SCOOP:
Decentralized and Opportunistic Multicasting of Information Streams

D. Gunawardena
Microsoft Research

T. Karagiannis
Microsoft Research

A. Proutiere
KTH

E. Santos-Neto
British Columbia

M. Vojnovic
Microsoft Research
Opportunistic Communication

- Aims at leveraging mobility for content delivery in networks of devices experiencing intermittent connectivity

- Applications: disaster recovery and challenged networks, DTNs, alleviate congestion in 3G / 4G cellular systems (?)
Routing / Relaying Strategies

• Main challenge in opportunistic communication: design optimal and decentralized relaying strategies

• **Forwarding** protocols: maintain a single copy of each message, e.g. Jain et al. (sigcomm’04)

• **Epidemic** routing: replicate messages, e.g. RAPID, Balasubramanian et al. (sigcomm’07)

• Drawbacks of existing approaches
  - infer mobility and track expected delays towards destination using simplifying assumptions on mobility: independence of delays through various paths, exponential inter-contact times
  - based on heuristics: not maximizing an a-priori well-defined global system objective
Our contribution: SCOOP

A novel relaying strategy that
• maximizes some global system objective,
• accounts for storage and transmission costs at relays,
• supports multi-point to multi-point communications,
• is decentralized (decisions based on local information),
• allows for general node mobility (correlated delays across paths and arbitrary inter-contact time distributions).
Outline

1. Analysis of mobility traces
2. SCOOP: Optimal relaying strategy -- Theory and Practice
3. Numerical experiments
1. Mobility traces: Path length and delay correlations
Multi-hop relaying

- Traces

<table>
<thead>
<tr>
<th>Name</th>
<th>Technology</th>
<th>Duration</th>
<th>Devices</th>
<th>Contacts</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSD</td>
<td>WiFi</td>
<td>77 days</td>
<td>275</td>
<td>116,383</td>
<td>2002</td>
</tr>
<tr>
<td>Infocom</td>
<td>Bluetooth</td>
<td>3 days</td>
<td>37</td>
<td>42,569</td>
<td>2005</td>
</tr>
<tr>
<td>DieselNet</td>
<td>WiFi</td>
<td>20 days</td>
<td>34</td>
<td>3,268</td>
<td>2007</td>
</tr>
<tr>
<td>SF Taxis</td>
<td>GPS</td>
<td>24 days</td>
<td>535</td>
<td>183M</td>
<td>2008</td>
</tr>
</tbody>
</table>

- Questions:
  1. How many hops do we need for acceptable performance?
  2. What are the statistical properties of the discovered paths? Are delays on different paths independent?
2 hops are enough

![Graph showing nodes reached (%) over time (hours). The graph includes four lines labeled Infocom, 1 hop, 2 hops, and infinite hops. The x-axis represents time in hours, ranging from 0 to 80, and the y-axis represents the percentage of nodes reached, ranging from 0 to 100.](image)
2 hops are enough
Paths positive correlations

![Graph showing CDF of correlation coefficients with mean, median, and percentiles marked.](image)
Paths positive correlations

Correlation coefficient vs. Time (hours) for SF Taxis
2. SCOOP: Optimal relaying scheme Theory and Practice
Network setting

• Multi-point to multi-point communication
• For a given information stream: a set of sources and a set of interested users
Network setting

- **Stream-\(i\) sources**: generate messages according to a stationary ergodic point process of intensity \(\lambda_i\)

- **General mobility model** (stationary ergodic processes)

\[
\begin{align*}
D_{i,u} & : \text{time it takes for a stream-}\(i\) message to reach user } u \text{ without the help of any relay} \\
D_{i,r,u} & : \text{time it takes for a stream-}\(i\) message to reach user } u \text{ through relay } r
\end{align*}
\]
Network setting

- **Relays**: buffer size of relay $r$, $B_r$
- **Probabilistic relaying scheme**: parameterized by $x \in [0,1]^{I \times U}$
  
  $x_{i,r}$ : probability that relay $r$ relays a message from sources of stream $i$

**Ex:** Relay $r$ meets a stream-i source at times $(T_1, T_2, ...)$
Consider messages in chronological order for upload

Stream-i deadline: $t_i$

max($T_1 - t_i, T_2$) $T_2$ message generation process of stream $i$ time

w.p. $x_{i,r}$
Network setting

• **Relays**: buffer size of relay $r$, $B_r$

• **Probabilistic relaying scheme**: parameterized by $x \in [0,1]^{I \times U}$

\[ x_{i,r} : \text{probability that relay } r \text{ relays a message from sources of stream } i \]

**Ex:** Relay $r$ meets a stream-$i$ source at times $(T_1, T_2, \ldots)$
Consider messages in chronological order for upload
Stream-$i$ deadline: $t_i$
FIFO buffer management

\[
\begin{align*}
\max(T_1 - t_i, T_2) & \\
T_2 & \\
\text{message generation process of stream } i & \\
\text{time} &
\end{align*}
\]
Network setting

- **Relays**: buffer size of relay \( r, B_r \)
- **Probabilistic relaying scheme**: parameterized by \( x \in [0,1]^{I \times U} \)

\( x_{i,r} \) : probability that relay \( r \) relays a message from sources of stream \( i \)

**Ex:** Relay \( r \) meets a stream-i source at times \( (T_1, T_2, \ldots) \)
Consider messages in chronological order for upload
Stream-i deadline: \( t_i \)
FIFO buffer management

\[
\text{max}(T_1 - t_i, T_2) \quad T_2
\]
Objective

• **Performance**: user-$u$ performance measured through

\[ \sum_i a_{i,u} P_x[A_{i,u} \leq t_i] \]

- $a_{i,u}$: binary variable indicating whether user $u$ is interested in stream $i$
- $A_{i,u}$: age of a stream-$i$ packet when arriving at user $u$

• **Global system objective**: Identify the strategy *optimally* exploiting mobility and buffer constraints at relays, i.e., solving:

\[
\text{maximize} \quad \sum_{i,u} a_{i,u} P_x[A_{i,u} \leq t_i] \quad \text{over} \quad x \in [0,1]^{I \times U}
\]
Sub-gradient algorithm

- The following updating rule converges to a solution:

\[ \frac{dx_{i,r}}{dt} = \sum_{j,u} a_{j,u} \frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] \]

- Problem: how to estimate \( \frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] \)?

- Key idea: Smoothed Perturbation Analysis (SPA) techniques -- see the paper for details
Gradient estimator

**Theorem**

\[
\frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{A_{i,u} > t_i} 1_{A_{i,r,u} \leq t_i} 1_{i=j} - E_x[I_{j,r,u} R_{j,r} \left( N_{j,r,u}^i 1_{N_{j,r,u}=B_r} + K_{j,r,u}^i 1_{N_{j,r,u}=B_{r-1}} \right)]
\]
Gradient estimator

Theorem

\[
\frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{A_{i,u}^r > t_i} 1_{A_{i,r,u} \leq t_i}] 1_{i=j} \]

\[- E_x[I_{j,r,u} R_{j,r} \left( N^i_{j,r,u} 1_{N_{j,r,u} = B_r} + K^i_{j,r,u} 1_{N_{j,r,u} = B_r - 1} \right)]\]

Positive effect of increasing \( x_{i,r} \). For stream \( i \) only, through

\[1_{A_{i,u}^r > t_i} 1_{A_{i,r,u} \leq t_i}\]

event “stream-\( i \) packet cannot reach user \( u \) before deadline without the help of relay \( r \), but could do it using relay \( r \)”
Gradient estimator

Theorem

\[
\frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{\hat{A}_{i,u} > t_i} 1_{\hat{A}_{i,r,u} \leq t_i}] 1_{i=j} - E_x[I_{j,r,u} R_{j,r}(N_{j,r,u} 1_{N_{j,r,u}=B_r} + K_{j,r,u}^i 1_{N_{j,r,u}=B_r-1})]
\]

Negative effect of increasing \( x_{i,r} \). For stream \( j \) through

\[
I_{j,r,u} R_{j,r}(N_{j,r,u}^i 1_{N_{j,r,u}=B_r} + K_{j,r,u}^i 1_{N_{j,r,u}=B_r-1})
\]
Gradient estimator

**Theorem**

\[
\frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{A_{i,u}>t_i}1_{i,r,u\leq t_i}] 1_{i=j} - E_x[I_{j,r,u}R_{j,r} (N_{j,r,u}^{i} 1_{N_{j,r,u}=B_{r}} + K_{j,r,u}^{i} 1_{N_{j,r,u}=B_{r}-1})]
\]

**Negative effect of increasing** \( x_{i,r} \). For stream \( j \) through event “stream-\( j \) packet cannot reach user \( u \) before deadline without relay \( r \), and the two hop delay via relay \( r \) is smaller than the deadline”
Gradient estimator

Theorem

\[ \frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{\hat{A}_{i,u,r} < t_i} 1_{\hat{A}_{i,u,r} \leq t_i}] 1_{i=j} \]

\[ - E_x[I_{j,r,u} R_{j,r} (N_{j,r,u}^i 1_{N_{j,r,u} = B_r} + K_{j,r,u}^i 1_{N_{j,r,u} = B_r - 1})] \]

Negative effect of increasing \( x_{i,r} \). For stream \( j \) through

\[ I_{j,r,u} R_{j,r} (N_{j,r,u}^i 1_{N_{j,r,u} = B_r} + K_{j,r,u}^i 1_{N_{j,r,u} = B_r - 1}) \]

binary variable indicating whether relay \( r \) uploads stream-\( j \) packet.
Gradient estimator

**Theorem**

\[
\frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{A_{i,u} > t_i} 1_{\hat{A}_{i,r,u} \leq t_i}] 1_{i=j} \\
- E_x[I_{j,r,u} R_{j,r} \left(N_{j,r,u}^i 1_{N_{j,r,u} = B_r} + K_{j,r,u}^i 1_{N_{j,r,u} = B_r - 1}\right)]
\]

**Negative effect of increasing** \(x_{i,r}\). For stream \(j\) through

\[
I_{j,r,u} R_{j,r} \left(N_{j,r,u}^i 1_{N_{j,r,u} = B_r} + K_{j,r,u}^i 1_{N_{j,r,u} = B_r - 1}\right)
\]

records the number of stream-\(i\) packets uploaded by relay \(r\) after stream-\(j\) packet was uploaded, given that the latter was dropped just before meeting user \(u\).
Gradient estimator

**Theorem**

\[
\frac{\partial}{\partial x_{i,r}} P_x[A_{j,u} \leq t_j] = E_x[1_{\hat{A}_{i,u} > t_i, 1_{\hat{A}_{i,r,u} \leq t_i}}1_{i=j} \\
- E_x[I_{j,r,u} R_{j,r} \left( N_{j,r,u}^i 1_{N_{j,r,u} = B_r} + K_{j,r,u}^i 1_{N_{j,r,u} = B_r - 1} \right)]
\]

**Negative effect of increasing** \(x_{i,r}\). For stream \(j\) through

\[
I_{j,r,u} R_{j,r} \left( N_{j,r,u}^i 1_{N_{j,r,u} = B_r} + K_{j,r,u}^i 1_{N_{j,r,u} = B_r - 1} \right)
\]

records the number of stream-\(i\) packets *observed* but not uploaded by \(r\) after stream-\(j\) packet was uploaded, given that this packet is at the head of relay-\(r\) buffer (next to be evicted) when meeting user \(u\).
Implementation

• All quantities involved in the gradient estimator are observable locally by users and relays
• ... it can be implemented by relays using **local information** obtained from users
• For every stream-$j$ packet $m$ observed by relay $r$, the latter collects feedback from user $u$ to compute: for all $i$,

  a. **term used to increase $x_{i,r}$**

  $$\alpha_{i,r,u}(m) = a_{i,u} 1_{i=j} 1_{\hat{A}_{i,r}(m) > t_i} 1_{\hat{A}_{i,r,u}(m) \leq t_i}$$

  b. **term used to decrease $x_{i,r}$**

  $$\beta_{i,r,u}(m) = a_{j,u} I_{j,r,u}(m) R_{j,r}(m) \left( N_{j,r,u}^i(m) 1_{N_{j,r,u}(m) = B_r} + K_{j,r,u}^i(m) 1_{N_{j,r,u}(m) = B_r - 1} \right)$$
Implementation

• **Online updates:**
  When receiving the $n$-th feedback, say from user $u(n)$ for a stream-$c(n)$ packet, relay $r$ updates:

  $$x_{i,r}(n + 1) = x_{i,r}(n) + \epsilon \left( \frac{\sum_{j \in \Omega(r)} \lambda_j}{\lambda_{c(n)}} \right) \left( \alpha_{i,r,u(n)}(n) - \beta_{i,r,u(n)}(n) \right)$$

  $\Omega(r)$: set of streams observed by relay $r$

• Refer to the paper for a detailed description of the protocols
3. Numerical experiments
Setting

• Comparison with R-OPT (optimized version of RAPID) that
  - has **perfect knowledge** of mean delays, and existing other packet replicas in the network, when taking relaying decisions,
  - is adapted to multi-point to multi-point communication
  - is adapted or not to restrict to 2-hop relaying schemes

• SCOOP
  - with $\varepsilon = 0.01$
Performance - DieselNet

Buffer size = 1

Message inter-publish time (hours)

Delivery Ratio

- R-OPT
- R-OPT (2 hops)
- SCOOP

0 10 20
0.4 0.6 0.8 1

Buffer size = 10

Message inter-publish time (hours)

Delivery Ratio

- R-OPT
- R-OPT (2 hops)
- SCOOP

0 10 20
0.4 0.6 0.8 1

Buffer size = 100

Message inter-publish time (hours)

Delivery Ratio

- R-OPT
- R-OPT (2 hops)
- SCOOP

0 10 20
0.4 0.6 0.8 1
Performance - DieselNet

Buffer size = 10, MIPT (Message Inter-Publish Time)
Performance – SF Taxis

Buffer size = 10, MIPT = 12 hours
SCOOP vs. R-OPT

• SCOOP performs almost as well as R-OPT (an idealized version of RAPID that assumes full global knowledge)

• Restricting relaying schemes to 2 hops does not impact the performance

• Results verified on other traces, for various system settings
Conclusion

- We proposed SCOOP, a decentralized relaying algorithm for information stream multicast that
  - provably maximizes some global system objectives
  - does not rely on some mobility assumptions that could not be met in practice (e.g. statistically identical and independent path delays)

- SCOOP learns how to optimally exploit nodes’ mobility accounting for their limited storage capacity

- SCOOP performs almost as well as relaying schemes having full knowledge of mobility and existing message replicas in the system when taking relaying decisions