# P2PSIP Peer Protocol Design Questions

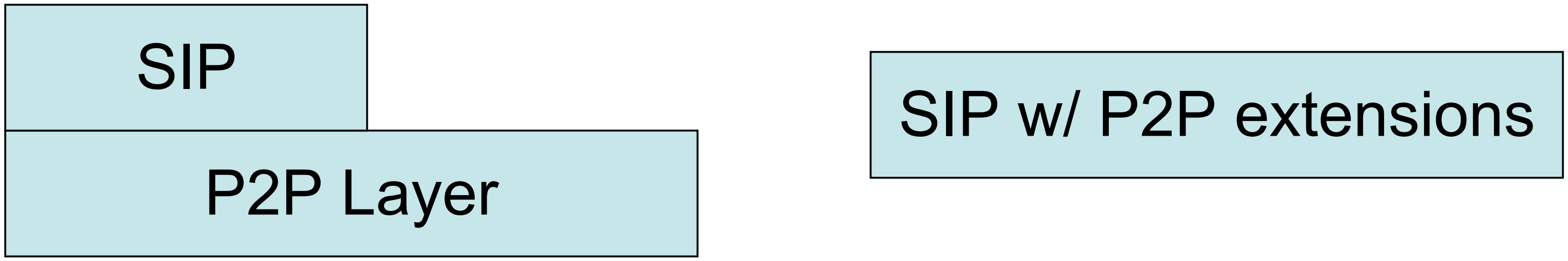
Presenter: Philip Matthews

(based on input from the authors of the various proposals)

# Intro

- This presentation: List of some key design questions WG needs to decide upon, and some of the choices and implications of those choices.
- Think about these questions as you listen to the individual presentations.
- Except perhaps for the first question (next slide), answers are still being explored and there is no consensus yet on the right answers.

# Should the P2P layer be distinct from the SIP layer?



SIP

P2P Layer

SIP w/ P2P extensions

Various things pushing for distinct P2P layer:

- Various concerns about the nature of the P2P extensions to SIP.
- Sense that a distinct layer is architecturally correct.

Do we have WG consensus that P2P layer should be distinct??

# Should P2P layer support apps other than SIP?

**NO**:

- Focus on a narrow, well-understood problem.
- This way, networks admins won't want to block P2PSIP.

**YES**:

- Strong push from some to do this.
- If P2PSIP is successful, it WILL happen.

**Note**: Charter says only "cannot work on issues not relevant to P2P-based SIP".

# Support multiple DHTs?

Options:

a) Support just one. (Which one?)

b) Support a number of similar DHTs.

c) Support most/all DHTs.

d) ???

If multiple DHTs, then:

- Is there a mandatory-to-implement DHT, or do we just specify one to use for testing?
- If MTI is X, but overlay currently running Y, what happens when peer that supports only X wants to join?

# DD Record Types

- How is format and meaning of a DD record type specified: in prose, in a formal notation, or … ?
- How are new record types added?
  - Software upgrade of DD code?
  - Agreed to before overlay forms?
  - Added dynamically when overlay is running?
- Who needs to know about and understand a type? All peers, or just peers doing put/get on that type?

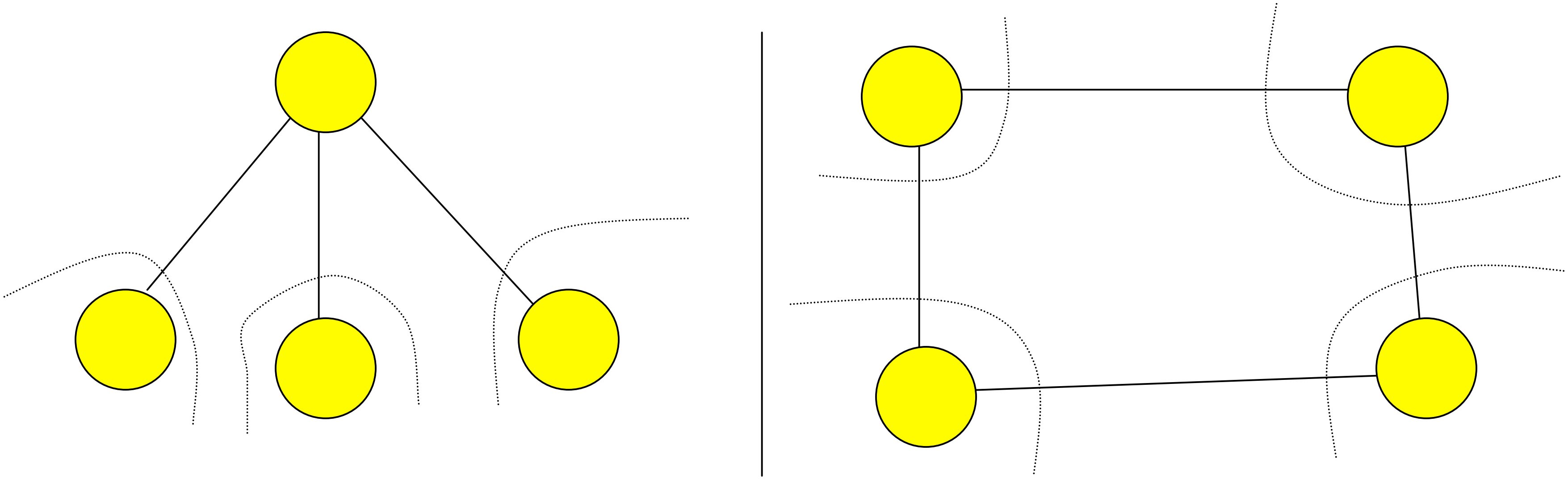
# DD: Soft state or hard state?

- Soft-state = Records must be refreshed periodically, otherwise they time out.
  - How long is timeout?
  - What if user/peer leaves overlay? Should their records be remembered? If so, refreshed how?
- Hard-state = Records remembered until explicitly or implicitly deleted.
- In general, soft-state protocols more robust and easier to get right, but more chatty. Use hard-state when chatty is a problem.
- Classic example. OSPF = DD proto w/ soft-state. Scales to about 600 peers. BGP = DD proto w/ hard state. Supports > 100,000 records, and developed because soft-state predecessor didn’t scale.

# Security

- Admission security
  - Who is allowed to join the overlay? How is this enforced?
- Database security
  - Who is allowed to create or modify a record in the DD?
  - How do we prevent a malicious peer from ignoring a record update, or replying with the wrong record contents when asked?
- Message security
  - Preventing inspection or modification of a message in transit.

# High-level NAT Traversal Approach?



- Do we assume that there will always be at least a few peers with public IP addresses?

# What transport should the Peer Protocol use?

- TCP - Provides reliability, segmentation
- UDP - No reliability or segmentation, but traverses NATs better
- SCTP - Like TCP, but message-oriented
- Add TLS/DTLS for security?

# Future-proofing

- Some things that can change:
  - New fields in messages
  - New DD types
  - New DHT algorithm
  - New admission procedure
- How can we mix old and new versions of the protocol in one overlay?

# Diagnostics

- Distributed systems are very difficult to troubleshoot. What can we do in the protocol to make this easier to check that:
  - DD is “correct”?
  - Overlay is properly formed?
  - Messages are getting delivered?
  - Etc.