# Grapevine System

- Message delivery service
- Naming
- Resource location
- Authentication
- Access control service

# Grapevine

- environment
  - 1500 computers, 50 local networks
  - involve gateway, phone lines and LAN
- Message delivery system
  - supports distribution list; support mailbox

# Requirements on Grapevine

- Separation between services and clients
  - clients are not trusted
- Reliable delivery
  - messages not lost in the face of user errors, client errors, server or network problems
- Fault tolerance
- Decentralized administration and Scalable
- Authentication of sender, secure communication

# Using Grapevine

user B wants to send message to user Y:

- accept message[sender, password, recipients, message]
- message polling[individual]
- retrieval message[name, password]
- authenticate[individual, password]

# Implementing Grapevine

Issues:

- Naming user B and user Y
- Locating user Y
- Transport & delivery

Design alternatives?

# Registration Databases

- Rname for individuals and groups
  - group: a collection of RNames
  - individual: password, inbox sites, connect site
- Two level hierarchy for Rnames: Y.R
- Additional functions:
  - membership[name, group]
  - resource location: group->members
    individual->connect site, inbox lists

# Servers

- Registration servers: authentication and resource location
- Message servers
- Registration database replicated
- Registration servers and message servers are each other’s clients

# Message Delivery

- acceptance
  - postmark
  - reliability
- transport
  - duplicate elimination
- replication
  - replication of acceptance service
  - replication of delivery service

# Resource Location

- locating registration servers
  - *.gv
- locating message servers
  - ms.gv; maildrop.ms
- resource location algorithms
  - resource representation
  - location

# Updating the Registration DB

- timestamps for registration entries
- primitive operations
- propogation
  - atomic update?
  - allow inconsistency?
- notification of changes

# Cache and Hints

- why
- Example: cache of RName to an inbox site
  - primary servers and changes in them
  - changes in inbox sites
  - cache flushing

# System Configuration

- adding registry replicas
- deleting registry replicas
- starting new servers
  - creation of name and changes to the database
  - server initialization

# Discussion

- distributed registration database
- trade atomic update for increased availability
- cache and hints

# Similar Issues in Distributed Systems

- Domain name servers:
  - nslookup & nstest
  - cache and hints
- Authentication in mail delivery
- Secure communication
- Consistency issue in wide-area distributed systems