Real-Time Database Systems

(Are They Real ?)

Jayant Haritsa

Database Systems Lab
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore 560012, India

{haritsa}@dsl.serc.iisc.ernet.in

Talk Organization

- What is an RTDBS?
- Where is it useful?
- Why is it challenging?
- How are the problems solved?
- What have been the achievements?
- Top 10 Research Problems

Real-Time Database Systems

- **Transactions have timing constraints**

Examples:

Sell all my Bofors stock before 4 pm

Register my auction bid within the next 5 minutes

Submit my workshop presentation before midnight, May 3, 1999

Objective

- **Meet transaction deadlines**

Origin of Time Constraints

- **Defined by the User**
  (QOS in Multimedia DB)

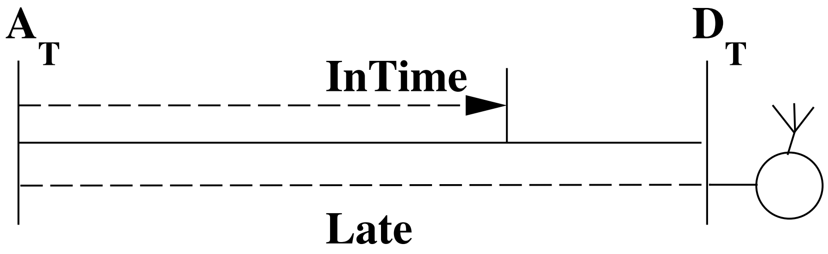
- **Required by the Client's Environment**
  (Battery in Mobile Computing)

- **Enforced by the System**
  (Flash Auctions on the Web)

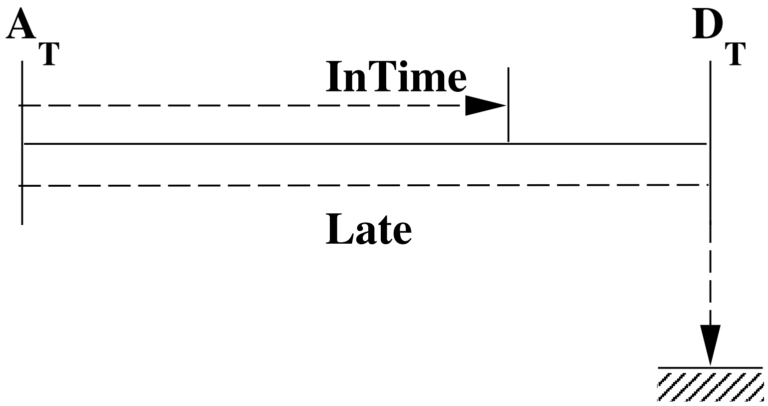
- **Derived from External World**
  (Sensor Value validity in
  Control Systems)

Types of Deadlines

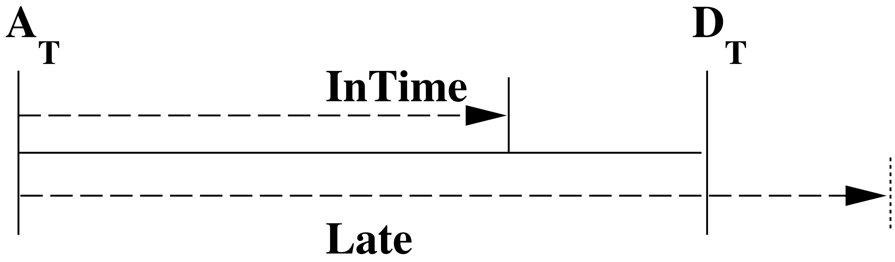
HARD



FIRM



SOFT



Application Areas

- **Financial Markets**
- **Electronic Commerce**
- **Multimedia Systems**
- **Manufacturing Processes**
- **Military Systems**
- **Safety-Critical Systems**

Integrity Constraints **Time Constraints**

Database Systems **Real-Time Systems**

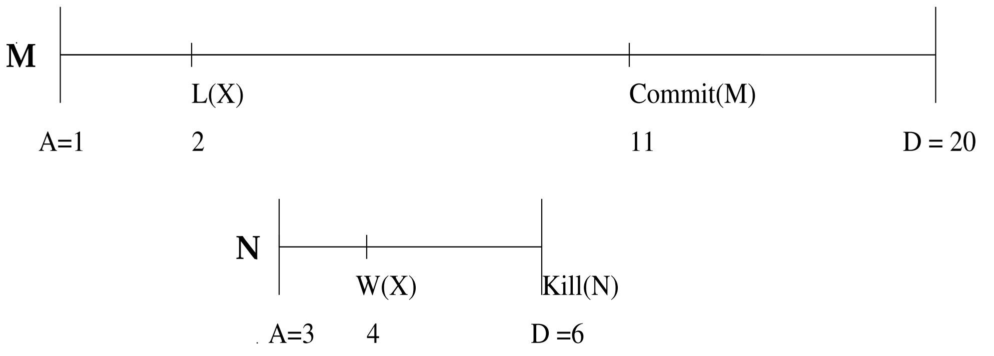
Real-Time Database Systems

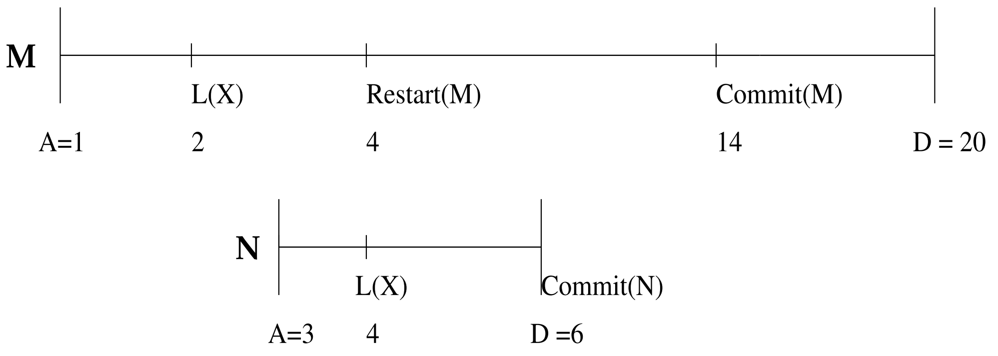
But That’s Easy !

- **Real-Time is equivalent to Fast (Put in better hardware, more efficient protocols, etc.)**
- **Make the database memory-resident**
- **Problems have been solved already in temporal database research**
- **Somebody must have done it already ! (RT and DB technology have been around for decades)**

Real-Time Example

Transaction M: Arrival at 1, Execution of 10, Deadline at 20
Transaction N: Arrival at 3, Execution of 2, Deadline at 6





RT vs RTDBS:

- **Fully characterized workload**

  Dynamic workloads (especially on Web!)

- **Single resource (processor)**

  Multiple resources – processors, disks, memory, data and meta-data

- **Main-memory processing**

  The disk is where the action is

- **Hard deadlines (Feasibility test)**

  Firm and soft deadlines (Best effort)

DBMS vs RTDBS:

- **Linear performance metric (throughput/response time)**

  Boolean performance metric

- **Fairness is essential**

  Priority is fundamental

- **All transactions eventually completed**

  Transactions may be thrown away (firm)

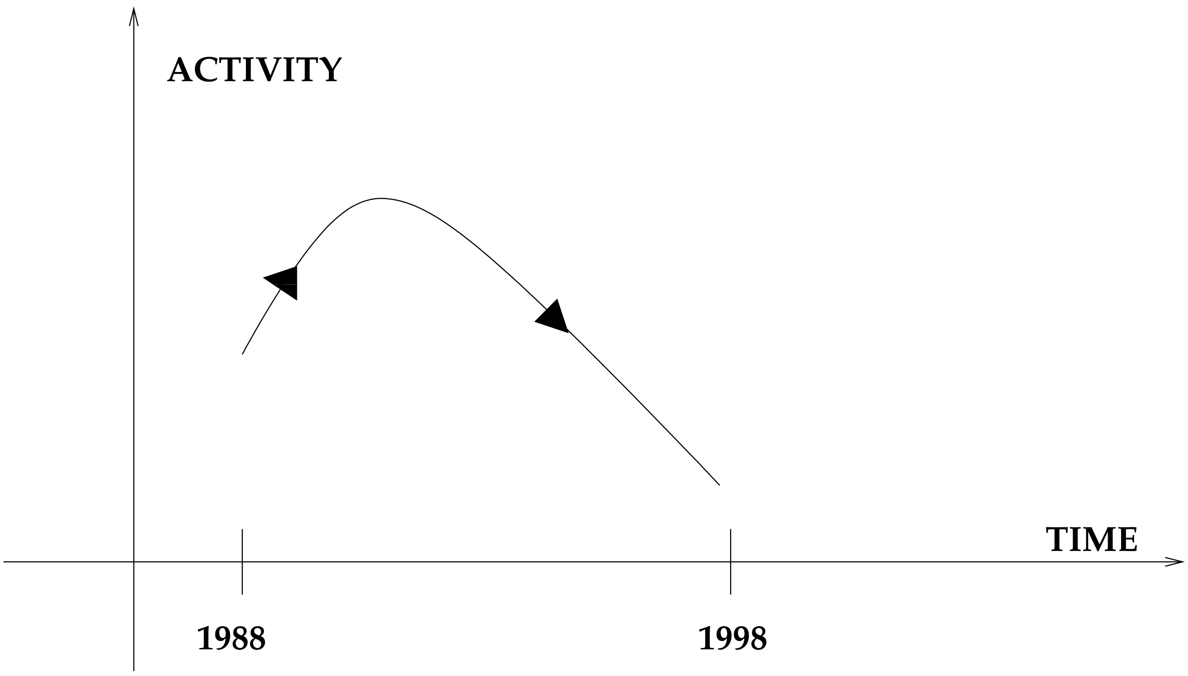
Research Issues

- **Transaction Management (Concurrency Control, Database Recovery, Memory Organization)**
- **Extending Query Language Syntax**
- **Interfacing with OS capabilities**
- **Integrating with Other Database Functionalities (Active, Security, Fault-Tolerance, …)**

History of RTDBS Research

- VLDB 88: Abbott and Garcia-Molina (Best Paper Award !)
- Over a hundred papers
- "Guaranteed" session in IEEE RTSS
- RTDB workshop since 1996
- Half-a-dozen books
- Special Issue of several journals

The Rise and Fall of RTDBS

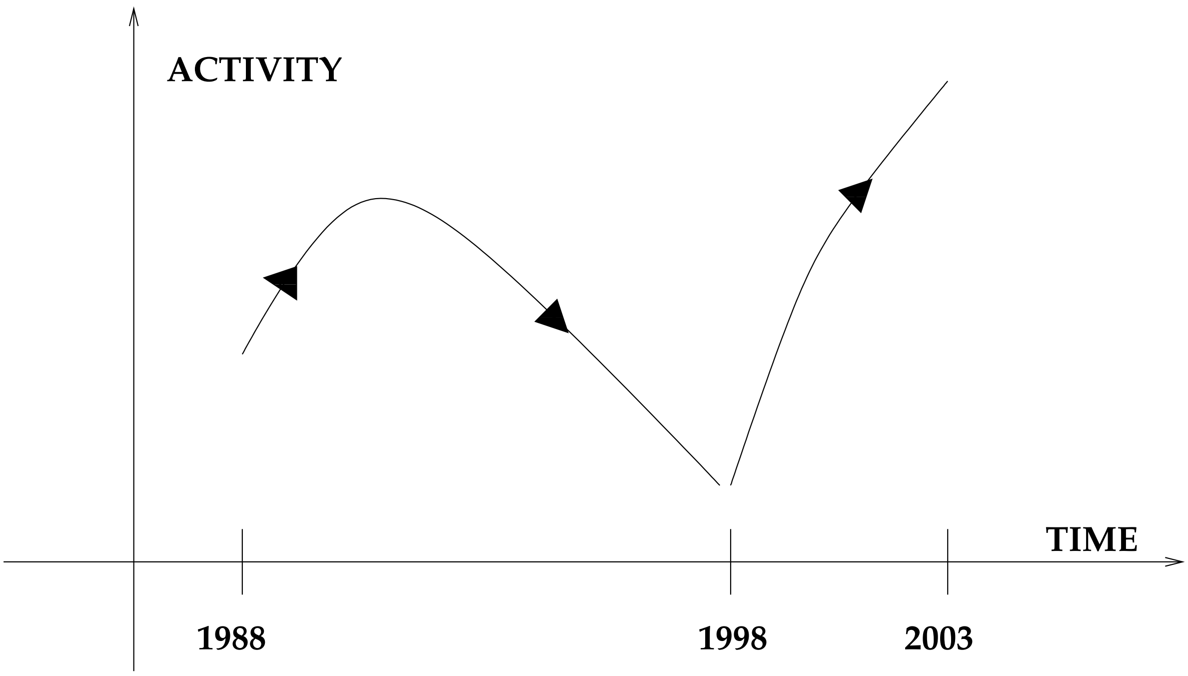


REASONS

- **Theory is hard, really hard !**
- **Software barriers to entry**
- **Academics – YACC papers !**
- **Publishing has become difficult**
- **Industry was underwhelmed**

Solutions in search of a problem (1990s)

… and Rise Again ?



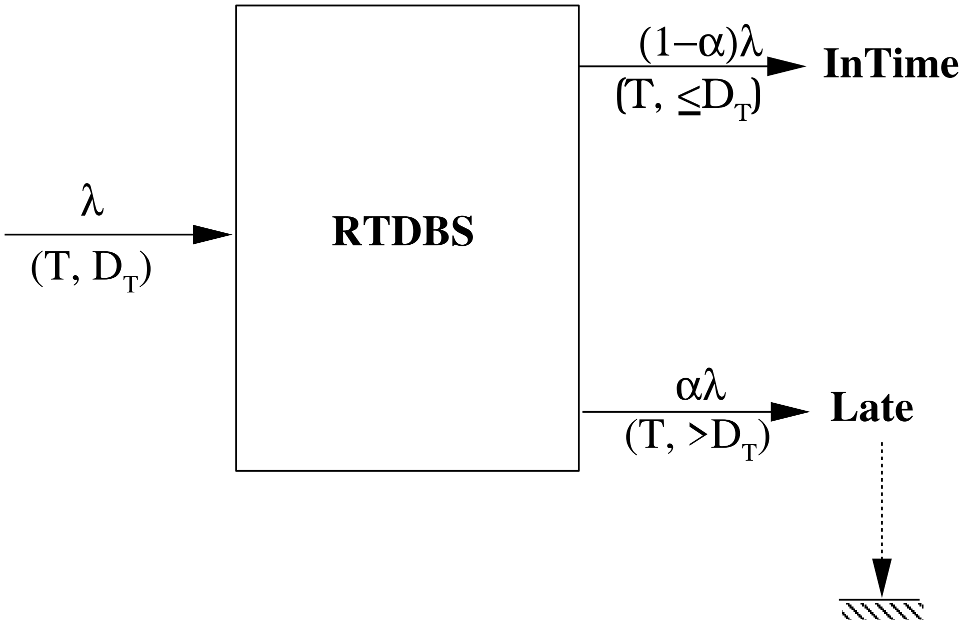
- **Prediction: A key technology of the next decade**

  (Web, Mobile, Multimedia, Control)

Real-Time Concurrency Control

(1990)

Firm RTDBS



- λ – **Arrival Rate**
  α – **MissPercent**
- **GOAL: Minimize** α

Secure Real-Time CC

(1998)

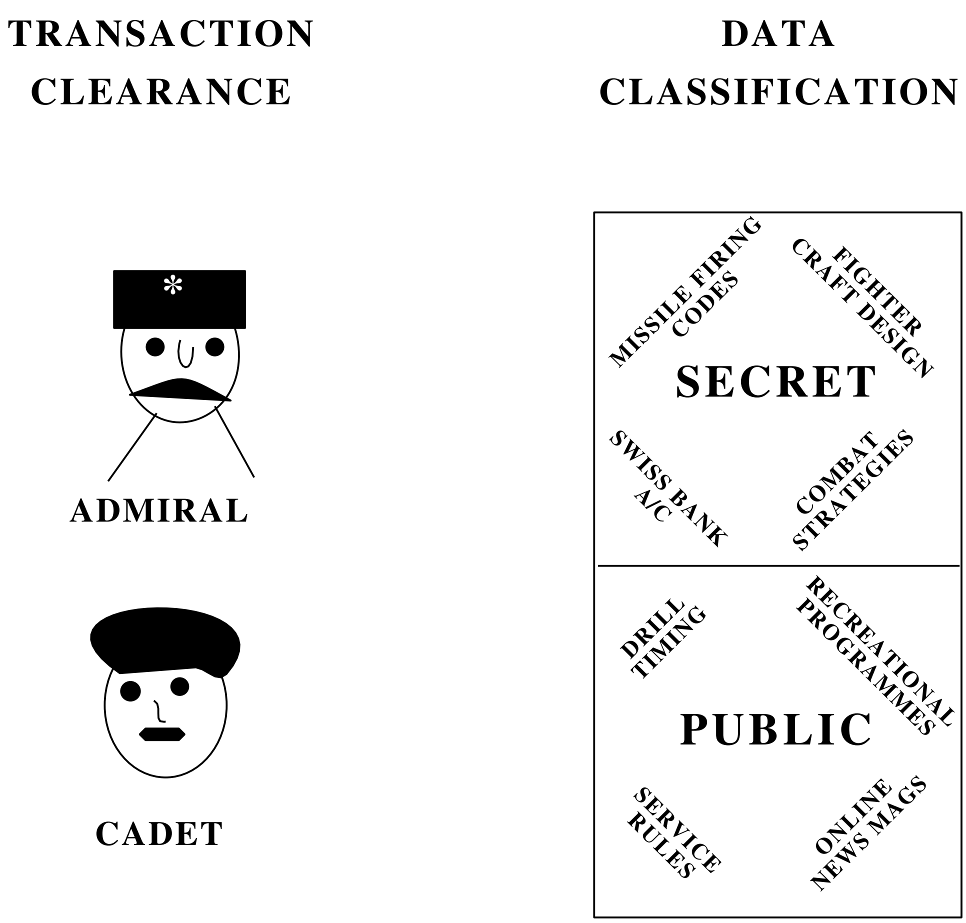
Real-Time Database Security

Many RT applications are security critical

(Military Systems, Stock Trade Systems, …)

- <u>Third</u> dimension of complexity

Security Model



Objective

- Prevent Admirals from passing Secret data to Cadets

SECURITY VIOLATION

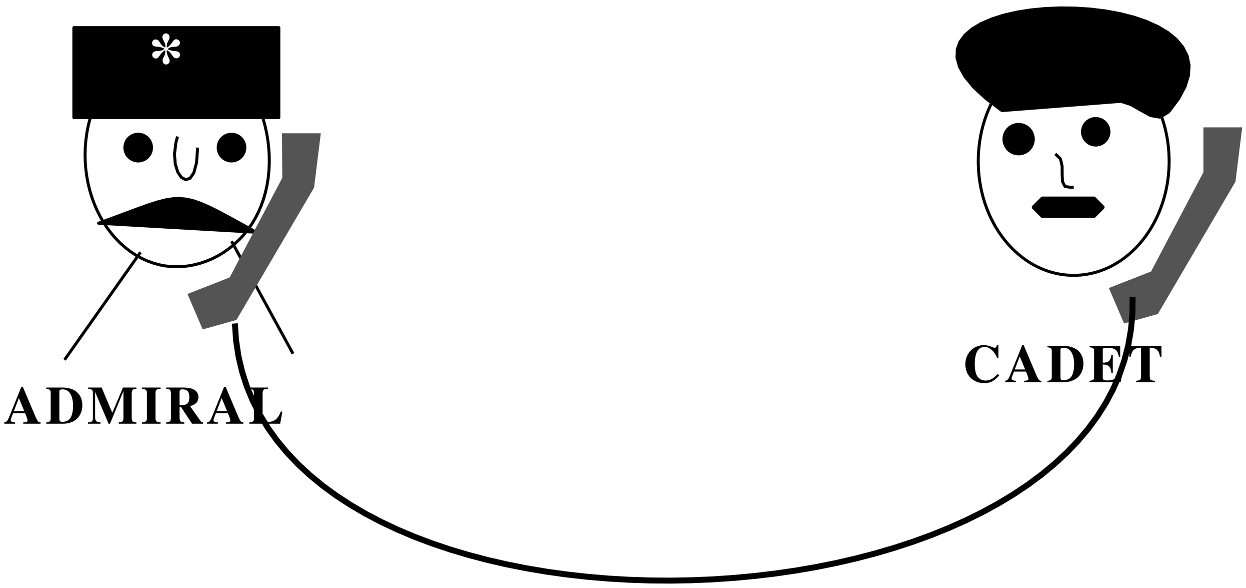
DIRECT

- Simple Solution
- Illegal

INDIRECT

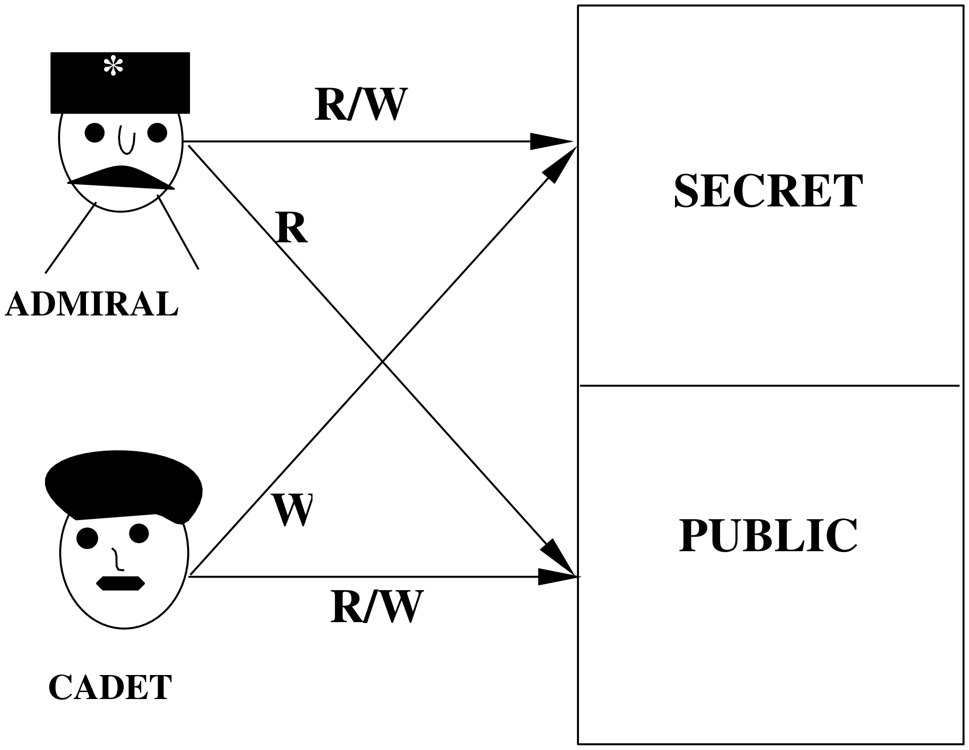
- Complex Solution
- Legal
- Prior off-line agreement

Direct Transfer



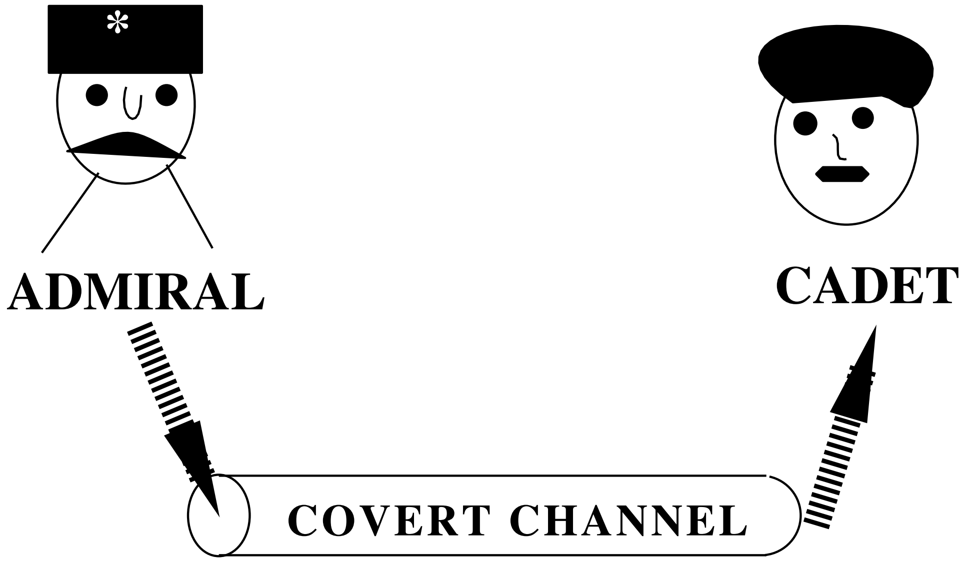
- Admiral copies the secret data to the public area
- Cadet reads the data subsequently

Solution



- Bell-LaPadula model prevents direct transfer of information
- Read from same level and below, Write to same level and above

Indirect Data Transfer



Covert Channel is an indirect,
but legal way of information transfer

Covert Channel - Example

- **Cadet accesses data item "d"**

Transaction Class	Time (ms)		
	05	10	20
ADMIRAL			
CADET		Lock(d)	

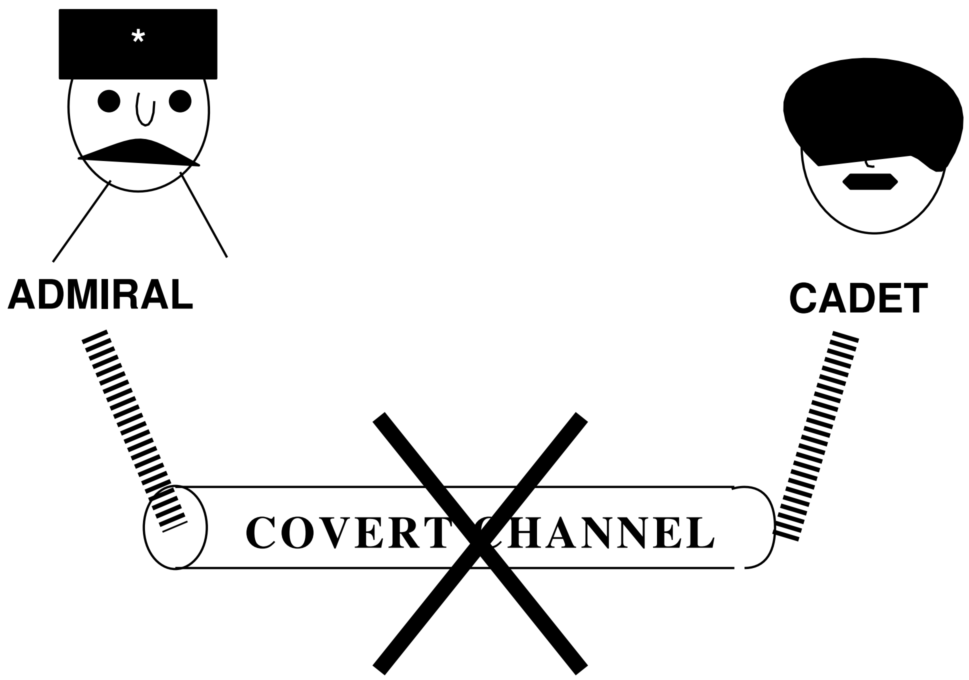
Admiral sending 0 to Cadet

Transaction Class	Time (ms)		
	05	10	20
ADMIRAL	Lock(d)		UnLock(d)
CADET		LockWait(d)	Lock(d)

Admiral sending 1 to Cadet

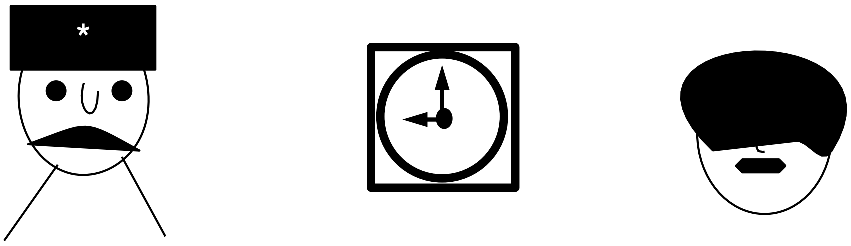
- **Presence or absence of delay encodes the information**

Solution



- Don’t let Cadets “feel” or “see” the existence of Admirals (Non-interference [GM82])
- Give higher priority to Cadets in <u>all</u> resource conflicts

Non-interference in RTDBS



ADMIRAL **CADET**

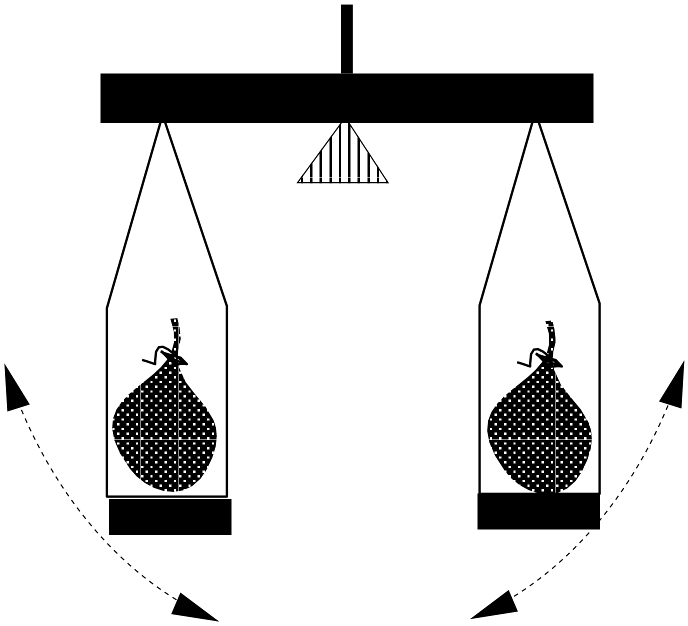
Deadline = 09:10am **Deadline = 10:00pm**

- **Admiral holds lock on data item X**
- **Cadet requests X**
- **Admiral has** TIGHTER **deadline.**
  Admiral is more IMPORTANT **than Cadet.**
  Let Admiral proceed while cadet is made to wait?

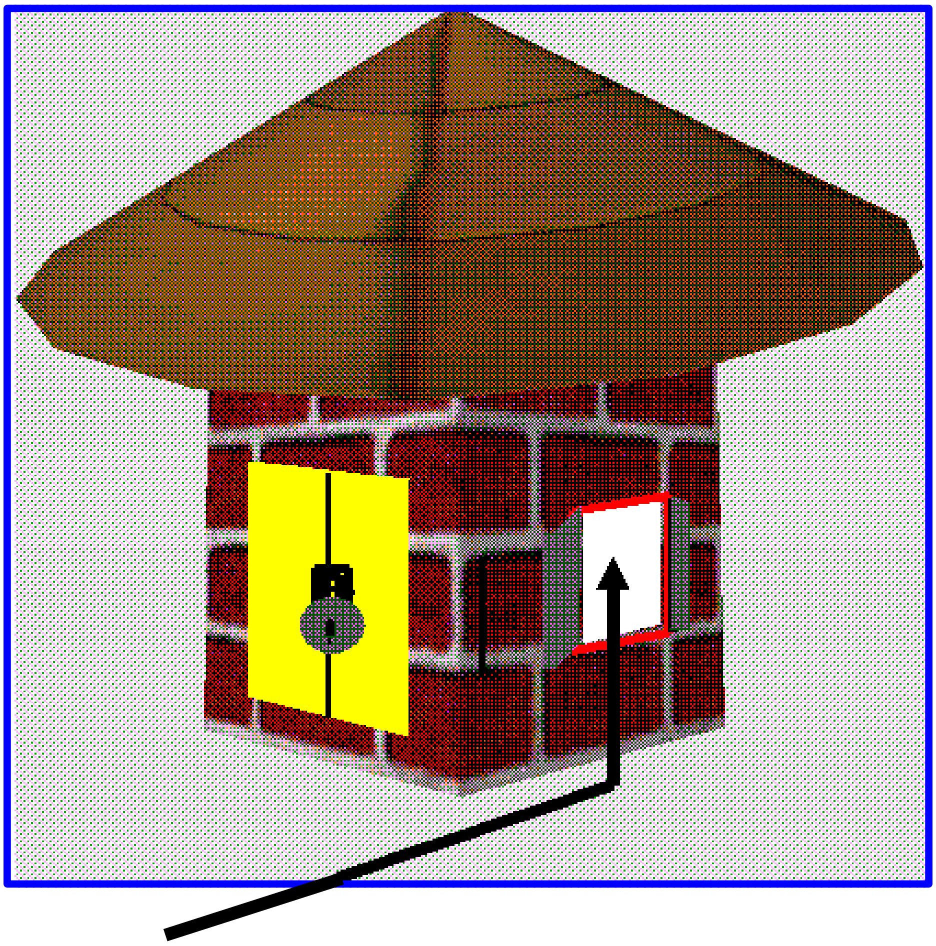
  NO! **Covert Security Violation**

- **Forced to pre-empt Admiral and provide X to Cadet**

TRADE–OFF APPROACH
(Son *et. al.*)

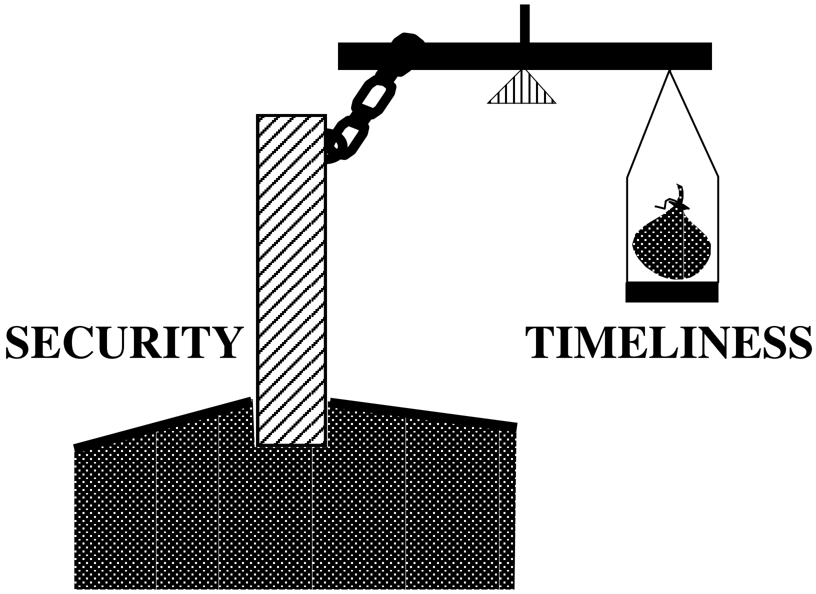


SECURITY **TIMELINESS**



The trade-off approach is like locking the door, but leaving the windows wide open

FULL SECURITY APPROACH



- **Security is a** correctness **constraint**
- **Timeliness is a** performance **constraint**

Complexity in the Secure Domain

- **Not all real-time CC algorithms support non-interference**

  e.g., 2PL-WP, 2PL-CR,
  2PL-OS/BI, WAIT-50, …

- **Multiple transaction classes corresponding to the various security clearance levels**

- **Data access patterns are constrained by Bell-LaPadula restrictions**

- **Conflicts are resolved based on security and timeliness**

- **Class fairness**

Secure CC Protocols

- **Real-time**

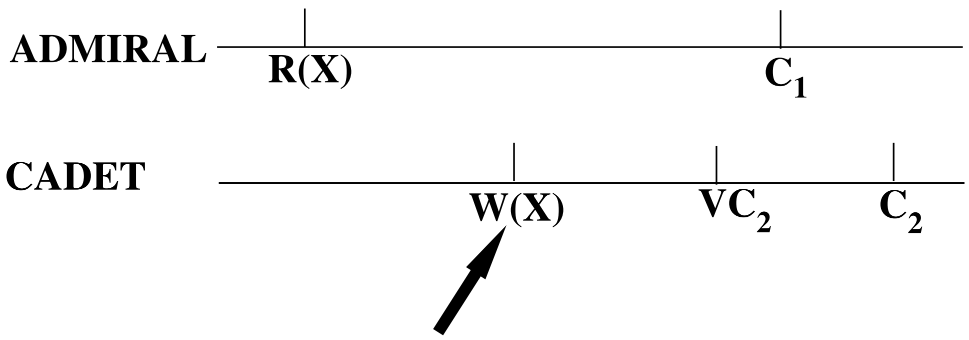
  2PL-HP
  OPT-SACRIFICE
  OPT-WAIT

- **Non-Real-Time**

  Secure2PL (No Restarts)

Secure 2PL [DS93]

- **Simulate 2PL with no blocking**
- **“Virtual” Lock for Cadet conflicting with Admiral**



A "Virtual" Lock is set and both Admiral and Cadet are allowed to proceed.

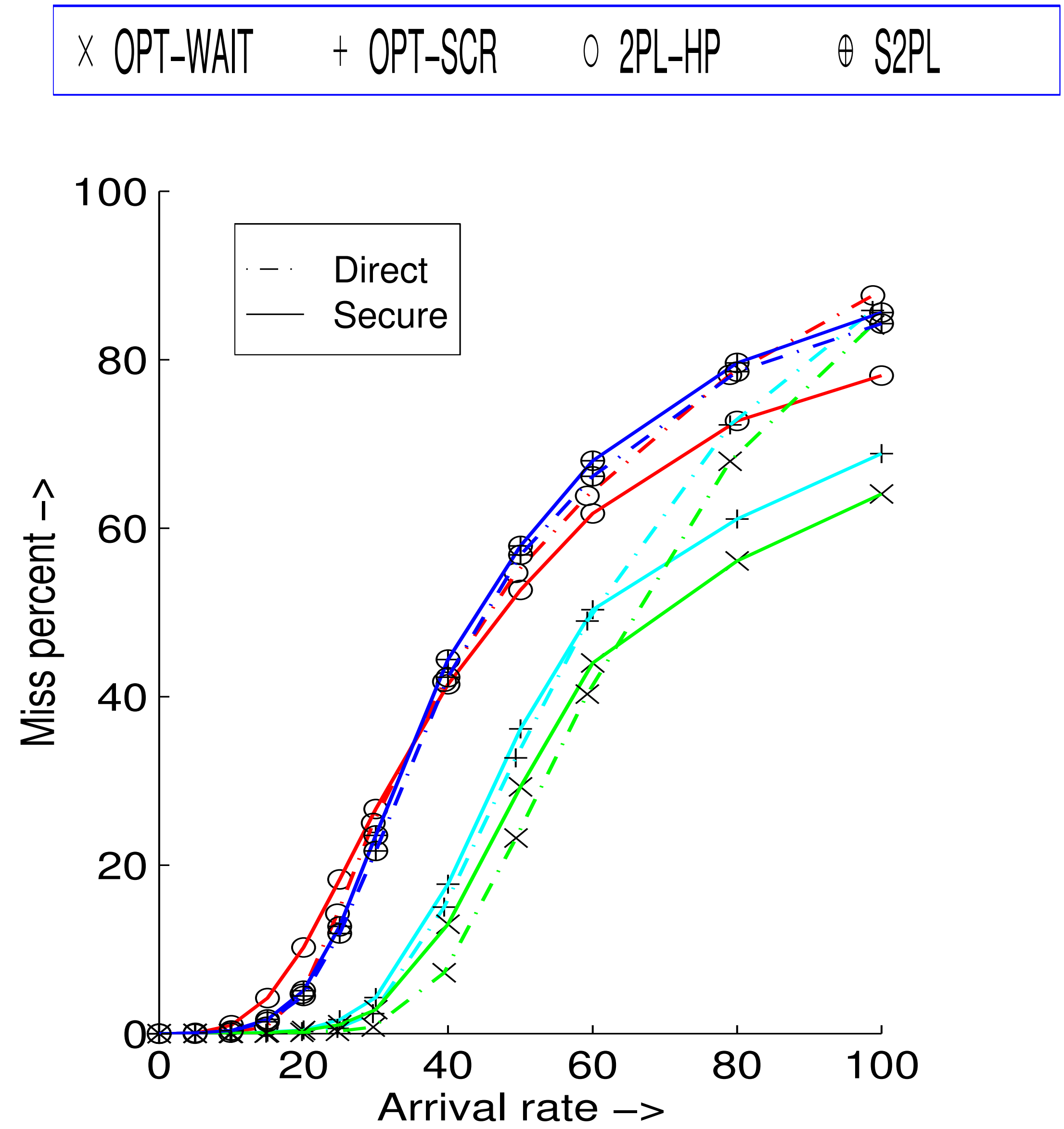
- **Virtual Commit is converted to Real Commit when Admiral commits**
- **Serialization list, Dependent Virtual Lock**
- **Non real-time**

Cost of Covert Channel security

- **Secure System**
  - **Direct Secure**
  - **Covert Channel Secure**
  - **Priority by (SecurityClass, Deadline)**

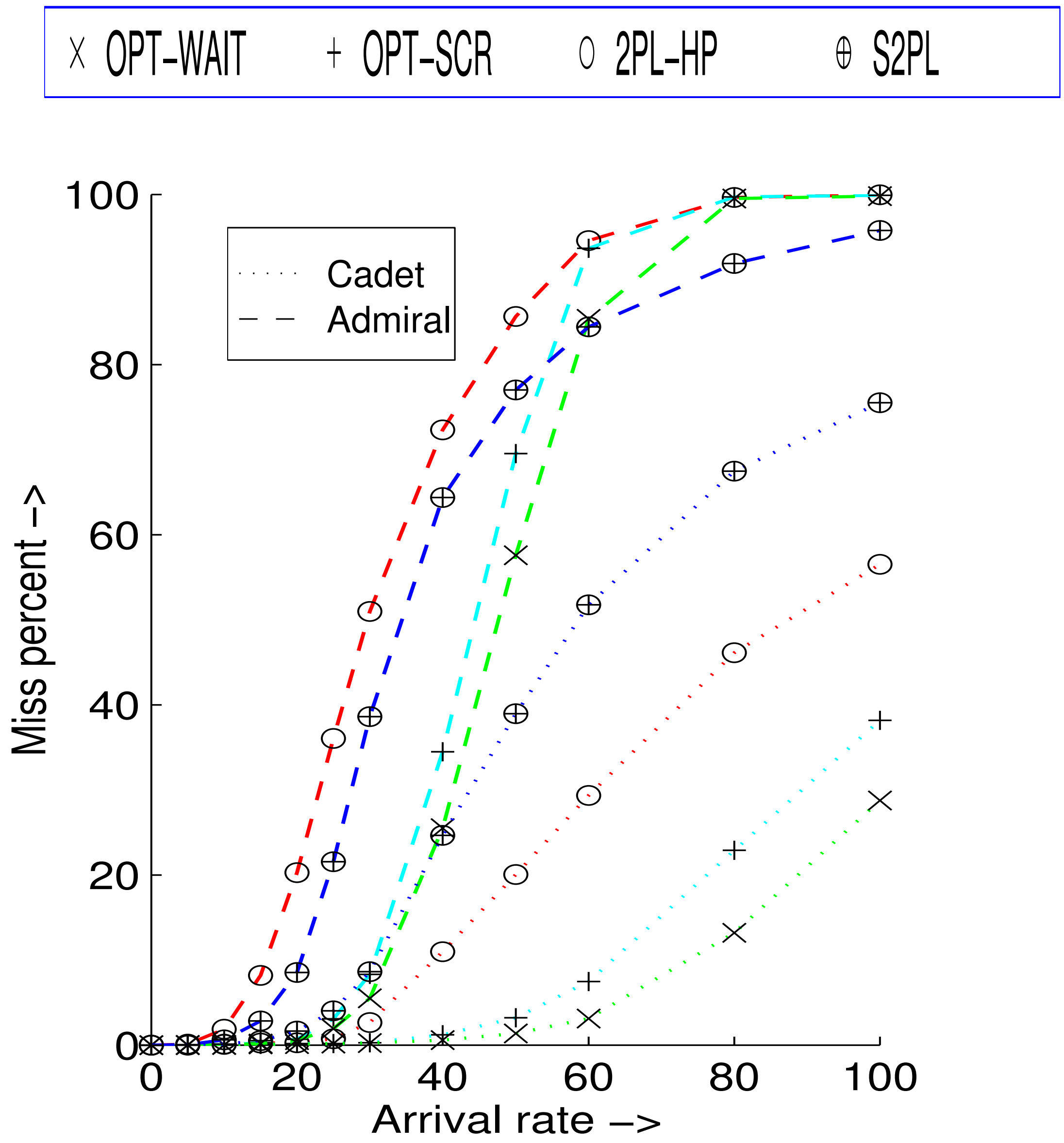
- **Baseline System**
  - **Direct (Bell–LaPadula) secure**
  - **No Covert Channel security**
  - **Priority by Deadline**

Covert Security Cost (RC + DC)



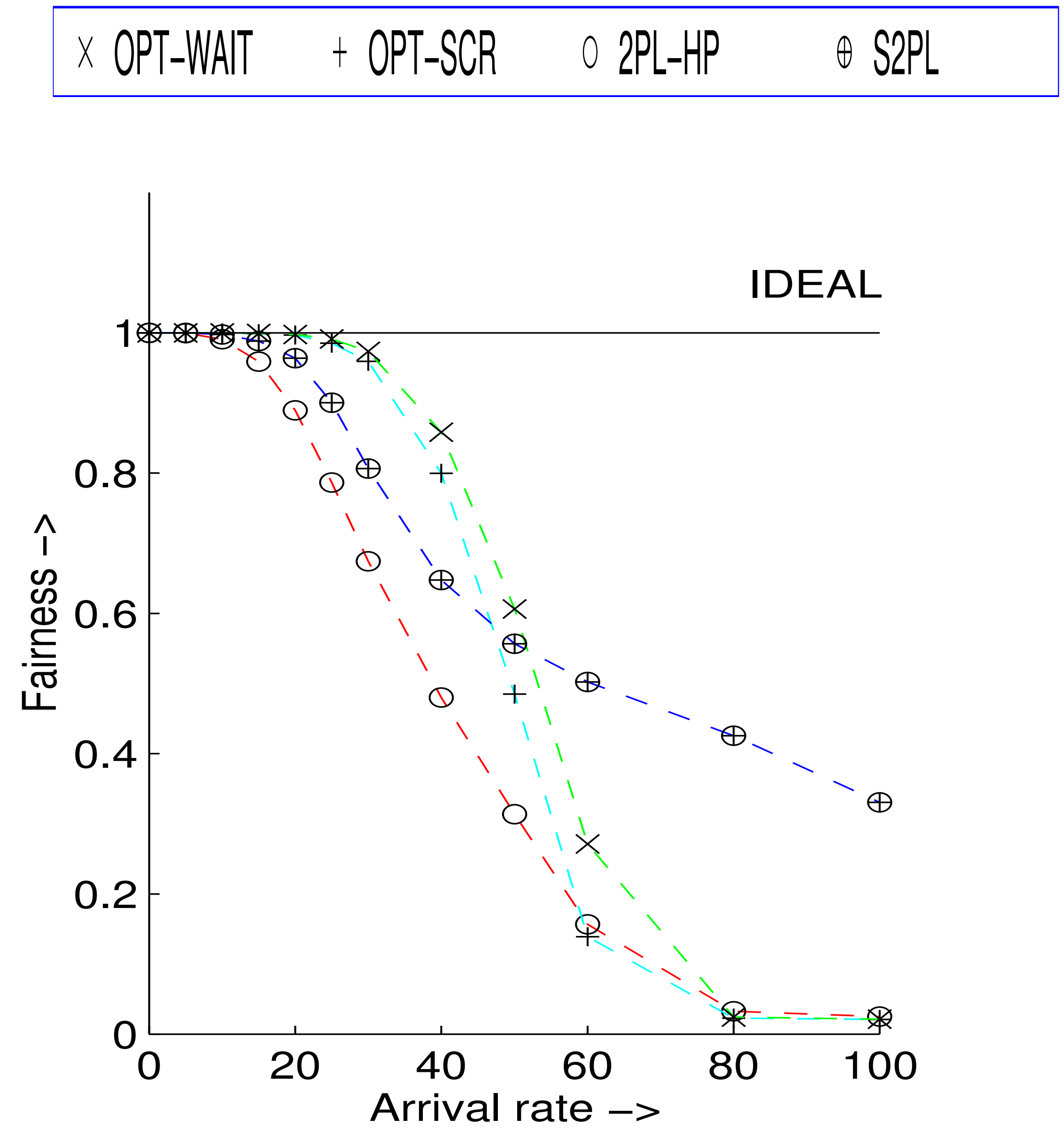
- Marginal cost for security
- OPT-WAIT is the best protocol

Level Miss Percent (RC + DC)



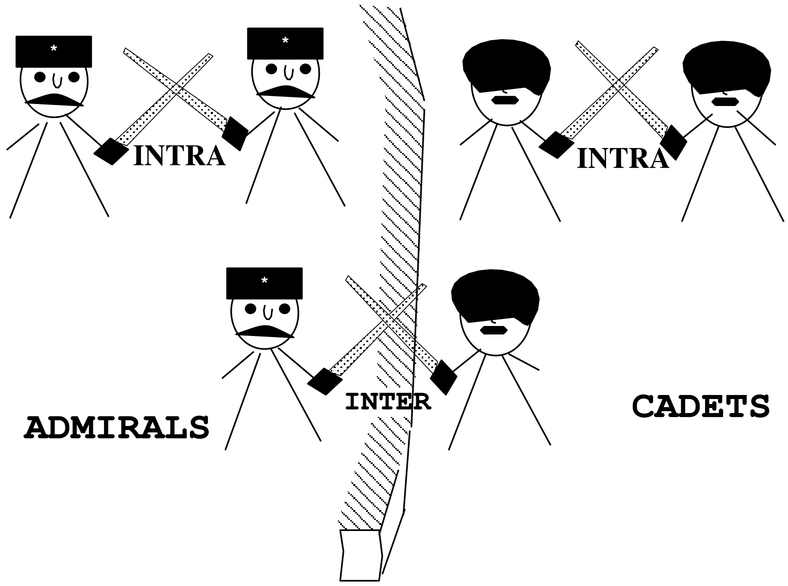
- Admirals much worse than Cadets
- Fairness is the major issue

Fairness (RC + DC)



- OPT-WAIT is most fair
- S2PL shows "virtual" fairness

Dual Approach



- **Two conflict types:**
  Inter-class & Intra-class
- **Different CC strategies for each type**
- **Inter-class Conflicts**
  – Resolve on Security
  Intra-class Conflicts
  – Resolve on Timeliness
- **Minimize Miss Percent performance** **<u>without</u> affecting security**

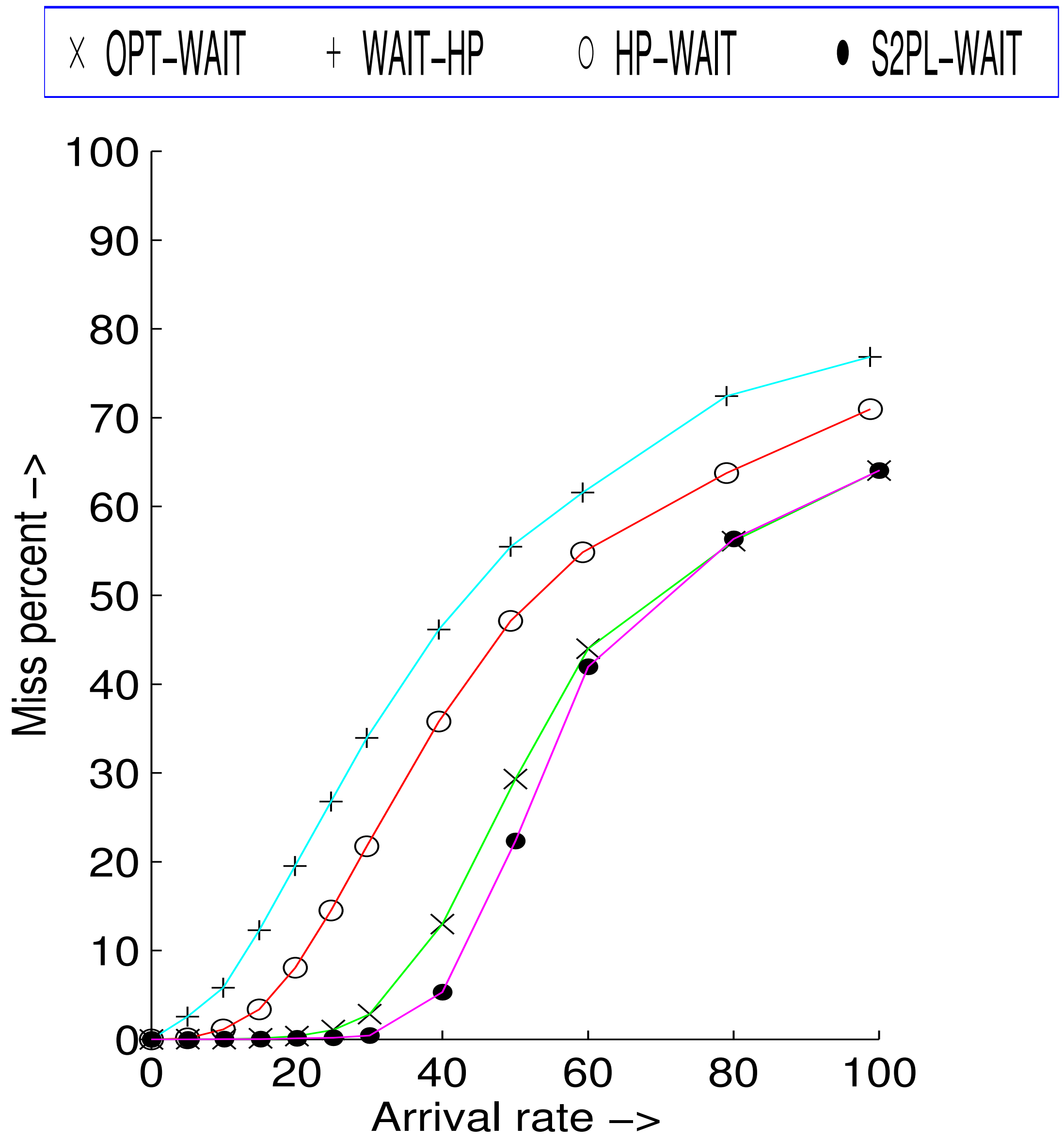
Dual Protocols

Dual Algo.	Inter	Intra
HP-WAIT	2PL-HP	OPT-WAIT
WAIT-HP	OPT-WAIT	2PL-HP
S2PL-WAIT	S2PL	OPT-WAIT

- Serializability?

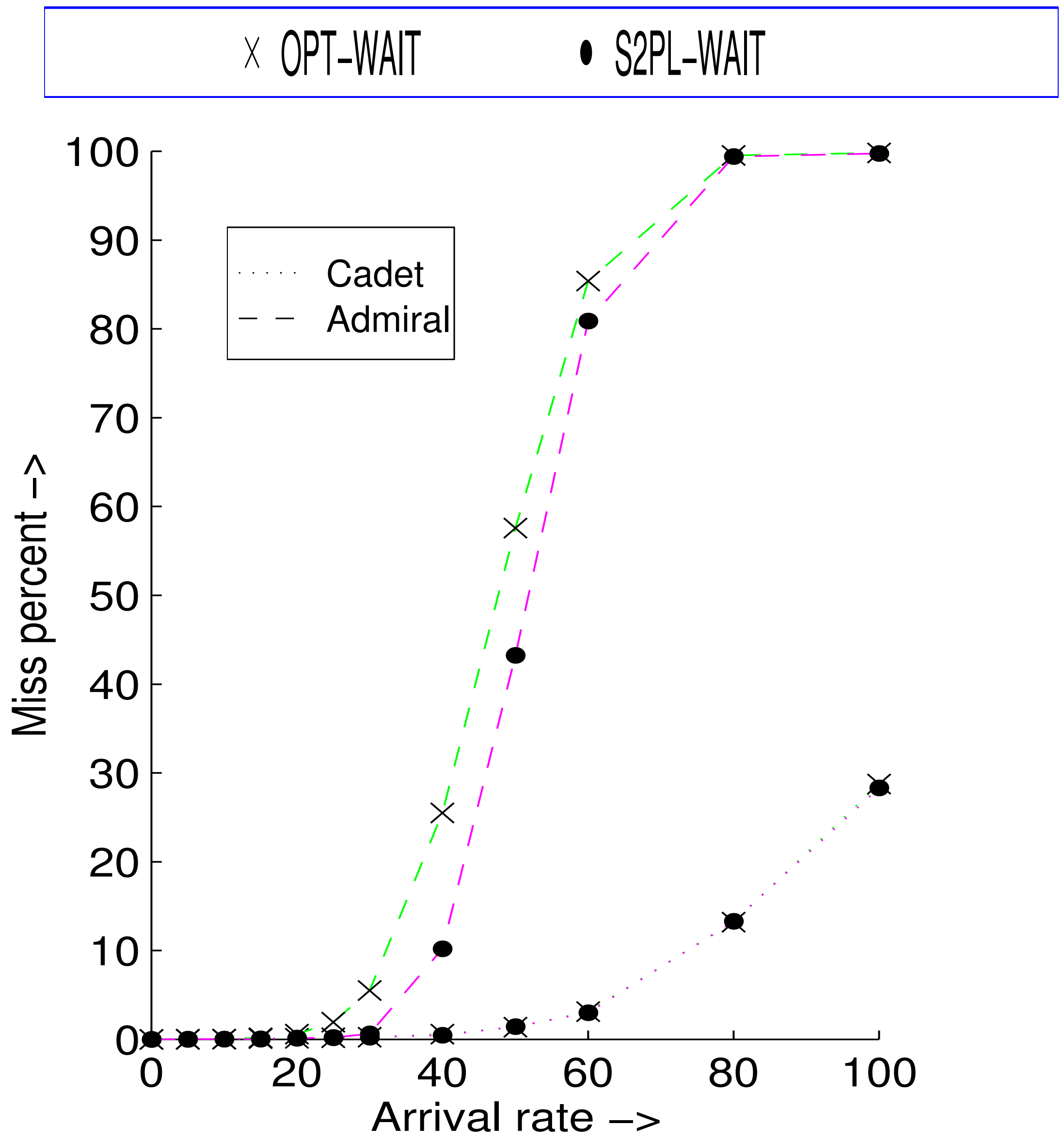
Information Systems, March 96
MCC Architecture
Solves this problem for n CC Managers

Miss Percent (Dual Protocols)



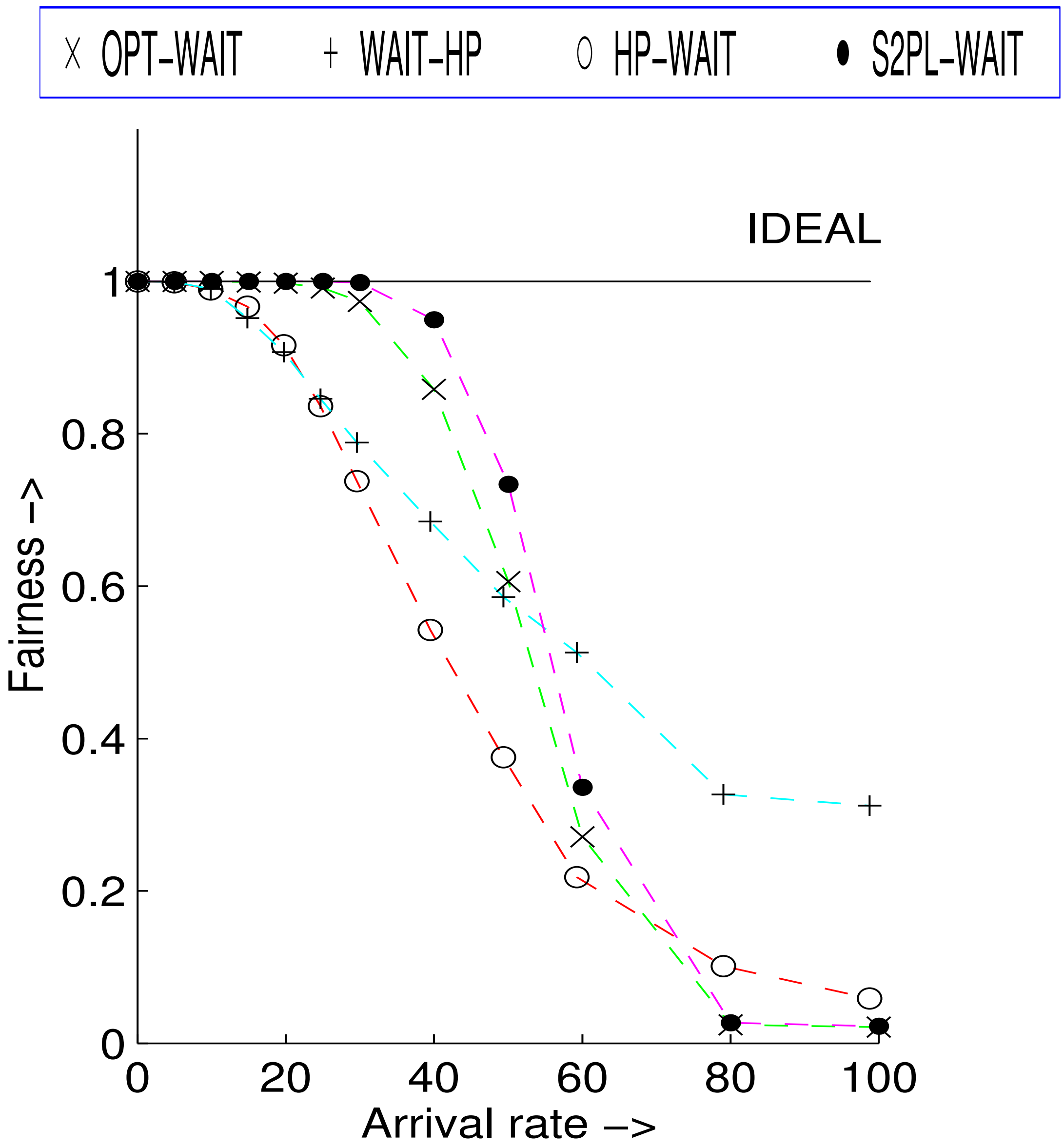
- **OPT-WAIT >> WAIT-HP and HP-WAIT**
- **S2PL-WAIT > OPT-WAIT**

Level Miss Percent (Dual Protocols)



- **Admiral performance in S2PL-WAIT improves**

Fairness (Dual Protocols)



- **S2PL-WAIT is most fair**
- **For WAIT-HP, WAIT dominates HP**

Conclusions

- **Marginal real-time cost for covert security**
- **OPT-WAIT is the best individual protocol**
  lowest Misspercentage
  best *true* Fairness
- **Dual Protocols**
  S2PL-WAIT shows better performance than OPT-WAIT

  Permits use of non-secure real-time CC protocols! (for intra-conflicts) e.g., S2PL/WAIT–50
- **Triple Protocols are possible !**
- **Fairness is a serious problem**

<u>Achievements</u>

- **All Aspects of Transaction Management for Firm Deadlines (Centralized and Distributed)**

  Data Concurrency Control,
  Index Concurrency Control,
  Replica Concurrency Control,
  Atomic Commit Processing,
  Buffer Management

- **Integration with Active Databases**

- **Support for Security Requirements**

Important Research Problems

- **Killer App! (E–Auctions?)**
- **Theoretical Frameworks**
- **Performance Benchmarks**
- **Data-Derived Deadlines**
- **Multi-Deadline Workloads**
- **Overload Management**
- **OLAP + imprecise RTDB**
- **Support for Mobile computing**
- **Protocols for Electronic Commerce**
- **Building of Prototypes**

Useful Sites

- www.eng.uci.edu/ece/rtdb
- www.cs.virginia.edu/ṽadb
- www-ccs.cs.umass.edu/rtdb
- www.cs.bu.edu/ best
- www.cs.cityu.edu.hk/research/rtds
- www.cs.ccu.edu.tw
- dsl.serc.iisc.ernet.in