A PERFORMANCE ANALYSIS FRAMEWORK FOR MOBILE-AGENT SYSTEMS

Marios D. Dikaiakos **George Samaras**
Department of Computer Science
University of Cyprus

Speaker: **Marios D. Dikaiakos**
mdd@ucy.ac.cy
http://www.cs.ucy.ac.cy/mdd

Workshop on Infrastructure for Scalable Multi-Agent Systems
The Fourth International Conference on Autonomous Agents 2000

Summary

- A conceptual framework to analyze the performance of MA systems quantitatively
- Materializing this framework as a hierarchy of benchmarks
- Benchmark implementation and experimentation
- Evaluation and restructuring of benchmarks and experiments

Acknowledgements

- Melinos Kyriakou
- Constantine Spyrou
- Dimitris Zeinalipour-Yiazti

Overview

- Motivation.
- A Performance Analysis Framework.
- Benchmarks and Experimentation.
- Conclusions and Future Work.

New Computing Paradigms

- Towards integrated services covering many dimensions:
  - Different levels of user interaction (push vs. pull,…)
  - Spectrum of "user experience" (rich vs. poor)
  - Alternative Connection modalities (wireless-fixed)
  - All different kinds of clients (thin, thick, portable, wearable, home)
- *A paradigm shift* from Client-Server computing towards more flexible schemes that adapt dynamically to the various dimensions of future integrated Internet services.
- **Mobile Agents.**

Objectives

A framework is required to:

- study & argue about performance issues of mobile-agent-based systems
- compare mobile-agent platforms quantitatively
- discover potential performance bottlenecks
- monitor MA-based systems' performance

The Need for Performance Evaluation

- Quantitative performance evaluation is the foundation for:
  - performance debugging and optimization
  - comparison of systems
  - extrapolation of properties of future systems.

- The more complex a system/application is, the harder its evaluation becomes. E.g., in multiprocessor systems:
  - What is a representative workload?
  - Software models not stabilized.
  - Many degrees of freedom in system/application configuration.
  - What are the appropriate metrics?

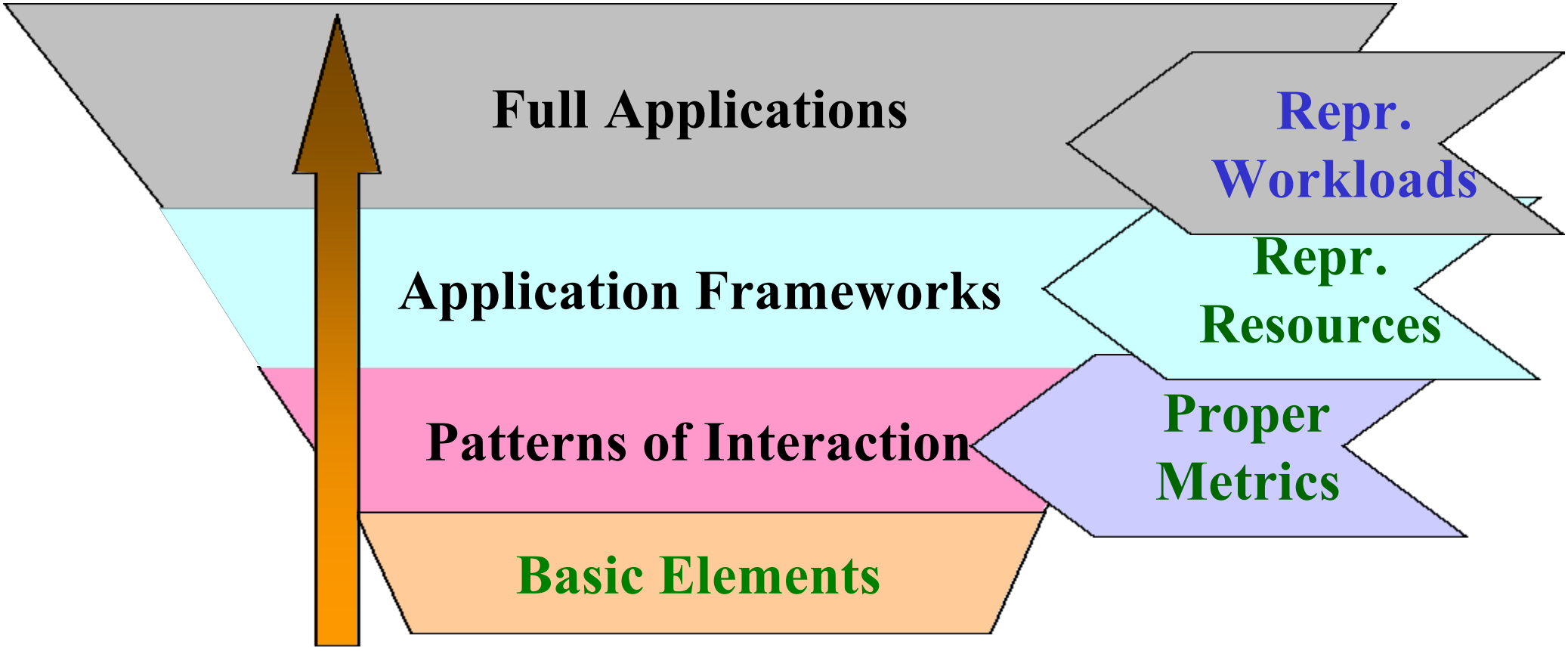
Evaluation of Mobile-Agent Systems

- Quantitative evaluation of mobile-agent-based distributed systems is even harder:
  - The absence of global time, control and state information.
  - The heterogeneity/complexity of platforms: difficult to describe performance properties via small sets of metrics.
  - The variety of distributed computing (software) models.
  - The diversity of operations found in distributed applications: hard to construct simple and portable benchmarks.
  - The flexibility of system configuration: hard to provide concise representation of system resources.
  - Issues affecting performance of JAVA.

Overview

- Motivation.
- **A Performance Analysis Framework.**
- Benchmarks and Experimentation.
- Conclusions and Future Work.

A Performance Analysis Spectrum



Establish causality relationships

A Performance Analysis Framework

- Identify & benchmark basic elements of mobile-agent systems.

  ➔ Agents, Places, Behaviors

- Identify & benchmark patterns of interaction appropriate for mobile-agent applications.

  ➔ Software models for Distr. Computing

- Formulate application frameworks that instantiate relevant software models and can be used in anticipated mobile-agent applications.

  ➔ Database access over the Web

Basic Elements of M.A. Platforms

- *Agents*:

  State, Implementation (code), Interface, Identifier, etc.

- *Places* (environment where agents execute):

  Engine, Resources, Location

- *Behaviors* (within and between places):

  Creation, Transfer, Arrival, Communication via messages and agents, Multicasting, Synchronization.

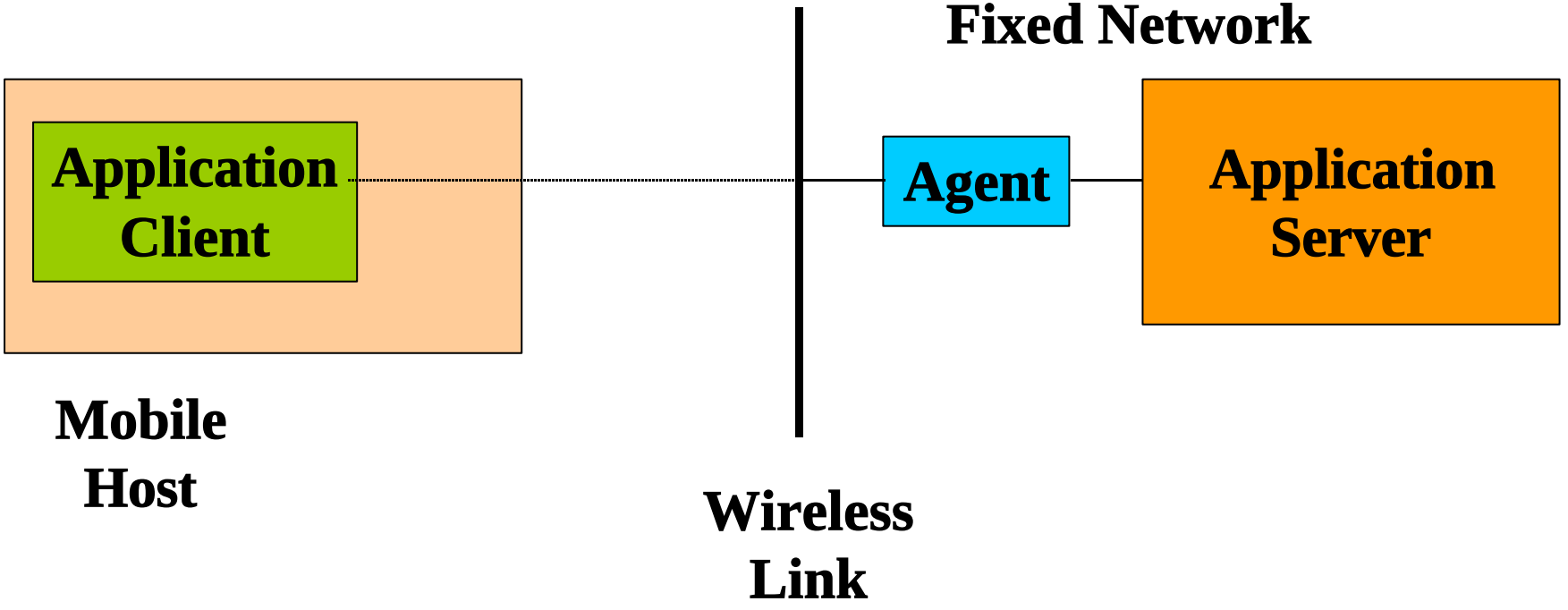
Software Models

- Patterns of Interaction or Agent Design Patterns:
  - Represent the synthesis of basic MA behaviors into more complex frameworks of MA behavior and interaction, which are common to many MA-based systems.
  - Encoded as *Software (Distributed-Computing) Models.*

- We focus on:
  - Distr. Computing Models: the Client-Server model and extensions: *C/S, C/A/S, C/I/S*
  - Agent Design Patterns: *Proxy*, *Router*, *Meeting*

Client-Agent-Server Model

Client-Agent-Server (C/A/S)



M.A. Application Frameworks

- Mobile-Agent Application Frameworks: implementation of software models, using MA, for:
  - Particular applications
  - Under characteristic workloads
- Application Frameworks are libraries of mobile-agent routines, materializing some software model and implementing core sets of services for a particular application.
- We examine application frameworks for Database-access provision over the Web. Generate characteristic workloads according to TPC-W benchmark suite.

Overview

- Motivation.
- A Performance Analysis Framework.
- **Benchmarks and Experimentation.**
- Conclusions and Future Work.

Benchmarking MA Systems

- Micro-benchmarks: short codes designed to isolate and measure performance properties of basic “behaviors” of mobile-agent-based systems for typical system configurations.

- Micro-kernels: short, synthetic codes designed to measure and investigate performance properties of software-model implementations, for typical applications and system configurations.

- Application kernels: instantiations of micro-kernels for particular application domains and for typical workloads derived from the **TPC-W** (Web Commerce) specification.

Micro-benchmarks

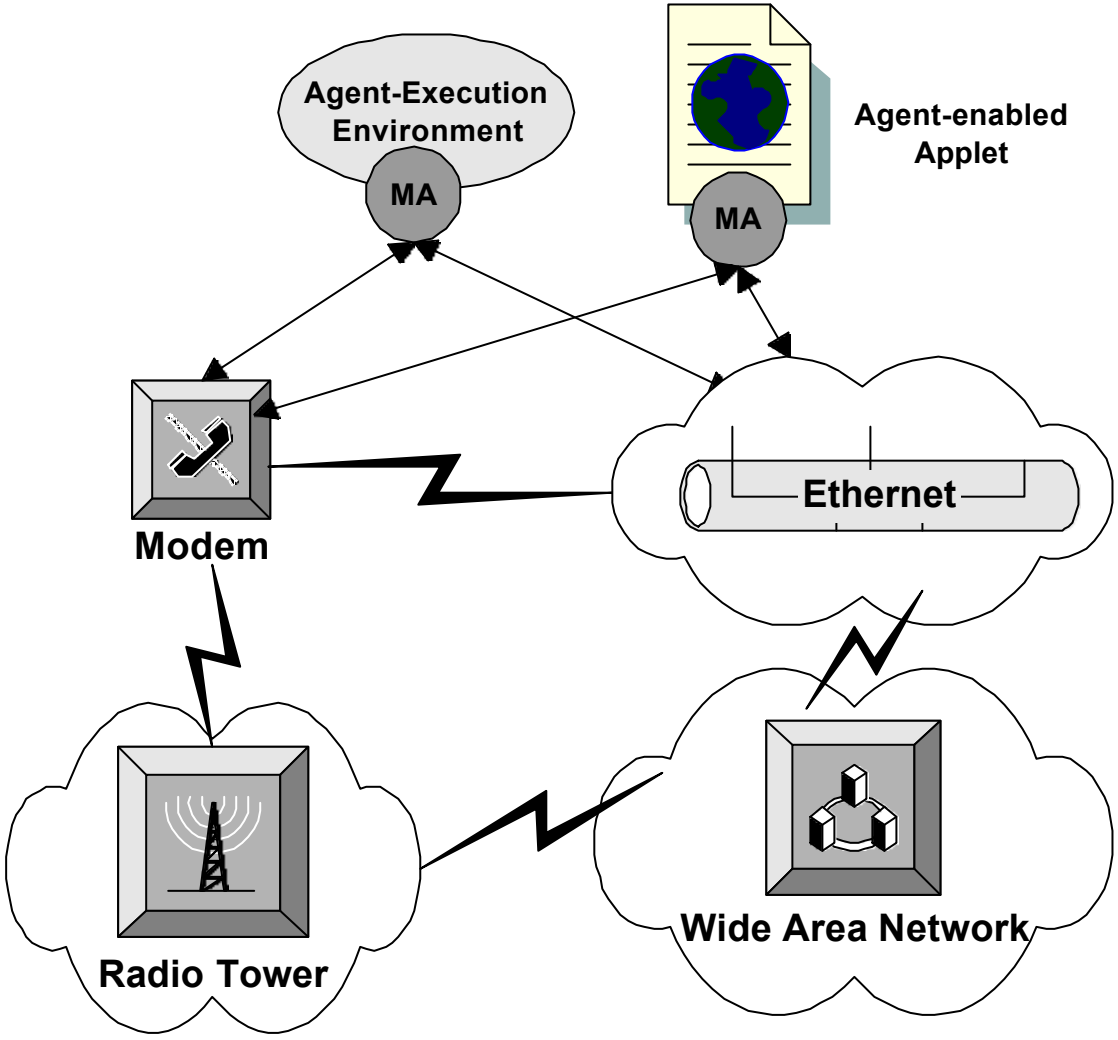
- Key software components:
  - *Mobile Agents* to materialize modules of C/S, C/A/S, etc.
  - *Messenger Agents* for flexible communication.
  - *Messaging* for efficient communication and synchronization.
- **[AC-L]**: captures the overhead of local agent-creation.
- **[AC-R]**: captures the overhead of remote agent-creation.
- **[AL]**: captures the overhead of agent-launching.
- **[AR]**: captures the overhead of receiving an incoming agent.
- **[MSG]**: captures point-to-point messaging overhead.
- **[MULT]**: captures multicasting overhead.
- **[SYNCH]**: captures synchronization overhead.
- **[ROAM]**: captures agent-travelling overhead.

Micro-benchmark Parameters

Tentative List:

- Configuration of *places* where agents reside, roam and perform basic behaviors.
- Configuration of *channels* used by agents in their movements from place to place.
- Number of iterations executed.
- Mobile Agent-size.

Places and Channels



Places

Channels

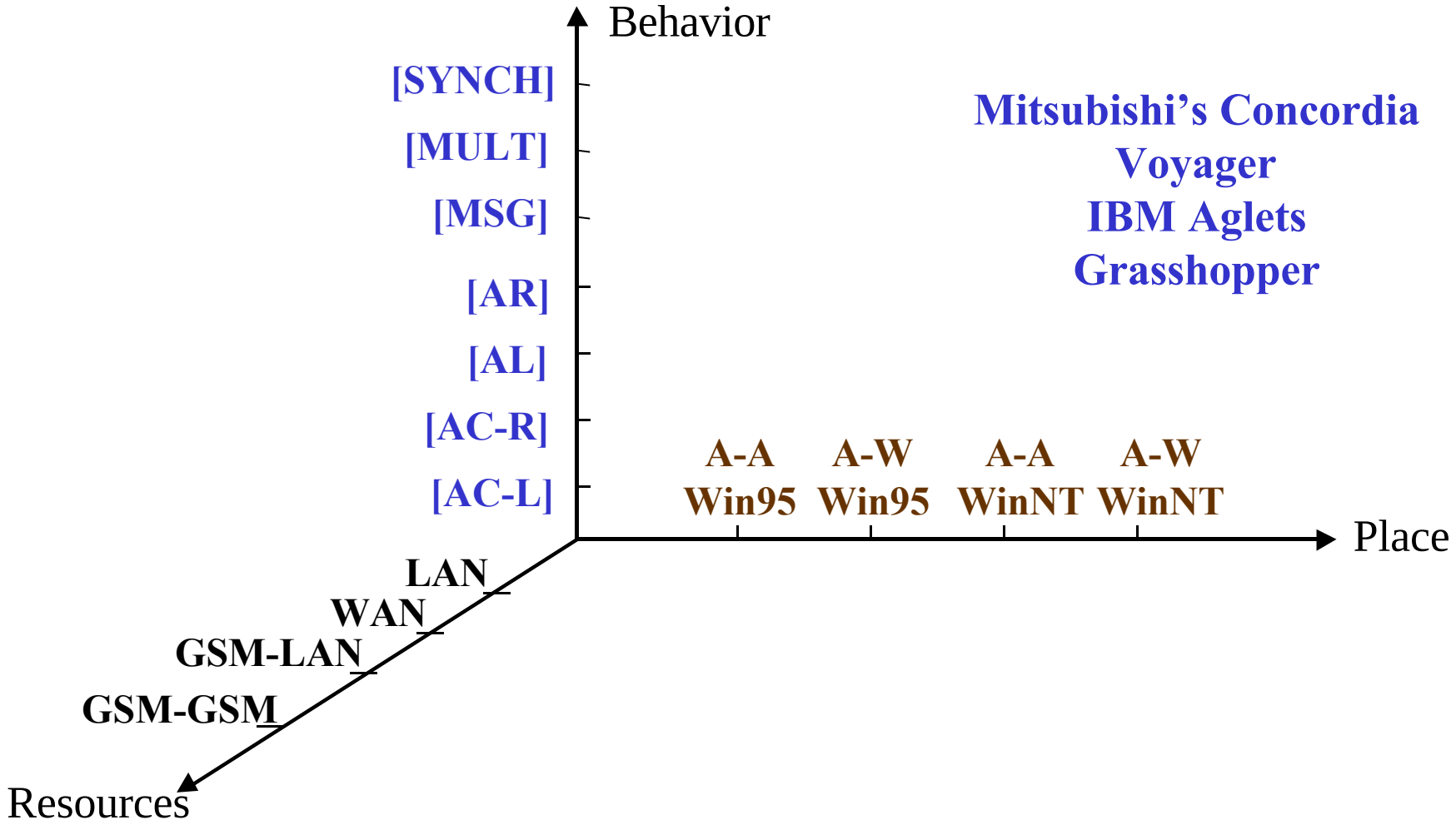
Metrics

- *Aggregate time to completion*:
  - Raw performance measurements.
  - Performance scaling under various load-conditions.
  - Identification of bottlenecks & performance problems.
  - Examination of platform-robustness.
- *Peak Rates*:
  - Sustained performance under "ideal" conditions.
  - Quantitative comparisons.

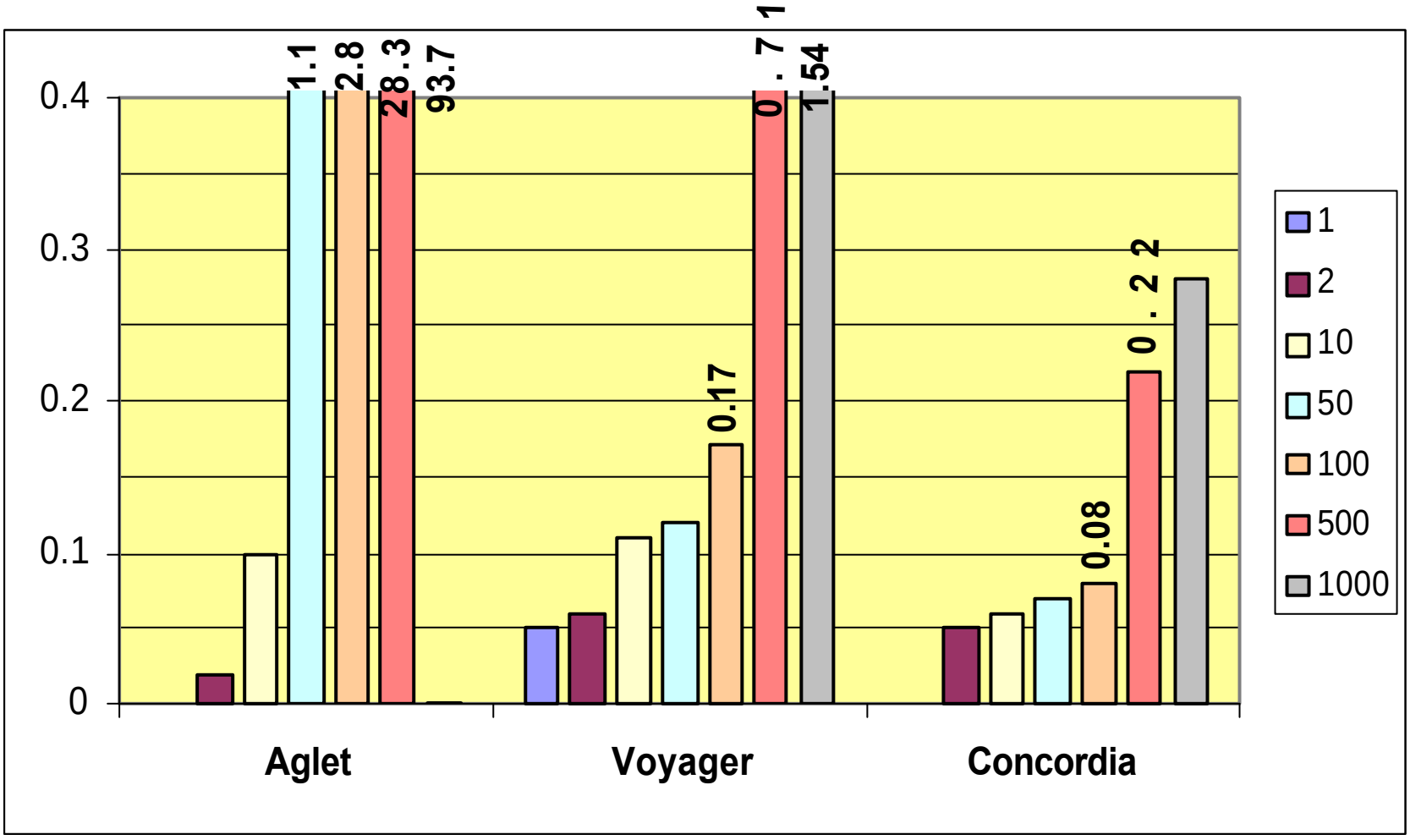
Micro-benchmark Experiments

- We have implemented the benchmarks with four Java-based platforms: IBM's Aglets, Mitsubishi´s Concordia, Voyager and Grasshopper.
- We are currently running tests on a LAN; in the near future we shall repeat them across different LANs of our WAN, and on top of GSM connections.
- We are testing two scenarios:
    1. Full agent-execution environment installed on client.
    2. Client with minimal resources-configuration downloads "agent-aware"applet

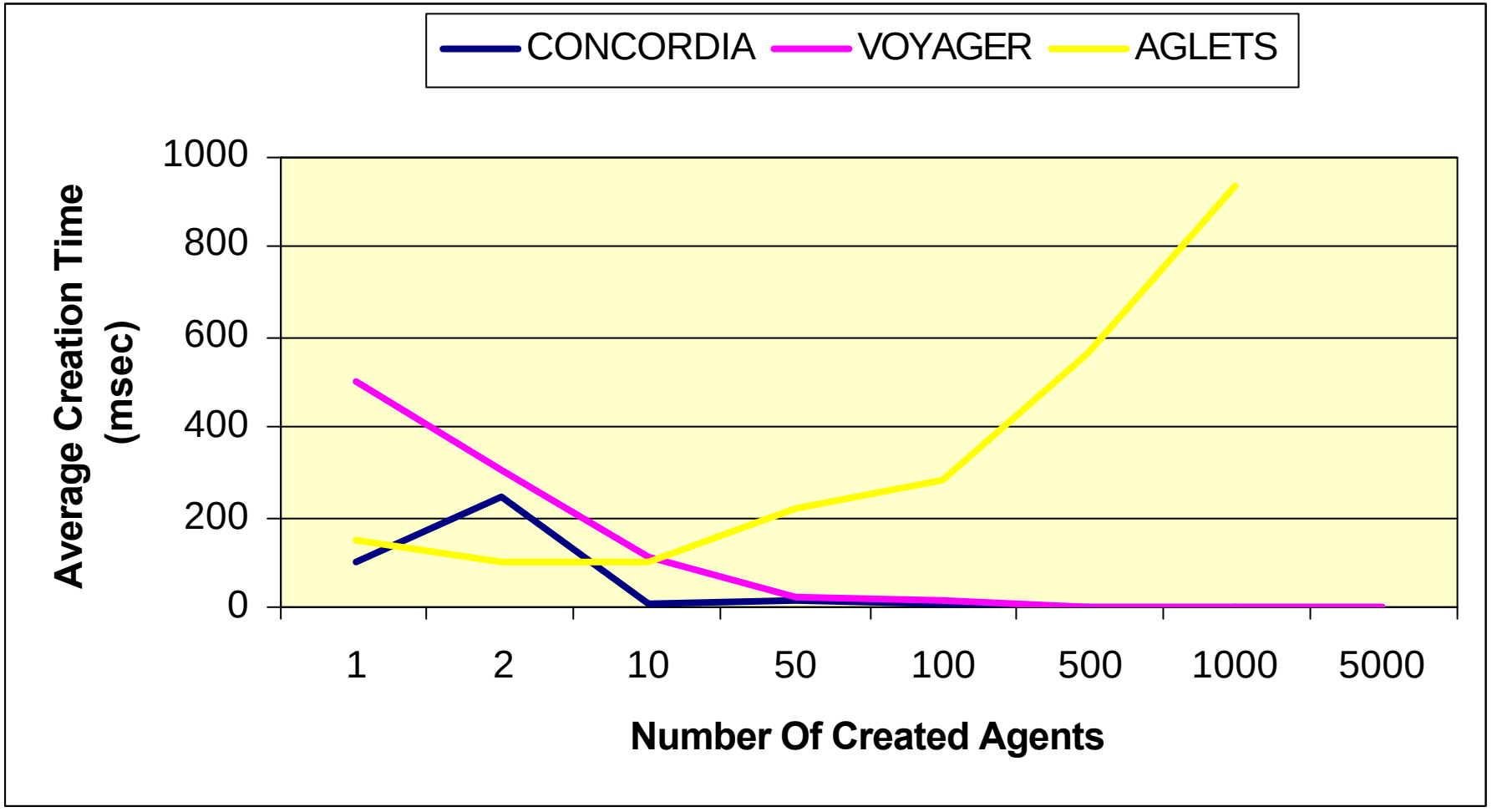
Micro-benchmark Experiments



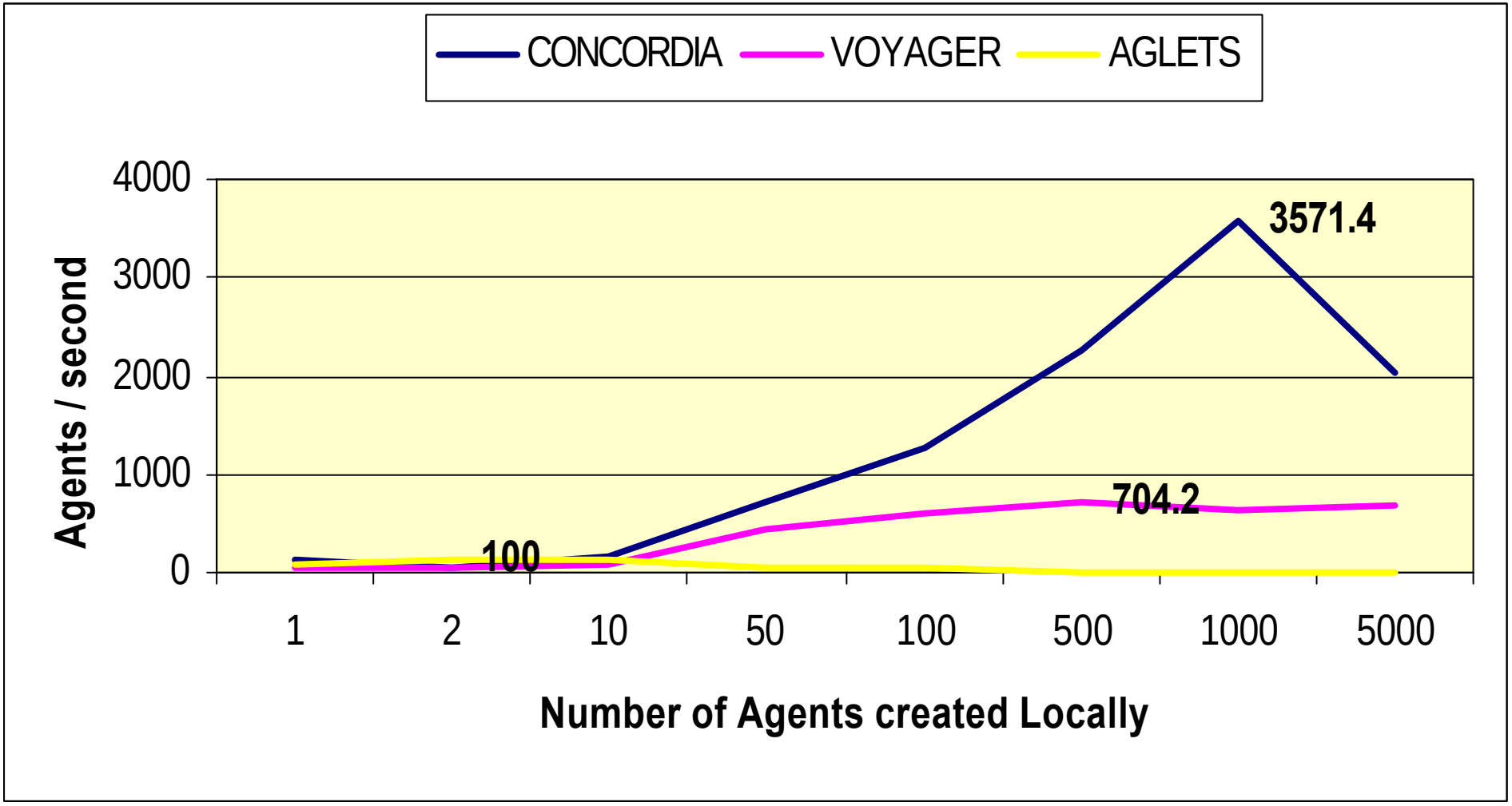
Agent Creation-Local (Win 95)



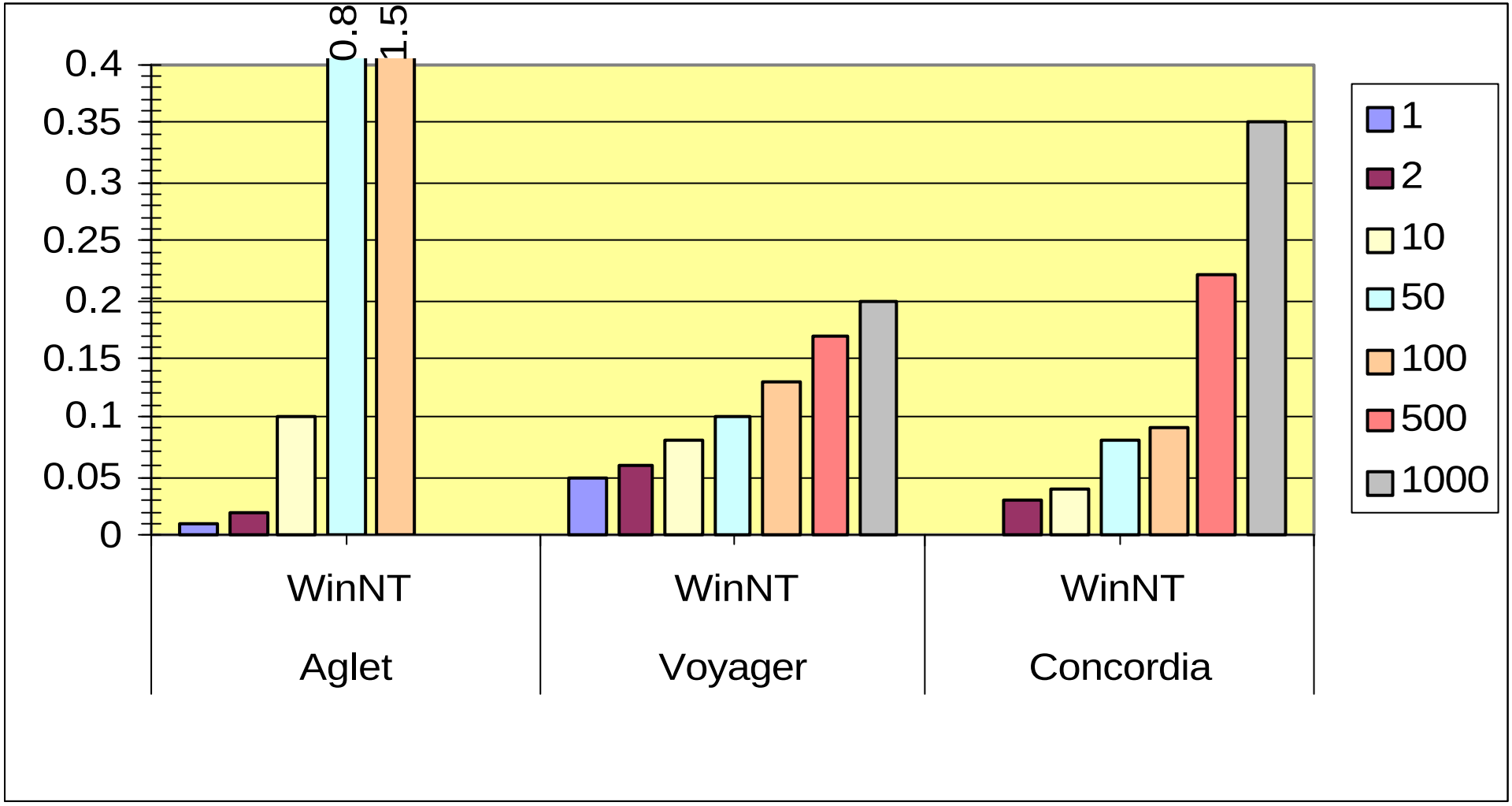
AC-L Benchmark: Average Timings



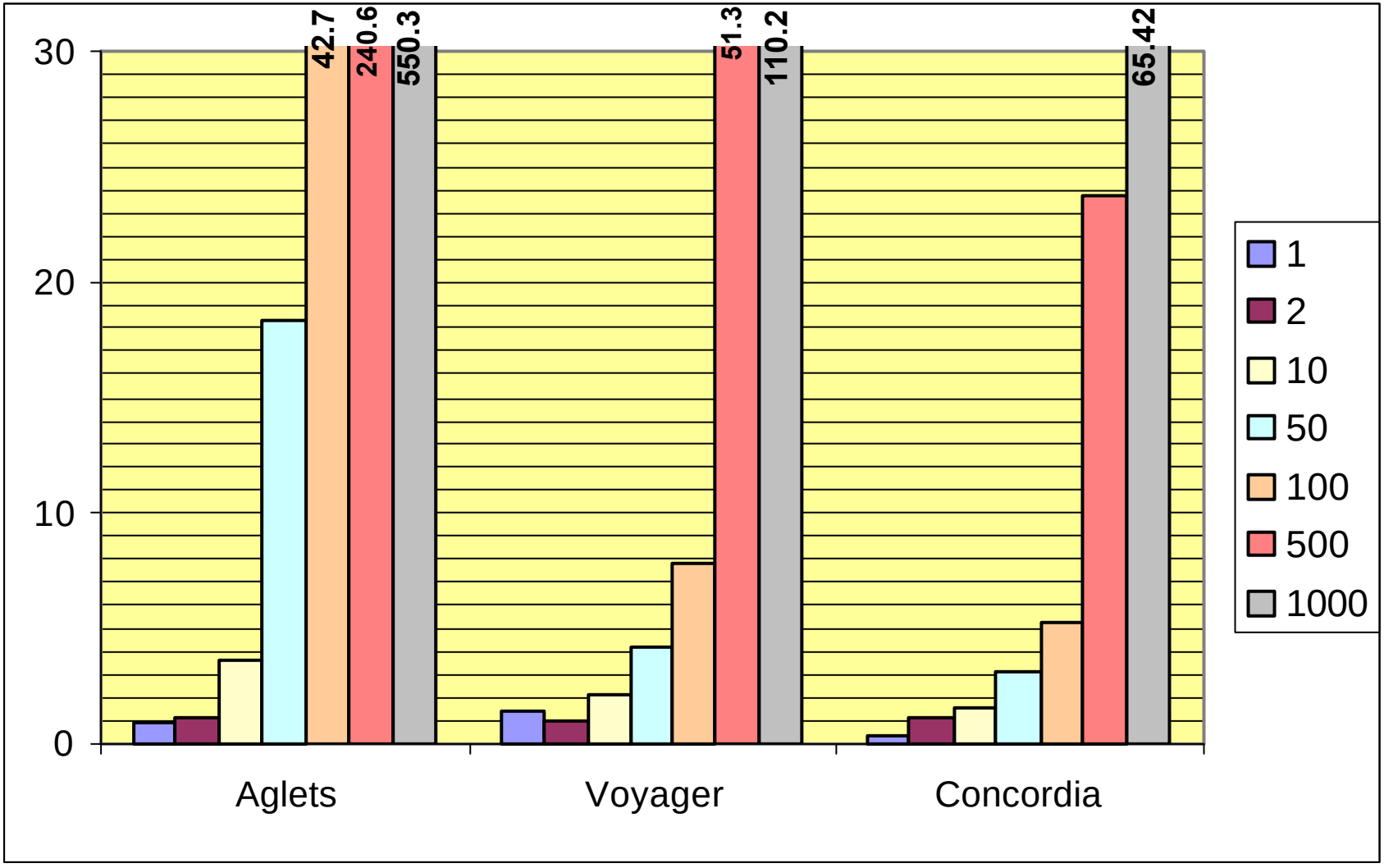
AC-L: Rates of Agent Creation



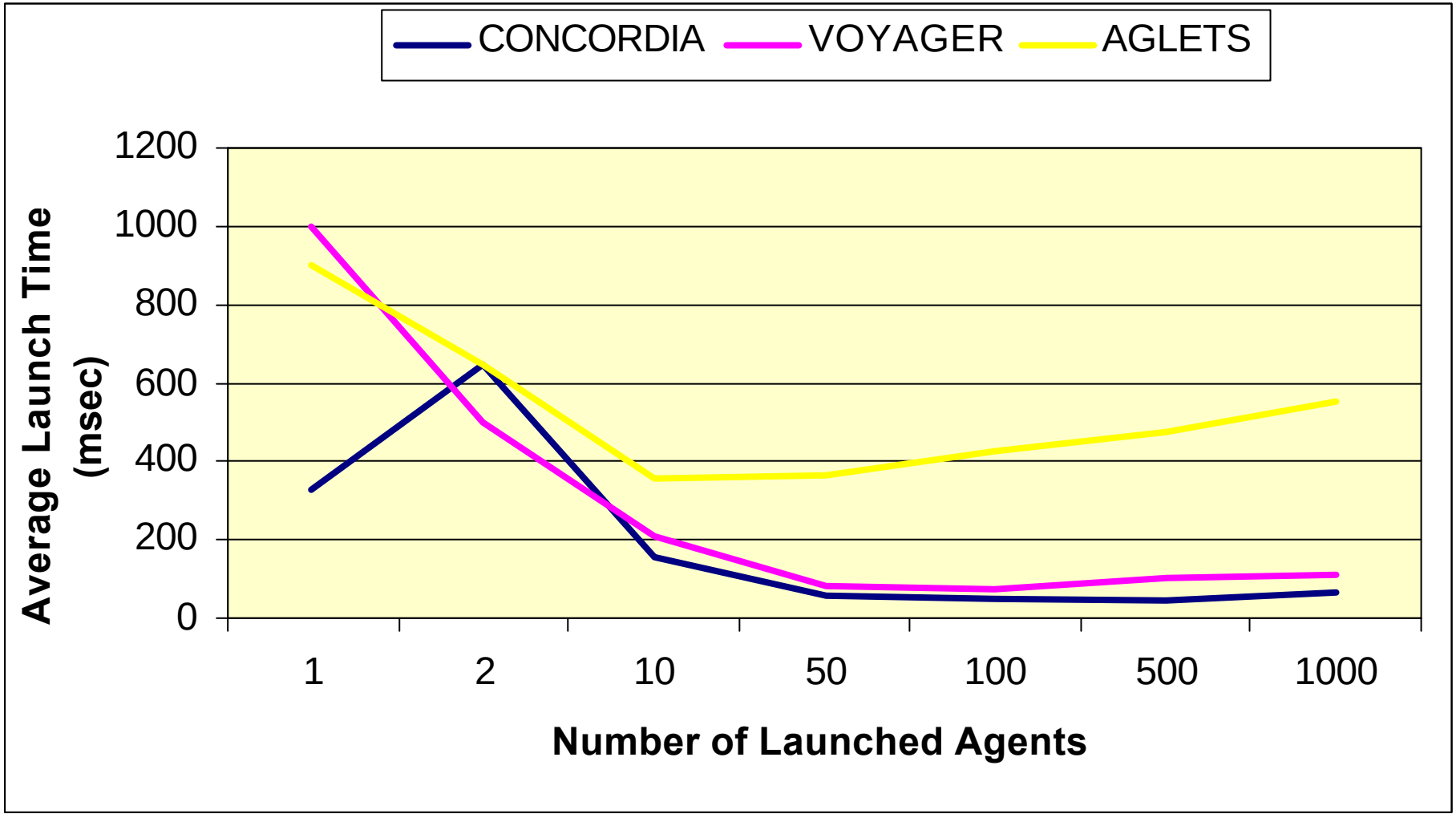
AC-L Benchmark (Win NT)



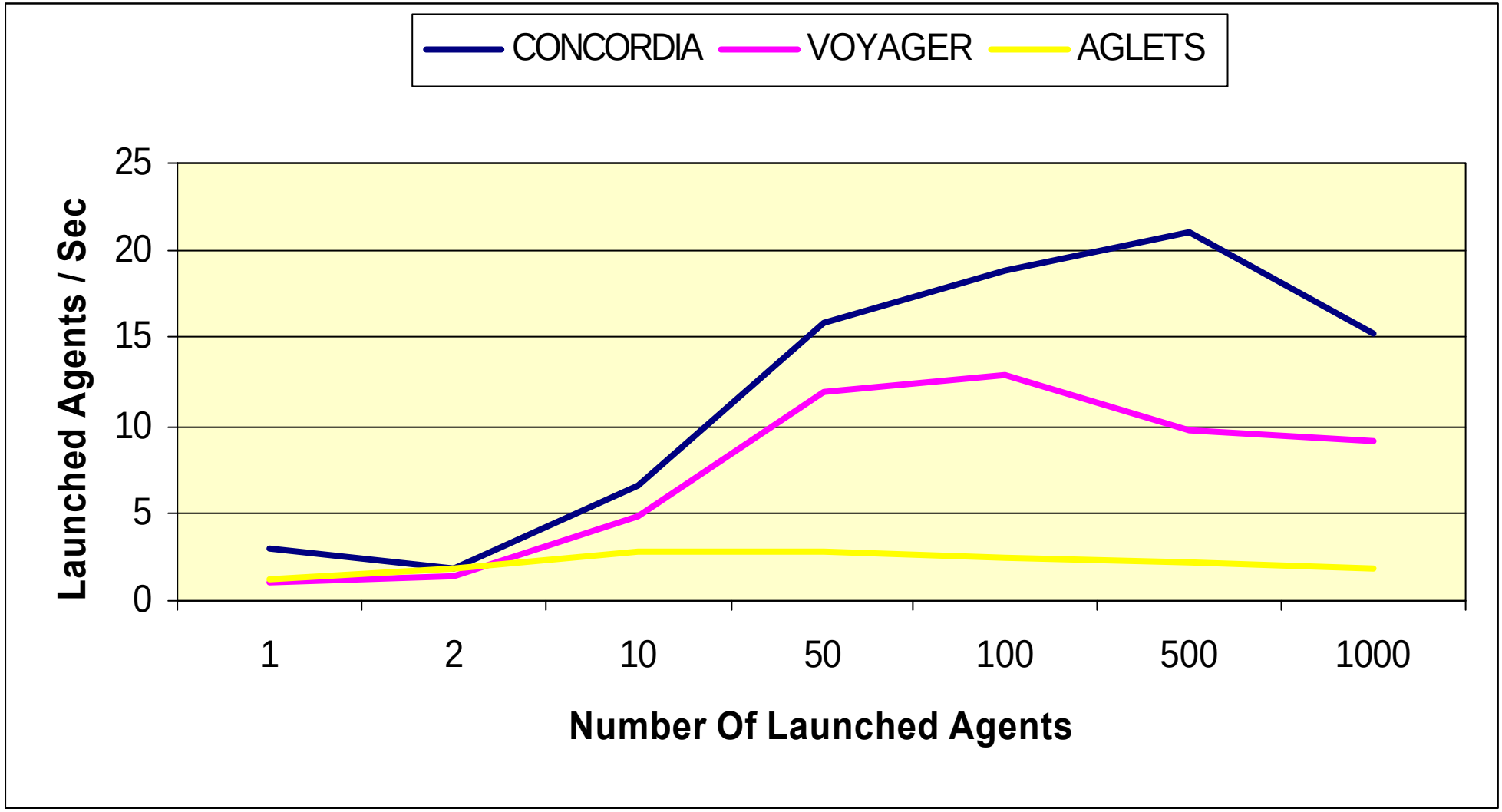
Agent Launching Benchmark (Win95)



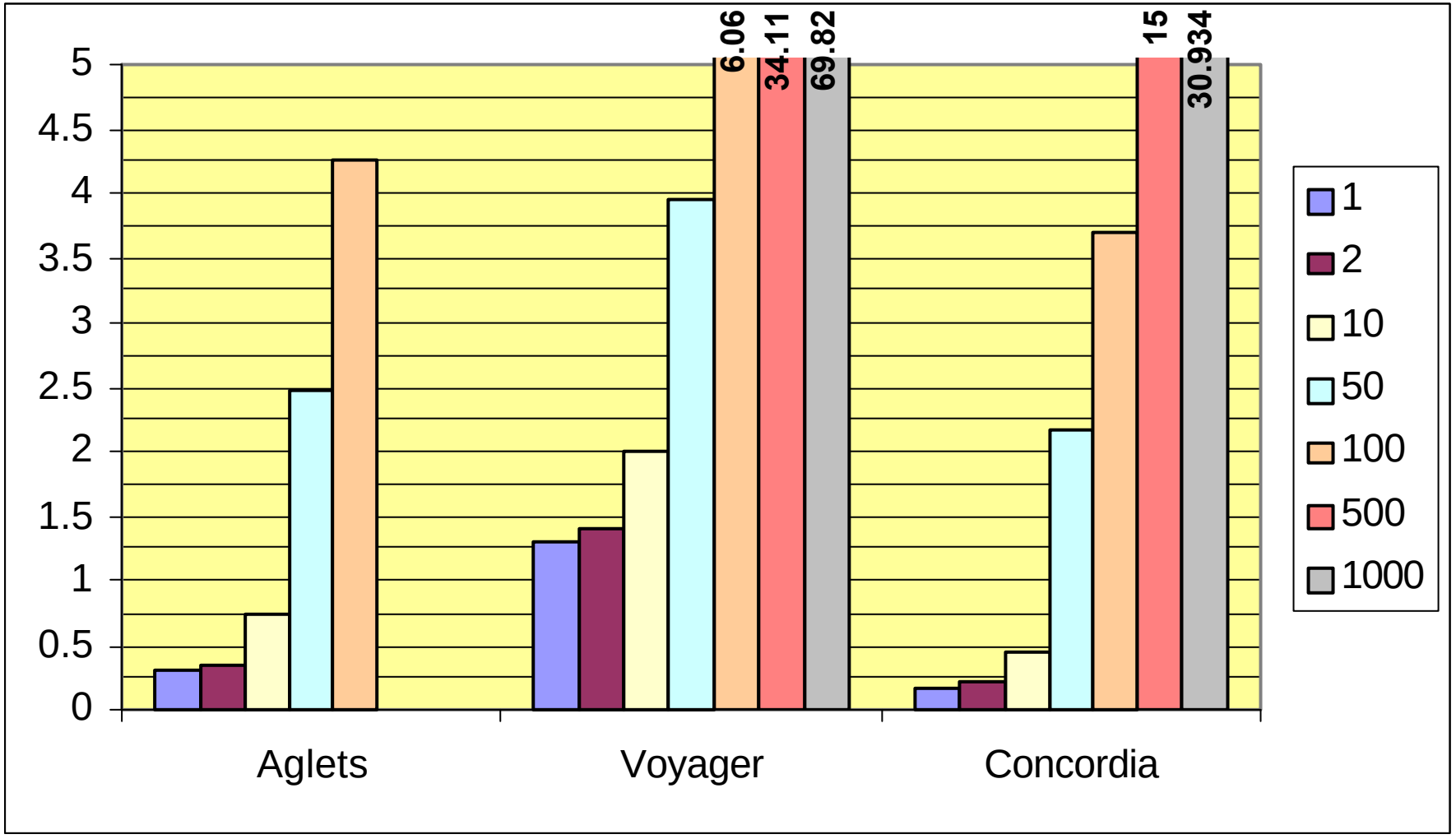
AL Benchmark: Average Timings



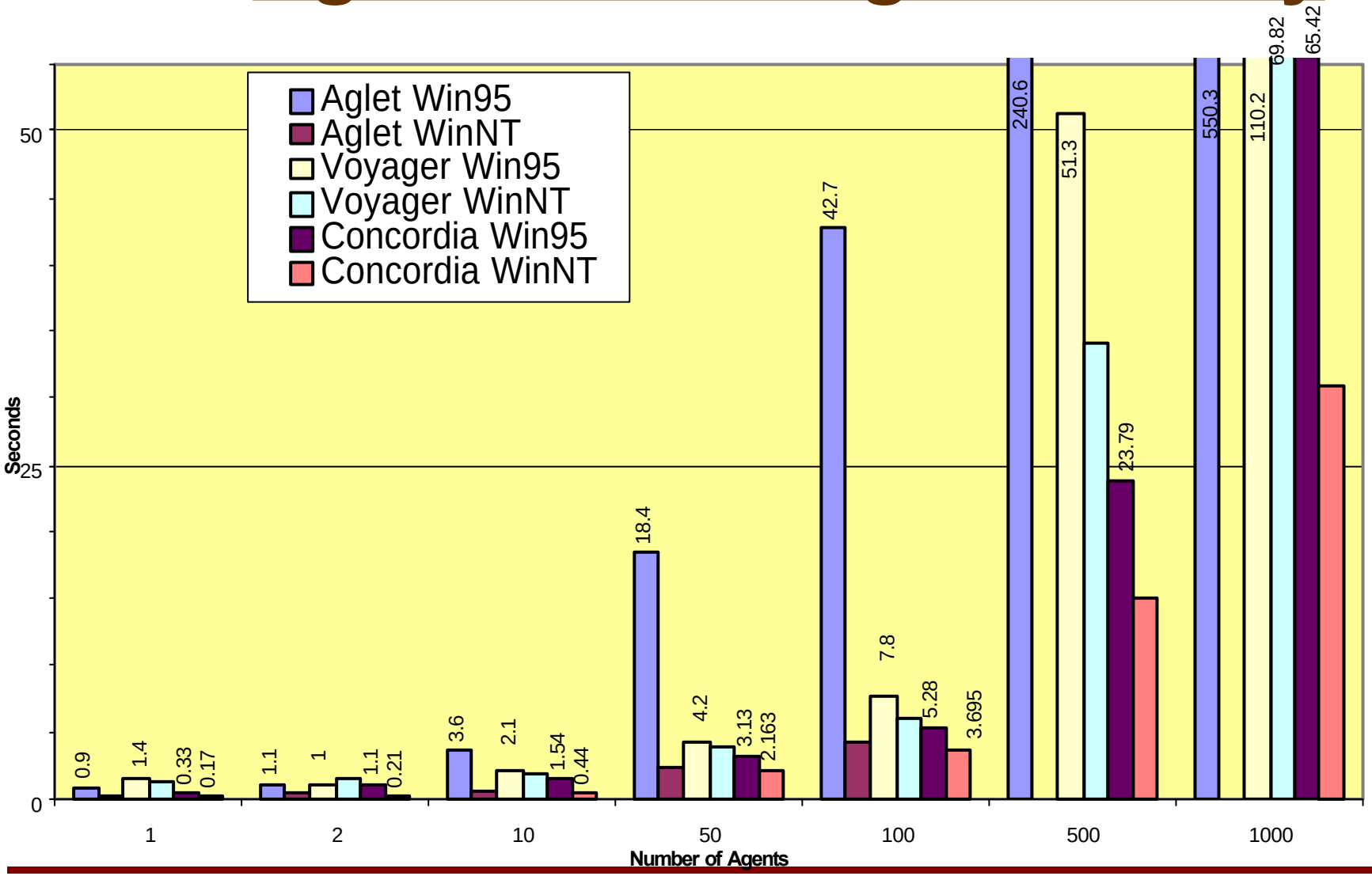
AL: Rates of Agent Launching



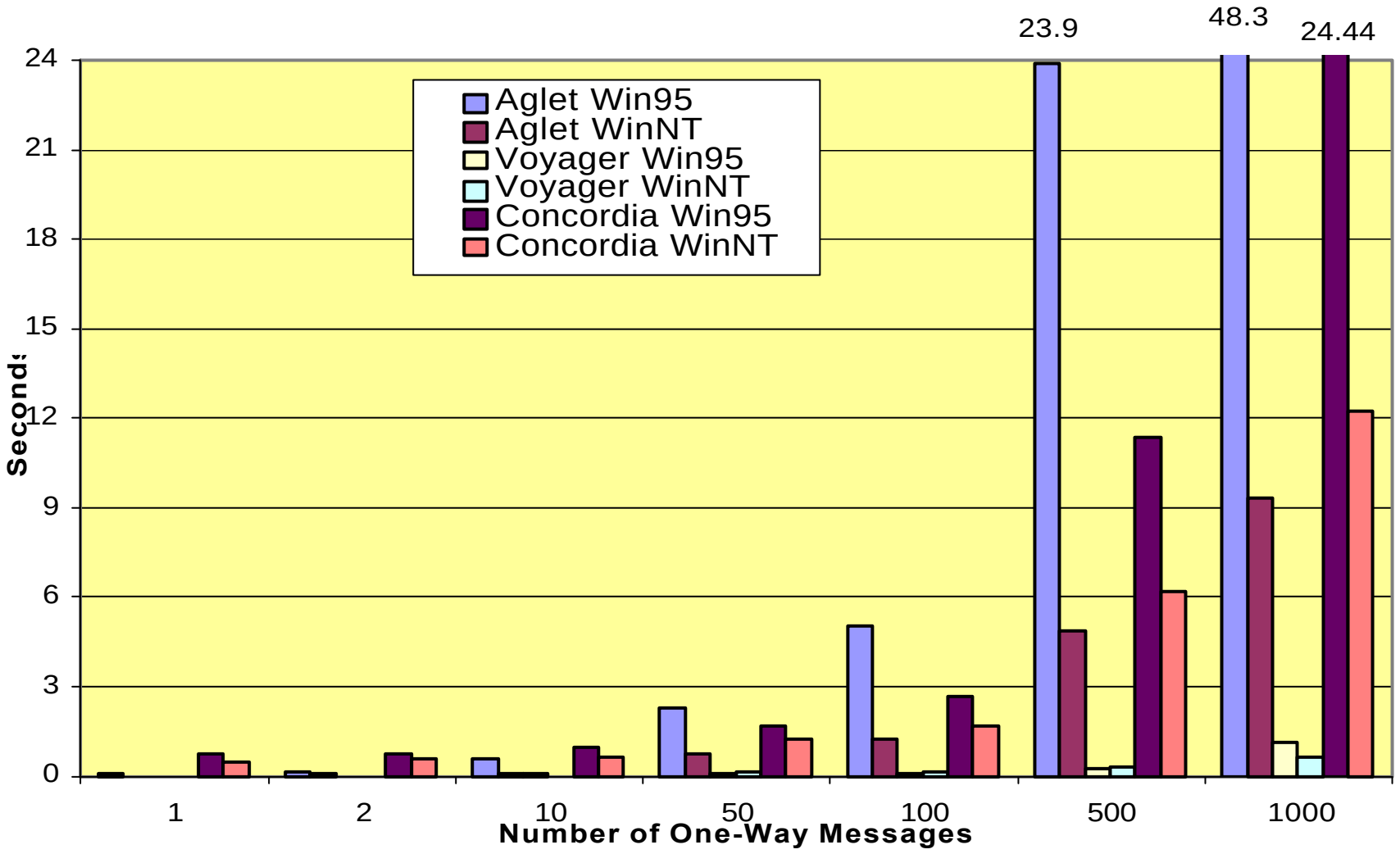
Agent Launch Benchmark (WinNT)



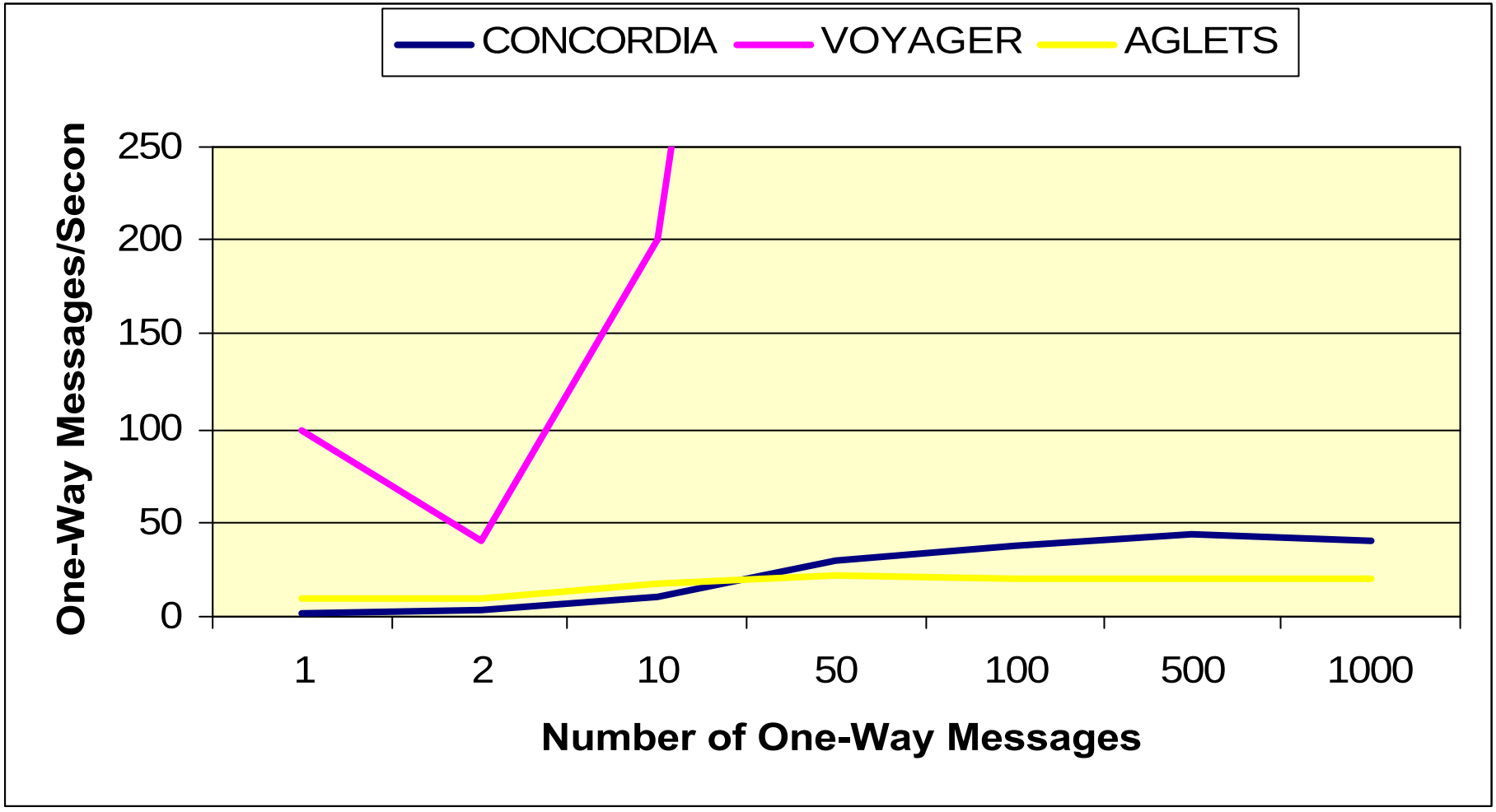
Agent Launching: Summary



MSG: One-way Messaging



MSG: Rate of Message Dispatch



An Assessment of Micro-benchmarks

- Micro-benchmarks provide useful insights into:
  - The behavior of Mobile-Agent Systems.
  - The factors that determine Mobile-Agent performance.
- Initial micro-benchmark results guide the redesign of old or the deployment of new micro-benchmarks.
- Expand our understanding on MA systems and their capabilities.
- Help us understand and explain the performance of micro-kernels.

More Conclusions

- Performance of "light" MA depends a lot on:
  - Loading and caching classes to main and remote memories.
  - Robustness and performance of MA servers under heavy load.
- Concordia shows a performance advantage over Aglets and Voyager in agent creation and launching.
- Voyager is a clear winner when it comes to messaging.
- Applets cannot sustain efficient and robust mobile-agent activity.
- WinNT provide more stable performance measurements than Win95.