# Analysis of Mobile Agents in Logistical Environments

## ITG FG 5.2.4 Workshop “Communication Applications for Logistics”

Teilprojekt B3 SFB 637
**Mobile Kommunikations-netze und -modelle**

**Prof. Dr. Carmelita Görg**

**Markus Becker**
Gulshanara Sayyed
Dr.-Ing. Andreas Timm-Giel

Arbeitsgruppe Kommunikationsnetze

Fachbereich Physik / Elektro- und Informationstechnik

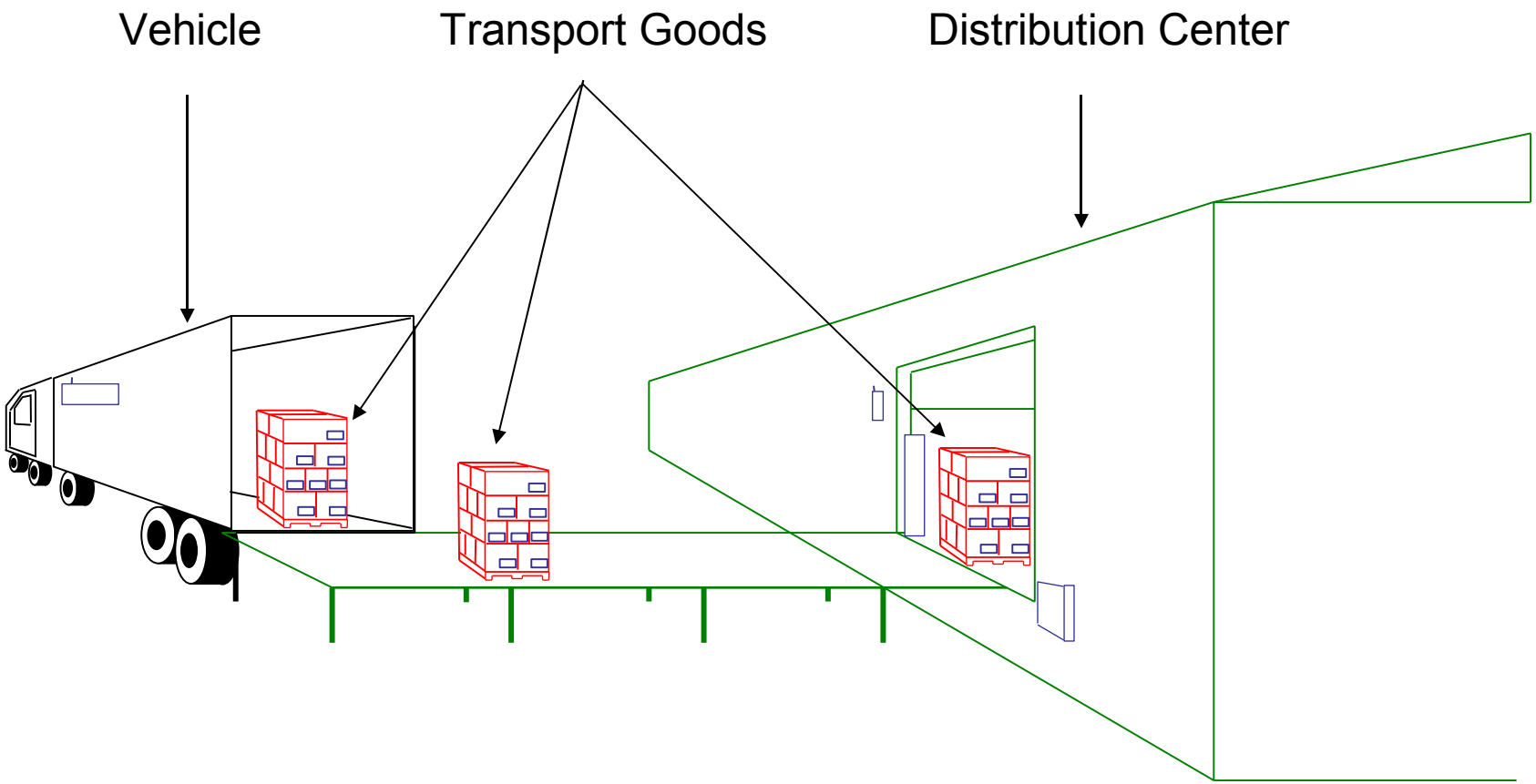
# Overview

- Introduction SFB 637
- Requirements of logistics
- Logistical Scenario
- Agents
  - Motivation for using Agents
  - What are agents
  - Introduction to JADE, JADE-LEAP
  - Mobile Agents
  - Analytical methods for Agent performances
  - Scenario & Scenario Notation
  - Results
- Conclusion and Outlook

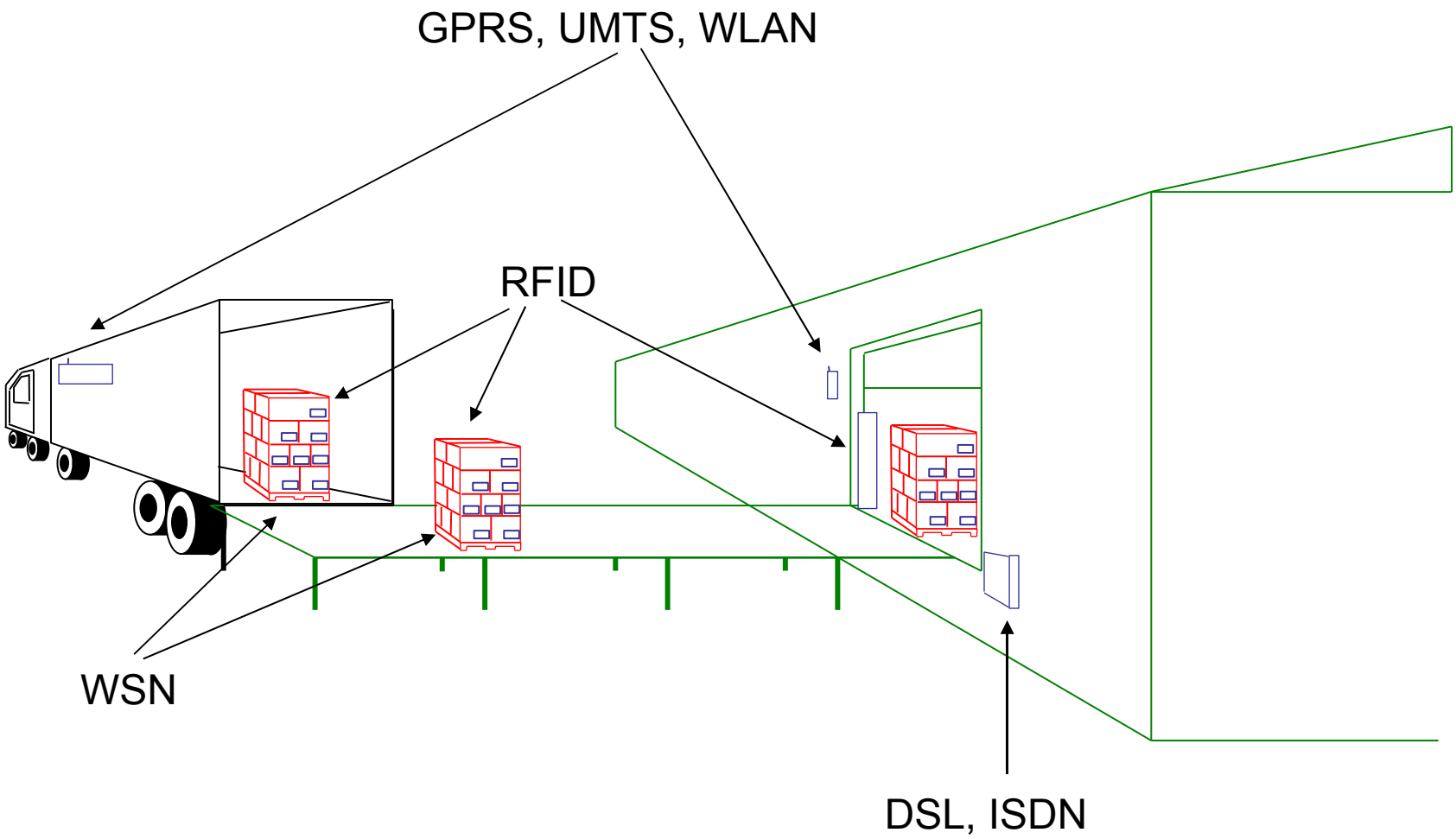
# Requirements of logistics

- High number of different end devices
- Heterogeneous Communication systems
- Heterogeneous Ressources
- Spatial Distributeness
- Mobility
- Robustness
- Heterogeneous Ownership (Security/Trust)

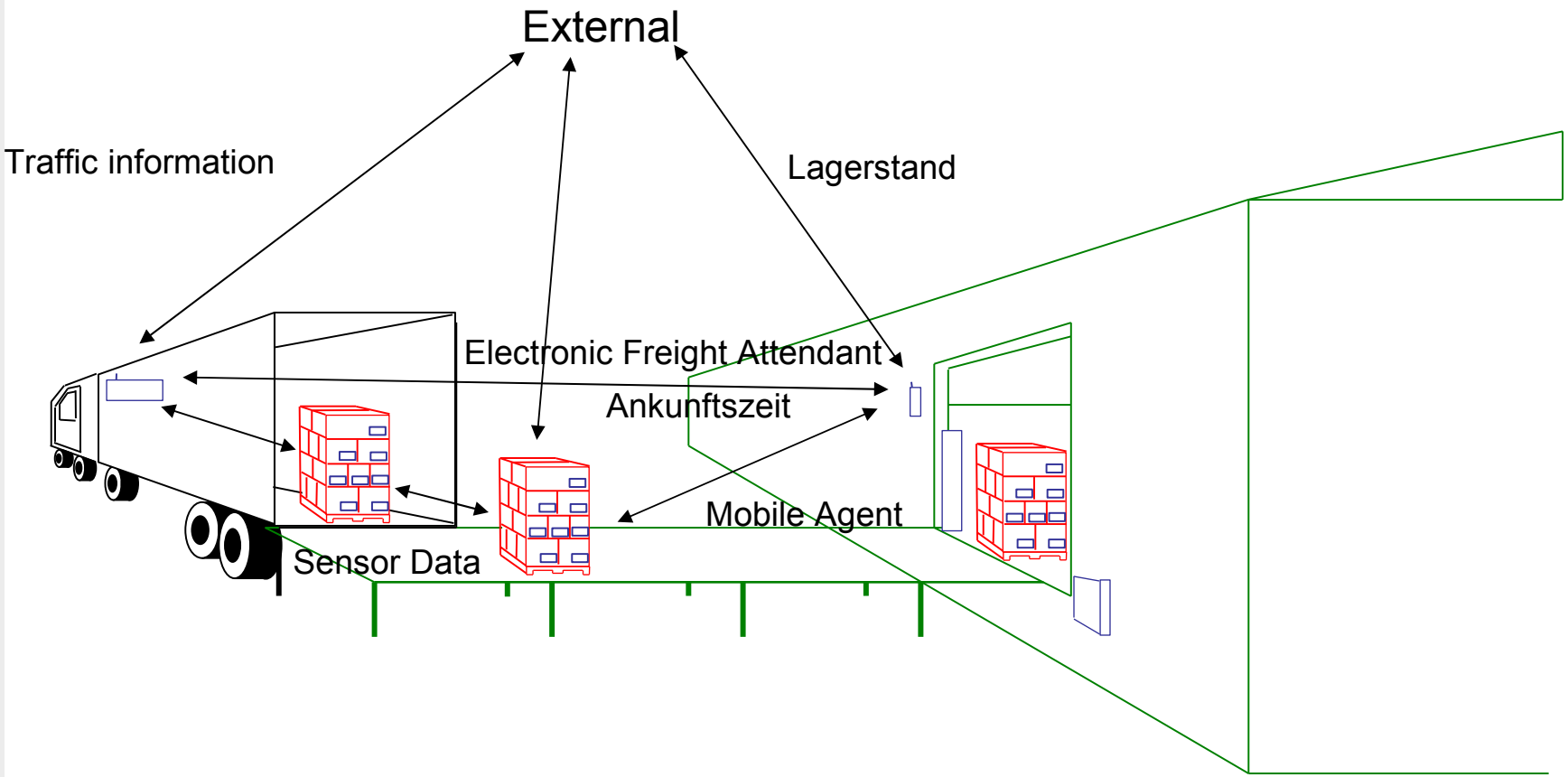
# Logistical Objects



# Logistical Objects – Communication systems



# Logistical Objects – Information model



# Motivation for using (Mobile) Agents

- The logistic systems and networks are dynamically and structurally complex
- Central planning and control of logistic processes has become increasingly difficult
- Emphasizes on decentralized and autonomous control of logistic process
- For autonomous logistic processes, communication between mobile agents is important
- Mobile agents present an important abstraction mechanism designed for distributed environments
- Hence need for a mobile agent architecture for logistic purposes

# What are Agents

- Most literature regarding Agents introduce a definition of what an agent is in terms of its properties.
- So for our purpose, an Agent possesses following properties:

| | |
|---|---|
| Autonomous | Exercise control over its own actions |
| Goal Oriented | Does more than just react to the environment |
| Reactive in | Can sense and respond to changes its environment |
| Flexible | Order of actions is not fixed |
| Communicative | Communicates with other agents |
| Adaptive/Learning | Can change behaviour based on experience |
| Mobile another | Can `migrate´ from one host to |

# Agent Platforms

- In 1996 The Foundation for Intelligent Physical Agents (FIPA) was formed to produce software standards for heterogeneous and interacting agents and agent-based systems
  - APRIL Agent Platform [non Java based, uses the `Agent Process Interaction Language´]
  - FIPA-OS [only theoretical mobility – the mobility is till a prototype]
  - Grasshopper
  - JACK intelligent Agents  [does not support mobility]
  - JADE/LEAP
  - Zeus [no mobility]

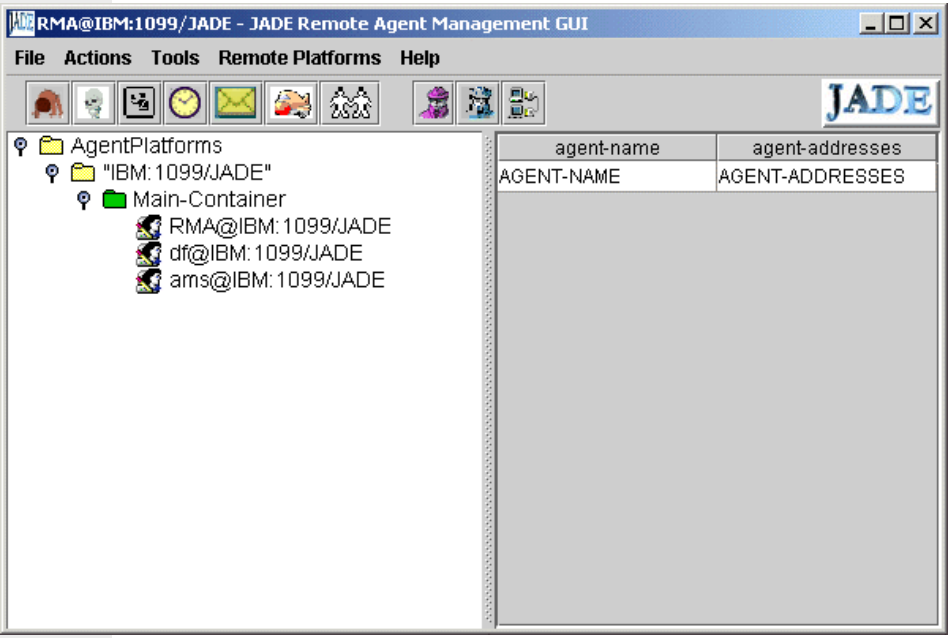
# Agent Platforms

- There are many `Agent Frameworks` available to produce Agent systems whilst many of these are based on `Java` Programming language (this is not always the case).
  - Ara Tcl — c, Tcl
  - Ajanta, Jade, Mole — Java
  - Telescript — Telescript
  - April — April Scripting Language

# JADE

- JADE (Java Agent Development Framework) is a software development framework aimed at developing multi-agent systems fully implemented in Java
- It includes:
  - a FIPA (Foundation for Intelligent Physical Agents) compliant agent platform, which facilitates a **runtime environment**
  - a **library** of classes to develop agents
  - a suite of **graphical tools**, that help in administrating and monitoring agents
- The agent platform can be distributed across machines running different operative systems, and the configuration can be controlled via a remote GUI.

# JADE



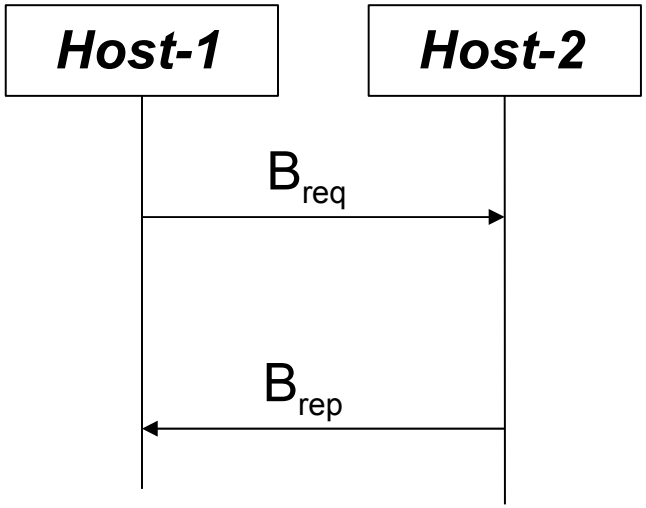

- JADE agent platforms have containers to hold agents.
- The Main Container resides on the host which also runs platform‘s RMI server.
- It contains the RMA (Remote Monitoring Agent), AMS (Agent Management System) and DF (Directory Facilitator)
- Agents residing have unique names AID (Agent ID)
- Message transport system is the software component controlling all the exchange of messages within the platform
- Communication between agents is by Agent Communicative Language Message (ACL Message class in JADE)

# Mobile Agents – Why?

- There are seven good reasons to use mobile agents
  - They reduce network load
  - They overcome network latency
  - They encapsulate protocols
  - They execute asynchronously and autonomously
  - They adapt dynamically
  - They are naturally heterogeneous
  - They are robust and fault tolerant

  (Danny B. Lange)

# Agent - Communication



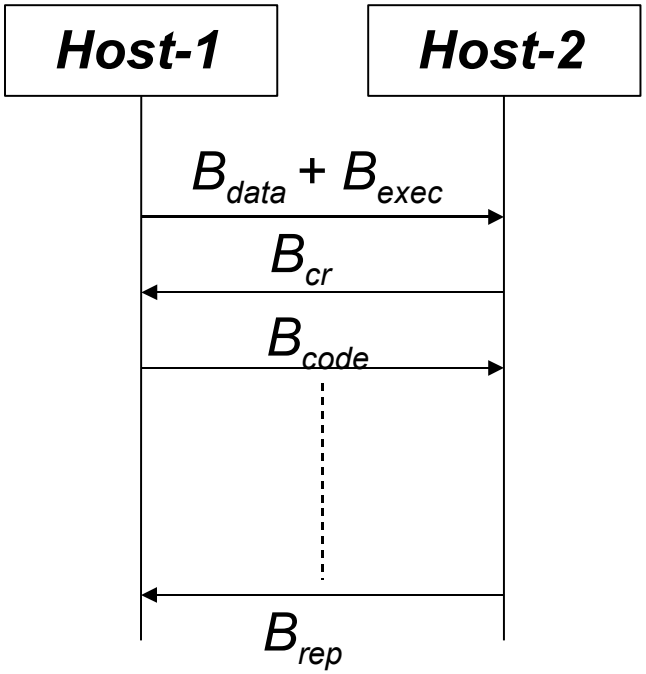

- $B_{req}$ - size of request from destination Host – 2
- $B_{rep}$ - size of reply from source Host -1

- Communication (Remote Communication): Two agents running on different hosts communicate with each other using either RPC (Remote Procedure Calling) or RMI (Remote Method Invocation)

$$N_{load} = B_{req} + B_{rep} \qquad \text{if } H_1 \neq H_2$$

$$T_{exec} = 2\delta(H_1, H_2) + (1/\tau) N_{load} \qquad \text{if } H_1 \neq H_2$$

# Agent - Migration




The agent moves from one host (source) to another host (destination) and interacts with the other agents using local procedure calling

$B_{data}$ - Data size of the agent

$B_{exec}$ - Execution State of the agent

$B_{cr}$ - Size of code request

$B_{code}$ - Code size of the Agent

$B_{rep}$ - Reply size

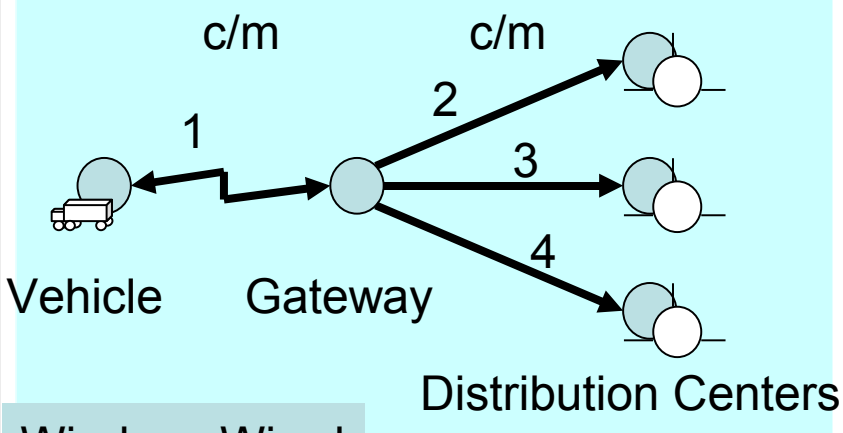
$$N_{load} = p(B_{cr} + B_{code}) + B_{data} + B_{exec}$$

$$T_{load} = (1 + 2p)\delta(H_1, H_2) + (1/\tau)(N_{load})$$

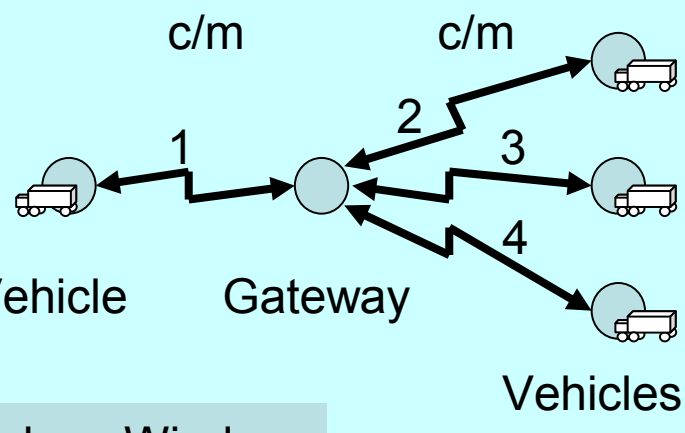
$$N_{selec} = \theta \cdot B_{rep} \; ; \; T_{selec} = \delta(H_1, H_2) + (1/\tau)N_{selec}$$

$$T_{exec} = (2+2p)\delta(H_1, H_2) + (1/\tau)(N_{load} + N_{selec})$$

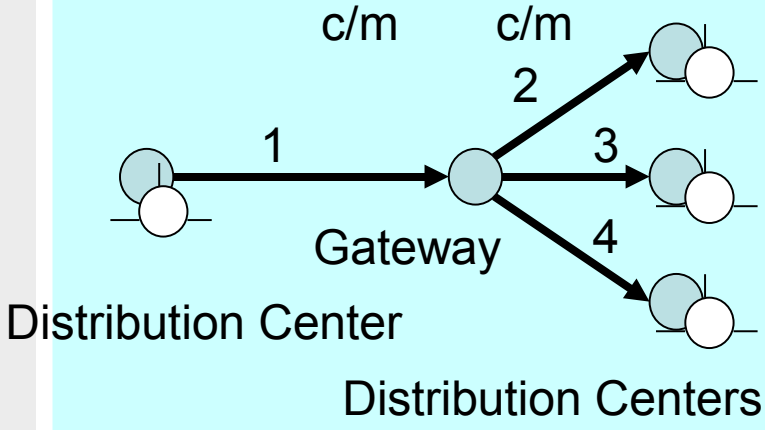
# Different Possible Scenarios



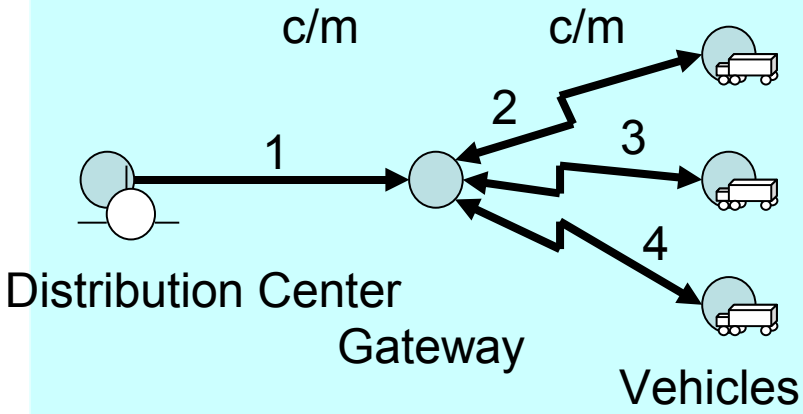

Wireless-Wired




Wireless-Wireless




Wired-Wired



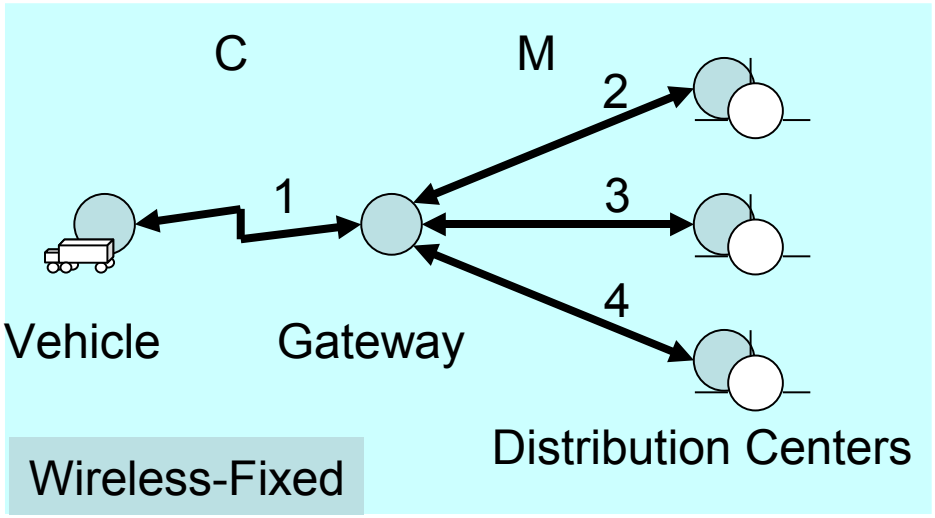

Wired-Wireless

# Mobile Agenten - Notation

X/nY-x/y-strategy

Strategy (Sequential, Parallel, Branch)
Network Operation (C-Communication, M-Migration)
Access Operation (C-Communication, M-Migration)
Type of network (F-Fixed, W-Wireless)
Number of legs in the network
Type of access (F-Fixed, W-Wireless)

e.g.:




**W/3F-C/M-sequential**

# Total Cost of Migration and Communication

- In summary, the cost of migration and communication measured by network load and execution time is calculated by $N_{Cost_{M_iR_j}} = N_{M_i} + N_{R_j}$

$$T_{Cost_{M_iR_j}} = T_{M_i} + T_{R_j}$$

- *Sequential Implementation*: Sequence communication implements each method (migration or communication) in turn $Cost_{seq} = \sum_{i=1}^{m} \sum_{j=1}^{c} Cost_{M_iR_j}$
- *Parallel Implementation*: It implements all methods (migration or communication) simultaneously, in our scenario for the distributed network only

$$T_{Cost_{par}} = \max_{i,j} Cost_{M_iR_j}$$

- *Branch Implementation*: It implements only one method (migration or communication) depending on the minimum cost $Cost_{branch} = \min_{i,j} Cost_{M_iR_i}$

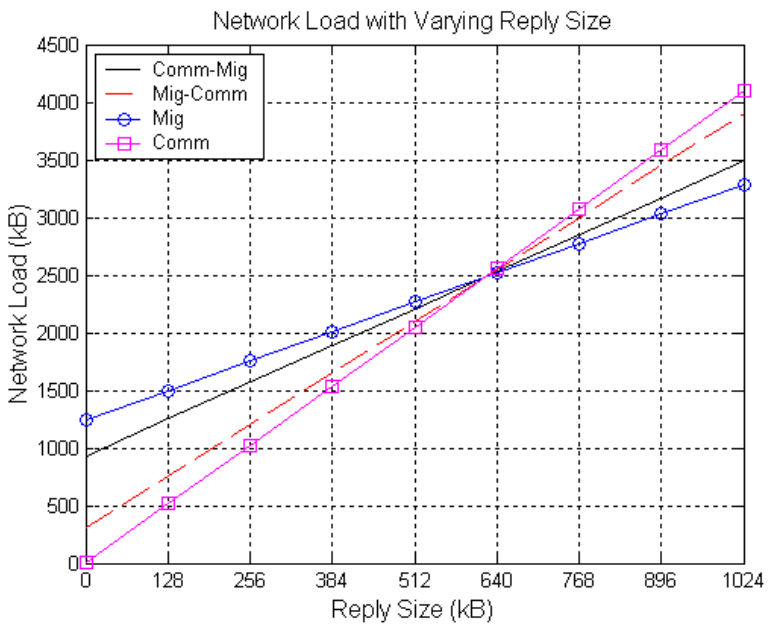
# Table for Network Characteristics

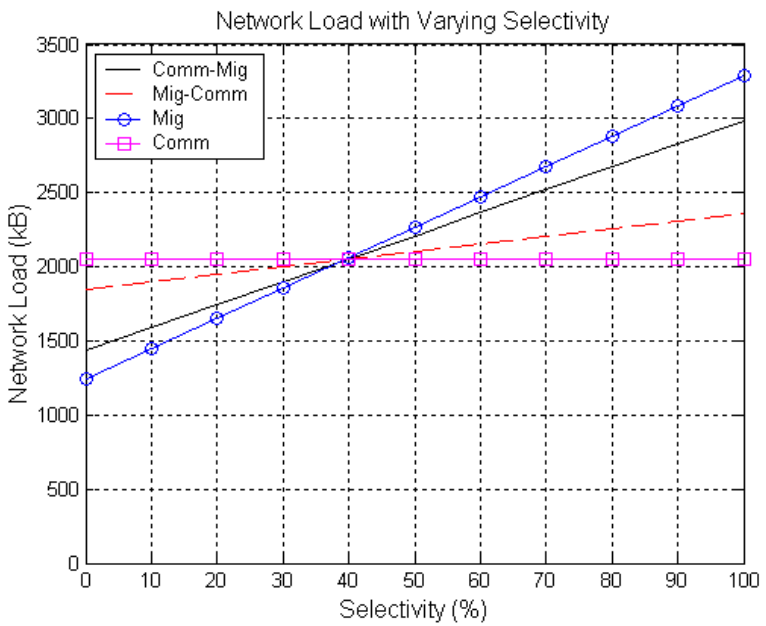
| Network Characteristics: | Values: |
|---|---|
| Throughput Wired | 560kBytes/s |
| Throughput Wireless | 9.6kBits/s |
| Delay Wired | 8ms |
| Delay Wireless | 400ms |
| Selectivity -1 | 50% |
| Selectivity -2 | 75% |
| Selectivity -3 | 75% |
| Selectivity -4 | 50% |

# Table for Agent Characteristics

| AgentCharacteristics: | Values [kB]: |
|---|---|
| Code blocks | 9.5 |
| Code block request | 0.5 |
| Data | 300 |
| Code for request -1 | 0.5 |
| Code for request -2 | 0.5 |
| Code for request -3 | 5 |
| Code for request -4 | 0.5 |
| Code for reply -1 | 512 |
| Code for reply -2 | 640 |
| Code for reply -3 | 10240 |
| Code for reply -4 | 5120 |

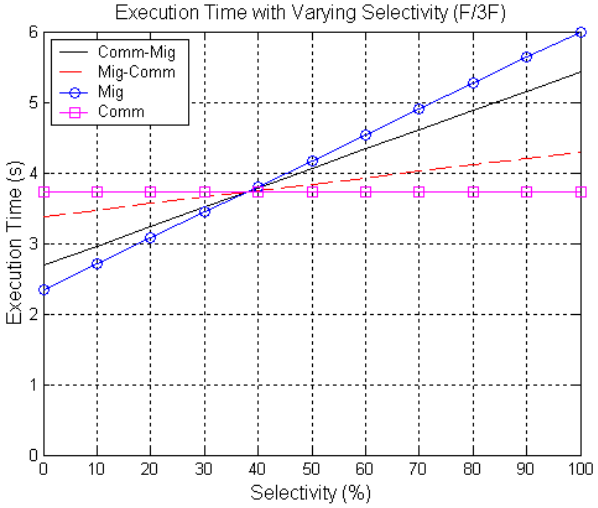
# Network Load in Sequential Strategy

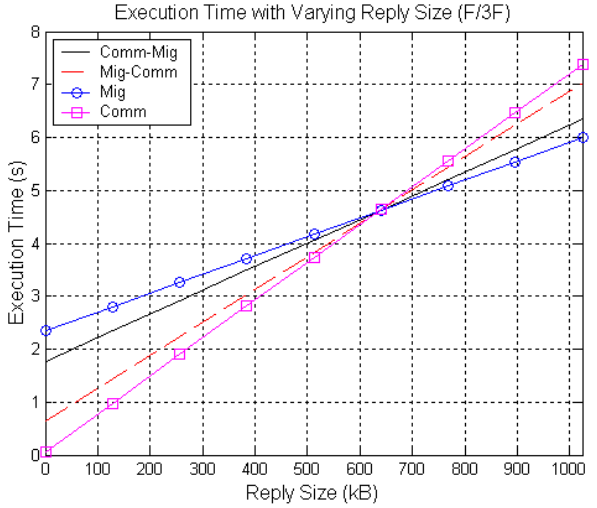


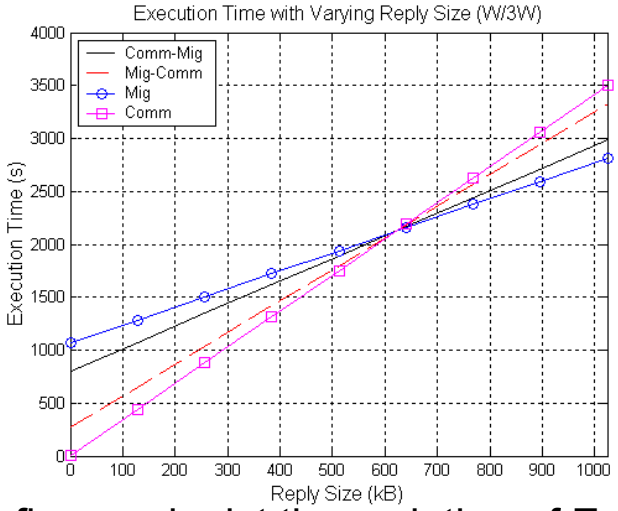



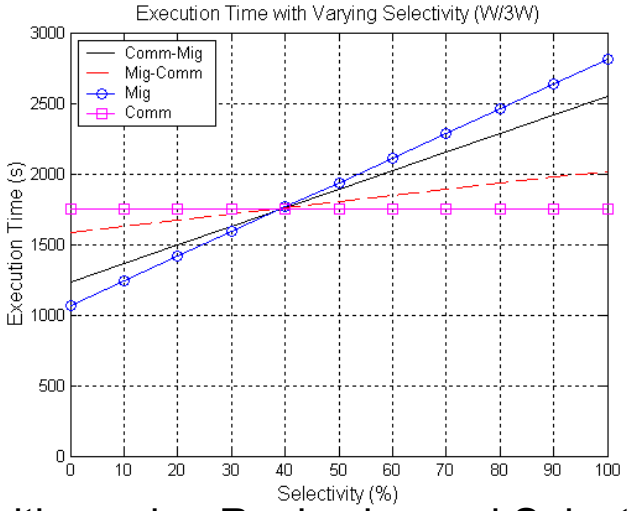

- Above figures show results of Network Load with respect to Reply Size and Selectivity
- Communication is the most sensitive with respect to Reply Size and is independent of selectivity
- Migration is relatively less sensitive with respect to Reply Size and it is also dependent on the selectivity at the same time

# Execution Time in Sequential strategy



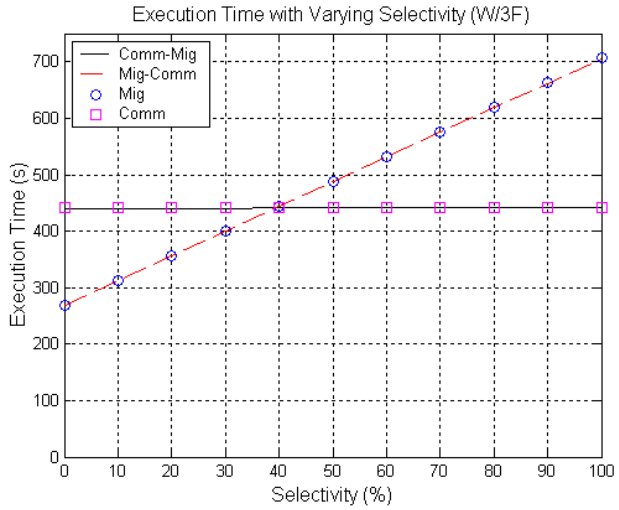



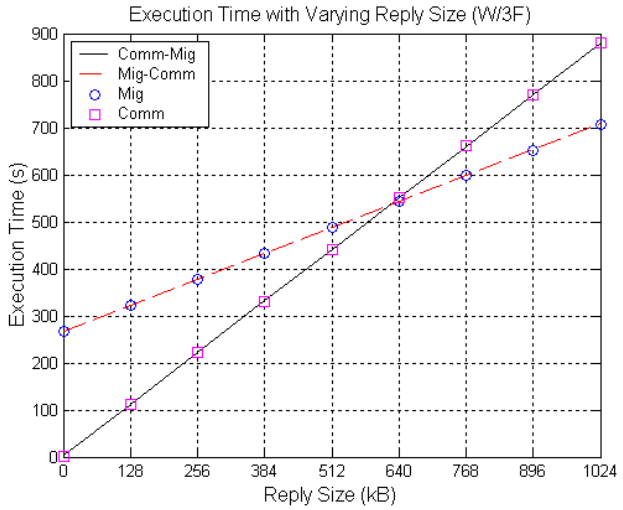



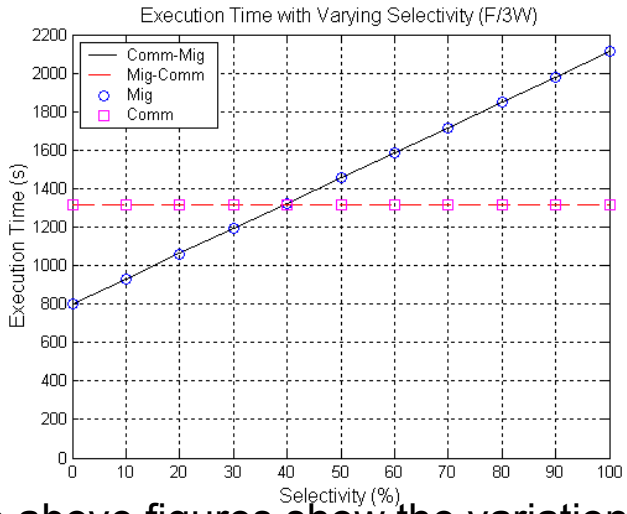



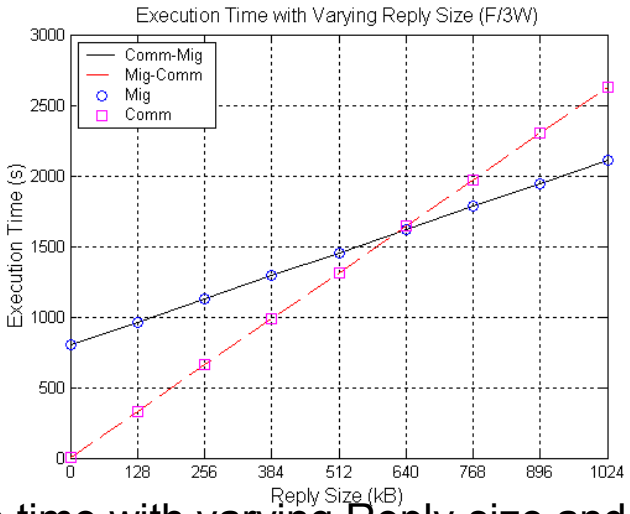

The figures depict the variation of Execution time with varying Reply-size and Selectivity for Wired for the first case and Wireless for other three cases respectively.

# Execution Time in Sequential strategy



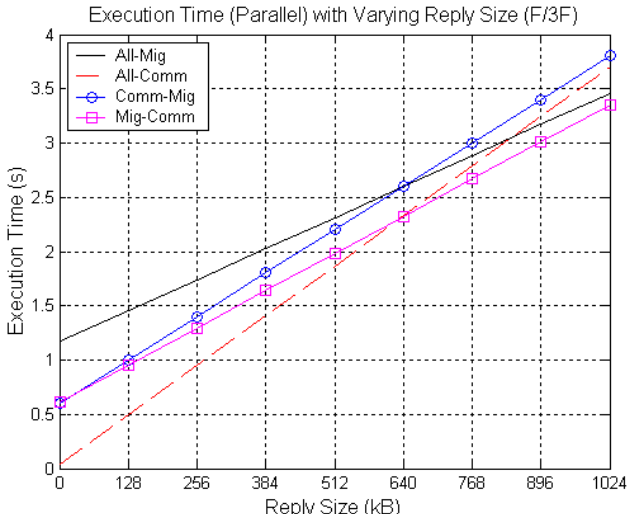



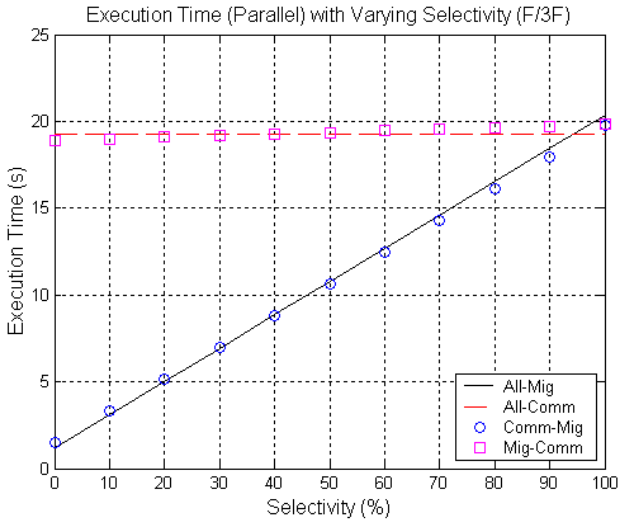



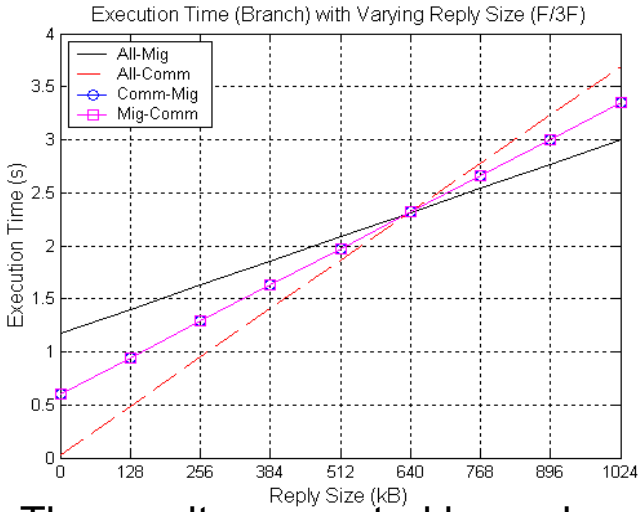



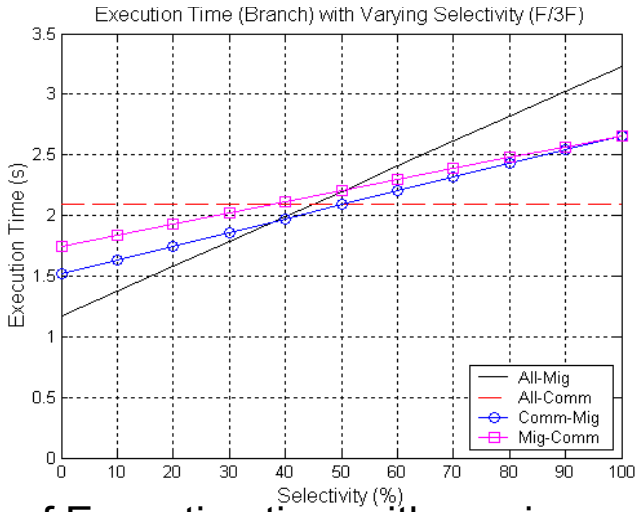

The above figures show the variation of Execution time with varying Reply-size and Selectivity, with single wired and three distributed wireless connections as per the scenario presented in the previous slides.

# Parallel and Branch Strategy



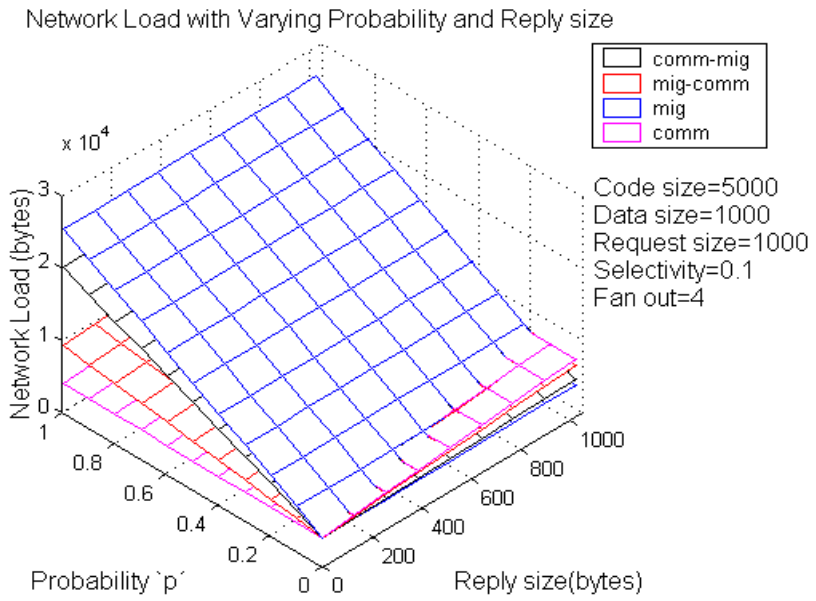



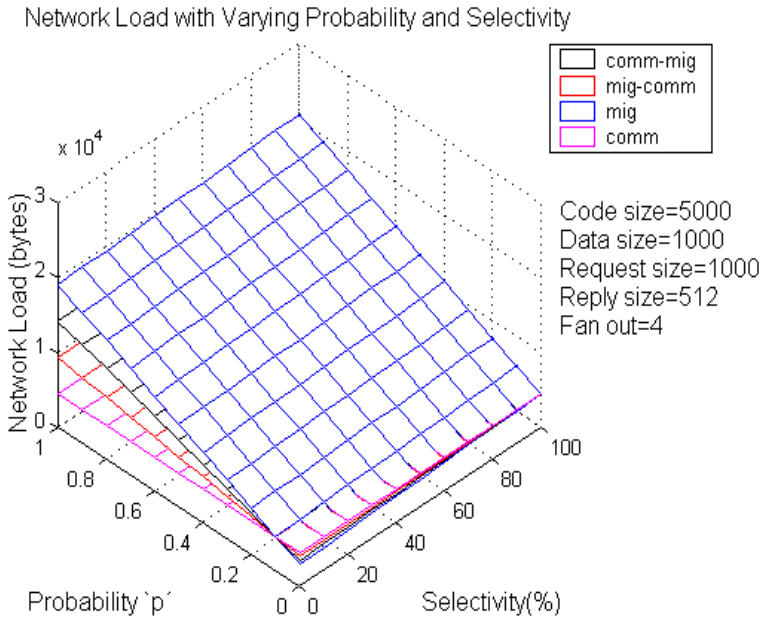




The results presented here show the variation of Execution time with varying Reply-size and Selectivity in the Parallel strategy and Branch strategy, respectively.

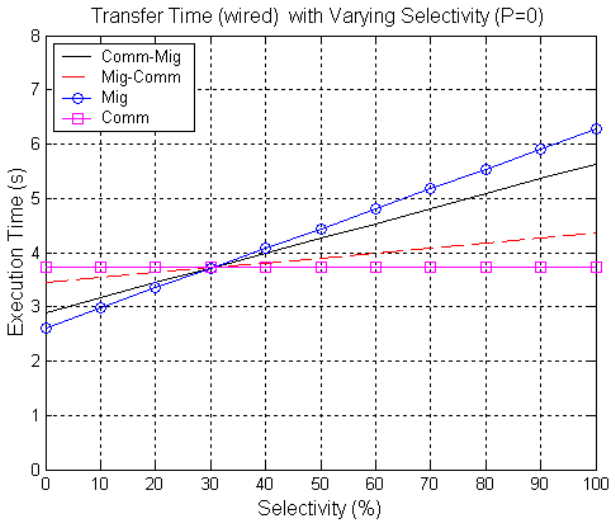
# Network Load (varying probability and reply size)

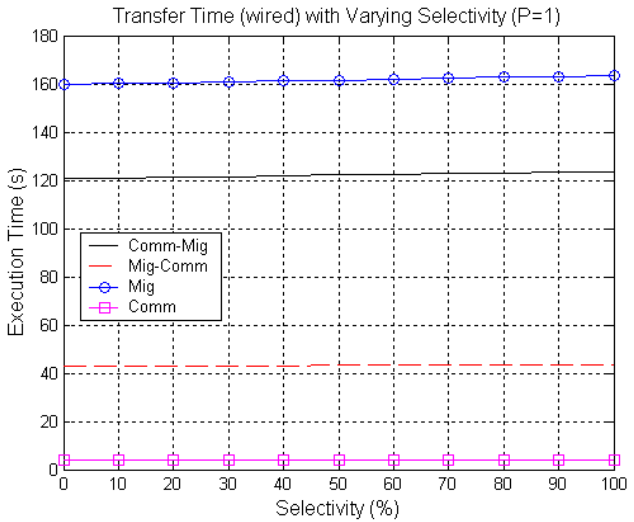


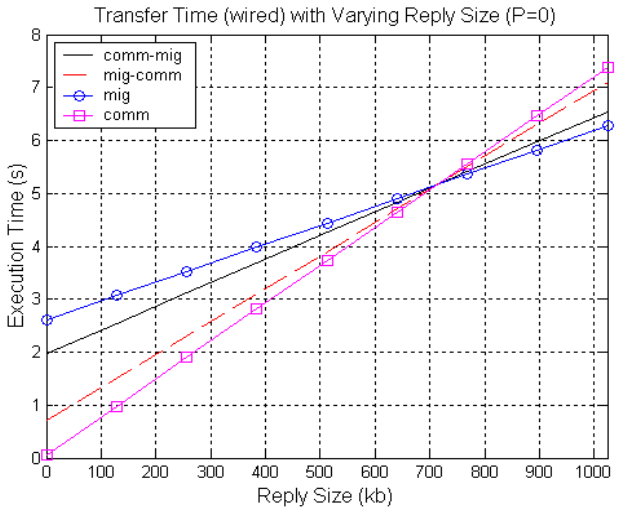



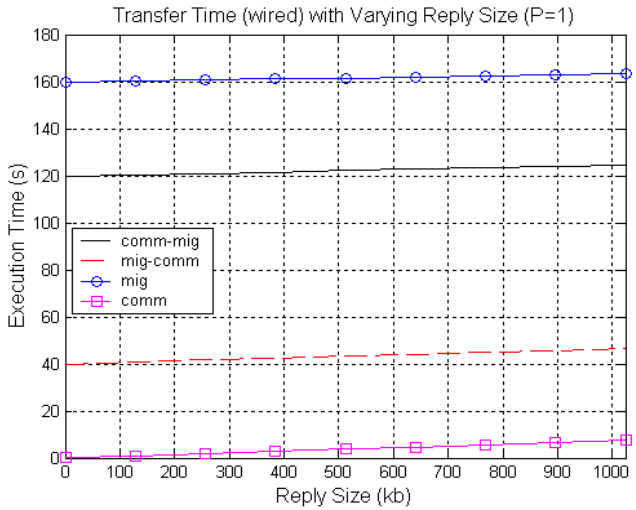

Here, we depict the network load for varying p (probability of code unavailability) and reply size

# Execution Time (probability P=0 and P=1)



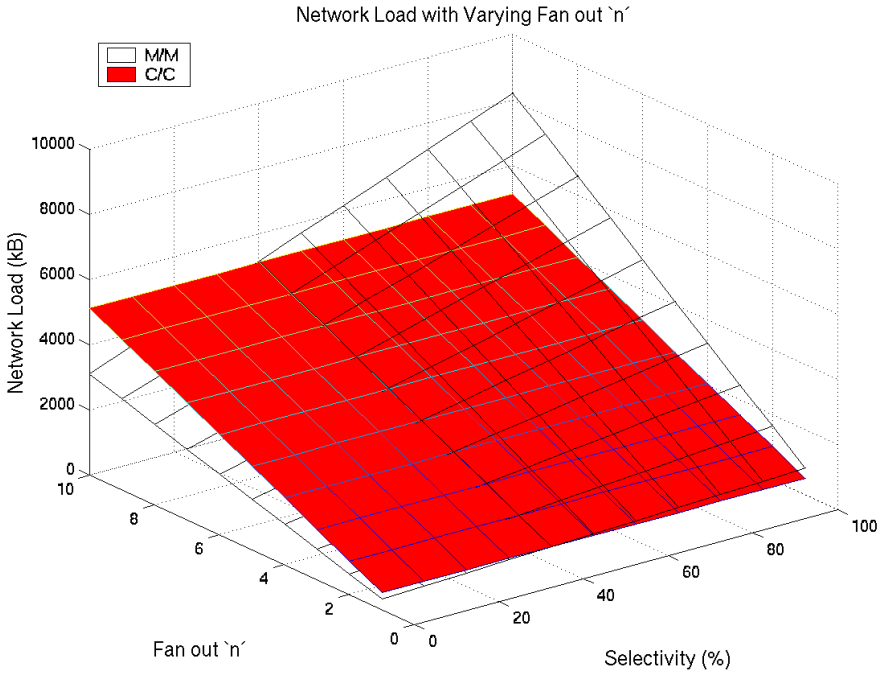



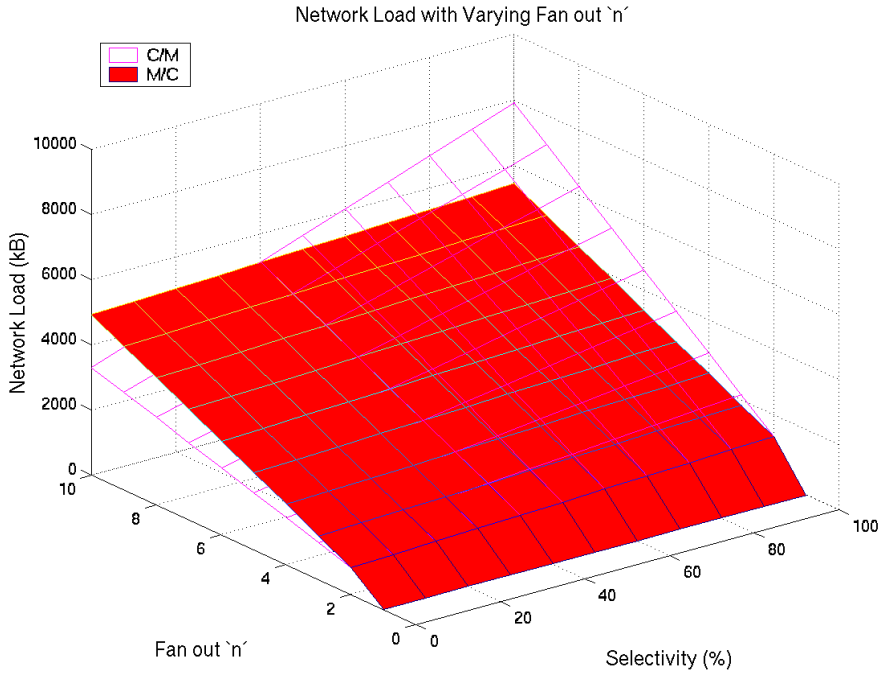




The execution time for varying reply size and selectivity is depicted with P=0 & 1

# Network Load (varying fan out/selectivity)




Here, we depict the network load for varying n (number of destination nodes) and selectivity

# Conclusion and Outlook

- A study was made to see the effect of migration of agents and their communication on the network load and the execution time
- It was also observed that communication was more sensitive with respect to Reply-size where as migration was relatively more sensitive with respect to Selectivity
- We studied three scenarios (Sequential, Parallel and Branch) and made comparisons of their performances
- A study was made to see the effect of varying probability of code inavailibility and reply size on the network load
- The analysis was carried on for varying fan out values and selectivity on the network load