A Plan to Save RSS :

Sharing Web micronews with peer-to-peer event notification

Dan Sandler
Alan Mislove, Ansley Post, Peter Druschel
Rice University

IRIS Student Workshop – Cambridge, MA
November 7, 2004

Surfing the Web

- Used to be fun, but now it's hard work!
  - Not enough just to reload a few bookmarks
  - Thousands of niche websites
  - Personal weblogs
- Trend: "Micronews"
  - Frequent, irregular updates

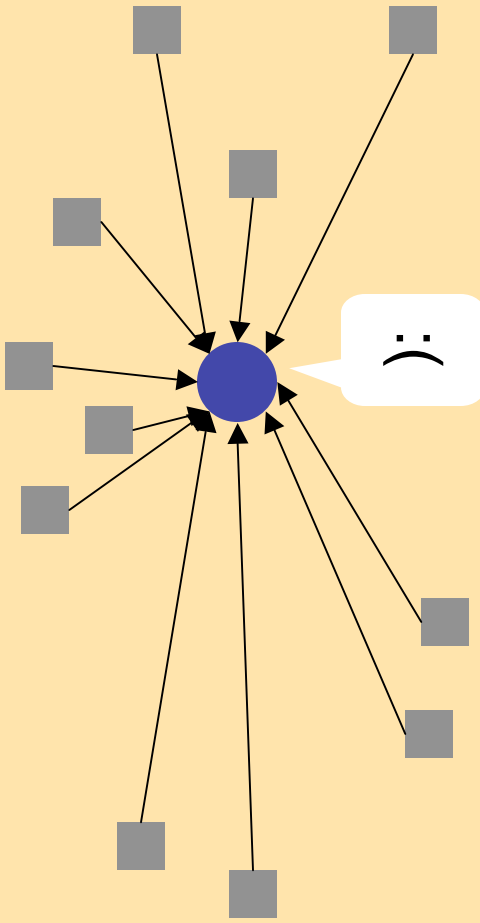
RSS News Feeds

- XML summary of headlines, links, stories
- 2004: Explosion in popularity of RSS feeds
  - 800,000+ feeds (Feedster)
  - Major news outlets
    - (NYT, BBC, MSNBC)
  - Minor news outlets
    - (Slashdot, Wired)
  - Weblogs

“Breaking News”

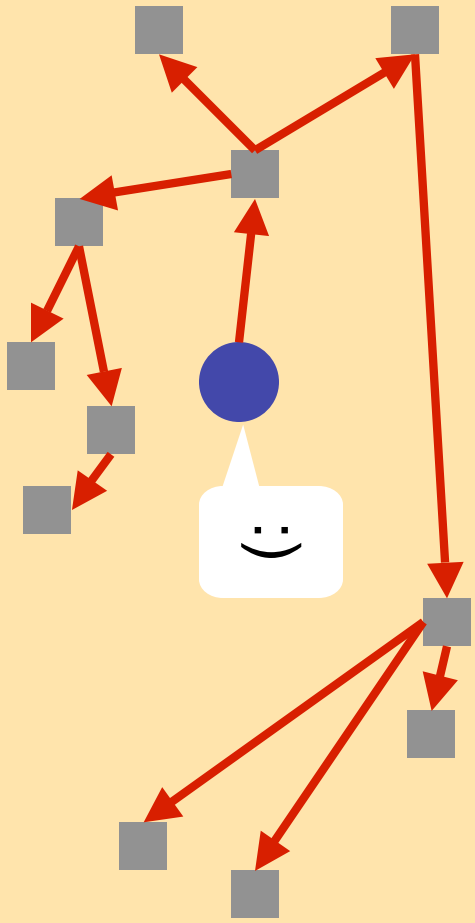
- Applications must p*oll* RSS feeds
  - Frequently (for freshness)
  - Polling one resource does not scale
  - **Bandwidth problem** for publishers
  - Server’s perspective: synchronized clients look like a DDoS attack!

- Some attempts to control the problem
  - Websites scaling back RSS service (*e.g.* MSDN, Slashdot)
  - The user base is only increasing…



“Fixing News”

- Perfect fit for a **p2p event notification** service
  - Many participants
  - Continuously running client applications
  - Don’t need extremely low latency
    - Still much better than polling: 10 sec. ≪ 30 min.
- Goal: cooperative p2p dissemination of micronews *as it happens*

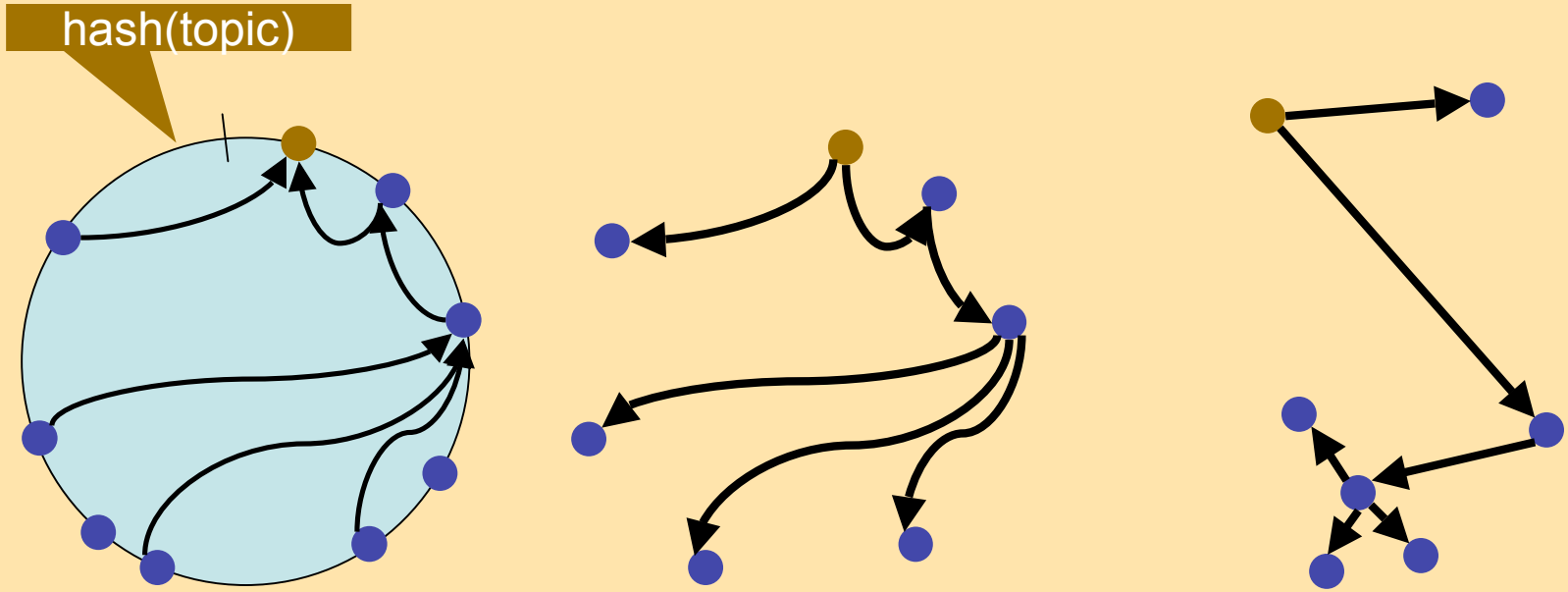


FeedTree

- Propose **FeedTree**, a p2p micronews distribution system built on Scribe
  - Use Scribe's event notification to disseminate RSS news, ASAP, without polling
- Scribe: Batteries included
  - Decentralized membership
  - Cheap maintenance of multicast trees
  - Efficient event multicast

Scribe

- Uses Pastry routing to create trees
- Tree = union(all routes to group root)



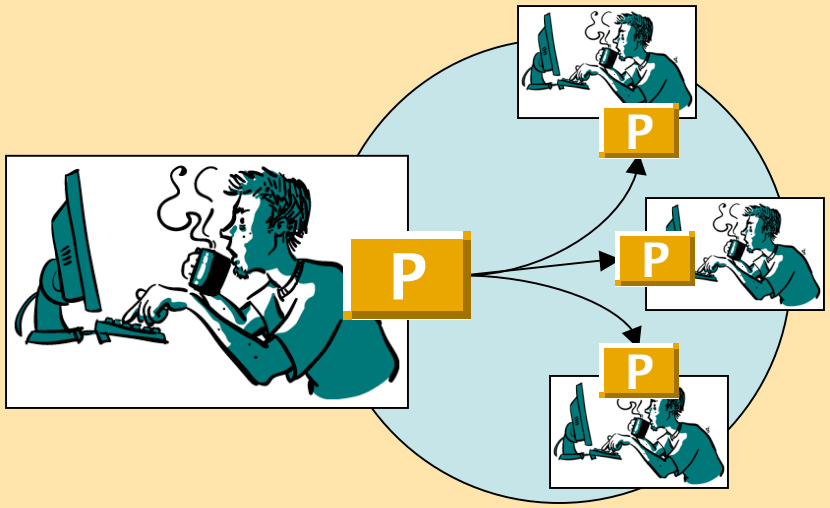
FeedTree Architecture

- Each RSS feed gets a Scribe group
  - Scribe topic = (prefix of) URL of RSS feed
- Feed author sends micronews to that group
  - Scribe event format: signed RSS document containing only new items
  - Send new items immediately
- RSS client software subscribes to this group
- Conventional RSS feed still useful
  - Recover recent lost items

Early Adoption Scenario

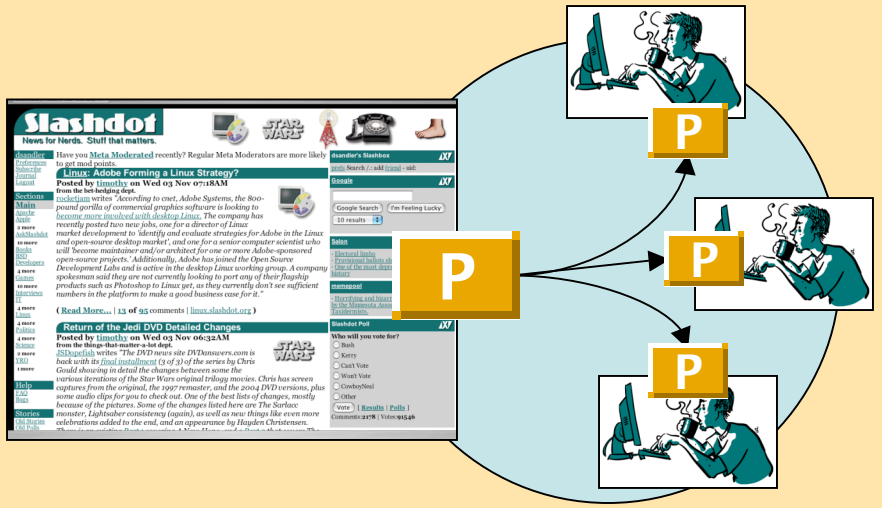
- Early adopters aim existing RSS clients at **local FeedTree proxies**
- Proxies look for relevant Scribe groups to join
  - If no group exists, poll the RSS as usual, and start a group, multicasting each new item

POLL
(yuck)



Publisher Adoption

- Desirable for stronger content authentication
- Publishers must join FeedTree
  - **Incremental:** Add a server-side proxy which polls existing RSS (very frequently)
  - **Ideal:** Add FeedTree to content-management tools, blog s/w



Incentives for Adoption

- Cheaper, richer offerings for publishers
  - Dramatically lower bandwidth demands
  - Offer richer or differentiated content
    - As compared with conventional RSS
- Better RSS service to end users
  - **Immediate delivery**
  - Richer content

Summary

- RSS
  - Users love it
  - Death of a thousand cuts for publishers
- p2p content distribution
  - A component of the “right” solution
  - Benefits for publishers and users
- FeedTree—a new application of structured overlays!
  - A path (with small steps) to a polling-free system
- We will build it
  - First steps: local FeedTree proxies
  - Bootstrap the overlay with live news
    - Immediate results for end users

Bonus Round!

Details: RSS Format

- Add new (optional) subscription info
  - Feed is available through FeedTree
  - Public key of author (for verifying items)

```
<rss version="2.0">
  <channel>
   <title>FooBlog</title>
   <link>http://foo.com/</link>
   <scribe:topic>http://foo.com/rss</scribe:topic>
   <scribe:pubkey> . . . </scribe:pubkey>
   . . .
  </channel>
</rss>
```

Details: **Publishers**

- Publishing software
  - Web servers, content management systems, server-side weblog tools
    - Anything that currently generates RSS
  - Join the overlay, becoming a long-lived Scribe node
  - Multicast new RSS items immediately
- Incremental solution: FeedTree "republishing" proxy on the server
  - Poll legacy RSS feeds and multicast new items

Details: **Clients**

- RSS client applications
  - Join the overlay
    - Distribute the RSS forwarding load
  - Examine conventional RSS feeds for FeedTree information
  - Subscribe to FeedTree feeds
    - Stop polling those feeds!
  - When updates are received, notify the user immediately
- Incremental solution: Local RSS client proxy
  - Existing apps can poll very frequently to get fresh news

Then What?

- Use p2p storage (DHT) to create an **RSS archive**
  - Offline clients will be able to “catch up
  - Recovery of (very old) missed items
- **Anonymous RSS feeds**
  - Use anonymizing p2p routing schemes
- Distribute more sophisticated RSS content
  - Media
  - Software

Some Not Entirely Fictional Data

- *Subscriber numbers from Bloglines.com*
- Slashdot: 17,700 subscribers
  - 2 KB headline-only feed * 30-min polling period
  - **1.7 GB daily**
- New York Times: 24,000 subscribers (all feeds)
  - 3 KB feeds -> **3.5GB/day**
- Boing Boing: 11,500 subscribers
  - Rich feeds: 40 KB
  - **22 GB/day**
- BBC News: 18,000 subscribers
  - "Updated every minute of every day"
  - Let's not even speculate about THAT polling schedule