Thoughts on the Current IPN Architecture Proposal

Kevin Fall

Intel Research – Berkeley, CA

kfall@intel.com

Presentation at JPL, October, 2001

Outline

- IPN goal and related Terrestrial work
- Bundling service and message format
- IPN Nodes and Bundle Routing
- Compatibility with existing Internet
- Security

IPN Vision

- Stated Goal:

  "*provide Internet-like services across interplanetary distances in support of deep space exploration*"

- Question: What does *Internet-like* mean?

How to be "Internet-like"

- "Classic" Internet characteristics:
  - Best-effort delivery of abstract datagram over stateless infrastructure
  - Unique, global, hw-independent topological addressing and dynamic routing
  - End-to-end reliability and flow control
  - Scalable, global naming associated with admin. or geo. domains and decoupled from addressing

Not so "Internet-like"

- Today's Internet characteristics:
  - Re-used addresses and lack of global connectivity
  - Stateful "gateways" above layer 3
  - Alternative, "tag-based" routing (MPLS)
  - Active data stream re-writing up to layer 7
  - Complex routing and filtering policies
  - Curious multi-layer encapsulations/tunnels

Future "Internet" [?]

- Datagram forwarding gives way to content forwarding friendly to NAT-style devices, multicasting/anycasting and data caching
- NAT-friendly IP-style routing:
  - IPNL (Tahoe Networks)
  - TRIAD (Stanford)
- Content routing and discovery:
  - FreeNet, Gnutella, Tapestry (UCB), CHORD (MIT), CAN (UCB), etc

IPN != Internet

- Internet service expectation:
  - Remote login, file transfer, e-mail, web access
  - RTTs consistent with interactivity (< 10s, typ << 10s)
  - E2E Authentication on as-needed basis
  - Undetermined QoS
- IPN service expectation
  - Remote messaging, file transfer, e-mail
  - RTTs beyond reasonable human wait-times
  - Delayed “Return receipts”
  - Authentication always
  - Some QoS [probably CoS] always

Architectural Context

- Today’s Internet interconnects distinct link layers by way of a common IP layer
  - Single packet abstraction, adaptation for datagram size and addresses via ARP and IP fragmentation
- IPN will interconnect IPN regions by way of common messaging layer (“Bundles”)
  - Single naming and delivery abstraction
  - Transport protocols terminate at region boundaries
  - “Gateways” span regions
  - Message switching a special requirement for IPN

Bundles

- Bundles
  - Arbitrarily long messages delivered end-to-end between IPN capable nodes over distinct (but possibly identical) transport layers
  - May have associated delivery characteristics. Thus, delivery is always at bundle granularity.
  - Bundles may be fragmentary and require reassembly to be complete.

IPN Nodes (currently)

- Agent
  - Build and consume bundles
- Relay
  - Agents, plus forwards bundles within or between regions
- Gateway
  - Relays, plus do routing between regions
- *Custody Transfer*
  - Orthogonal and optional vs. node type

IPN Nodes (Alternative)

- Non Persistent Node [NP node]
  - no stable storage
  - Build/consume bundles, forwards bundles, participates in time synchronization
  - May forward or cache bundle or bundle parts
  - Never assumes custody
- Persistent Node [P node]
  - stable storage
  - Does everything an NP node does
  - Always accepts custody of a bundle on success
  - Notifies prior custodian of custody transfer
- Exception: SRC/DST accept custody always

Routing, Forwarding and Custody Transfer

- "Classic" Concepts (Internet):
  - *Routing*: selecting best next hop for every possible destination
  - *Forwarding*: sending packet to best next hop
    - Typically, "on demand" [statistical multiplexing]
    - Forwarders know *a-priori* next hop for every destination
- IPN Concepts:
  - *Routing*: selecting best next IPN hop for destination
  - *Forwarding*: sending a bundle p2p on demand
  - *Custody Transfer*: reliable intra-IPN delivery (with storage)

Forwarding

- Applicable to NP nodes
- Knows next-hop name for each destination name
  - Decide if a-priori or can learn on-demand
- Sends as soon as possible
- Transport layer will assure p2p reliability
- Does not verify bundle integrity, only access control check and CoS

Custody Transfer

- Applicable to P nodes (incl SRC and DST)
- Node dispatchers operate using link schedule:
  - A table of (T, L, Op, Args) tuples
    - Op: SendMsg *or* SteerLink
    - Args: NC/NH *or* Direction
  - At time T, send message M over link L to IPN Next-Hop H with next custodian NC
- Expect custody transfer ACK from NC

Info at Bundle Layer

- Currently, this is proposed to be:
  - `BundleID, Dest, Source, Auth Info, Source APP Handle, Dest APP Handle, Data Size, Handling Instructions, Data Descriptor, TTL, Source Route, Bundle Custodian, User Data`
- `Auth Info, Handling Instructions, Data Descriptor` are not really defined yet

Current IPN Naming Scheme

- Entity names are of the form:

  { *admin-part, routing-part* }
- *routing-part* is topologically significant
- *admin-part* is opaque outside the region specified by the routing part
- Names are carried E2E in bundles

Alternative Structure

- `Destination`, `Reply-To`, `Last Custodian` using URL-like syntax
- `AuthInfo` is crypto material containing delivery CoS, sender, and bundle digest
- Source timestamp replaces `Bundle ID`
- Data offset and length for bundle frag.
- Optional delivery info (e.g. delivery path)
  - Needs further thought

Small Comment on DNS

- DNS names are of the hierarchical form

$$n_1.n_2....n_k$$

- Existing naming is administrative and/or geographical, not topological. (It is a poor "source route").
- But, DNS names do not necessarily need to be used with the existing distributed DNS database structure (consider early transition to DNS names)

Small Comment on URLs

- URL syntax:

$$p://n_1.n_2...n_k/a$$

  - *p* – app access protocol, implies transport protocol and default port ID (enumerated type)
  - *n* – globally unique, hierarchical name, (arbitrary length)
  - *a* – locally significant identifier (unstructured)
- Two name spaces: one global, one local

URL-like IPN Entity Ids

- URL-like syntax: $p://n_1.n_2...n_k/a$
- Can easily construct an { *admin-name, routing-name*} tuple from this structure:
- Example:
  - { www.ipnsig.org, earth.sol} becomes
  - http://www.ipnsig.org//mars.sol/ or maybe
  - http://34-8-45.118-7-56.nw.latlong.earth.sol/

Postage Stamp Proposal

- Each bundle contains a cryptographically-signed "postage stamp"
  - Similar to Kerberos tickets
- Provides authorization to use the IPN at a particular class of service for a particular message
- Postage stamps are verified at each P node
  - NP nodes may not store any complete bundle
  - Endpoint P nodes are special (later)

USPS Options

Option Name	Mailing Receipt	Delivery Record	Air Delivery	Recipient Pays	Moves Money	Delivery Confirm	Return Receipt	Careful Handling	Insurance	Restricted Delivery	Signature Confirm
Cert. Of Mailing-RM	Y		(w/PAL)					(w/SH)			
ParcelAirLift (PAL)			Y								
Special Handling SH			(w/PAL)	(w/COD)		(w/DC)	(w/RR)	Y	(w/IM)		(w/SC)
Certified Mail CM	Y	Y					(w/RR)			(w/RD)	
COD	(w/RM)	Y		Y		(w/DC)	(w/RR)	(w/SH)	(w/RM)	(w/RD)	(w/SC)
Delivery Confirm DC				(w/COD)		Y	(w/RM)	(w/SH)	(w/IM or RM)		
Insured Mail IM			(w/PAL)			(w/DC)		(w/SH)	Y		(w/SC)
Money Order					Y						
Return Receipt RR	Y	Y	(w/PAL)			(w/DC)	Y	(w/SH)		(w/RD)	(w/SC)
Registered Mail RM	Y	Y		(w/COD)		(w/DC)	(w/RR)		Y	(w/RD)	(w/SC)
Restricted Delivery RD			(w/PAL)			(w/DC)	(w/RR)	(w/SH)		Y	(w/SC)
Sig. Confirm		Y				Y					Y

USPS Mail Services

- First Class, Priority/Express, Parcel Post, Printed Matter, Media Mail
  - 1st: Sealed against inspection, max 13 oz weight
  - Priority/express is faster delivery
  - Parcel post/printed/media is cheaper/bulk delivery
- Relevant Special Services: Certificate of Mailing, Delivery Record, Delivery Confirmation (opt signature), Insured, Restricted Delivery

IPN CoS Proposal

- Proposal:
  - Types: Expedited, Regular, Bulk
  - Options: send notification, keep delivery record, inform on delivery
- Stamps encode CoS, are not forgeable, and are obtained by sender from trusted service
- IPN routers can verify CoS in stamp using IPN "forwarding service" key

Security Proposal

- Assumptions:
  - Require: access control/DOS prevention
  - Nice to have: data secrecy and traffic analysis resistance
- Approach:
  - Capabilities created on per-bundle basis
  - Used for authentication and integrity check

Authentication Model

- Similar to Kerberos system. Initially:
  - Send sends [sender name, lifetime] to KDC
  - KDC returns $\{T_{tgs}, K_{tgs\text{-}sess}\}K_{user}$
  - T_{tgs} is $\{uinfo, K_{tgs\text{-}sess}\}K_{tgs}$
- User thus obtains TGT (T_{tgs}) and $K_{tgs\text{-}sess}$
- User obtains network service tickets (stamps) using TGT
- IPN P and NP nodes know the IPN service key; P nodes check message integrity, NP only checks authentication info

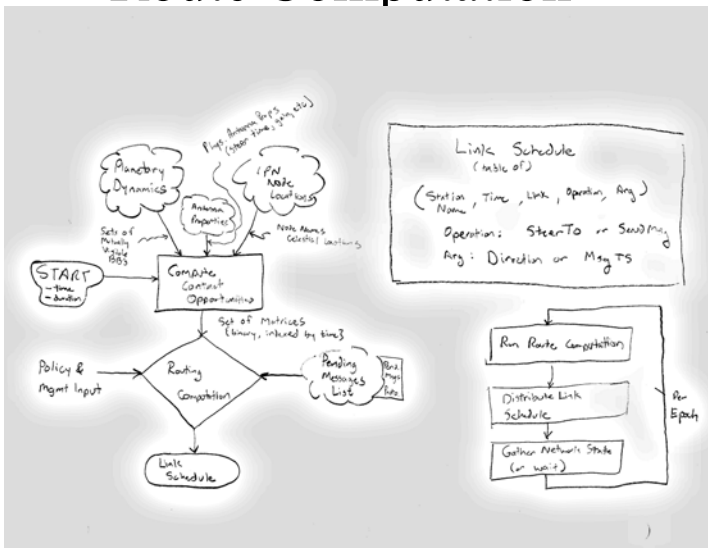
Using Stamps (Detail)

- Stamp is essentially a Kerberos service ticket for the "IPN Forwarding Service"
- Stamping a bundle:
  - First, sender requests stamp from TGS:
  - { TGT, sender, bundle-hash, CoS, send TS }$K_{tgs\text{-}sess}$
  - TGS provides the stamp for sender to use:
  - {{sender, TS, cos, hash, K_{sess}}K_{ipn}, K_{sess}}$K_{tgs\text{-}sess}$
- Sender then sends the following:
  - {sender, TS, cos, msg hash, K_{sess}}K_{ipn}, Message

End to End Delivery (A to B)

- Preparing to send:
  - A determines IPN next hop H, next custodian C, and sending time from IPN route server [or itself]
  - Using send time, A obtains IPN service ticket
  - A arranges for receipt of ACK from C
- A sends to IPN next hop H:
  - If H is a P node, H will return a custody transfer notification and A can free its resources
  - If H is an NP node, H will in turn forward to next hop

Route Computation



Summary

- Only “somewhat Internet-like” service expectation
- URL-like naming
- Bundling data re-structuring
- Authentication model based on Kerberos
- Alternative node types and routing function
- Security

Some Questions

- What exactly is the nature of the time synchronization requirement?
- What sort of policies need be expressable?
- Is data secrecy support fundamental?
- Is there a maximum (min?) bundle size?
- Where is a delivery log kept?
- Re-visit the assumptions about proxies?
- When/how does bundle layer re-try?
- How to re-sequence pending msgs on LS change?
- Do IPN GW's *need* more than 1 name?

Other Protocols Required

- Pending messages, IPN Node List with locations, Link Schedule Distribution, Custody Transfer indication, Error Indications, user/KDC exchange, Policy/Mgmt distribution

… End…