I3: Internet induction infrastructure.

Point to point abstraction

L \rightarrow not suitable to support other communication primitives.
- Multicast
- Anycast
- Mobility
- content centrality?

Why is point to point assumption one sender, single receiver at fixed well known locations?

How do current approaches solve it? Indirection

If Multicast group
Home agent / foreign agent in mobility.

Create induction as the primitive
Build an efficient induction layer on top of I3.

Build using an overlay \rightarrow no need to change I3
\rightarrow incrementally deployable.

I3 basics: Packets have ID
- to receive ID, receiver B maintains a trigger (ID,E)
Service model: API: send packet (P)
insert trigger (t);
remove trigger (t) \rightarrow \text{optional}

Best effort delivery, just like IP.

Triggers periodically refreshed by end-hosts.
ID \rightarrow 256 bits.

How to support various services.

Mobility: update with new mapping.

Multicast: receivers insert triggers with the same identifier.
- can dynamically switch unicast to multicast.

Anycast: use longest prefix match instead of exact match.

\[
\begin{align*}
P(\text{id}\text{,} \text{data}) \\
\text{data, r} \\
id, r \\
\end{align*}
\]
Service composition, sender initiated.
- Use a stack of ids to encode a sequence of operations to be performed.

Sender

data, id_s, id

Transcoder (T)

data, id

Receiver

Receiver initiated:

Firewall

Sender

(data, id)

Receiver (R)

Implementation / practice

- Collection of infrastructure nodes
- IP space is partitioned across nodes
  - A node is responsible for a region of IP space.

Use chord to route triggers and packets to nodes responsible for their IDs,
- O(log n) hops.
Example:

In space \([0, 63]\)

partitioned across 5 nodes

Each host knows one is node

Optimization, path length.

Sender receives cache is node mapping a specific IR.

Triangle routing:
- Use well-known trigger for end initial rendezvous
- Exchange a pair of private triggers well-located
- Use private triggers to send data traffic
Security challenges

- Eavesdropping
- Loop
- Confidence
- Dead end

Trigger byack $\rightarrow$ (id, x) (x, s)

DoS attacks $\rightarrow$ challenges to verify that trigger was inserted by end-host claiming to want

$\rightarrow$ Resource allocation

$\rightarrow$ loop detection