Heterogeneous Addressing in DTN

Rick Taylor

John Dowdell

Airbus Defence and Space

Based on discussion with Scott Burleigh

What’s the problem?

- DTNs can be addressed by a URI scheme such as `dtn:<`*`address`*`>` (RFC 4838), `ipn:`*`node-number.service-number`* (RFCs 6260, 7116) or IPv6 or some other format depending on how constrained the bearers are. (draft-templin-dtn-numid)
- However, format of <*address*> is not explicitly specified
  - “Use a naming syntax that supports a wide range of naming and addressing conventions to enhance interoperability.” from RFC 4838 section 2.
- Organisations can and do use different schemes of <address> (a namespace) to suit their use cases, e.g. numeric ID, textual ID, geolocation ID, …
- So … How to forward Bundles between different namespaces using different schemes of <*address*>?

Proposal

- Continue to allow organisations to use their own address schemes within their own borders.
- If a DTN Node does not understand the destination address scheme, forward the Bundle to a Resolver, located in its understandable namespace.
- If the Resolver understands the destination address scheme, forward the Bundle directly, via another Resolver, or via a Border Gateway for onward delivery.

How?

- A new Secondary Block, the Endpoint Resolution Block (ERB):
  - Has Previous and Next endpoint IDs, in the local namespace.
  - Added, removed, updated or stacked Hop-by-Hop.
  - Only used when endpoint IDs cannot be resolved by a Forwarding Agent.
  - Orthogonal to topological routing
- Remember: The Primary Block is immutable, so intermediate nodes cannot alter it.

Endpoint Resolution Block (ERB)

Flags	Previous EID	Next EID

- The Previous and Next Endpoint ID MUST be in the same namespace.
- Both Endpoint IDs are optional; if no resolution is required, no entry required.
- This is a work in progress!

Let’s break that down (1)

- DTN node in namespace Alice wishes to send a Bundle to an Endpoint in namespace Bob.
- Alice uses an naming scheme `<`*`scheme-a`*`>`
- Bob uses an naming scheme `<`*`scheme-b`*`>`
- An Endpoint in Bob could be addressed through a standardised traffic exchange mechanism as `dtn:<`*`scheme-b`*`>/<`*`name in scheme-b`*`>`

Let’s break that down (2)

- The sending node in Alice constructs a Bundle primary block containing source and destination EID, but the EIDs have different schemes and (say) cannot be resolved locally. The Bundle must be sent to a Resolver to process the EID further.
- The sending node and Resolver share the same address scheme (are in the same namespace). An Endpoint Resolution Block must be added to allow the Bundle to reach the Resolver.
- The block is forwarded towards the Resolver via whatever routing topology is in place.

Let’s break that down (3)

- As the Block is forwarded, each Node examines the Endpoint IDs in the Primary Block.
  - If it understands the address scheme in use by the Primary Block, remove any ERB, and forward as normal.
  - If there is an ERB, forward to the ERB destination EID.
  - Option to nest ERBs as a stack.

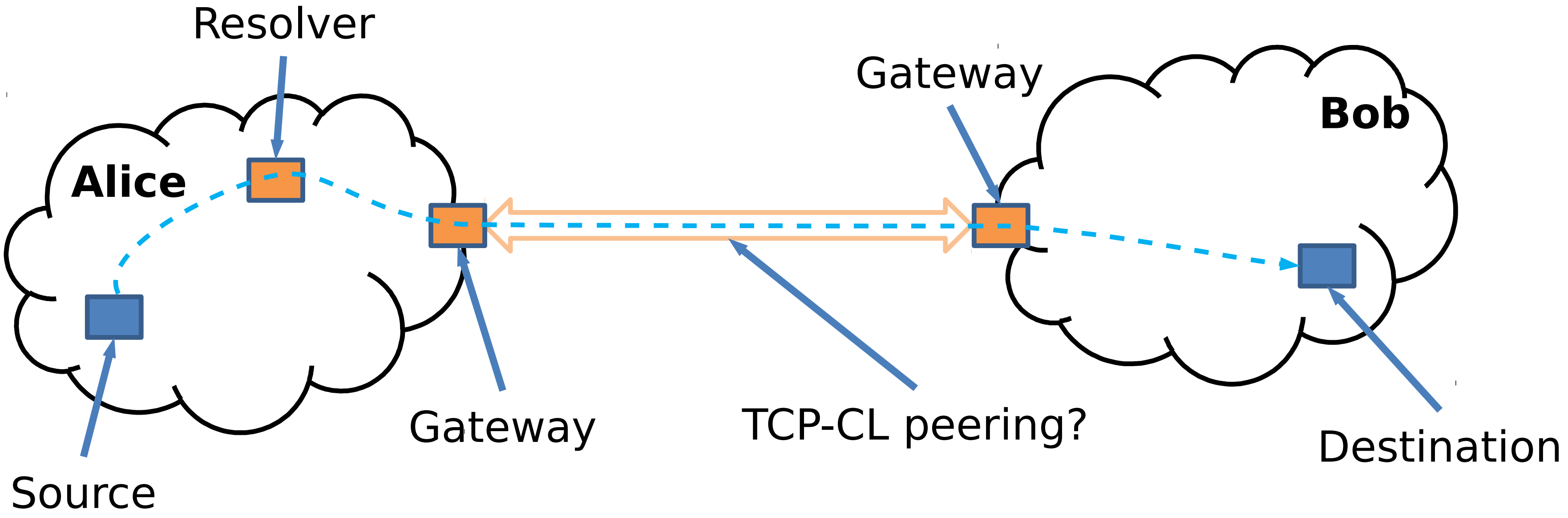
Let’s break that down (4)

- Eventually the Bundle arrives at the Resolver, via whatever routing is in effect.
- The Resolver removes the original ERB, and may need to replace with a second ERB to allow the Bundle to reach Alice’s border gateway.
- Alice’s border gateway swaps the second ERB for a third ERB to allow the Bundle to reach Bob’s border gateway.
- ERB-swapping or nesting continues until the Bundle reaches it’s intended destination.

Return path

- As Bundle traverse address scheme boundaries, the Previous Endpoint ID in the ERB can be updated to point to a suitable Resolver for the return path, for use by Status Reports or other signals.
- Nesting provides a backwards trail, if required.

Multi-namespace DTN



Next steps

- We believe this problem space is in scope of the Static Routing charter item.
- Should we write an I-D on ERBs?
- Yes, this is Locator/ID separation.

`dtn:` scheme

- Proposal

  `dtn:`*`<addr_scheme>`*`/`*`<any*>`*

- Exisiting implementations?

  `dtn://`*`<any*>`*

  - `//` reserved for ‘current local scheme’

- ipn?
  - Only needed when bridging ipn namespace to dtn namespace
  - ipn scheme is top-level already, so it needs a mapping:

    `ipn:`*`x.y`* → `dtn:ipn/`*`x.y`*

- Does `dtn:` need to be included in the bundle header, or is it implied?