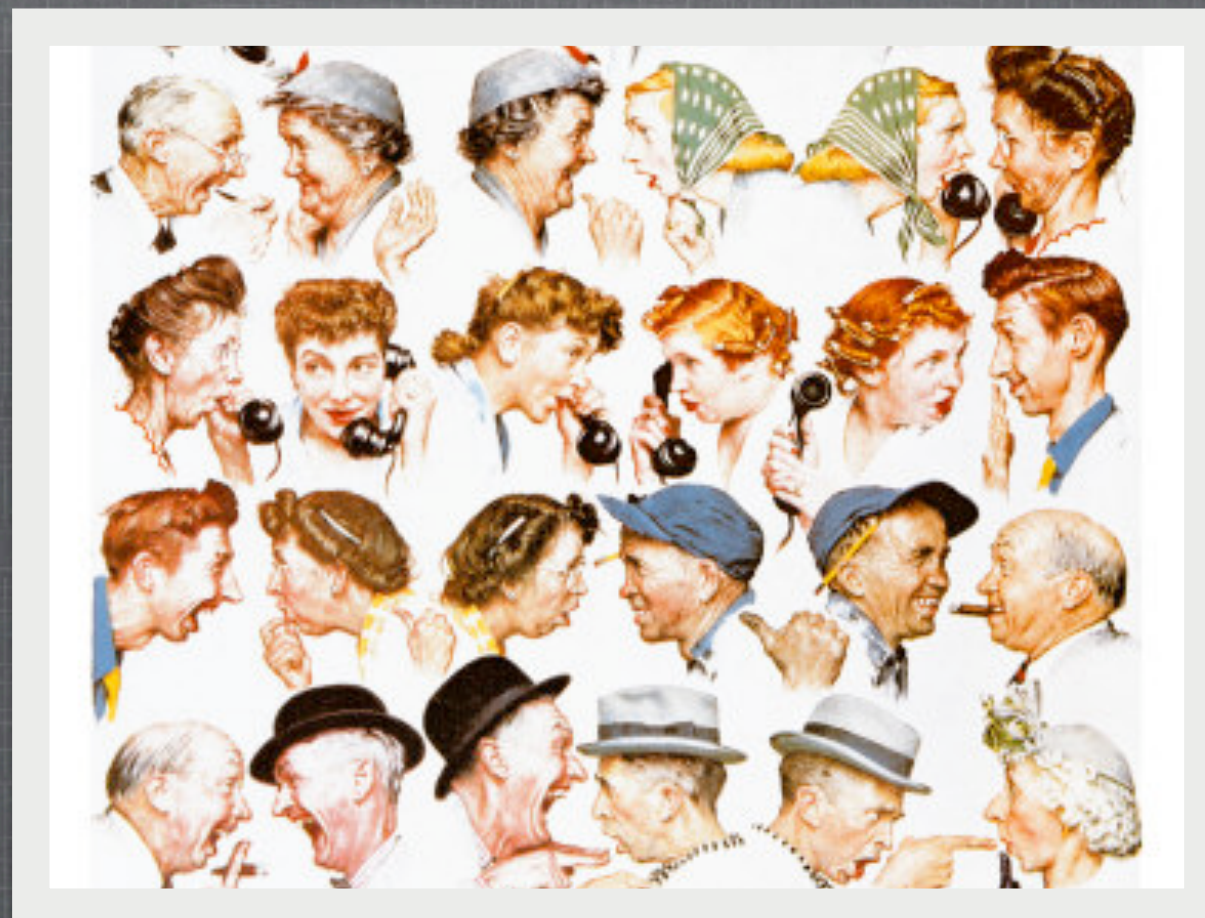


DISCOVERY IS STILL HARD GOSSIP IS STILL EASY

draft-hardie-p2psip-p2p-pointers-00.txt



P2P POINTERS

- Discovering P2P networks using only P2P tools is hard.
- This draft puts forward 3 mechanisms that provide ways to share information about P2P networks using existing protocol contexts (web, email, other textual media).
 - They can be combined
 - They may end up being useful in different contexts

OVERLAY

- Allows you to get an object that tells you about the p2p network using existing URI schemes / attachments.
- XML-based (using RelaxNG) with a single “availableOverlayDetails” container
- Similar in some ways to the reload draft’s enrollments server XML format; with some work, they could converge.
- <http://introducer.example.net/>

OVERLAY POINTER URI SCHEME

- Provisional registration of a scheme “overlay” for pointers to overlay networks’ enrollement servers or introducers
 - Notes the DHT types and available services in parameters
 - `overlay://
enrollment.example.org/;otype=Pastry;service=mass-
storage`
 - otypes are IANA registered; services are not
 - currently looks for AAAA or A records

OVERLAY NODE POINTER URI

- This points to a particular node in an overlay or a resource at a particular node.
- The authority section of this URI scheme uses the “registered name” construct and the node identifiers of a specific overlay
 - `overlay-node: / / 22301203 / ?
resource=example.iso`
 - obviously, this is not useful without an established overlay context

EXAMPLE COMBINATIONS

- overlay-node: / / 22301203 / ;context="overlay: / / enrollment.example.org / ;otype=pastry" \ ? resource=example.iso
- overlay-node: / / 22301203 / ;context="http: / / inducer.example.net / example.odd" ? resource=service-instance
- You can use either pointer mechanisms to set context
- “service instances” relate to the services overlay parameters

WHAT NEXT?

- Do folks want to progress work on pointers in the working group?
 - How much specificity to reload, if so?
 - Should the next draft have all three mechanisms, if so?
- Are there other textual pointer mechanisms that need to be included?