


Extended Shim6 Design for ID/Locator Split and Traffic Engineering

draft-nordmark-shim6-esd-00.txt

Erik Nordmark
erik.nordmark@sun.com

A close-up photograph of a brick wall. The bricks are reddish-brown and arranged in a standard running bond pattern. A central mortar joint is visible, showing some white residue. In the lower right, there is a circular hole in the brickwork, surrounded by a white, star-shaped mark. The text "Is Shim6 a brick wall?" is overlaid in the center of the image.

Is Shim6 a brick wall?



**Or part of an interesting
architecture?**

What do we have?

- HBA and CGA to satisfy “first, do no harm” security
- Using locators as upper layer IDs
- Incrementally deployable
- Very basic, static TE support (priority and weight)

What is missing?

- HBA and CGA to satisfy “first, do no harm” security
- Using locators as upper layer IDs
- Incrementally deployable
- Very basic, static TE support (priority and weight)
- Complete ID-locator separation?
- Managing static TE per site?
- More dynamic TE control?
- Running over IPv4?

Why discuss this draft?

- To determine whether the proposed extensions could increase the applicability of shim6
 - and as a result reduce *the pressure* for IPv6 PI space
- To determine whether the WGs documents makes some unwise assumptions
- To stimulate others to think about the bigger picture
- As input to a rechartering discussion?

Complete ID-Locator Separation

- The protocol mechanisms don't assume the ULID is reachable
 - But it is a key optimization and necessary for deferred context establishment
- If we had a unreachable ULID format that fits in 128 bits, then the existing (socket) APIs can be reused
- If we also could lookup a ULID to find a set of locators, the application could use referrals and callbacks as today
- Many possible details – have examples

Unreachable ULID format

- Pick a relatively short prefix from the IPv6 address architecture
- We only know how to do scalable lookups from a hierarchically allocated “name”
 - Think of 10^{15} hosts using this scheme
- The result is something very similar to the registered unique-local addresses
 - That were abandoned as ULAs moved to RFC
 - Using HBA/CGA for the bottom 64 bits to handle security

Need for Lookup of ULID?

- If a ULID is reachable we can just send packets to it to find out the set of locators
- For unreachable, we need a way to get packets flowing by finding out the locators
- We could piggyback this on the DNS lookup of www.example.com. But that is insufficient since
 - The shim6 state might be lost and the ULPs just have the ULID
 - Application referrals, callbacks and long-lived application handles

Example: Using DNS

- Host looks for ID RRtype for www.example.com
 - Result is a 128 bit unroutable identifier
 - If no ID RR, looks for AAAA just like with shim6 today
- The ULID is mapped to locators using a reverse lookup in e.g., ip6.arpa
 - Can be creative and place AAAA records in the reverse tree
 - Can be even more creative and place SRV records in the reverse tree in order to express static priority and weight

Example: using DNS

- Syntax:

`_Service._Proto.Name SRV Priority Weight Port Target`

- Example:

`$ORIGIN 10.6c.36.fe.ff.6b.0b.02.bc.00.9a.00.78.56.34.12`
`.ip6.arpa.`

`; 3/4 on to fastpath locator, 1/4 on slowpath`

`_shim6._ip SRV 0 1 0 slowpath-www.example.com.`
`SRV 0 3 0 fastpath-www.example.com.`

`; fallback if the above are broken`

`_shim6._ip SRV 1 0 0 fallback-www.example.com.`

Walkthrough (1)

- Application calls `getaddrinfo()` which finds ID RR
 - returns this as the IPv6 address to the application
- Application calls `connect/sendto`
- TCP/UDP sends packet to IP
- Shim looks at packet and finds “unreachable ULID prefix”
 - Looks for shim6 context state
 - If none found, must setup context state before sending ULP packet (can't send to ULID)

Walkthrough (2)

- Shim6 uses lookup of ULID to find set of locators
 - Can take priority and weight into account if we have a SRV like capability
- Shim6 uses ULID-pair option in setup packets
 - No changes to protocol; sends I1 etc
- If one locator doesn't work for the context setup, then try other locators at the shim
- Once the context is established, again shim6 works unchanged
 - Might need to carry ULID-pair options on keepalive and probe messages etc.

Traffic Engineering

- Can already carry priority and weight (defined as for DNS SRV records) for the locators once the context is established
- But no way for the host to know what values to use for its locators
 - And manual configuration not likely to be sufficient
- Could easily define a DHCPv6 option to allow side-wide configuration
 - Might be useful
 - Can use with stateless address autoconfiguration

Static Traffic Engineering

- Need some TE input before the shim6 context is established
- Possible to use DNS SRV for the application protocol
 - E.g., `_http._tcp` type SRV records
 - Requires application changes in most cases
- If non-routable ULID, see previous slides
- Combined with the DHCPv6 option, this provides the site with the ability to specify static load spreading weights and primary/fallback locators

Dynamic Traffic Engineering?

- A possibility would be to add support for routers rewriting (source) locators on shim6 packets
 - Based on idea in Mike O'Dell's GSE draft
- Shim6 (more or less by accident) allows this on packets that have the Payload Extension header
- We could add this for shim6 control messages
 - I-D has example “Sent locator-pair” and “Received locator-pair” options so hosts can learn from routers
 - These are used on I1, R1*, I2*, R2 and perhaps other shim6 control messages

Locator rewriting by routers

- Routers would be free to rewrite every packet with
 - `nextheader == IPPROTO_SHIM6`
 - thus every ULP packet should have payload ext hdr
- If the ULID is CGA, then the hosts can learn new locators from the routers based on the rewriting
- There are unsolved issues around which locator to use e.g., when
 - probe mechanism says that A1 works and A2 fails
 - routers rewrite the source to be A2

IPv4 addresses as locators

- Observation
 - If the applications are using the 128-bit APIs
 - And the ULID is CGA
 - Then the locator can be anything (that is known to the local host and meaningful to the peer)
- Thus we could easily define a way to carry IPv4 addresses as locators
- Note: this does not “solve IPv4”, since IPv4 is likely to have NATs
 - And this WG shouldn't re-invent those square wheels

Conclusions

- Using non-routable ULIDs doesn't place any new requirements on the shim6 mechanism
 - Need to discuss DNS vs. some other lookup system
- If we want to allow router rewriting for IPPROTO_SHIM6, then it makes sense to define this sooner rather than later
 - Avoid “installed base” of shim6 that can't handle this
 - Details need to be worked out
- IPv4 locators (with CGA ULID) are easy
 - But leave NAT discussions outside the door (in some other WG please)