



# The Convergence of Heterogeneous Internet-Connected Clients Within iMASH

*Thomas Phan, George Zorpas, and Rajive Bagrodia, UCLA  
IEEE Wireless Communication, June 2002*

徐延源  
2002/8/30



# Outline

- Introduction
- The iMASH architecture
- Application session handoff
- A scalable distributed middleware service
- Content adaptation pipeline
- Case study: the teaching file client application
- Conclusion



# Introduction

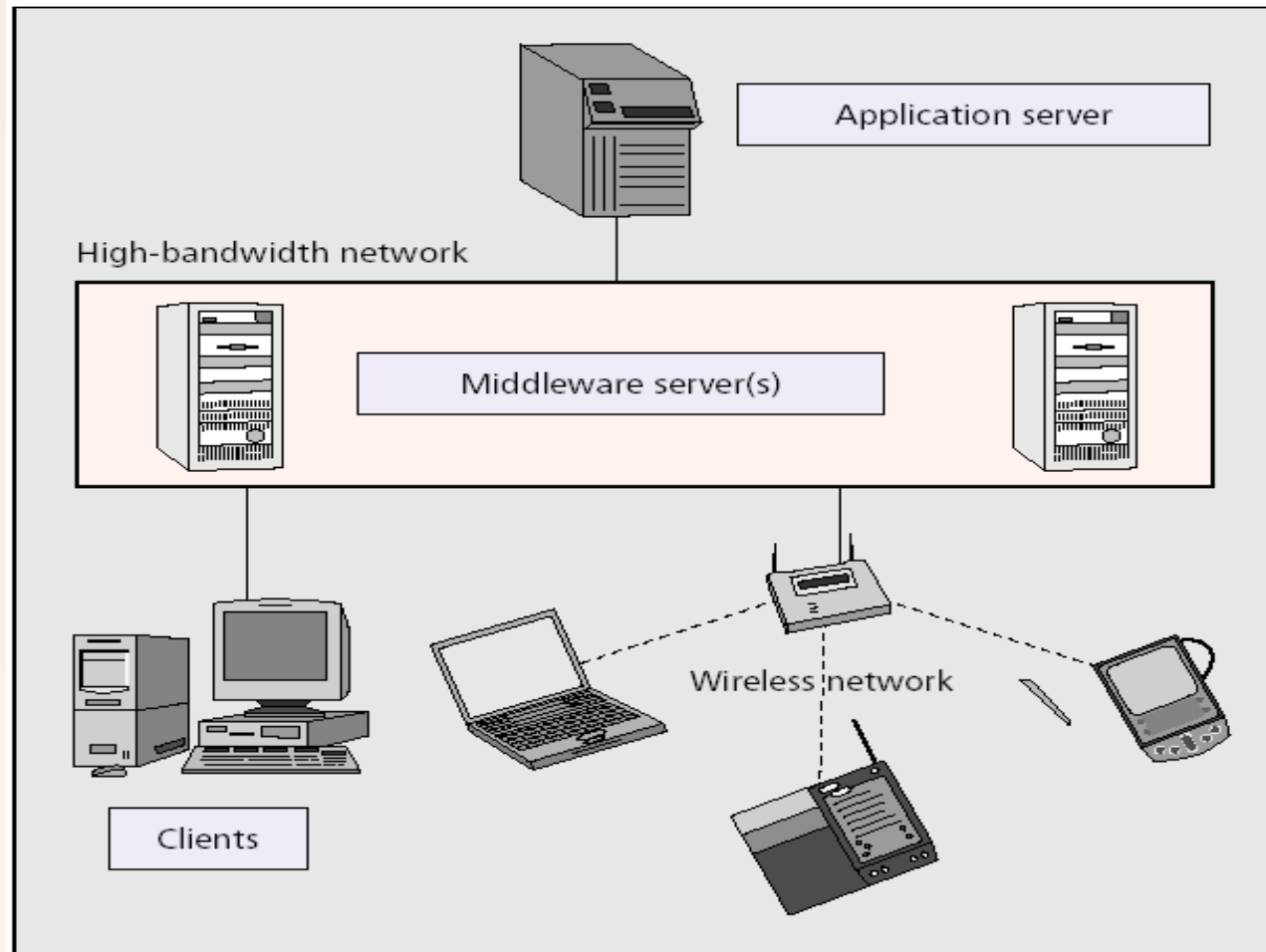
- Interactive Mobile Application Support for Heterogeneous Clients (iMASH)
- A computing environment that will allow user to have seamless, uninterrupted access to information across multiple platforms
- Convergence of desktop and mobile applications



# The iMASH architecture (1/2)

- Application server
- Heterogeneous clients
- Middleware service layer
  - A distributed set of middleware servers (MWSs)

# The iMASH architecture (2/2)





# Application session handoff (1/4)

- Application session can be summarized as simply the state of the crucial data structure on which the application acts
- Application session is never the entire address space
- Application session handoff (ASH) is different from general process migration



# Application session handoff (2/4)

- Types of ASH:
  - One-way noninteractive session transfer
  - One-way interactive session transfer
  - Two-way interactive session transfer (TWIST)

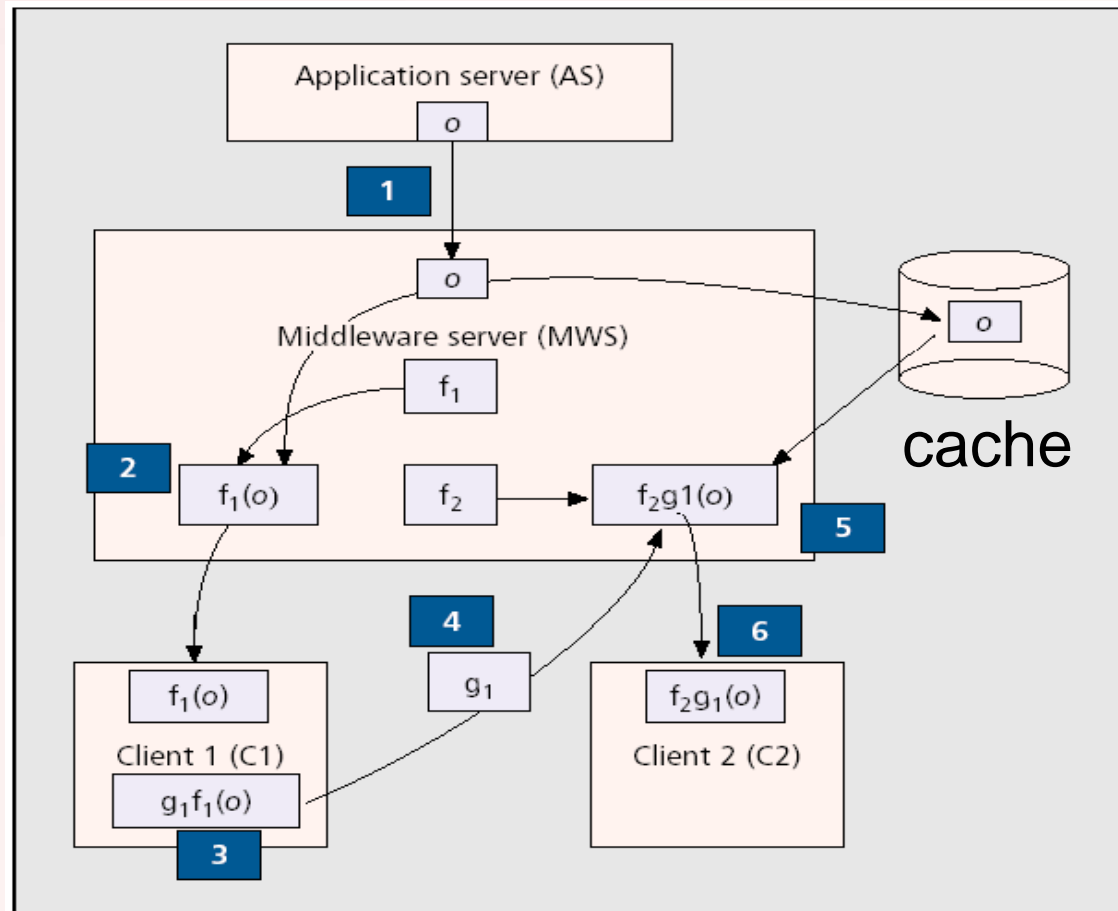


# Application session handoff (3/4)

- TWIST involves the handoff of possibly modified data from one client to another
- TWIST shields the application server (AS) from the handoff details
  - No modification to AS is needed
  - The application programming interface (API) remains unchanged
- No save to intermediate medium is required



# Application session handoff (4/4)





# A scalable distributed middleware service (1/3)

- The purpose of middleware layer is to reduce the load on the AS and provide variety of services for the clients
- Middleware service (MWService) layer contains a distributed set of individual server



# A scalable distributed middleware service (2/3)

- Each MWS determines autonomously if it is capable of providing service
- A lookup service, or registry, allows clients to discover an MWS within the MWService that can perform work on the client's behalf
- Middleware-Aware Remote Code is used in the application as a client-side proxy that interacts with the registry



## A scalable distributed middleware service (3/3)

- When a MWS become unavailable, the MWS will perform a middleware-to-middleware handoff of the client's session state to another available MWS



# Content adaptation pipeline

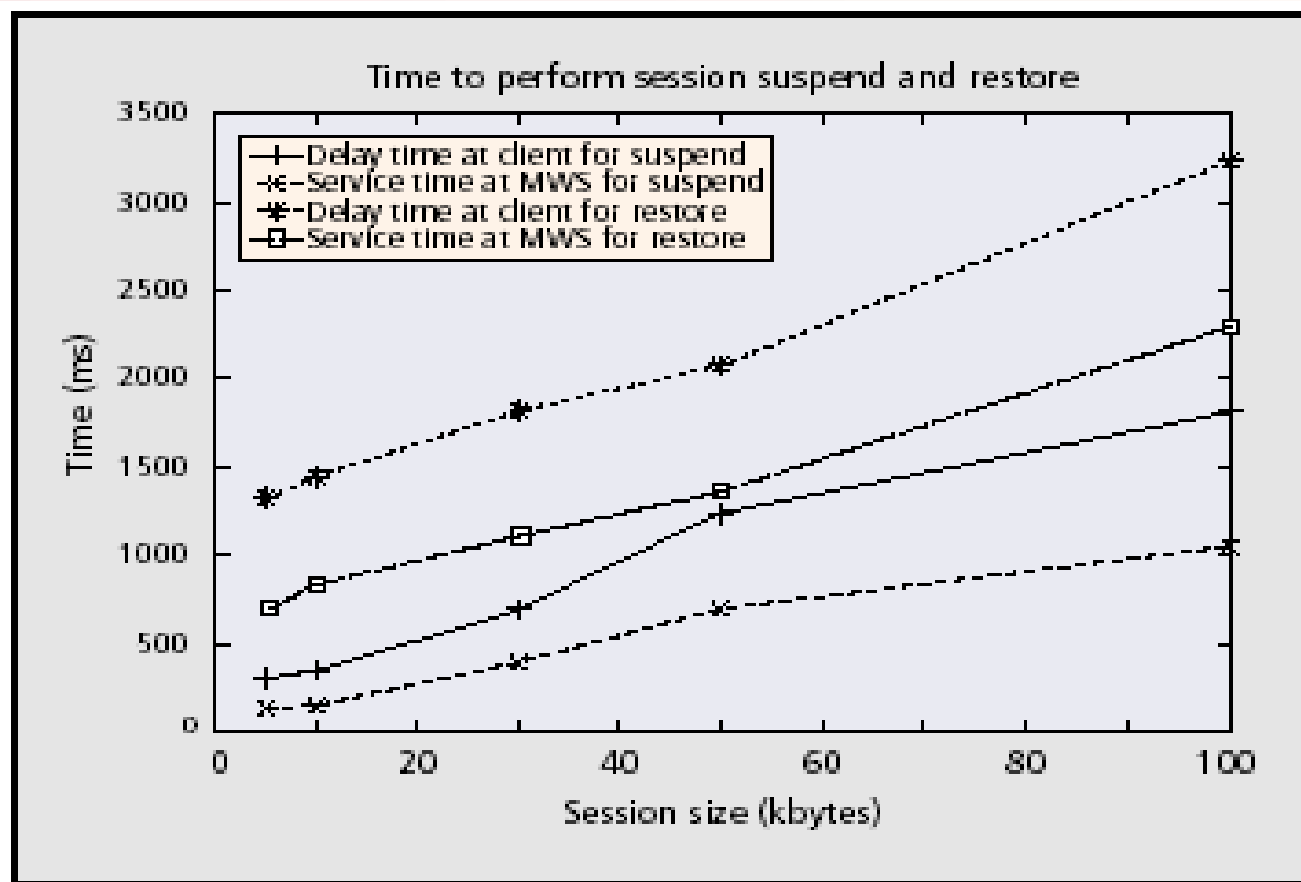
- Four stages:
  - Collecting client's profile
  - Identifying the characteristic of the data objects
  - Generating commands to adapt the object based on heuristics
  - Producing the resultant adapted data



# Case study (1/2)

- The teaching file client application
  - A java applet
  - Retrieves data from an AS via HTTP
  - Displays images and allows user to add textual annotation to the images
  - The user modification is stored in XML format

# Case study (2/2)





# Conclusion

- Lack of integration between similar applications running on different machines
- ASH capability allows seamless transfer of application data across multiple OS platforms