The Real Ad-hoc Multi-hop Peer-to-peer (RAMP) Middleware: Content sharing and Social Networking

Paolo Bellavista
Carlo Giannelli
Stefano Poli
Scenario
- Smart devices and spontaneous networking
- Online social networks, social relationships, and content sharing

RAMP middleware
- Solution for spontaneous networking and content sharing
- Resource access filtering based on social relationship tightness
- Preliminary experimental results
Widespread availability of **smart devices** (smartphones, tablets, notebooks, smart TVs, etc.)

- heterogeneous wireless connectivity
- powerful computing capabilities
- exploited to access services and generate/share content
Widespread use of Online Social Networks (OSNs)
- Creating relationships between users
- Interaction between users
  - Message exchanging, photo tagging, etc.
- User-generated content sharing

**Sensitive issue:** users lose control on resources when uploaded on OSNs
Resource sharing
- **Middleware** solution allowing to *share content and services* in a *peer-to-peer* way while *fully preserving content ownership*
- Content sharing based on legacy protocols such as HTTP and UPnP

Visibility Filtering
- Resource filtering system based on relationships
- **Issue**: OSN relationship does not allow to discriminate among friends
- **Social Relationship Tightness evaluation** to tune/differentiate **access** to shared **resources**
Resource sharing **without centralized servers**

- OSN only to advertise the new content
- Resources sent from storing device to requesting one
- Resource ownership maintained by resource owner
Automatic creation of **Spontaneous Networks (SNs)**

- **Impromptu** interconnection of mobile and fixed nodes
- **Node cooperation** (single/multi-hop connectivity, P2P services)
- Example: node A interacts with node E via intermediate nodes C and D
RAMP: User Centered Network (UCN)

Automatic creation of **User Centered Networks (UCNs)** based on **OSN identities**
RAMP: Federation of UCNs

- **Home to home** resource sharing
- **Dynamic federation** of **UCNs** based on social relationships

---

**Bob**
- Bob’s OSN account
- Bob’s UCN

**Alice**
- Alice’s OSN account
- Alice’s UCN

**GW_1**: OSN relationship (e.g. Facebook friendship)

**GW_2**: Internet

**UCNs Federation**: Bob’s UCN and Alice’s UCN
Filtering System

- Filtering system to **control access** to resources
- E.g. Bob and Alice are friends on Facebook; node A of Bob requests a resource to node E of Alice: requests will be accepted or denied
Very **coarse grained** filtering

- Checking **only the relationship type** (e.g. Facebook relationship, Twitter relationship)
- E.g. Alice specifies a rule for Facebook relationships: that rule will be applied to all requests coming from her Facebook friends

Rules
- **Facebook** friends: allowed
- **Twitter** friends: denied
**New Filtering System: insights**

- **Idea:** if two users are close friends, it is likely they are willing to share services and resources one another
- Close friends should be able to access more resources
- Filtering rules based on *relationship tightness* or specific *friend ID*

![Diagram showing different relationships and filtering rules]

**Rules**
- Twitter friends: allowed
- Facebook close friends: allowed
- Facebook regular friends: denied
- Carl: denied
RAMP: new Filtering System

- Dynamically evaluated **relationship tightness**
- Well-defined grammar to specify **high-level filtering rules**
- Runtime **visibility** tuning of shared resources

- Gathering social meta/data from OSNs
- Social relationship tightness evaluation

- Web GUI
- Tightness Evaluator
- Visibility Tuner
- RAMP Packet Dispatcher

- Users can customize the middleware behavior

- Active monitoring of RAMP traversing packets
- Runtime enforcement of filter rules

Users can customize the middleware behavior.
• **Idea**: if two users interact very frequently on a OSN, it is likely that their relationship is tight

• Three tightness categories: “**tight**”, “**regular**”, “**loose**”

E.g. Alice and Bob have exchanged

• more than 50 messages => tight
• between 5 and 50 messages => regular
• less than 5 messages => loose
Users typically exploit **OSN interaction tools** in different ways (private/public message exchanging, quotes in texts, photos, videos, etc.)

- E.g. Alice exploits only private messages to interact with her tight friends, while Bob is used to exploit public posts

- **Survey** (20 Facebook users, 10 Twitter users) to identify useful data and build to decision model

- **Weka** to analyze gathered data
Tightness Evaluator

Single attribute is not enough to classify relationships

Intersection of different attributes

Five most significant Facebook attributes

1. # of messages friend has sent to the user
2. # of messages user has sent to the friend
3. # of user’s photos friend is tagged in
4. # of user’s posts friend is tagged in
5. # of posts friend has sent to the system user
• **Default Decision Tree Model** useful for the first evaluation
• Users can **improve** the Decision Tree Model specifying the tightness of (part of) their relationships
Tightness Evaluator: implementation insights

- Social meta/data gathered from OSNs
- Producing suitable information to support Decision Tree Generator
- Periodic data updating

- Decision tree based on gathered information
- J48 (C4.5) algorithm
- Weka APIs to perform classification

- Usage pattern computation to customize the decision tree
- User’s direct customization

Specific OSN APIs
Visibility Tuner: Packet Filters

- Packet monitoring and runtime enforcement of visibility rules
- **Discovery Filter**: to hide/expose nodes and services
- **Browse Filter**: to tune the visibility of shared contents
- **Action Filter**: to tune the access to supported features and resources

**Ingress packet filtering**
(dropping a request)

**Egress packet filtering**
(selective forwarding of only replies coming from given devices)
Visibility Tuner: Rules grammar

**Discovery rule:**
1. DF_FacebookFriend_tight = /, 5545 [/, +FS]
2. DF_GenericSocial_loose = /

**Browse rule:**
3. BF_TwitterFriend_tight = *
4. BF_TwitterFriend_57349 = /, 5545 [/, FS ".*?\.(mp3|divx)$"]

**Action rule:**
5. AF_FacebookFriend_regular = /, 5545 [/, FS [/, +get]]
6. AF_Generic = *, 124878 [/, Light [/, +SwitchPower]]
Visibility Tuner: implementation insights

- Adding/removing filters and de/activating them
- Performing filtering actions at runtime applying filter objects
- Storing Java objects implementing actual filters
- Ex: Discovery Filter for RAMP-based File Sharing service, Action Filter for UPnP devices
Experimental results: testbed

- Required times to evaluate tightness
- Required times to filter packets

- RAMP-based File Sharing server
- UPnP Media Server

Facebook

Data updating

GW

UPnP AV packets

UPnP Control Point

RAMP-based File Sharing packets

RAMP-based File Sharing client

Bologna, Italy - 11.11.2013
Stefano Poli – Mobilware ’13
Experimental results: Tightness Evaluator

- Required time to process **User**, **Friend List** and **Friend Data**
- **Friend Data sub-phase** greatly influences the total time

Required time to **classify relationships** linearly grows with the amount of friends
Experimental results: Visibility Tuner

Filtering of ingress “browse” request packets (File Sharing 53 bytes; UPnP 52 bytes)

- Filtering of egress “browse” packets (responses to “browse” requests; 60 bytes) for the File Sharing service
- Parsing and modification of responses → Almost linear growing with the packet rate
Conclusions and ongoing work

- **Automatic creation of SNs and UCNs**
- **Federation of UCNs based on Social Relationships**
- **Automatic Social Relationship tightness evaluation**
- **Feature and resource visibility tuning** based on Social Relationship Tightness (Packet Filters)

Ongoing work

- GUI to facilitate rules definition
- Support to NAT-T environment
- Hybrid cloud-based deployment to improve availability
The Real Ad-hoc Multi-hop Peer-to-peer (RAMP) Middleware: Content sharing and Social Networking

THANKS FOR YOUR ATTENTION

Questions time...

Paolo Bellavista
Carlo Giannelli
Stefano Poli

see also

P. Bellavista, C. Giannelli, L. Iannario, L. Goix, C. Venezia:
"Peer-to-peer Content Sharing Based on Social Identities and Relationships",
IEEE Internet Computing (accepted for publication)