

Application Mobility with HIP

Teemu Koponen

1.12.2004

Motivation

- *“Process migration is the act of moving a process between hosts during its execution. It enhances load management, fault resilience, data access locality, and system management.”*
[Milojičić et al.]
- A reason for failure: the problem of maintaining (communication) transparency for applications.
- Process migration or checkpointing? Neither one, but **application mobility** instead!

How HIP fits in?

- Exploit HIP to solve the problem of communication transparency.
- Clearly, a relationship must exist between hardware mobility and application mobility, but:
 - *What kind of implications does HIP have to network communications of process being migrated and can HIP offer at least a partial remedy to a classical problem of process migration, namely, transparency of communications?*
 - *What implications the above has to HIP i.e., does the above [process migration] create new requirements for HIP?*

Migration transparency

- Roles: **source, destination & remote peer.**
- Approaches without “connection migration” result in **residual dependency** -- the problem of “an extra hop”.
- Key challenges in providing transparency for the transport layer [Su]: **state inconsistency, state conflicts, and state synchronization.**
- Related work:
 - Abstractions built on top of the transport layer
 - Extensions to the transport layer
 - Extensions to layers below the transport layer

Architectural issues (1)

- Migration simply does *not* equal to multi-homing; private key material could cause residual dependencies.
- Granularity of an end-point defines the unit of moving
- Identity **moving** is attractive, though, but not really an option.
- **Delegation** of HIP associations seems more attractive, but it's an unsupported concept in HIP.
- Socket API challenges with delegation; an address translation, or similar, mechanism needed?

Architectural issues (2)

- Finding a service after delegation is still easy (rendezvous + DNS), but how to validate and avoid a service identity? Transform the “delegation orders” semi-persistent.
- Global DNS view may contain multiple versions of a service identity.
- Delegation renders end-point identifiers stateful.

Design (1)

- A new **certification parameter** to allow usage of an end-point identifier not being a hash of **responder's** identity (public key).
- A rendezvous association update with a rendezvous server in similar manner: now initiator/target sends the new certification parameter.
- A new **delegate parameter** to replace HIP associations on-the-fly.
- A source asks a remote peer to discard an association with itself and communicate with a destination host instead.
- Security: the destination must prepare associations.

Design (2)

- Maintaining transparency for applications becomes management of mappings between endpoint identifiers and IPsec SPIs.
- Symmetric usage of certification parameters requires coordination in SPI selection; source and destination hosts share an identifier!
- Migration protocol transferring a process state runs on top of HIP, too.
- Implementation shall be a hybrid of process migration and checkpointing; instead of processes, so called **process domains** are migrated.

Discussion (1)

- Delegation has become a primitive in HIP architecture, as in DOA [Walfish et al.]
- Outsourcing of services to out-path middle boxes becomes easy (certification parameter).
- Nodes of a service farm behind a load-balancer don't need to share private key material anymore.

Discussion (2)

- The certification parameter re-enables usage of a class of high-availability protocols (VRRP, CARP); identity can be “stolen” again!
- The parameter targets the second core mapping of HIP: mapping of an identity to a host-identifier.
- Two layers of mobility: mobility due to a) physical movement, and b) administrative aspects.

Conclusion

- HIP integrates with application mobility **almost** seamlessly.
- Yes, HIP can offer a remedy for the problem of communications transparency.
- We introduced a concept of delegation to HIP; middle boxes are no more considered evil, but actually facilitated!
- End-points are more about applications than about hosts, from application mobility point of view.

Questions or comments?