Persistent Personal Names for Globally Connected Mobile Devices

Bryan Ford, Jacob Strauss, Chris Lesniewski-Laas, Sean Rhea,
Frans Kaashoek, and Robert Morris
Massachusetts Institute of Technology

OSDI 2006

Presented by: Matthew Tan Creti

PURDUE
UNIVERSITY

Unmanaged Internet Architecture

- *Unmanaged Internet Architecture* (UIA) provides zero-configuration connectivity among mobile devices addressed by *personal names*
- Provides non-technical users a simple and intuitive way to connect their mobile devices
- Addresses mobile devices with convenient *personal names* organized into *personal groups*
- Share resources with trusted acquaintances
- Routing algorithm exploits user's social network for connectivity over both ad hoc networks and global infrastructure

User-friendly Naming Based on Social Networking

- Instead of using a long globally unique name like **ipod.mysite.myisp.com** we can name a device just **ipod**
- Users can introduce their device names to each other
- For example, if Bob has a friend he refers to as **Alice**, and Alice has a VoIP phone she calls **phone** then Bob can call Alice using the name **phone.Alice**
- Names are organized into groups and groups can be shared amongst a multiple users

The Users Experience

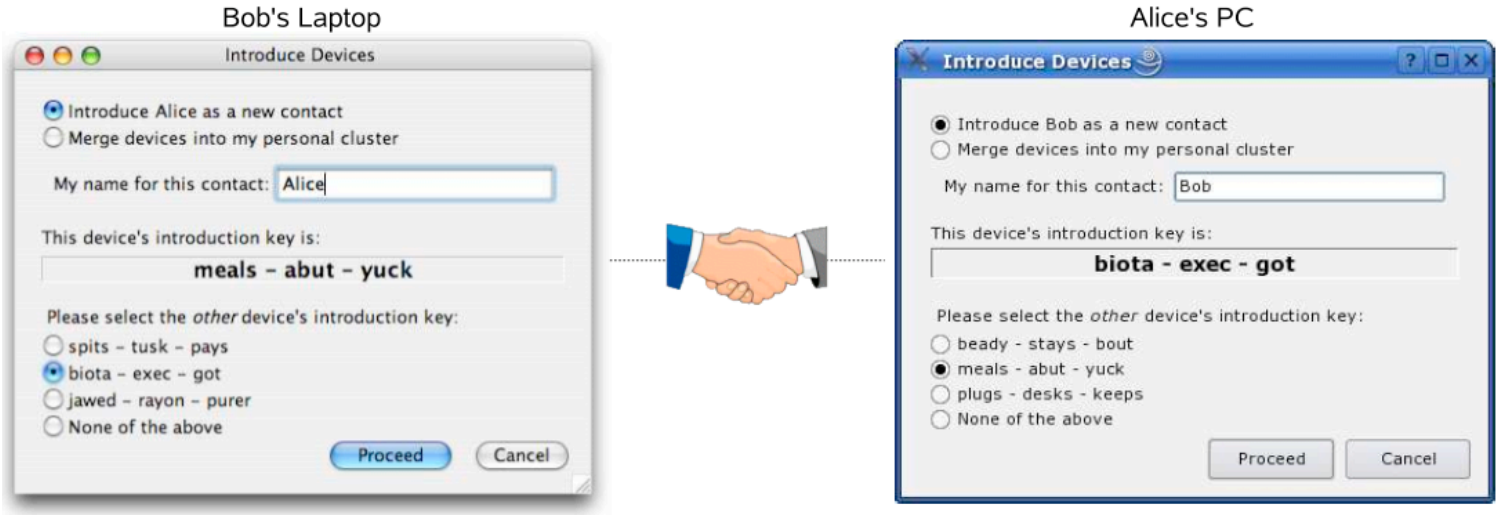
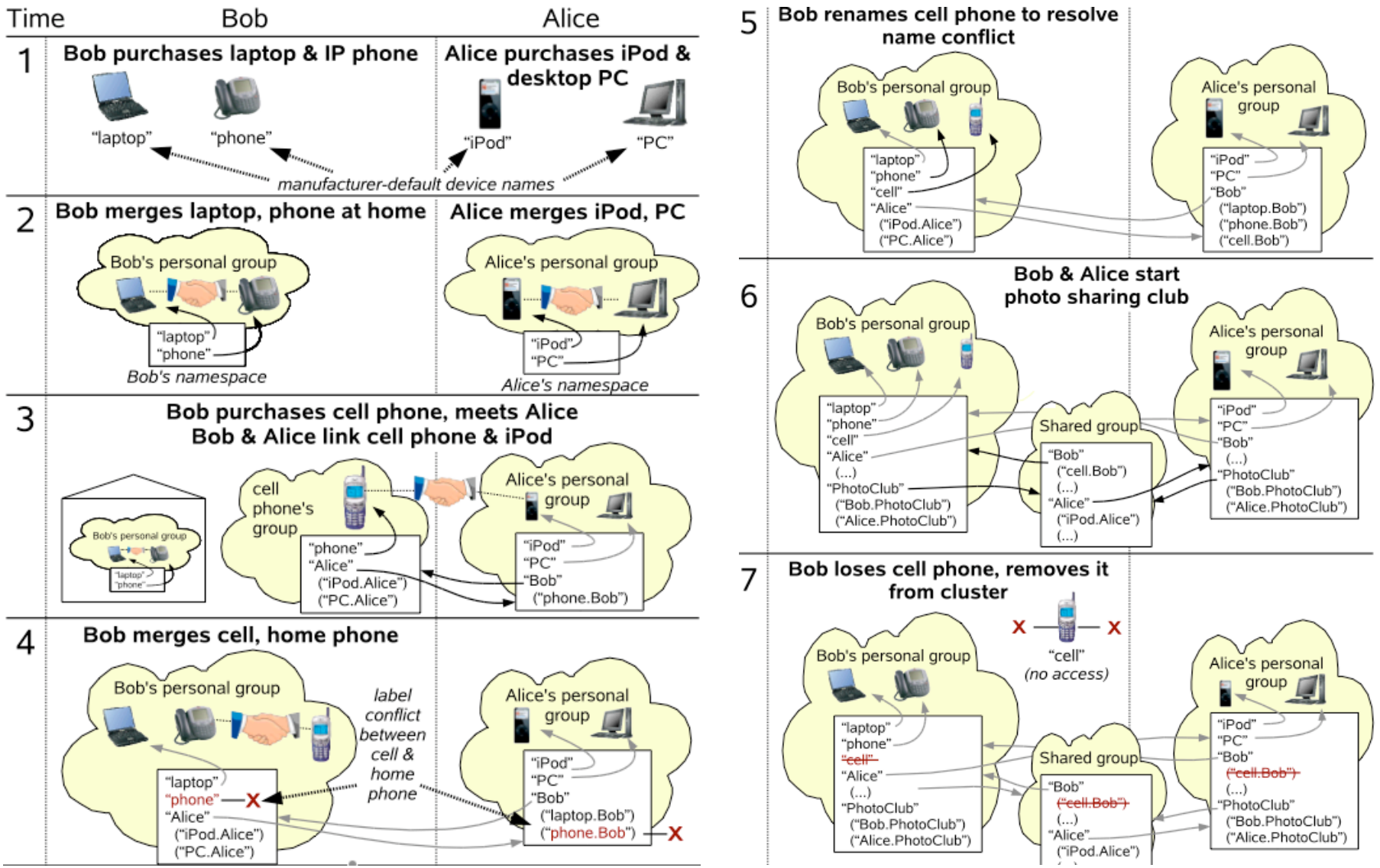


Figure 1: Bob and Alice introduce their devices

- A device is sold with a simple name like “laptop” or “phone”
- Names can be changed by the users
- During the introduction of two devices the user physically brings the devices together or connects them to a local network
- Each device displays and *introduction key* consisting of three randomly chosen words
- The user enters the introduction key of the opposite device
- This could also happen over the Internet with keys shared though some trusted second channel such as over phone

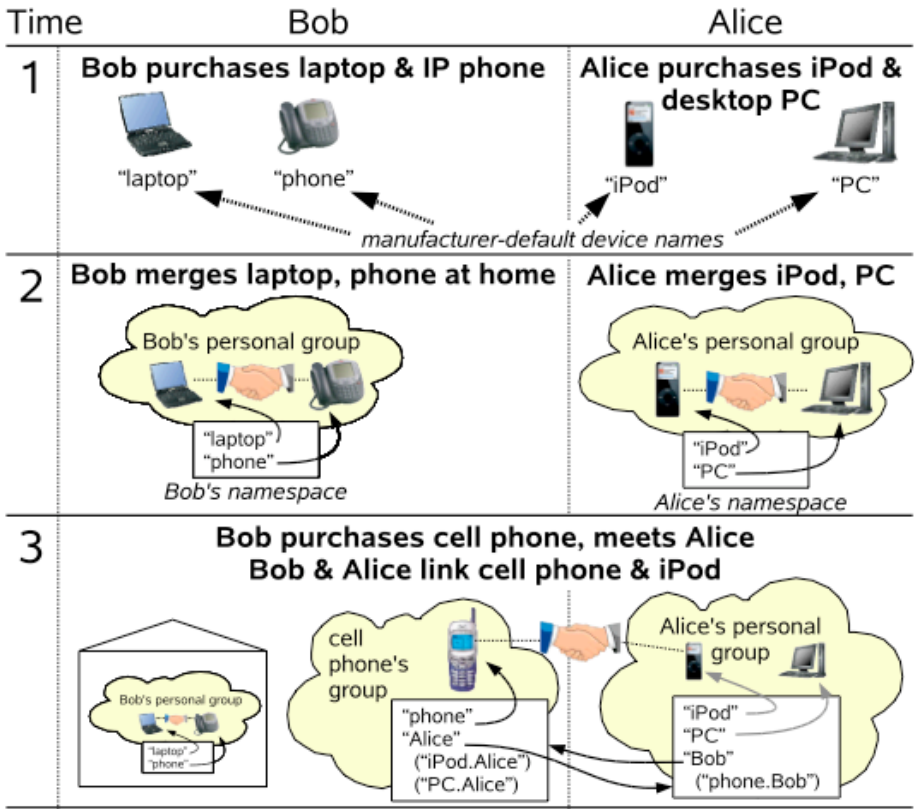
Introduction to Groups



PURDUE UNIVERSITY

Functionality

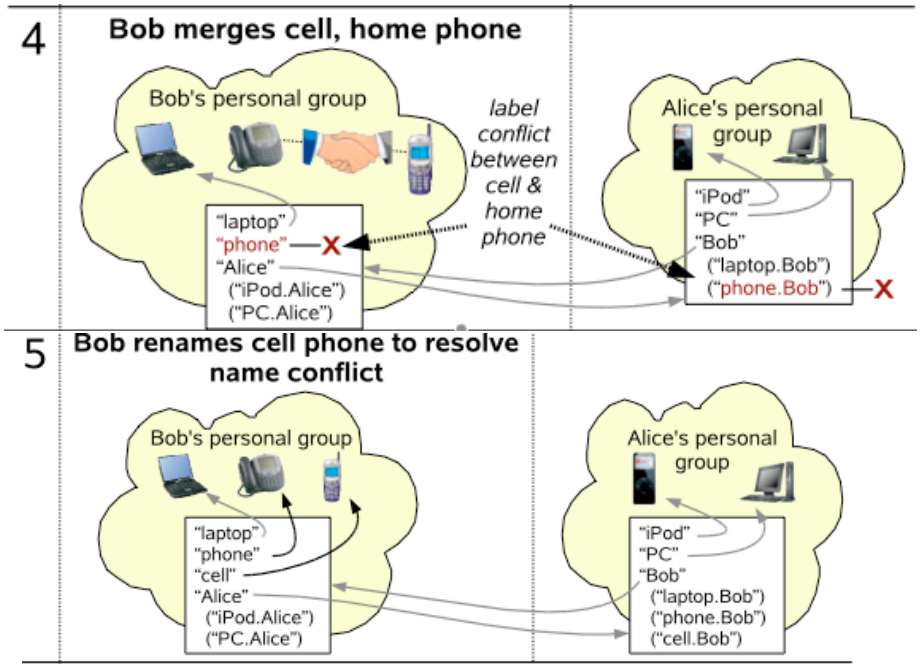
- Introducing Devices
  - Given manufacture default names
- Device Names and Personal Groups
  - When a device introduces itself it can merge a new device into one of its groups, gossiping lets all the other devices in the group know about the new device
  - Name changes to a device in a group are also gossiped
- User Names and Social Networking
  - Each device has a user name, if Bob already knows an Alice he can refer to the new Alice as Alice-Smith
  - Once Bob know Alice he can then refer to any of her devices using the dotted notation (e.g. **PC.Alice**)



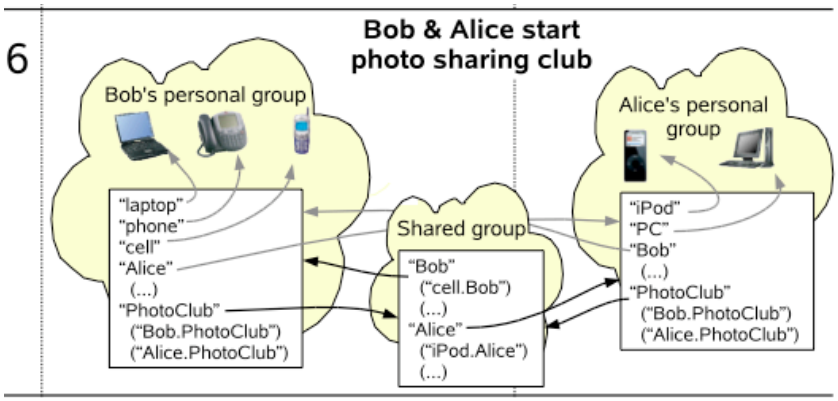
PURDUE UNIVERSITY

Functionality (Cont.)

- Transitive Merging and Gossip
  - Bob's laptop can learn about Alice from Bob's cell phone. Later when Bob is in the bus with his laptop and he meets Alice his laptop will connect to her cell phone even though they have never meet before and they can gossip changes to their groups
- Resolving Conflicts
  - Bob has two devices named phone, when he resolves the conflict the result is gossiped to all interested groups



Functionality (Cont.)



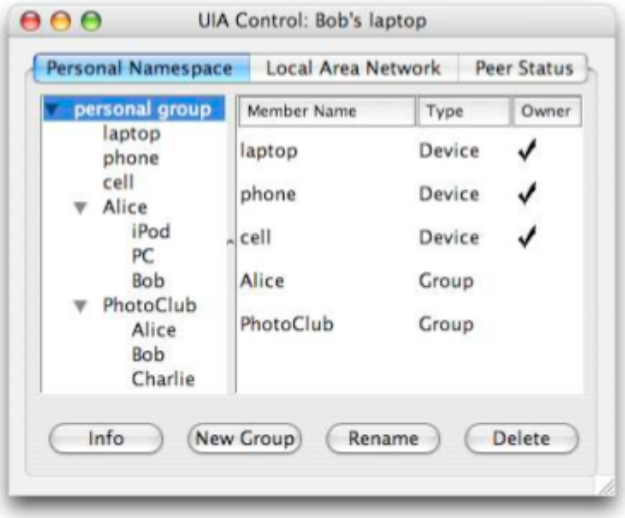
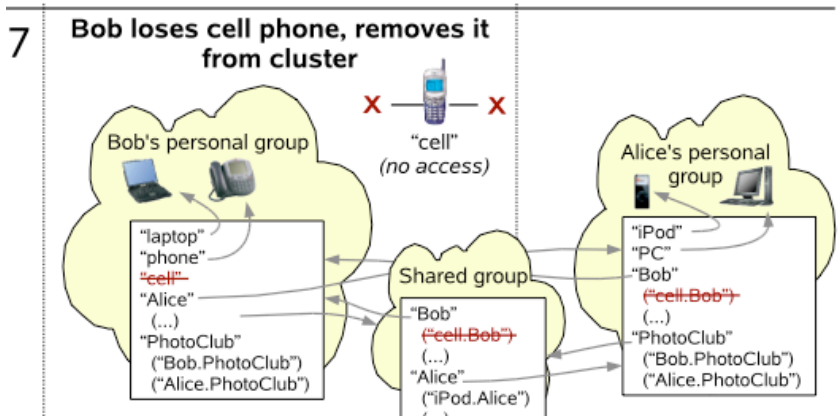


Figure 3: Groups and Ownership

- Shared Groups
  - Bob creates a group called "PhotoClub"
  - Bob adds Alice to the group
  - Alice sees the group as PhotoClub.Bob but she can give it a different name on her devices if she wants
- Group Ownership
  - One or more members of a group may be designated as owners
  - For example, laptop, phone, and cell are owners of Bob's personal group
  - Ownership is transitive (e.g. Bob's laptop is an owner of PhotoClub)

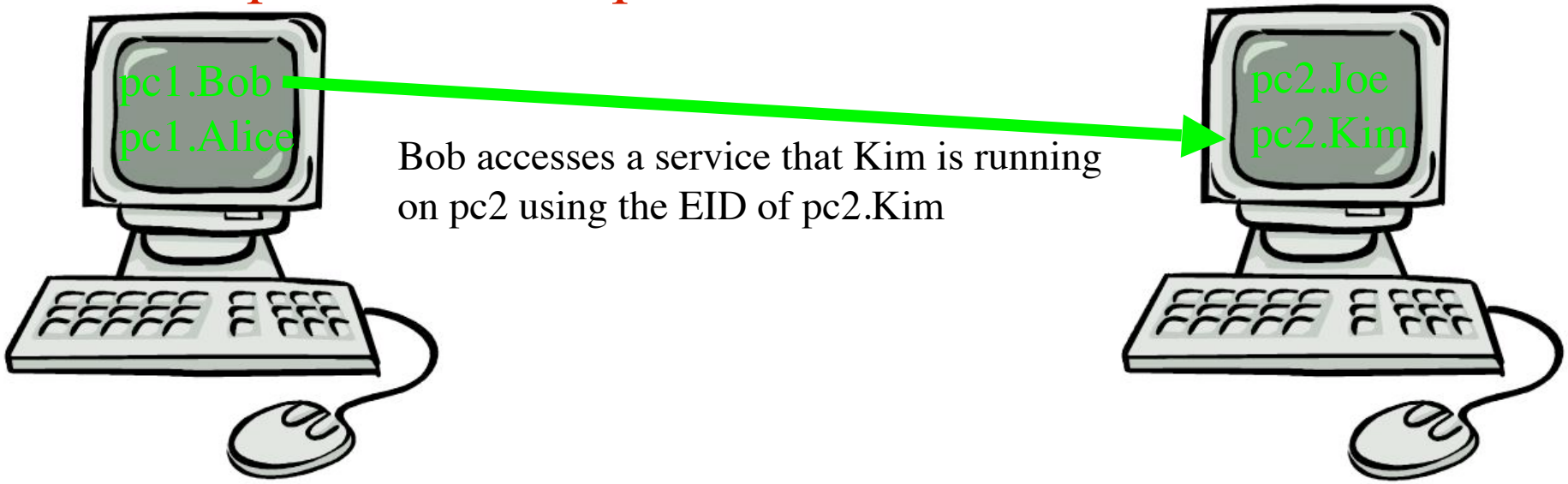
Functionality (Cont.)



- Security and Ownership Revocation
  - If a device is stolen or lost a user can revoke its ownership of his personal group
  - If the device is found the revocation can be undone
  - If Bob's phone is revoked it will also lose access to share resources with other users
- Ownership Disputes
  - Any device can dispute another device's revocation of its ownership
  - When Bob disputes revocation of his laptop, his personal group is split into two new groups
  - All of the devices in Bob's old group go to his new laptop's group
  - All names referring to Bob's old group are no longer valid and must be reintroduced

How it Works: Endpoint Identifiers

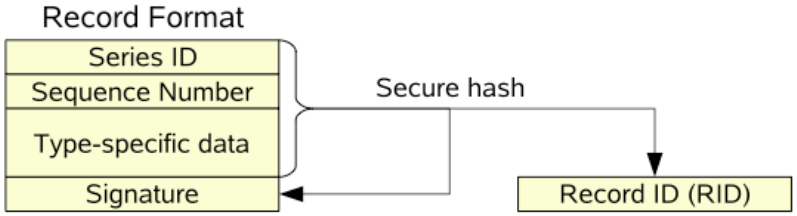
- Every name is mapped to an *endpoint identifier* (EID)
- An EID is generated by:
  - Generating a private/public key pair
  - Computing a cryptographic hash of the public key
- EIDs are *personal*; an EID corresponds to a particular user's presence on a particular device



State Management

- State is managed in append only log
- Optimistic replication (eventual consistency)
- Push/pull epidemic protocol
  - In push phase a new record is pushed by a device to other random devices until it finds a device that already has the record (known as rumor mongering)
  - In the pull phase each device periodically contacts a randomly-chosen peer to obtain missing records

Device Log Structure



Type	Type-specific Record Content
Create	*Owner:* endpoint ID (EID) of owner device *Nonce:* ensures uniqueness of new series ID
Link	*Label:* human-readable string *Target:* device (EID) or group (series ID) *OwnerFlag:* grants group ownership if true
Merge	*Target:* series ID (SID) to merge with
Cancel	*Target:* record ID to cancel

Figure 4: Log Record Format

- A series contains the sequence of changes made to a particular group by a particular device
- A device that owns a series signs each record with its private key

Record Types

- Create Record
  - The create record initiates a new series (group/device pair)
  - The create record contains the owner's EID
  - This is signed using the owners private key
  - A hash of the record is used as the record ID
  - The record ID will be used as the series ID of the new series
- Link Record
  - Binds a human-readable label to an endpoint ID (link to device) or series ID of some series in the group (link to group)
  - Owner flag indicates whether the link grants ownership to the link's target

Record Types (cont.)

- Merge
  - Joins two series to form a single group
  - The union of all link and cancel records in all merged series determines the set of names that appear in the group
- Cancel
  - Nullifies effect of previous link record

Namespace Operations

- Device Initialization
  - Device writes a create record that will form a new series to represent the user's personal "root" group on the device
  - Device writes a link record to the new series giving it a default name like "laptop"
  - Owner flag is set to make the device the sole initial owner of the group
- Merging Device Groups
  - When two devices merge, each device writes to its own root series a merge record pointing to the other device's root series
  - The merging causes the root series of both devices to be gossiped to each other

Namespace Operations (cont.)

- Meeting Other Users
  - Devices exchange their root series IDs
  - Each device writes a link record to its own root series referring to the other device's root series
  - Because owner flag is not set the two users have read-only access to each others namespaces
- Transitive Merger
  - A single merge is always between a pair of records
  - The group is made up of the union of all merges series
  - Gossiping allows devices to learn about each other transitively
  - If names conflict then the devices flag a label conflict and refuse to resolve the name

Namespace Operations (cont.)

- Renaming Labels and Resolving Conflicts
  - When a label is renamed to resolve a conflict, a cancel record is written to the root series and a new link record is written with the new name
  - Because one of the two conflicting links is now renamed, the conflict no longer exists
- Creating Groups
  - A create record is written to generate a fresh series ID
  - The device writes a link record with a name like “PhotoClub” in the root series pointing to the new series
  - The device writes a link record with a name link “Bob” in the new series pointing to the root series
  - The owner flag is set on this second link record

Namespace Operations (cont.)

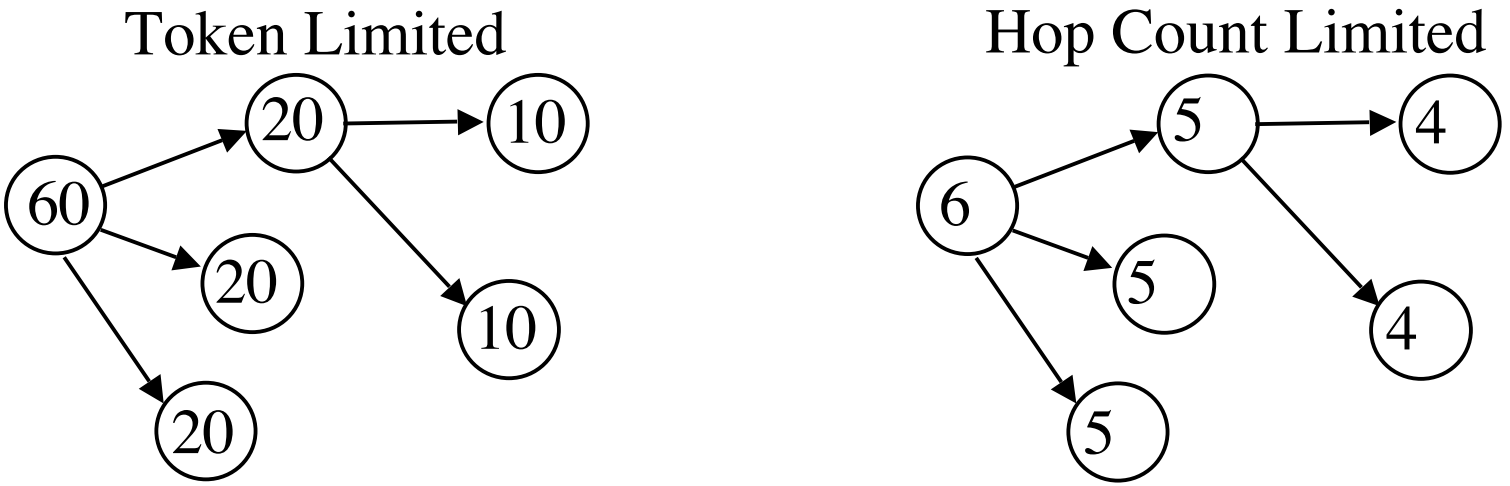
- Revoking Ownership
  - Bob's laptop cannot just put a cancel record in his cell phones root series
  - Instead Bob's laptop creates a new personal group for Bob and copies the original group's name content into it
  - The create record itself has the series ID of the old series to indicate to all interested devices that the new series is meant to replace the old series
  - If the cell phone disputes the revoking then Bob will have to reintroduce the correct new series with interested devices

Routing and Forwarding

- Optimized for connecting to devices in the user's immediate *social neighborhood*
- If a device does not know the IP address of the device it is trying to contact, then it will flood a location request through the overlay network between devices in its social neighborhood
- If a device is behind a firewall or NAT then data can be forwarded to it though open TCP connections in the overlay network

Overlay Construction and Maintenance

- Each device maintains an open TCP connection with a configurable number of overlay peers
- A device prefers peers in this order
  - *stable* peers (e.g. 90% available at the same public IP in the last week)
  - Closest in *friendship distance*
- Flooding is token limited rather than hop count limited



PURDUE UNIVERSITY

Experimental Results

- Used social network relationships from Orkut, users in Orkut represent devices in the experiments
- Devices are randomly chosen to be stable or mobile
- Ideal hop count-limited has oracle information on the minimum number of hops needed to reach a device

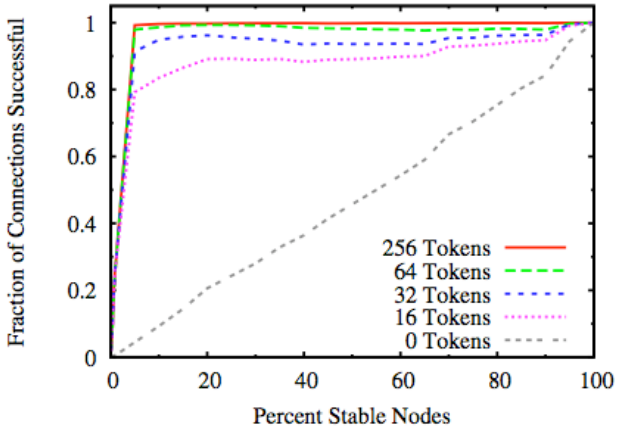


Figure 8: Location request success rate

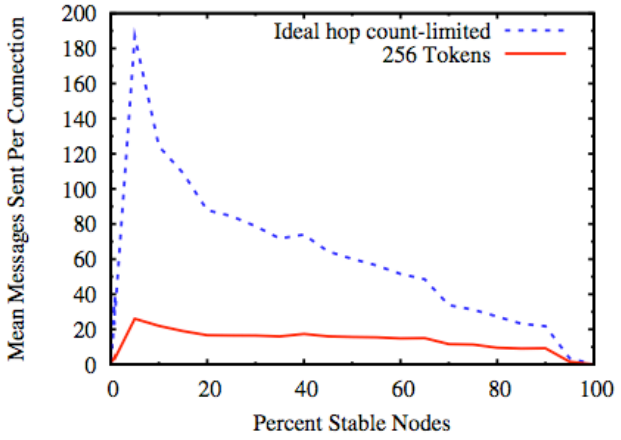


Figure 9: Mean messages sent per location request

PURDUE UNIVERSITY

Conclusions on UIA

- Practical way of managing devices over the existing Internet
  - e.g. when they take a laptop home they don't need to re-establish an ssh connection
  - This can already be done with existing protocols like mobile-IP but they require a lot of work to configure
  - UIA requires zero-configuration
  - Works with existing Internet