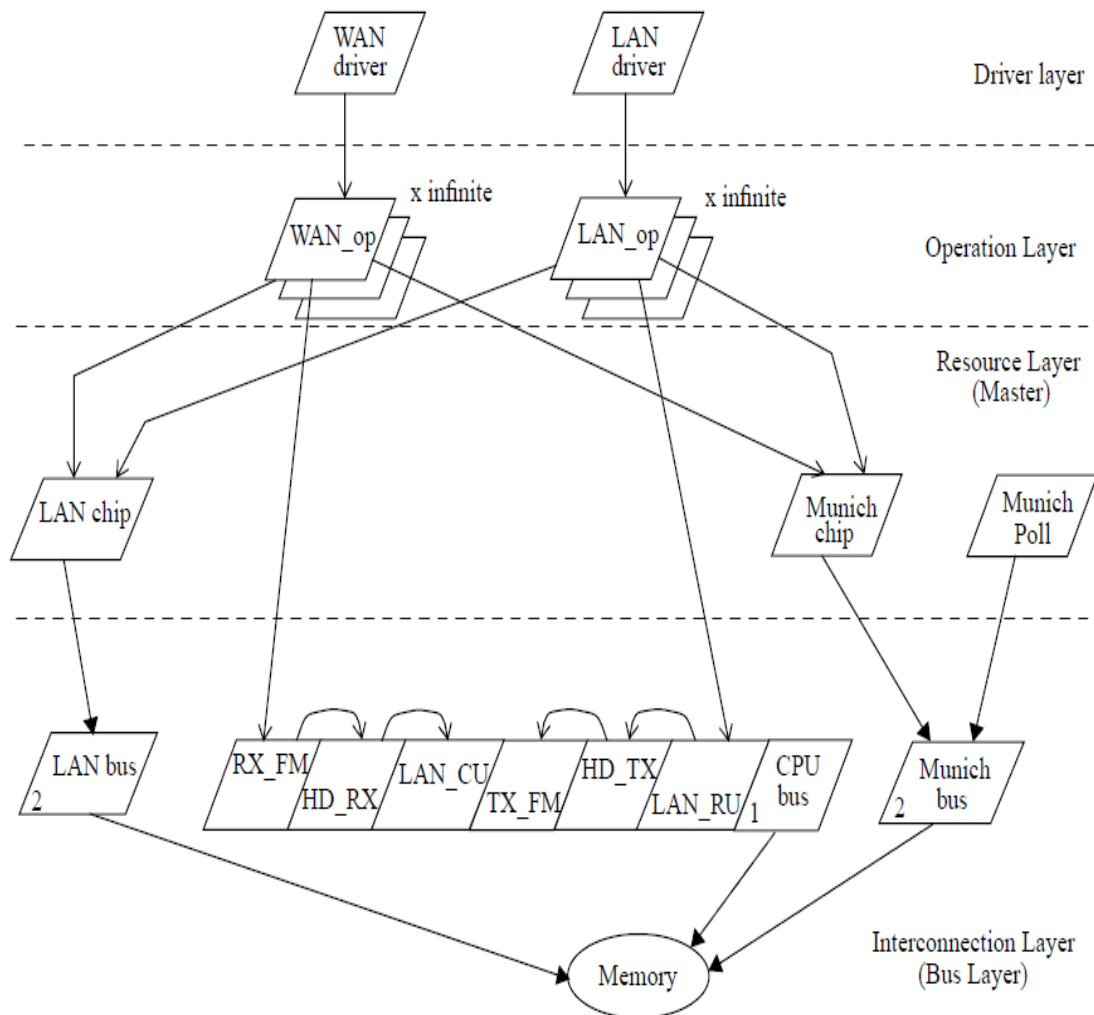
REf Peter Maly Masters thesis, paper:

P. Maly, C.M. Woodside, "Layered Modeling of Hardware and Software, with Application to a LAN Extension Router", Proc 11th Int Conf on Computer Performance Evaluation Techniques and Tools (TOOLS 2000), Chicago, March 2000, pp 10-24.

The model is for a small office-scale ISDN switch with a WAN and LAN backend connection for internet data. The so-called Munich chips interface to the ISDN (telephone line voice + data) connections.

The interesting thing about this model is that the processor bus is a resource modeled as a task. The bus uses a polling discipline which has to be simulated.

FIGURE 39. A modified model of the RLAN with the new extensions



EX11 Optical Ring Failover Software (deadlines, determinism and priorities)

Ref Hesham PhD thesis, Sigmetrics paper

Hesham successfully used priorities to achieve deadlines, using LQN

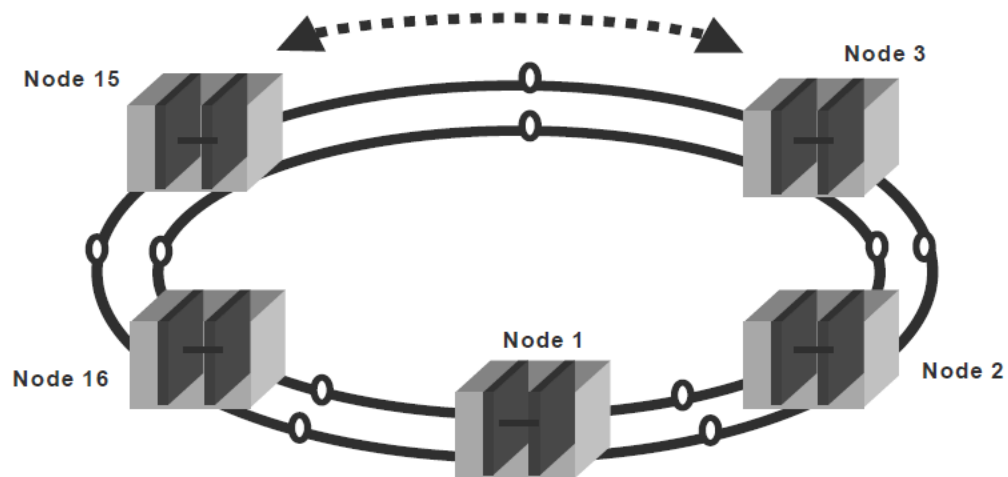


Figure 6.1 A 16-node 2-fiber BLSR network

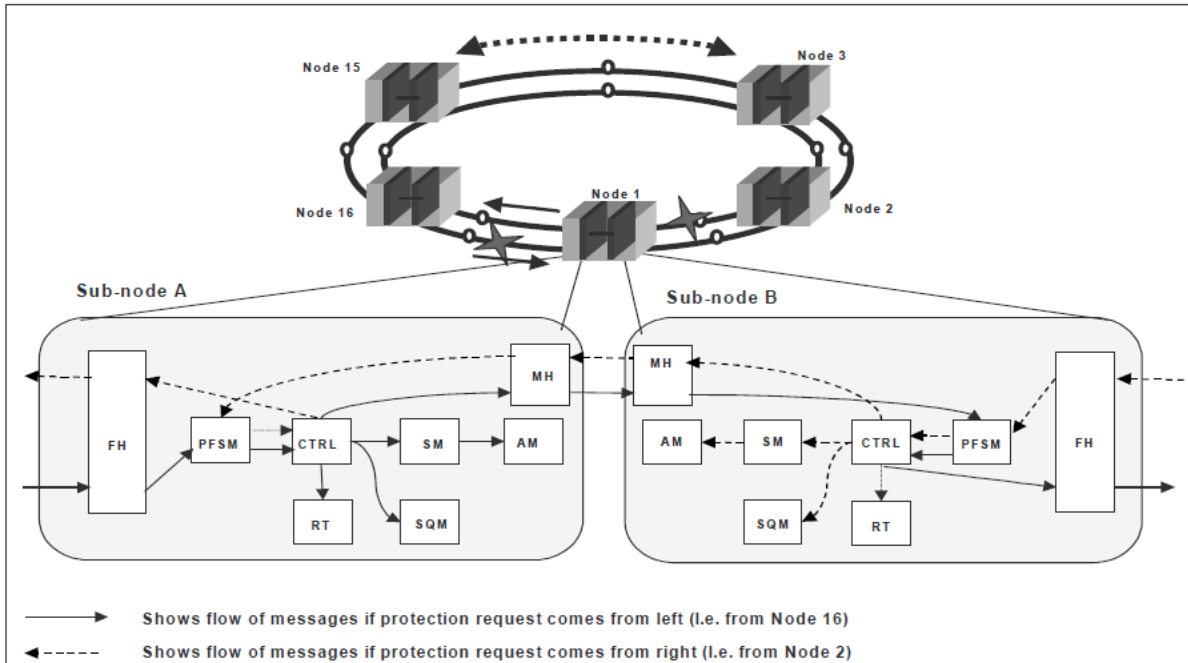


Figure 6.2 Software Architecture of The Automatic Protection Switching System

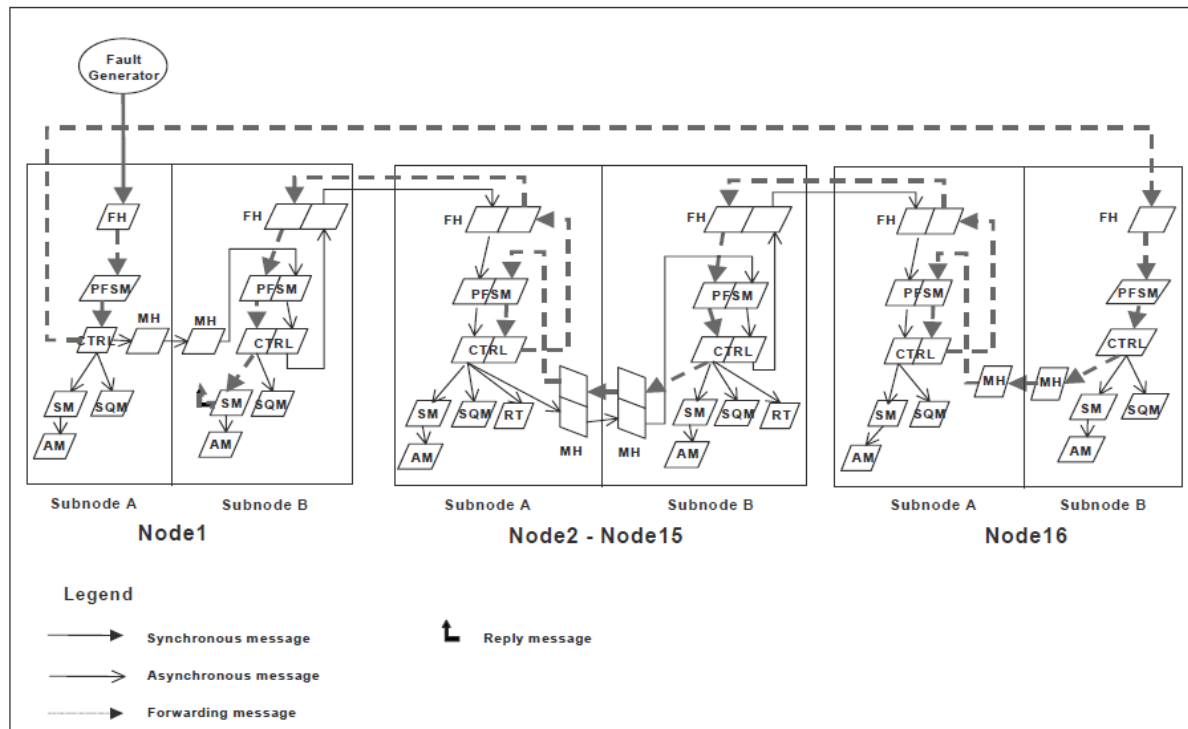


Figure 6.6 LQN model of the automatic protection switching system