Routing in Opportunistic Networks

Silvia Giordano
silvia.giordano@supsi.ch
work in collaboration with:
Marco Conti, Jon Crowcroft, Pan Hui, Hoang Anh Nguyen, Andrea Passarella
and the whole Haggle project

thanks to Cecilia Mascolo and Martin May

Outline

- Internet: where are we?
  - ubiquitous computing/communication
  - clean-slate approach
- New Networking Technologies
  - Haggle
  - DTN
  - Opportunistic Networks
- Routing in OppNets
  - context based routing
  - C-BR classification
  - SANETs
Ubiquitous Computing: Business Users

- users must interrupt their work when they move
- users can work without interrupting their work even when they move

Ubiquitous Computing: Other Users

- visiting people
- young people
First dream

Example: realize the ubiquitous computing with MANET technology.


Missing a pragmatic approach!

Example: realize the ubiquitous computing with MANET technology.

But pervasiveness is a reality

Chatpen

- Digital Information recognized and transmitted by pen

Ceiva

- Digital Picture Frame
  - Plug & Play
  - 640 x 480 VGA
  - JPEG Image Format
  - Storage: 10 photos/250 on line

Clean-slate approach

- Heterogeneous networks technologies
- Heterogeneous applications
- New architecture and communications paradigms:
  - DTNs
  - ANA
  - Haggle:
    - opportunistic networks,
    - social networks
The Internet Hourglass

Voice, Video, P2P, Email, youtube, ....
Protocols – TCP, UDP, SCTP, ICMP, ....

Changing/updating the Internet core is difficult or impossible!
(e.g., Multicast, Mobile IP, QoS, ...)

Disruptive approaches need a disruptive architecture

IP on Everything
IPv6
Everything on IP

Ethernet, WIFI (802.11), ATM, SONET/SDH, FrameRelay,
modem, ADSL, Cable, Bluetooth...

Approach 1: ANA

- We have to widen the waist and host more paradigms
- Federation instead of homogeneous abstraction
- May the best concept wins

We need the framework for these multiple networks
ANA ≠ "one-size-fits-all"

- A network compartment implements the operational rules and administrative policies for a given communication context

**ANA framework**

- Publish Subscribe Compartment
- PodNet compartment

---

**Approach 2: DTN**

- Support interoperability across 'radically heterogeneous' networks
- Tolerate delay and disruption
  - Acceptable performance in high loss/delay/error/disconnected environments
  - Decent performance for low loss/delay/errors

Still IP based
Delay-Tolerant Networking Architecture*

- Components
  - Flexible naming scheme
  - Message abstraction and API
  - Extensible Store-and-Forward Overlay Routing
  - Per-(overlay)-hop reliability and authentication

Still IP based

*from DTN group slides
DTNs: What?

- Network with intermittent connectivity
- Data are disseminated exploiting host mobility
  - long and variable delays
  - high error rates
  - heterogeneity

DTNs: Why?

- To cover regions with no infrastructure
- Human social networks
- Intelligent highways
- Emergency rescue
- Sensor networks
- Wildlife monitoring
Approach 3: Haggle

- new autonomic networking architecture
  - enables communication for intermittent network connectivity,
  - exploits autonomic opportunistic communications (i.e., when no end-to-end communication infrastructures).
- support interoperability across 'radically heterogeneous' networks

No more layers!

Haggle (2006-2010)

- Started January 2006 - EU FP6 - FET
- developed 3 prototypes (Infant, Child & Young Haggle)
- code on sourceforge
- http://www.haggleproject.org
- a revolutionary paradigm for communicating without end-to-end connection
- relax the end-to-end paradigm

Haggle

User

Haggle control

HIS

Resource control
Incentive
Authentication

Forward

Classifier

802.11 Bluetooth GPRS Connectivity management Cradle Ethernet

- a context based routing
- forwarding manager with more strategies
Opportunistic Networks

Opportunistic networking: we are surrounded by communicating devices and most of applications are not real time constrained

- relax the end-to-end paradigm
- communication is “on-the-fly”
- heterogeneous devices
  - mobile phones
  - laptops, palm
  - sensors, etc...
- exploit mobility
- real best-effort

Opportunistic Networks - The Missing Link?

In oppnets, diverse systems—not deployed originally as oppnet nodes—join an oppnet dynamically in order to perform certain tasks they have been invited (or ordered) to participate in

- Not always connected, “internet connectivity islands”
- Huge amount of untapped resources in portable devices
  - Local wireless bandwidth
  - Storages
  - CPUs
- A packet can reach destination using network connectivity or user mobility
Objectives

- Oppnets are envisioned to provide, among others:
  - Bridges between disjoint communication media
  - Additional platforms for offloading tasks
  - Additional sensing modalities by integrating existing independent sensory systems
- Oppnets can exploit context information

OppNets routing: context information

- the *context* in which the users communicate.
- Examples:
  - users’ working address and institution, personal information, habits
OppNets routing: context information

- Thus:
  - given context information about the destination
  - suitable forwarders can be found based on:
    - the probability of meeting with other users
    - the probability of visiting particular places

Context aware routing classification

- context-oblivious:
  - basically exploit some form of flooding

- partially context-aware (aka mobility-based):
  - exploit some particular piece of context information to optimise the forwarding task.

- fully context-aware (aka social context-aware):
  - not only exploit context information to optimise routing, but also provide general mechanisms to handle and use context information.

I’ll wait the school bus and give the message to someone in it, who is going to school, who is likely to meet Jack in few minutes.
Context-oblivious Protocols

- exploit some form of flooding.
- a message should be disseminated as widely as possible.
- might be the only solution when no context information is available.
- generate a high overhead
- may suffer high contention and potentially lead to network congestion

To limit this overhead, the common technique is to control flooding
- by limiting the number of copies allowed to exist in the network
- by limiting the maximum number of hops a message can travel.
- In the latter case, when no relaying is further allowed, a node can only send directly to the destination when (in case) it is met.
Context-oblivious Protocols

- **Epidemic Routing**

- **Network Coding**

- **Spray&Wait**

---

When two nodes meet exchange summary vectors with compact unambiguous representation of the messages currently stored in their local buffers.

Then, each node requests from the other the messages it is currently missing.

hop count limits dissemination (hc = 1 -> only to dest.).
Context-oblivious Protocols

- Network Coding
  - an original approach to limit message flooding.

- In general, network coding-based routing outperforms flooding, as it is able to deliver the same amount of information with fewer messages injected into the network.

---

Context-oblivious Protocols

- Spray&Wait
  - two phases: the spray phase and the wait phase.
    - spray phase: L copies of the same message are spread over the network
    - wait phase: each relay node stores its copy and eventually delivers it to the destination when (in case) it comes within reach.
  - L can is chosen based on a target average delay.
  - spray can be performed in multiple ways (e.g. binary)
- **Partially Context-aware**

- exploit some particular piece of context information to optimise the forwarding task.
  - e.g. mobility of nodes

- Main difference with fully context-aware:
  - generally is customized for a specific type of context information
  - does not provide a full-fledged set of algorithms to gather and manage any type of context information

**Probabilistic ROuting Protocol using History of Encounters and Transitivity (PROPHET)**


**MV and MaxProp**


Partially Context-aware

- **Last Encounter Routing and Spray & Focus**

- **MobySpace Routing**

- **Bubble Rap**

---

**PROPHET**

- evolution of Epidemic: during a contact, nodes also exchange their delivery predictability (DP) to destinations of messages they store in their buffers
- messages are requested only if DP is higher than that of the node currently storing the message.
- DP is the probability for a node to encounter a certain destination
  - increases when the node meets the destination
  - decreases (according to an ageing function) between meetings
- context information used by PROPHET is the frequency of meetings between nodes
- **MV and MaxProp**
  - in addition, also exploit information about the frequency of visits to specific physical places.

- **Last Encounter Routing and Spray&Focus**
  - exploit the fact that the decreasing gradient of the time lag identifies a suitable path towards the destination.

- **MobySpace Routing**
  - the mobility pattern of nodes is the context information used for routing.

---

- **Bubble-Rap**
  - automatically inferring the parameters of the underlying social structure.
  - exploiting the structure properties to select paths.
  - dynamically identifies users’ communities, and ranks the nodes “sociability” (measured as the number of links) inside each community.
Fully Context-aware

- provide general mechanisms to handle and use context information
- more general than PCA:
  - works with any context information
  - can be customized for the specific environment it must work on

Context-Aware Routing (CAR)

History Based Opportunistic Routing (HiBOp)

Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (PROPICMAN/Spatio-Tempo)
Context-Aware Routing (CAR)

- Nodes are divided in groups
  - Nodes inside the same group have **social links** between each other
  - Are connected by an underlying MANET routing protocol
- Nodes of **different groups** can have social links as well
- Nodes reconfigure

**CAR**

- A node selects the fwder node/group
  - **Probabilistically**
  - Based on the **social attraction** among nodes
- multi-attribute utility-based framework is defined to combine context information for computing the delivery probabilities
- **CAR** compute delivery probabilities proactively, and disseminate them in their ad hoc cloud.
- context information is exploited to evaluate probabilities just for known destinations
CAR: conjunction between PCA & FCA

- The framework is general enough to accommodate different types of context information.
- Context information in CAR consists of the logical connectivity of nodes, such as the rate of connectivity change, the delivery probability of the neighbor nodes to the destination, and device information, such as residual battery life.
- However, the social context information is not taken into consideration as in social context-based category.
- CAR can be seen as a conjunction between mobility-based protocols and social context-based protocols.

HiBOp

- Mimic the establishment of social relationships among people
- Users describe themselves
  - Users decide what information about themselves and their habits they want to make publicly available
  - Users store this information into their own devices
  - HiBOp is not bound to any predefined set of information

<table>
<thead>
<tr>
<th>Personal Information</th>
<th>attributes</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Donald</td>
<td></td>
</tr>
<tr>
<td>Surname</td>
<td>Duck</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:d.duck@cnr.it">d.duck@cnr.it</a></td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>3403434398</td>
<td></td>
</tr>
<tr>
<td>Work</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td>Via Moruzzi, 1</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>Pisa</td>
<td></td>
</tr>
<tr>
<td>Institute</td>
<td>IIT-CNR</td>
<td></td>
</tr>
</tbody>
</table>
HiBOp in a nutshell

- Sender does several copies ($k$) of the message
- To achieve a target delivery probability

$$k = \min \left\{ K \mid \prod_{i=0}^{K} (1 - p_{\text{succ}}(i)) \leq p_{\text{max}} \right\}$$

- Messages are discarded after being handed over to the next hop
- i.e., message spread is controlled by the target delivery probability

Propicman: Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network

- Exploit the context information of nodes to select the best next hop candidate.
- Each node has a common node profile with evidence/value pairs.
- Evidences have different weights.
- Compute the deliverity probability to the destination.
- Zero-conf
Propicman social approach

- Common set of evidences for every node - T

<table>
<thead>
<tr>
<th>Evidence names(E)</th>
<th>Weights(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>W1</td>
</tr>
<tr>
<td>E2</td>
<td>W2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>En</td>
<td>Wn</td>
</tr>
</tbody>
</table>

- Node profile:

<table>
<thead>
<tr>
<th>Evidence names</th>
<th>Values</th>
<th>Hashed values</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>V1</td>
<td>H(E1,V1)</td>
</tr>
<tr>
<td>E2</td>
<td>V2</td>
<td>H(E2,V2)</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>En</td>
<td>Vn</td>
<td>H(En,Vn)</td>
</tr>
</tbody>
</table>

An example:

<table>
<thead>
<tr>
<th>Evidence names(E)</th>
<th>Weights(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work place</td>
<td>10</td>
</tr>
<tr>
<td>City</td>
<td>5</td>
</tr>
<tr>
<td>Profession</td>
<td>7</td>
</tr>
<tr>
<td>Nationality</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evidence names</th>
<th>Values</th>
<th>Hashed values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work place</td>
<td>SUPSI</td>
<td>26ea4b976c47c022b2383...</td>
</tr>
<tr>
<td>City</td>
<td>Lugano</td>
<td>14e126e02d9e0...</td>
</tr>
<tr>
<td>Profession</td>
<td></td>
<td>c056817c8f1771...</td>
</tr>
<tr>
<td>Nationality</td>
<td>Swiss</td>
<td>8bb8105e7b76cc201775...</td>
</tr>
</tbody>
</table>
Propicman

- Neighbors do local matching between received $h_w$ with its node profile.
  
  Message header $h_w$

  
<table>
<thead>
<tr>
<th>Hashed values in node profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(E_{D1}, V_{D1})$</td>
</tr>
</tbody>
</table>

  Hashed values in node profile

  - Then neighbor node (A) can compute its $P_A$:

  $$0 \leq P_A \leq 1$$

  - $\sum W_M$: sum of the weights of all the evidences that S knows about D.
Similarity!

One cycle

7am-8am  8am-9am  9am-12am  12am-1pm  1pm-5pm  5pm-8pm  8pm-7am

---

SpatioTempo – Design

- Each period can have different contact opportunities. In some periods source node encounters few devices, some encounter bunch of devices with high DP.

$$T^{c-1}_i$$: period $i$th of the cycle $c-1$

$$S^{c-1}_i$$: set of encountered nodes in period $i$th of the cycle $c-1$
Variation of the sets of encounter nodes

- The variation of the set of encounter nodes in a period is indicated by $EV_i$.
- $EV_i$ is computed with respect to the past cycle (same period) and to the previous period (same cycle).

$$EV_i = \frac{|S_i^{c-1} \cup S_{i+1}^{c-1} - S_i^{c-1} \cap S_{i-1}^c|}{|S_i^{c-1} \cup S_{i-1}^c|}$$

SpatioTempo – Design

- Each person is supposed to have a set of frequent contact persons (FD).
- The source node $S$ wants to send a message $M$ to the destination $D$. There are 2 cases:
  - $D$ belongs to FD
  - $D$ is an occasional destination ($D$ belongs to OD)
Recording task

Compute the number of retransmissions

- \( D \) belongs to OD: Basically use Propicman to select the next hop with the number \( F \) of rebroadcasting \( M_H \), depends on the battery level \( b \), the priority level of \( \mu \) message (\( \beta \)) and \( EV_i \) in each period.

\[
F = \begin{cases} 
0 & \text{if } b < \beta \\
\text{round}(EV_i, \mu, t_i^*) & \text{otherwise}
\end{cases}
\]

: number of units of time in the period
D belongs to FD

- Delivery probability of a period: $DP_i^C$
  
  \[
  DP_i^C = \frac{DP_i^{C-1} + \eta \cdot DP_{i-1}^C}{1 + \eta}
  \]

  Delivery probability to reach the destination if the sender sends the message in this period.

- The number of retransmissions:
  
  \[
  F = \text{round} \left( \frac{F_M \cdot DP_i^C \cdot EV_t + (F_M - F_m) \cdot DP_i^C + F_m}{DP_i^C \cdot EV_t + 1} \right)
  \]

  $F_M$: maximum and minimum number of times that S is able to send during the period.

- Given $i$ is number of units of time in the period
- And if S can resend $M_i$ each unit of time in the period, then $F_M$ equals $i$

  \[
  F = \text{round} \left( \frac{t_i \cdot DP_i^C \cdot EV_t + (t - 1) \cdot DP_i^C + 1}{DP_i^C \cdot EV_t + 1} \right)
  \]
OD/FD management & Message lifetime

- After one cycle, nodes review its OD, FD lists
- Each member in FD, OD of S has a frequency level (FL), which indicates how often S has a message to it.

$$FL_D > \delta : \text{removed} \quad FL_D > \delta : \text{promoted}$$

- Message lifetime: $$M_{TTL} = \mu * C_T$$
  - $$C_T$$: The duration of time of a cycle
Predict next best interval

- Having information on the past cycle and past interval, we can predict next best interval

\[ S_{i-1}^{c-1}, S_{i+1}^{c-1}, S_{i+1}^{c-1}, D_{i}^{c} \]

Opportunistic Sensor Networks

- Context oblivious: Data Mules, SWIM, ZebraNet, Message Ferrying, DFT-MSN
Opportunistic Sensor Networks

- Context aware
  - SCAR = Sensor based Context-Aware Routing
  - PROSAN: PRobabilistic Opportunistic Routing in SANets

Motivation

- Pervasiveness and repeating behavior are true also in SANET:
  - people who repeatedly stay and pass in area A are the ones who most likely want, need and will support a reaction.

- By naturally acting, people can enable the communication between sensors and actuators, and trigger the reaction.
Motivating Scenario

- same for animals

SCAR
- Mobile sensors collect information and try to deliver this opportunistically to one of the sinks
  - Energy is an issue (batteries cannot be changed so easily)
  - They exploit "best" carriers to deliver the information back to sinks
    - Choice of the best carriers is key to the protocol: SCAR uses utility theory and Kalman Filter based prediction of various attributes to choose
Assumptions

- Data is reasonably delay tolerant
- Clocks of the sensors are (reasonably) synchronized and the sensors wake up at regular intervals
- Sensors are all cooperating (possibly owned by same developer)
- Mobility is not source of energy consumption

The key ideas of SCAR

- Each sensor collects and stores its data
- Each sensor will try to deliver its data in bundles to a number of neighbouring sensors which seem to be the best carriers to reach a sink
- Decision process based on the prediction of the evolution of the system:
  - mobility of nodes
  - colocation with the sinks
  - battery level
- The forecasted values of the context attributes are then combined to define a probability $P(s_i)$ of the sensor $s_i$ of delivering the bundles to one of the sinks
Multiple-carrier Selection

- Each node maintains a list of the neighbouring nodes (including itself) ordered according to their delivery probabilities.
- Each source node then replicates the bundle to the first $R$ nodes ($R$ nodes if the node itself is in the first $R$ positions of the list).
  - Data bundles are only replicated on $R$ nodes by the source, while they are forwarded (deleted from a node and copied on another) if the carriers, while roaming, find either a sink or a better carrier.
  - The value of $R$ is specified by the user and can be considered as a priority level associated to the data retrieved by the sensor.

Bundle Forwarding

- Each node keeps monitoring if there are neighbours with better probability of delivery than its own.
- If this is the case the data bundles are shifted to the other nodes.
- Key problem: intelligent replication.
Forecasting Techniques for Probabilistic Routing

- Each node calculates its delivery probability using multi-criteria decision theory over the evolution of its context described by a set of attributes.
- In particular, we consider three attributes:
  - Colocation with the sinks
  - Change degree of connectivity (i.e., how many nodes have been seen in a fraction of time)
  - Battery level
- Our aim is to maximize the combined utility function (i.e., selection of the nodes with the best trade-off between these attributes)

Prosan design

- Based on user profiles
  - Using users’ social behavior - node profile - to predict the mobility of nodes.
- Routing with Zero Knowledge
  - Sender does not need to know any information about other nodes - no synchronizing, no local table.
  - Only senders broadcast helloing messages.
- No sharing user information
  - Nodes do not need to share, periodically update their information to others.
  - Keep privacy amongst members.
Forwarding strategy

- Delivery probability (DP): The probability of a node to meet and forward a message to other nodes toward destination.

- The sender S wants to send a message M to the destination actuator D, S will choose the route that has the highest *delivery probability* (DP) to send M to D.

- Before sending the message content M to the destination D, S sends M’s header to its neighbors.

  \[ h_M = \text{Concat}(H(E_{Di}, V_{Di}), MAC_S, SN_M) \]  

*Example:*

```
26a48976c47c0d2b2383a3403de937e6bf8d16c554b9f72143d12b3c2d9c5cf50d086e7967327b607d52b008
ff1c110c09b62d36a395c27b76627564e92f52603c6599f05f087f3110b7f171871af402b7adbbf88055e
0ad74c0a08419e9f11b077b2c244566817c8c4ce423e0b923b70c9c5a61758486.
```
Forwarding strategy (cont)

- Neighbors do local matching between received $h_M$ with its node profile.

  Message header $h_M$

  \[
  \begin{array}{ccccccc}
  H(M), V & H(R), V & H(R), V & \ldots & H(R), V & MAC & SN \\
  \end{array}
  \]

- With each matching, neighbor nodes look up in $T$ to get the corresponding weight. Then neighbor node (A) can compute its $DP_A$ as follows:

  \[
  0 \leq DP_A = \sum_{i}^{\text{matched}} W_i \leq 1
  \]

- $\sum W_M$ : sum of the weights of all the evidences that $S$ knows about $D$.

- $S$ will choose the route(s) with highest $DP$ it receives from its neighbor nodes.

- $S$ only sends the message to these routes if $DP_{\text{route}} > DP_S$.
Simulation environment

- Hybrid P-MULEs
  - nodes with higher capacity
  - use Prosan
  - move as MULEs
- Simulation area
  - rectangle 870x520 meters
  - divided in 6 areas
  - 50 nodes (1 p-mule)
  - use sleep mode
- Simulation approach
  - static actuator
  - mobile actuator

Simulation results for static actuator

![Graph showing simulation results for static actuator](image-url)
Conclusion

- Context information in opportunistic routing can be very advantageous.
- Exploit effectively the mobility, the social connections and the community relationships:
  - Reduce significantly the number of nodes involved in
  - Reduce the number of broadcasts
  - Efficiently control message lifetime
- Assumption for effectively using context information
  - People move in a predictable fashion based on repeating behavioral patterns at different timescales (day, week, and month)
  - Or these patterns are known
- -> Mobility as expression of the users social behavior
- Haggle architecture for supporting context aware routing
Thanks