A Mobile Agents based Distributed Speech Recognition Engine for Controlling Multiple Robots

Mayank Gupta, Pulkit Verma,

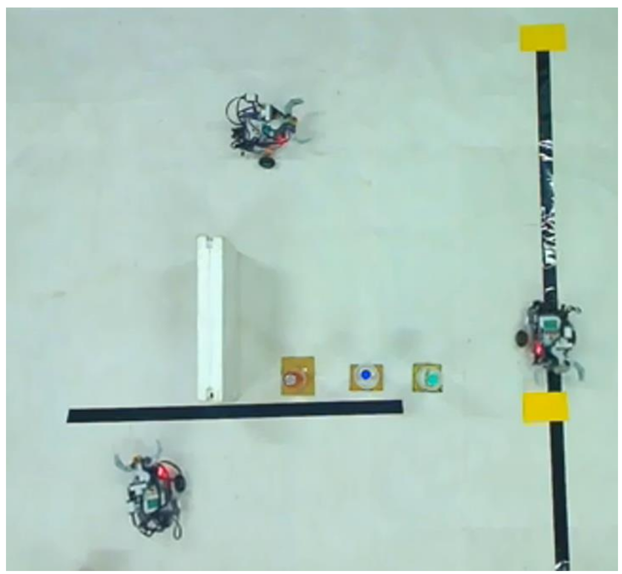
Tuhin Bhattacharya, Pradip K. Das

Robotics Lab

Indian Institute of Technology Guwahati

IIT Guwahati

Department of CSE, IIT Guwahati



Overview

- Introduction
- Proposed System
- Architecture
- Experiments
- Results
- Conclusion

Overview

- Introduction
- Proposed System
- Architecture
- Experiments
- Results
- Conclusion

Introduction

- Talking to a robot to control it.
- Speaking to one robot is doable but what if there are multiple robots, languages and people.
- Challenges:
  - Not possible to have a full-fledged speech recognition engine installed on a robot.
  - A shared speech engine can be used, but not when the number of robots is more (will slow down the whole system).

Introduction

- A distributed recognition system is needed.
- Should be able to recognize different languages.
- Adding new commands should be easy and on-the-fly.

Overview

- Introduction
- Proposed System
- Architecture
- Experiments
- Results
- Conclusion

Proposed Approach

- A distributed and scalable speech recognition engine.
- Based on a mobile agents based framework.
- The number of robots can be increased on the fly.
- The recognition algorithm can also be changed or a new algorithm can be added whenever needed.

Proposed Approach

- Different algorithms work well in different environments and for different languages.
- The given approach can support multiple algorithms in parallel.

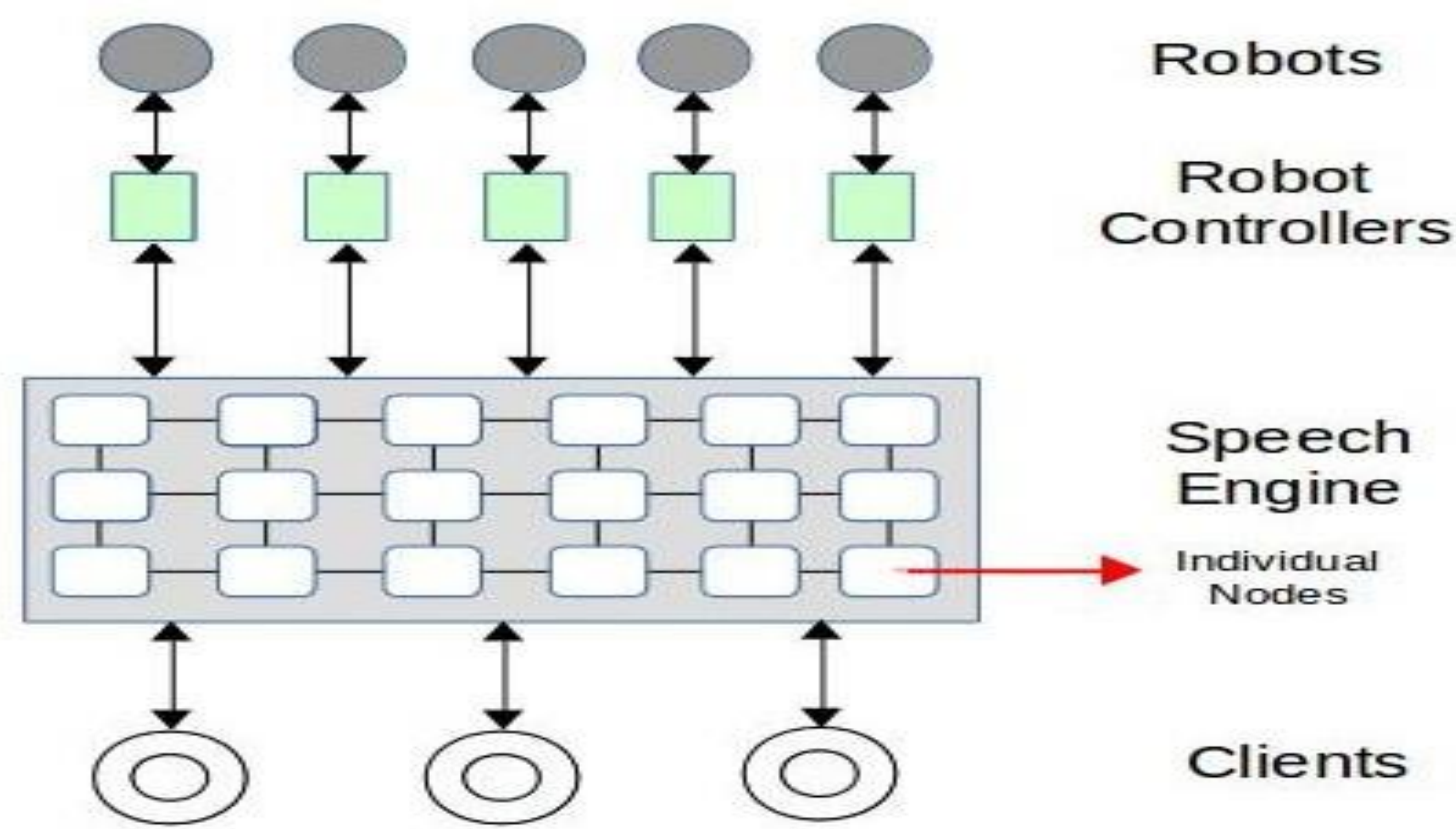
Mobile Agents

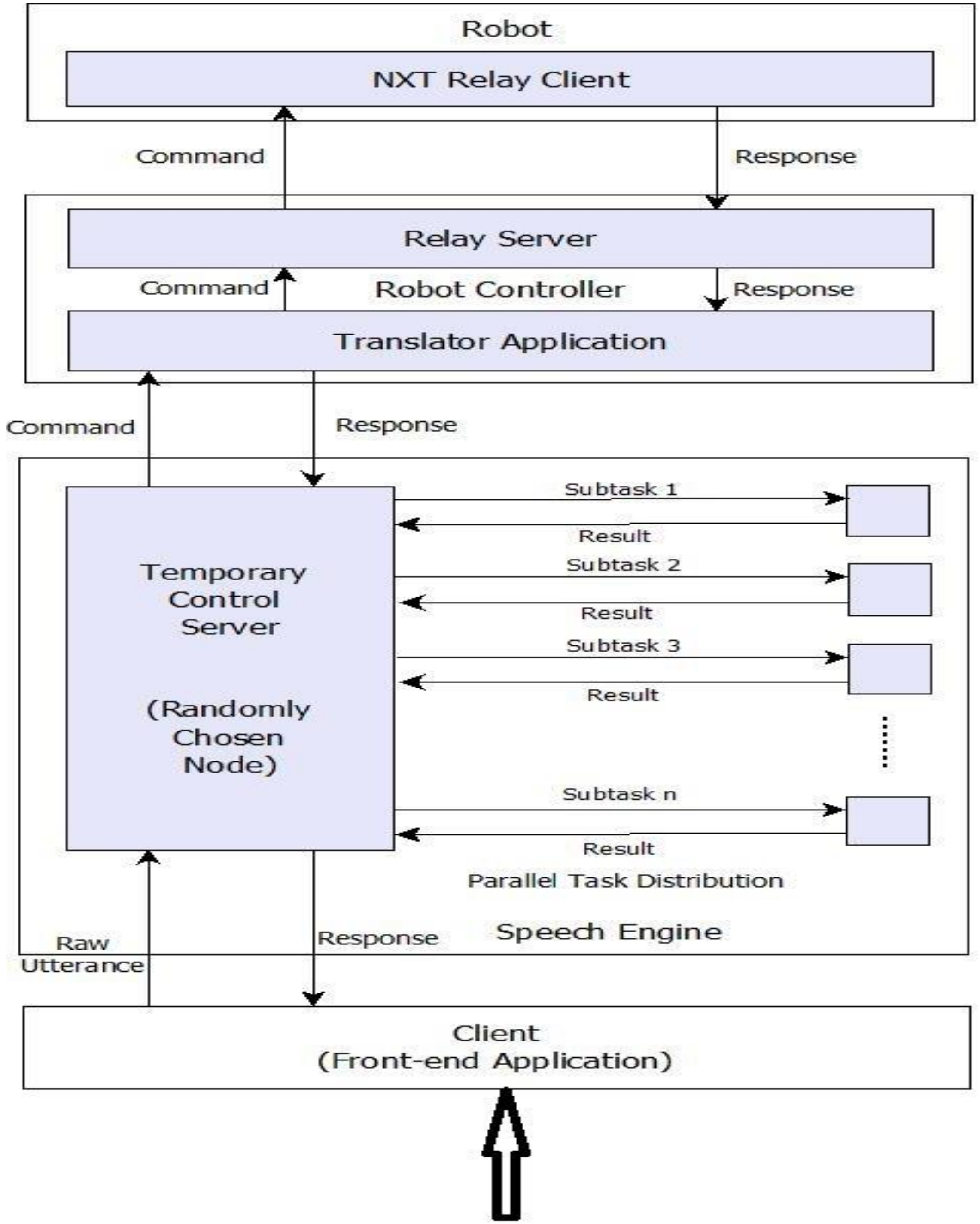
- Autonomous piece of computer programs.
- Can migrate anywhere in a network.
- Can make decisions locally.
- Asynchronous and Distributed.
- Encompass features of static agents.

Overview

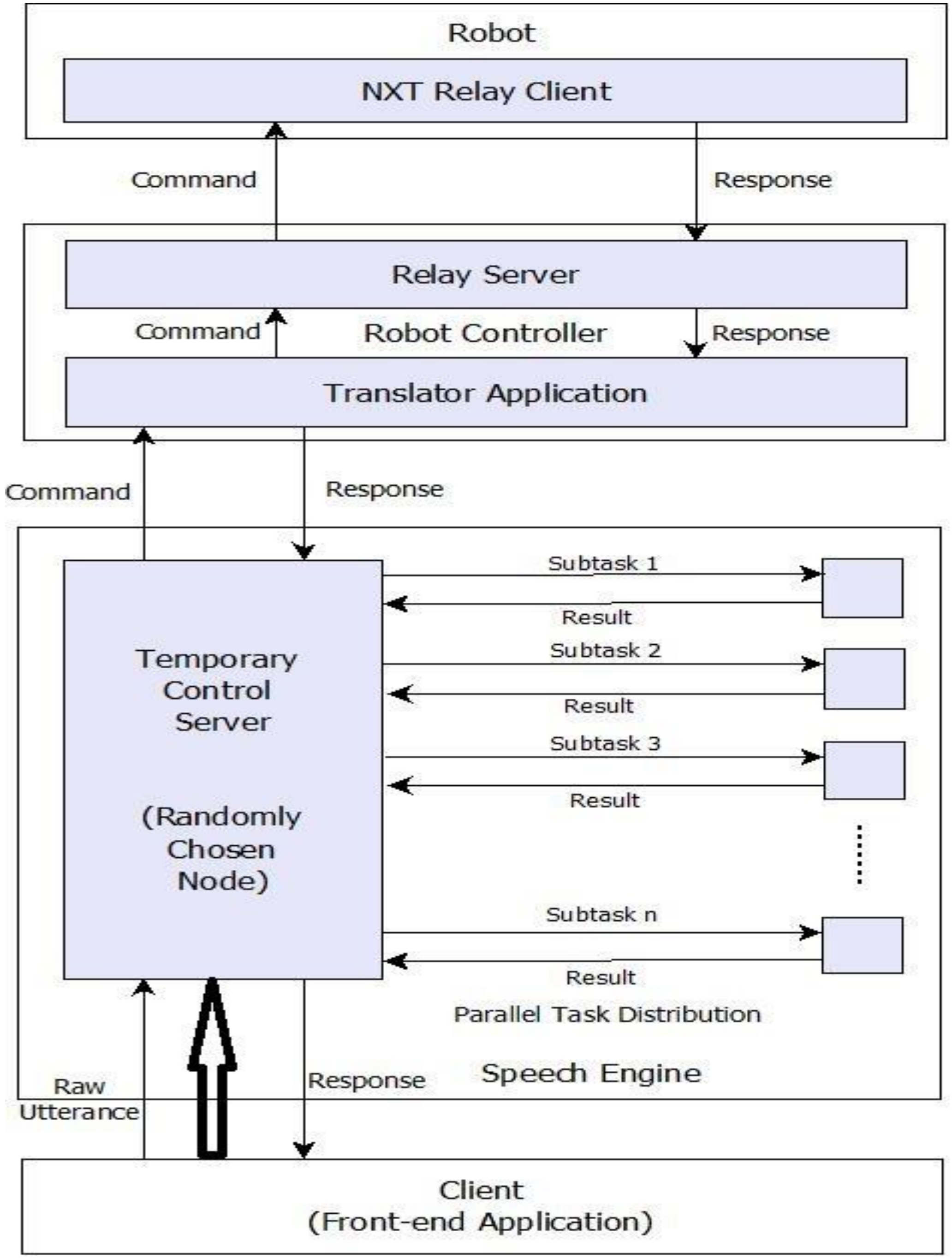
- Introduction
- Proposed System
- Architecture
- Experiments
- Results
- Conclusion

Architecture

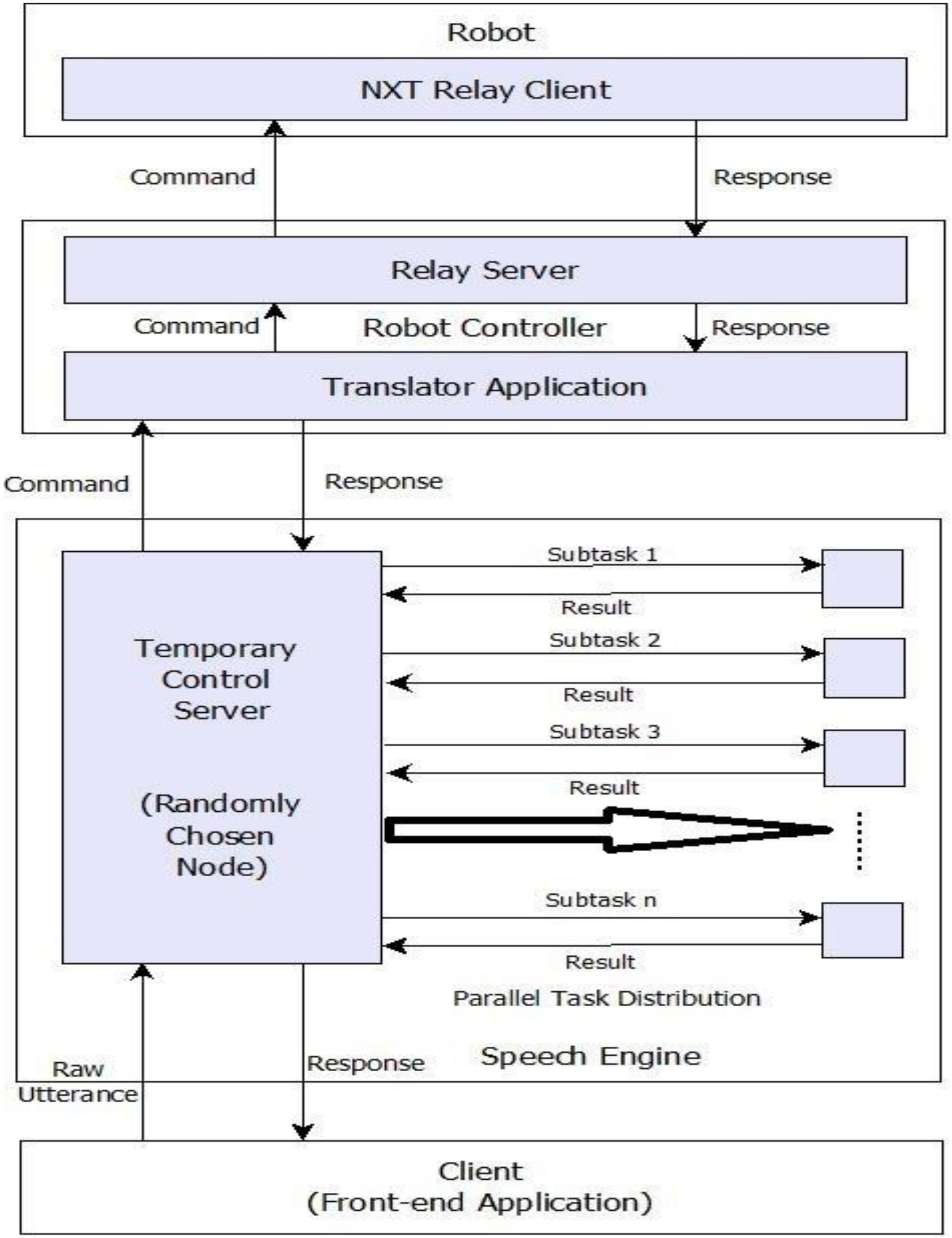




The spoken command is sent over the network to the speech engine.



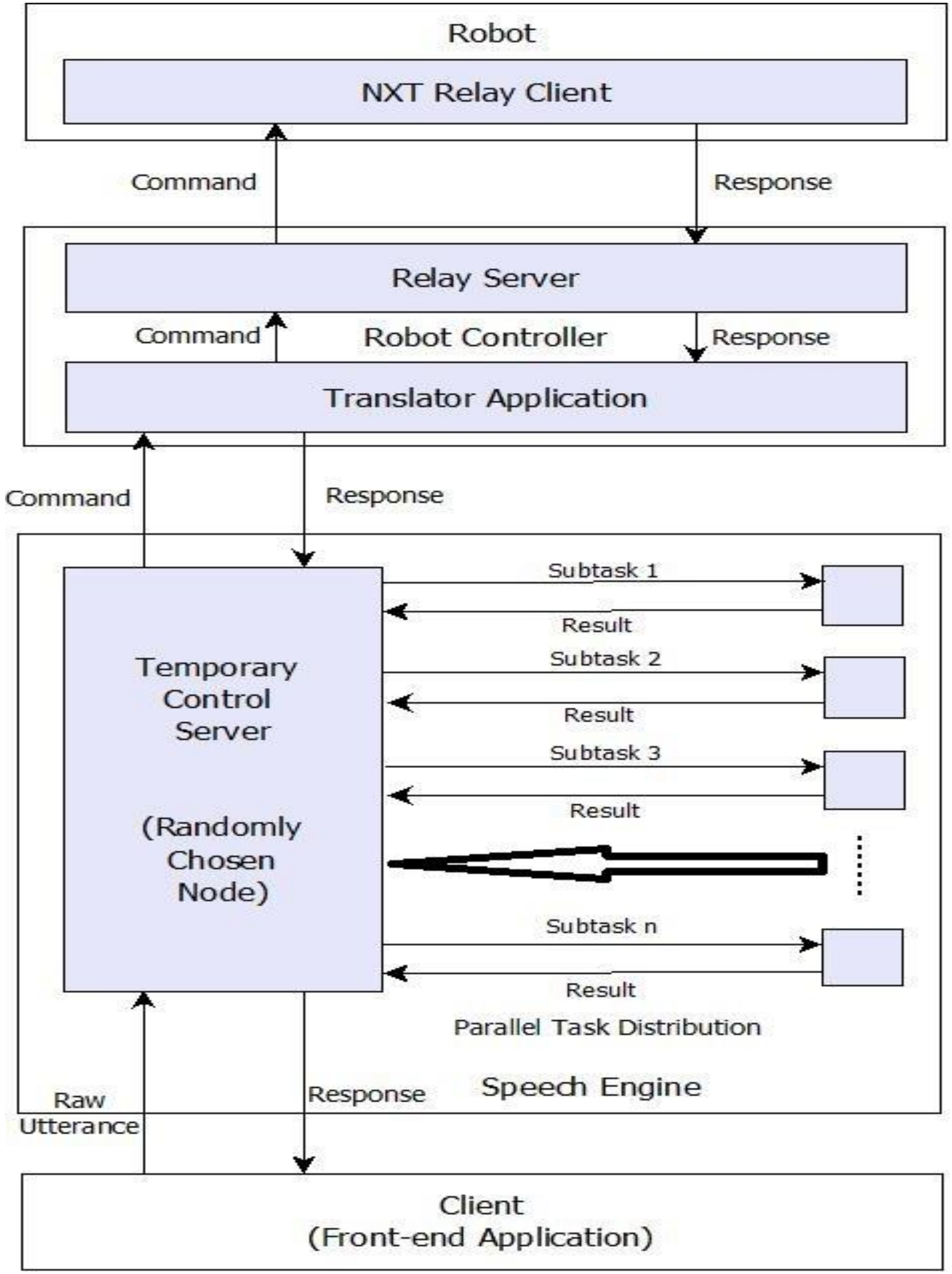
Each new command is sent to a random node within this engine.



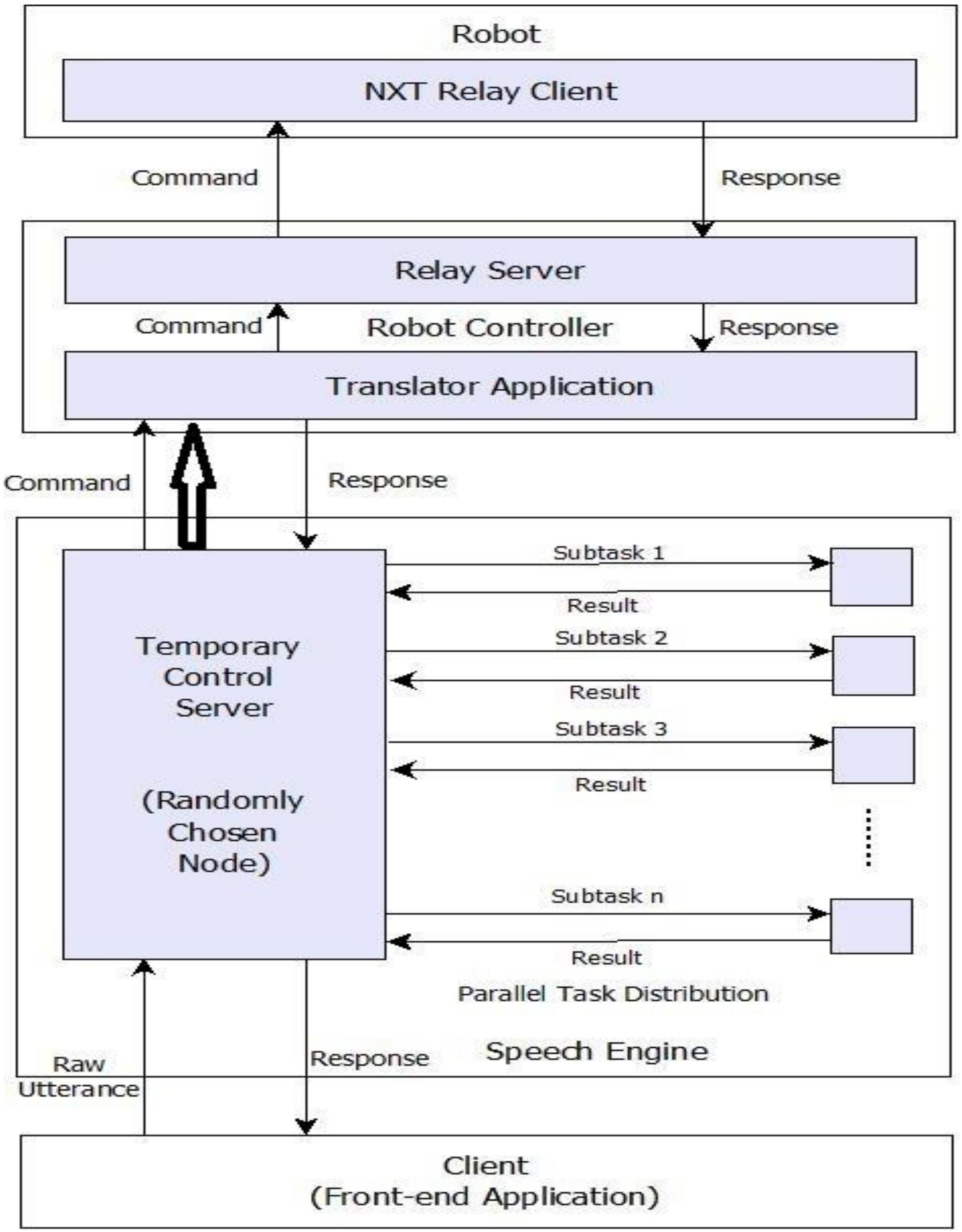
This node acts as the control server for that particular request.

The control server performs the task of parallelizing the recognition based on word models.

Mobile agents carry the spoken phrase/command as a payload to different nodes of the recognition engine.



The maximum probabilities at each node are then returned to the control server, which compares them and decides the most probable command. This is sent back to the user for verification purposes.



If the user replies positively, a positive score/feedback is sent to the model and the command is interpreted and the corresponding opcodes are forwarded to the robot controller.

Overview

- Introduction
- Proposed System
- Architecture
- Experiments
- Results
- Conclusion

Experiments

- Used LPA Prolog and Typhon - A Mobile Agents Framework.
- Nine commands were used for experimentation: Acquire, Forward, Backward, Left, Right, Stop, Release, Grab, Move.

Experimental Setup

- Used an HMM toolkit that has a predefined set of models. The toolkit calculates the percentage match between the spoken word with each of the models. The model with the highest match is chosen as the spoken word.
- Took two instances of the toolkit.
  - One of these compares the models with the words sequentially.
  - The other one was tweaked to compare the word with just one model. So, we could run multiple instances of the engine in parallel - one for each model and return all the results and compare them.
- The command to control a robot is of the following format:

  <Operation> <Robot ID(s)>.

Experimental Setup: Mobile Agents

- Used Typhon - A Mobile Agents Framework for Real World Emulation in Prolog.
- The mobile agent takes the observation sequence (the recorded audio in vector quantized form) as payload to different nodes.
- Each node calculates and returns the probability measure.

Overview

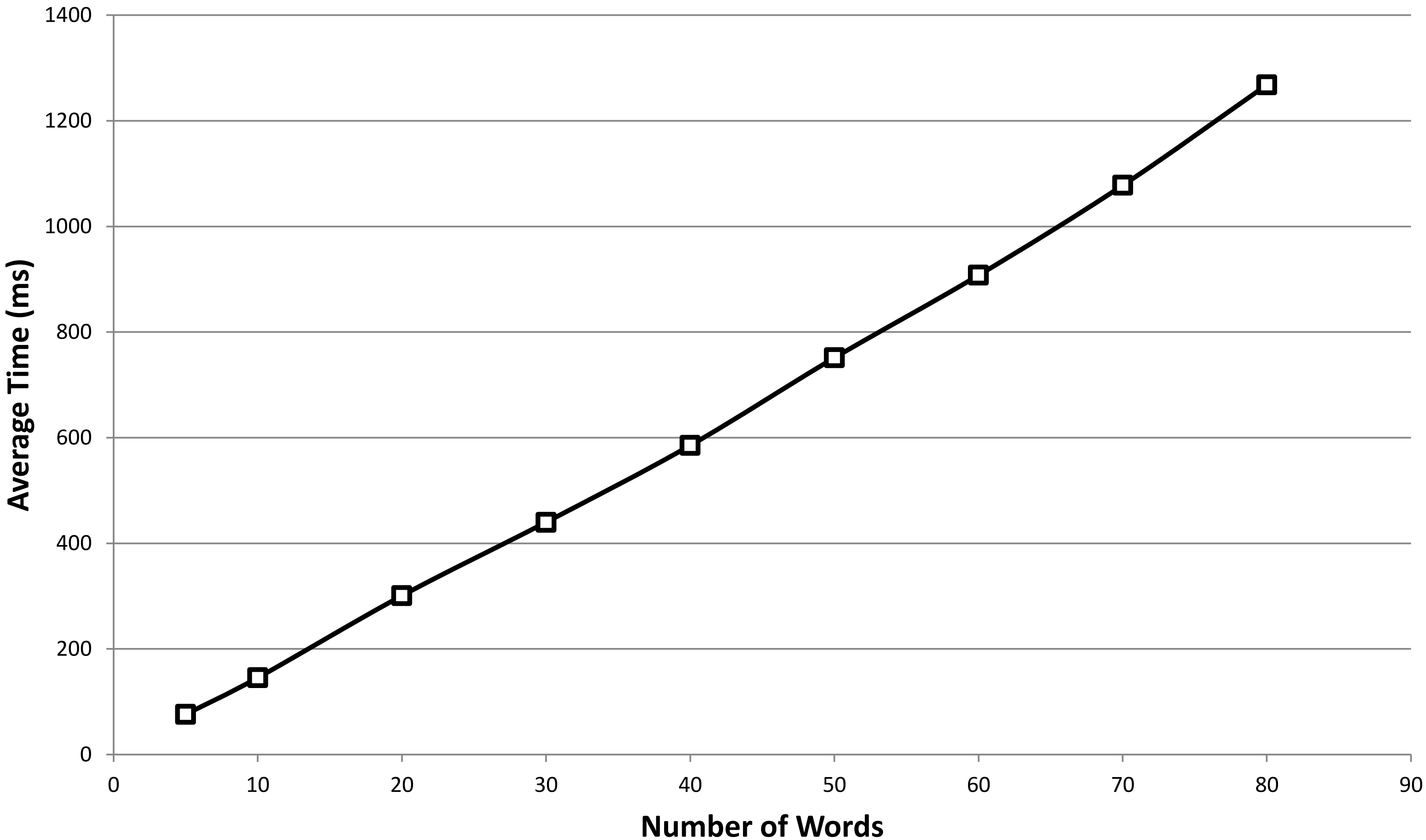
- Introduction
- Proposed System
- Architecture
- Experiments
- **Results**
- Conclusion

Results

- The performance was measured as the time taken to return the result for a given number of models.

Number of Words	Average Time (ms)
1	80
20	112
50	138
100	180
500	578
1000	1082
2500	2566
5000	5020
10000	10168

Average Time Taken for the first Approach



Comparison between the two Approaches

Vocabulary Size	Number of Models per node	Parallel Time (ms)	Serial Time (ms)
2500	250	5020	482.2
	50	5020	889.2
	25	5020	1694.4
1000	100	1082	325.2
	50	1082	438.6
	20	1082	863.2
500	100	578	255.8
	50	578	283.2
	10	578	853.2
100	100	180	200.4
	20	180	187.8
	1	180	1657.4

Results

- As the number of models on a single node decreases, there is a performance increase as compared to the "Serial" approach.
- This trend is not followed if we decrease this value beyond a certain threshold.
- Reason: The time taken for mobile agents' traversal overshadows the gain achieved by the parallelization of the system.

Results

- The recognition accuracy of the HMM toolkit was found to be around 88%.
- However it is dependent on various factors like the algorithm used, the amount of data used for training, noise etc. and is not relevant to the purpose of this work.

Overview

- Introduction
- Proposed System
- Architecture
- Experiments
- Results
- Conclusion

Conclusion

- The time taken grows linearly as the number of words increases.
- Whereas the same task takes less time after the number of words cross a certain threshold if done in parallel.
- Many ways are possible to implement the parallel approach. This work uses mobile agents which ensures the **scalability** and **decentralization** of the system.
- This approach supports adding new algorithms to the speech engine dynamically. Different nodes in the speech engine may host different recognition techniques.
- Any new language can be added to the system dynamically just by hosting the new models of the commands.

Future Work

- The present model supports only one relay server at each robot controller node. Currently work is underway to create a multi-threaded version of this relay server so that multiple robots can be connected to a single host.
- Data losses due to network errors have not been accounted for in this work. These losses may affect the usability of the system in noisy environments like in wireless ad hoc networks.
- This work also does not handle the problems that may arise due to inactivity of the client for a certain time.

Thank You

Robotics Lab.

Dept. of Computer Science & Engg.

Indian Institute of Technology Guwahati

Guwahati – 781039

Assam INDIA

Dr. Pradip K. Das

Email : pkdas@iitg.ernet.in

Home Page : www.iitg.ernet.in/pkdas

Robotics Lab : www.iitg.ernet.in/cse/robotics