Content discovery in Information Centric Networks

*When to flood? Where to flood?*

Suzan Bayhan
Dept. of Computer Science, Collaborative Networking Research Group (CoNe)
University of Helsinki, Finland

http://www.hiit.fi/u/bayhan

December 28, 2015 Bogazici University, Department of Computer Engineering
Briefly Information-centric networking (ICN)

- Van Jacobson: congestion avoidance, traceroute, tcpdump
- 2006 Google Tech Talk on *A New Way to look at Networking*
- Many things have changed since IP specification RFC 791, 1981
- Mostly not conversational traffic but networks distribute content (mostly bandwidth-hungry video) in an inefficient unicast manner to mobile devices
- *What if telephony is not the only way to route data packets?*
- *What if IP is not the only way to distribute content?*
Information-centric Networks (ICN)

- Today’s host-centric IP interconnecting *machines*
Information-centric Networks (ICN)

- ICN interconnecting *information*
- Address content
Information-centric Networks (ICN)

- ICN interconnecting *information*
- Address content
- Add *in-network storage* everywhere, not only at the edge
- Apply *pervasive caching*
- Facilitate *nearest replica routing*
Internet vs. ICN

Host-centric communications (host-to-host)  
Content-centric communications (host-to-content)
Content is decoupled from its containers and move in the network.

Challenge: How to discover the requested content?
Scoped-flooding

- Propagate the content discovery message in a scope, i.e. number of hops message can travel
- Benefits of (scoped) flooding in the network
  - Low state maintenance, low protocol complexity, etc.
  - A scalable solution or not?
Scoped-flooding

Requested content not in the cache
Initiate content-discovery using scoped-flooding with scope 2
Scoped-flooding

- 1-hop flooding content discovery packet
Scoped-flooding

- 2-hop flooding content discovery packet
Technically we want to know
- How to set the flooding scope optimally?
- How a network topology impacts the scope?
- How content availability impacts the scope?

In short, we want to flood on the right content at the right place with the right scope.
Pro-Diluvian: Understanding Scoped-Flooding for Content Discovery in ICN

Liang Wang (Cambridge University), Suzan Bayhan (University of Helsinki), Jörg Ott (TU Munich, Aalto University), Jussi Kangasharju (University of Helsinki), Arjuna Sathiaseelan (Cambridge University), Jon Crowcroft (Cambridge University)
Three key components

1. **Content**

- The content (can be anything), only its value matters.
Three key components

1. **Content**
   - The **content** (can be anything), only its value matters.
   - The representation of **gain/cost** as a function of # of nodes and content (value).

2. **Utility** = gain-cost
Three key components

1. **Content**
   - The content (can be anything), only its value matters.

2. **Utility = gain-cost**
   - The representation of gain/cost as a function of # of nodes and content (value).

3. **Network topology**
   - The network model based on which, we can tell how the # of nodes increases as a function of # of hops (scope).
How are these components connected?

A node-centric ring-based model

Utility (ring r) = Gain(ring r) - Cost(ring r)

Ring 1: 1-hop neighbors

Ring 2: 2-hop neighbors
Utility modeling

Utility = Expected value of content – expected cost of content discovery
Utility modeling

Utility = Expected value of content – expected cost of content discovery

Content discovery success probability:
probability that the requested content is hosted by at least one of the nodes reached

# of total transmissions * cost,
cost: bandwidth, energy, delay
Utility modeling

Utility = Expected value of content – expected cost of content discovery

Content discovery success probability:
probability that the requested content is hosted by at least one of the nodes reached

Total number of nodes receiving the message = n
Content availability = p (uniform distribution)
Gain = 1 – q^n where q = 1-p

Cost = n*c

# of total transmissions * cost, cost: bandwidth, energy, delay
Utility modeling

Utility = Expected value of content – expected cost of content discovery

Content discovery success probability: probability that the requested content is hosted by at least one of the nodes reached

Total number of nodes receiving the message = \( n \)

Cost = \( n^c \)

Content availability = \( p \) (uniform distribution)

Gain = \( 1 - q^n \) where \( q = 1 - p \)

\[
U = (1 - q^n) - n \cdot c
\]
Effect of $p$ and $c$ on flooding behavior

- What is the critical cost $c$ below which the node will initiate scoped flooding for content with availability $p$ and given $n$?
  - The lower cost for higher $n$
  - A node with a large neighborhood is reluctant to flooding
Effect of p and c on flooding behavior

- What is the critical cost c below which the node will initiate scoped flooding for content with availability p and given n?
  - The lower cost for higher n
  - A node with a large neighborhood is reluctant to flooding

- How does availability p affect flooding?
  - Higher p, worth flooding to more neighbors
  - Lower p, conservative flooding
What is the relation between n and scope r?

- Given scope r, n is the number of nodes that will receive the message:
  \[ n = f(r) \]
- Graph with a given degree distribution \( \rho \), i.e., \( G = (N, \rho) \)
- h-hop neighborhood of a node: \( n_h \)
- \( n = \sum n_h \) where \( h \leq r \)
1-hop neighbors

- $k$: random variable representing node degree
- $<k>$: expectation of node degree variable
- $\rho_i = \Pr(k=i)$: the probability that a randomly selected node has degree $i$
1-hop neighbors

- $k$: random variable representing node degree
- $\langle k \rangle$: expectation of node degree variable
- $\rho_i = \Pr(k=i)$: the probability that a randomly selected node has degree $i$

- Expected number of 1-hop neighbors: $n_1 = \langle k \rangle = \sum_{k=0}^{\infty} k \rho_k$
1-hop, 2-hop neighbors

- Expected number of 1-hop neighbors: \( n_1 = \langle k \rangle = \sum_{k=0}^{\infty} k \rho_k \)

- Expected number of 2-hop neighbors:
1-hop, 2-hop neighbors

- Expected number of 1-hop neighbors: \( n_1 = \langle k \rangle = \sum_{k=0}^{\infty} k \rho_k \)

- Expected number of 2-hop neighbors:

The probability of \( v_j \) having \( k \) new next-hop neighbours is:

\[ \tau_k = \Pr[\text{deg}(v_j) = k|\rho] = \frac{(k+1)\rho_{k+1}}{\sum_m m\rho_m} \]

Therefore, the average number of new nodes from \( v_j \) is:

\[ \sum_{k=0}^{\infty} k \tau_k = \frac{\sum_{k=0}^{\infty} k(k+1)\rho_{k+1}}{\sum_m m\rho_m} = \frac{\sum_{k=0}^{\infty} k(k-1)\rho_k}{\sum_m m\rho_m} = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \]

- \( \langle k^2 \rangle \): second moment of node degree variable
1-hop, 2-hop, …, r-hops

- Expected number of 1-hop neighbors: \( n_1 = \langle k \rangle = \sum_{k=0}^{\infty} k \rho_k \)
- Expected number of 2-hop neighbors: \( n_2 = \langle k^2 \rangle - \langle k \rangle \)
- Expected number of r-hop neighbors:
  \[
  n_r = n_{r-1} \sum_{k=0}^{\infty} k \tau_k = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} n_{r-1} = \left[ \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \right]^{r-1} \cdot \langle k \rangle
  \]
1-hop, 2-hop, ..., r-hops

- Expected number of 1-hop neighbors: \( n_1 = \langle k \rangle = \sum_{k=0}^{\infty} k \rho_k \)

- Expected number of 2-hop neighbors: \( n_2 = \langle k^2 \rangle - \langle k \rangle \)

- Expected number of r-hop neighbors:

\[
n_r = n_{r-1} \sum_{k=0}^{\infty} k \tau_k = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} n_{r-1} = \left[ \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \right]^{r-1} \cdot \langle k \rangle
\]

\[
n_r = \left[ \frac{n_2}{n_1} \right]^{r-1} \cdot n_1
\]
Neighborhood growth rate

A node can estimate its neighbourhood with 2-hop knowledge.

\[ n_r = \left[ \frac{n_2}{n_1} \right]^{r-1} \cdot n_1 \]
Neighborhood growth rate

A node can estimate its neighborhood with 2-hop knowledge.

\[ n_r = \left[ \frac{n_2}{n_1} \right]^{r-1} \cdot n_1 \]

- Average network growth rate \( \beta \)
- The ratio of \# of ring \( r+1 \) nodes to \# of ring \( r \) nodes

\[ \beta \triangleq \frac{n_2}{n_1} \triangleq \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \]
Neighborhood growth rate

A node can estimate its neighbourhood with 2-hop knowledge.

\[ n_r = \left( \frac{n_2}{n_1} \right)^{r-1} \cdot n_1 \]

- Average network growth rate \( \beta \)
- The ratio of \# of ring \( r+1 \) nodes to \# of ring \( r \) nodes

\[ \beta \triangleq \frac{n_2}{n_1} \triangleq \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \]

\[ n_r = \left[ \beta \right]^{r-1} \cdot n_1 \]
**Neighborhood growth for random graphs**

\[ \beta \triangleq \frac{n_2}{n_1} \triangleq \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \]

– Erdös-Renyi graph (ER)
  
  • Every node is equally likely to be connected with every other node with prob. \( p \)
  
  • Poisson degree dist. for large \( N \), small \( p \)

– Scale-free graph
  
  • Some nodes are tightly connected (hub), some have only a few connections
  
  • Power-law degree dist. parameter \( \alpha \)
Neighborhood growth for random graphs

\[ \beta \triangleq \frac{n_2}{n_1} \triangleq \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \]

- Erdös-Renyi graph (ER) \( \beta = \langle k \rangle \)
  - Every node is equally likely to be connected with every other node with prob. \( p \)
  - Poisson degree dist. for large \( N \), small \( p \)

- Scale-free graph \( \beta = \frac{1}{\alpha - 3} \quad \forall \alpha > 3 \)
  - Some nodes are tightly connected (hub), some have only a few connections
  - Power-law degree dist. parameter \( \alpha \)
How accurate can this model predict?

Synthetic topologies with 10,000 nodes and analyze the largest connected component
Calculate the degree distribution parameters from the actual graphs and find $n_r$
How accurate can this model predict?

Synthetic topologies with 10,000 nodes and analyze the largest connected component. Calculate the degree distribution parameters from the actual graphs and find $n_r$.

Table 1: Overestimation of the model at each hop for various network graphs. $V$: Number of nodes and $E$: Number of nodes in the generated instance of the graph, $l$: average path length. Shaded cells represent the cases where the error is below 0.20.

<table>
<thead>
<tr>
<th>Id</th>
<th>Topology</th>
<th>$V$</th>
<th>$E$</th>
<th>$\langle k \rangle$</th>
<th>$l$</th>
<th>Clustering</th>
<th>$n_r$</th>
<th>$r = 2$</th>
<th>$r = 3$</th>
<th>$r = 4$</th>
<th>$r = 5$</th>
<th>$r = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Random</td>
<td>339</td>
<td>338</td>
<td>1.994</td>
<td>23.07</td>
<td>0</td>
<td>0.327</td>
<td>1.046</td>
<td>2.359</td>
<td>4.692</td>
<td>9.092</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Random</td>
<td>8030</td>
<td>9761</td>
<td>2.431</td>
<td>12.03</td>
<td>0</td>
<td>0.152</td>
<td>0.371</td>
<td>0.642</td>
<td>0.972</td>
<td>1.399</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Random</td>
<td>9426</td>
<td>15068</td>
<td>3.197</td>
<td>8.30</td>
<td>0.00040</td>
<td>0.060</td>
<td>0.130</td>
<td>0.212</td>
<td>0.332</td>
<td>0.565</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Random</td>
<td>9811</td>
<td>20073</td>
<td>4.091</td>
<td>6.75</td>
<td>0.00049</td>
<td>0.023</td>
<td>0.053</td>
<td>0.106</td>
<td>0.259</td>
<td>0.873</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Random</td>
<td>9928</td>
<td>25060</td>
<td>5.048</td>
<td>5.88</td>
<td>0.00048</td>
<td>0.004</td>
<td>0.017</td>
<td>0.079</td>
<td>0.419</td>
<td>2.79</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Random</td>
<td>9989</td>
<td>35020</td>
<td>7.011</td>
<td>4.95</td>
<td>0.00066</td>
<td>0.003</td>
<td>0.030</td>
<td>0.229</td>
<td>2.139</td>
<td>54.124</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Scale-free, $\alpha = 3.24$</td>
<td>7141</td>
<td>9648</td>
<td>2.70</td>
<td>7.88</td>
<td>0.00057</td>
<td>0.093</td>
<td>0.271</td>
<td>0.529</td>
<td>1.069</td>
<td>2.599</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Scale-free, $\alpha = 3.35$</td>
<td>5869</td>
<td>7347</td>
<td>2.50</td>
<td>8.66</td>
<td>0.00076</td>
<td>-0.115</td>
<td>-0.174</td>
<td>-0.194</td>
<td>-0.16</td>
<td>0.013</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Scale-free, $\alpha = 3.50$</td>
<td>5960</td>
<td>7357</td>
<td>2.47</td>
<td>8.99</td>
<td>0.00013</td>
<td>-0.356</td>
<td>-0.555</td>
<td>-0.68</td>
<td>-0.757</td>
<td>-0.794</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Change in neighbourhood in real ISP networks.
How accurate can this model predict?

Synthetic topologies with 10.000 nodes and analyze the largest connected component
Calculate the degree distribution parameters from the actual graphs and find \( n_r \)

Table 1: Overestimation of the model at each hop for various network graphs. \( V \): Number of nodes and \( E \): Number of nodes in the generated instance of the graph, \( l \): average path length. Shaded cells represent the cases where the error is below 0.20.

<table>
<thead>
<tr>
<th>Id</th>
<th>Topology</th>
<th>V</th>
<th>E</th>
<th>\langle k \rangle</th>
<th>l</th>
<th>Clustering</th>
<th>Overestimation of the model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( r = 2 )</td>
</tr>
<tr>
<td>1</td>
<td>Random</td>
<td>339</td>
<td>338</td>
<td>1.994</td>
<td>23.07</td>
<td>0</td>
<td>0.327</td>
</tr>
<tr>
<td>2</td>
<td>Random</td>
<td>8030</td>
<td>9761</td>
<td>2.431</td>
<td>12.03</td>
<td>0</td>
<td>0.152</td>
</tr>
<tr>
<td>3</td>
<td>Random</td>
<td>9426</td>
<td>15068</td>
<td>3.197</td>
<td>8.30</td>
<td>0.00040</td>
<td>0.060</td>
</tr>
<tr>
<td>4</td>
<td>Random</td>
<td>9811</td>
<td>20073</td>
<td>4.091</td>
<td>6.75</td>
<td>0.00049</td>
<td>0.023</td>
</tr>
<tr>
<td>5</td>
<td>Random</td>
<td>9928</td>
<td>25060</td>
<td>5.048</td>
<td>5.88</td>
<td>0.00048</td>
<td>0.004</td>
</tr>
<tr>
<td>6</td>
<td>Random</td>
<td>9989</td>
<td>35020</td>
<td>7.011</td>
<td>4.95</td>
<td>0.00066</td>
<td>0.003</td>
</tr>
<tr>
<td>7</td>
<td>Scale-free, ( \alpha = 3.24 )</td>
<td>7141</td>
<td>9648</td>
<td>2.70</td>
<td>7.88</td>
<td>0.00057</td>
<td>0.093</td>
</tr>
<tr>
<td>8</td>
<td>Scale-free, ( \alpha = 3.35 )</td>
<td>5869</td>
<td>7347</td>
<td>2.50</td>
<td>8.66</td>
<td>0.00076</td>
<td>-0.115</td>
</tr>
<tr>
<td>9</td>
<td>Scale-free, ( \alpha = 3.50 )</td>
<td>5960</td>
<td>7357</td>
<td>2.47</td>
<td>8.99</td>
<td>0.00013</td>
<td>-0.356</td>
</tr>
</tbody>
</table>

- Pretty accurately for big networks for 3 - 4 hops (finite network size in contrast to large \( N \) in our model).
- Better accuracy for larger networks (small networks, small network diameter).
Theory does not always match reality

Test our network growth model with real ISP topologies from Rocketfuel

Accuracy analysis on real ISP topologies

Fast growth till 4-5 hops! Then drops due to limited network size and small diameter.

Summary: our model and analysis on real ISP topologies show that neighborhood growth is fast at the first hops.
back to content discovery
Q: When to flood?

A: Flood when $U > 0$

\[ U = (1 - q^n) - n \cdot c \]
Q: When to flood?

A: Flood when $U > 0$

$$U = (1 - q^n) - n \cdot c$$

We have a model for $n$ as a function of $r$ and network topology.
Q: When to flood?

A: Flood when $U > 0$

$$U = (1 - q^n) - n \cdot c$$

How about $q = 1 - p$?

We have a model for $n$ as a function of $r$ and network topology.
We consider two cases of a given content set.

- The availability is given as *a priori knowledge*.
- The availability is *unknown*, so we apply Bayesian inference* to estimate.

What is the availability given that I receive this discovery message with hop count $h$, i.e., $h$ nodes do not have the content?

How to calculate the optimal scope?

\[ U = (1 - q^n) - n \cdot c \]
How to calculate the optimal scope?

A good flooding strategy:

- Node is aware of its neighborhood with an accurate topological inference
- Node is aware of the content’s availability with an accurate inference on user requests
Optimal scope for each node or for the whole network?

- **Static Flooding \((r)\)**
  - Same optimal scope for all nodes
  - A priori knowledge on availability
  - Scope is optimised over the whole network using average # of 1-hop and 2-hop neighbours of the network

- **Dynamic Flooding \((r_i\ for\ node\ i)\)**
  - Scope calculated for each node based on that node’s neighbors
  - With content availability, only flood on popular content
  - Without content availability, always flood 1-hop neighbours by default to infer availability
Do graph generative models matter?

Synthetic graphs with 10000 nodes and 60000 edges

- Optimal scope 1-3 hops
- For higher content availability, scope can be larger as there is a high chance that the content will be in the network
- Nodes in a scale-free network have more diverse optimal scope setting
A closer look to the optimal scope

- Optimal scope calculated by static flooding

- **Scale free:** more heterogeneity $\rightarrow$ divergence from network wide optimal scope.
A closer look to the optimal scope

- Scale free: more heterogeneity $\rightarrow$ divergence from network wide optimal scope.
- Negative correlation $\rightarrow$ nodes at the network core have smaller optimal scope.
Is dynamic flooding always effective?

Improvement = (Utility of dynamic flooding - utility of static flooding) / utility of static flooding

- Very little utility improvement (10% of the nodes) in ER graph because network-wide optimal scope matches node-based optimal scope.
- ER: homogenous structure
Is dynamic flooding always effective?

Correlation between growth and the utility improvement on random network (10% of the nodes) is close to zero, indicating that the significance of improvement is irrelevant of a node’s growth rate and its position in the network.

Correlation on scale-free network (30% of the nodes) is much stronger, with Pearson correlation being 0.53.
How utilities are distributed in the network?

- **Strong negative correlation** between the utility and betweenness centrality.
- In the dense area, a node has a high betweenness centrality, it may include more neighbours than necessary (the optimum) even just for 1-hop neighbours.
- In the sparser area, growth rate is lower, so nodes have a better control over the neighborhood size by fine-tuning their scope leading to smaller cost and better utility.
Comparison of dynamic, static, and network-wide flooding

- Four realistic ISP networks and a community network.
- Each node has a 4GB cache with LRU algorithm.
- Content set is based on a Youtube video trace.
- Nodes of degree 1 are clients.
- 10 to 20 servers are randomly selected in a network.
- The collective request trace is generated using a Hawkes process*, which is controlled by both temporal and spatial locality factors.

A. Dabirmoghaddam, et al., “Understanding optimal caching and opportunistic caching at “the edge” of information-centric networks,” in ACM ICN’14, 2014
Byte hit rate, cost, and average hops

<table>
<thead>
<tr>
<th>AS</th>
<th>Byte hit rate</th>
<th>Cost</th>
<th>Avg. hops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nw</td>
<td>st</td>
<td>dy</td>
</tr>
<tr>
<td>1239</td>
<td>0.44</td>
<td>0.40</td>
<td>0.43</td>
</tr>
<tr>
<td>2914</td>
<td>0.49</td>
<td>0.42</td>
<td>0.47</td>
</tr>
<tr>
<td>3356</td>
<td>0.42</td>
<td>0.39</td>
<td>0.42</td>
</tr>
<tr>
<td>7018</td>
<td>0.47</td>
<td>0.41</td>
<td>0.45</td>
</tr>
<tr>
<td>Guifi</td>
<td>0.51</td>
<td>0.44</td>
<td>0.49</td>
</tr>
</tbody>
</table>

- Network-wide flooding always achieves the best byte hit rate, the improvement is marginal at the price of 2 to 3 times increase cost.
- Dynamic flooding consistently outperforms static one.
- Most content are discovered within 2 hops. Network-wide flooding has the highest values due to its inherent aggressiveness.

nw: network-wide flooding
st: static flooding
dy: dynamic flooding.
What are the limitations of this model?

- **Clustering coefficient** is not considered in the network model, so it overestimates the neighbourhood growth.

- Cost of retrieving a content is not considered.

- **Sublinear** growth in gain and **exponential** growth in cost, this needs to be verified and justified in reality.

- Only evaluated with LRU, we do not know whether other in-network caching algorithms will change our story or not.
Key take-aways

- The neighbourhood (of a medium scope) can be very well approximated with a node’s 2-hop information.
- Accurate estimation for 3-4 hops on the network growth
- Analysis on ISP topologies shows the fast network growth
- The choice on static or dynamic flooding depends on the network structure. I.e., random or scale-free networks.
- When to flood: If expected utility is positive, higher content availability
- Where to flood: Better at the network edge
Thanks

The slides and the paper are available at http://www.hiit.fi/u/bayhan/
References

- Dima Mansour, Information-Centric Networking, 17.11.2015