

Comments on
**Secure Naming and Addressing Operations
for Store, Carry and Forward Networks**

John Day

Before I start let me say that this is a good start and you are covering ground that needs to be covered. From our work, you are largely on the right track. It is unfortunate that you are trying to use to what is a fundamentally flawed model as the context for the work. It compromises your ability to reach your goals and creates unnecessary complexity.

The second point I would make is that so-called DTN and SCF configurations are not networking. They are distributed applications, but they are not networking. One of the implications of Richard Watson's work is that to be interprocess communication or networking, it is necessary to be able to bound maximum packet lifetime (MPL). It doesn't matter whether MPL is 5 ms, 5 sec, 5 hours, 5 days, or 5 weeks. What does matter is that it is bounded. Of course, the longer the bounds greatly impacts what can be said about such systems and when it can be said. If MPL can't be bounded, then it is remote storage, e. g. file transfer. SCF is one of these and very similar to a distributed electronic mail application.

This is a good example of a case where what an external observer imagines is the case, is not how the system must perceive it (see page 87 of PNA). The fact that people have just assumed these were networks has lead to a lot of sloppy thinking. It is far from clear what it is we can know about such configurations. In particular, my sense tells me that there are pathological interactions between the security mechanisms and their requirements with Watson's bounds that create problems for this class of distributed storage applications. These issues must be address otherwise everything is built on sand.

With that said, my comments:

- 1) Page 2, column 1. Timer based mechanisms can be used without requiring global time synchronization. Even so, tight time synchronization is seldom necessary and a bad requirement in any distributed system. It is equivalent to introducing a single point of failure.
- 2) Page 2, column 2. Middle of the column. Your blind use of Salzer will lead you to the same mistake he made: that there may be multiple paths between adjacent nodes. Hence, routing must be based on nodes not points of attachment. (See PNA).
- 3) Page 3, Beginning of Section 3: Your characterization of the Internet naming is incorrect. Only one entity is named in Internet architecture, the interface. It has names taken from two namespaces: MAC addresses and IP addresses. Domain

names in DNS are macros for IP addresses. (URLs are not part of the Internet architecture but part of the web.) There are no other named entities in the Internet. (A path to a new instantiation of certain applications can be designated by an IP address and a local identifier often called a well-know port or socket. This is analogous to what was seen in very early OSs, where applications were accessed by a branch through fixed memory locations in low memory. It can be seen as a path to application process. But it is not a name of an application process or other higher level entities.)

I don't understand the last sentence of this paragraph. All identifiers in computing locate an object in a given context (See Saltzer's definition of "resolve."). So-called flat identifiers are merely identifiers used outside their context, i.e. by entities that do not understand their locator-related semantics. Hence all identifiers are locators and all locators are identifiers.

- 4) I also don't understand your claim that "both namespaces have centralized (though hierarchical) allocation" There are sub-authorities for allocation for both IP addresses and Domain names which make it decentralized. There does not seem to be anything centralized about it.
- 5) Top of page 3 column 2. Your claim that namespaces are "virtually cost free" is a bit of exaggeration. We have done a lot of work into the nature of these sorts of systems and have found that while costs are not exorbitant, they are not virtually free.

There is no such thing as unlimited scope. Such a concept is nonsense. The scope of a namespace is the set of all elements that may be members. The definition of that membership limits the scope. Merely creating another namespace with the rule "not members of the other one" would limit its scope. If by this statement you mean, that the names are not bounded. This also does not make sense, since the process of resolving the name must halt and hence be finite. Point 3 at the end of this paragraph I do not understand. As an aside I would note that it seems that the authors have adopted the false distinction widely held in the Internet about separating locators and identifiers. As we have noted elsewhere, this is not only a false distinction but a flawed view of the problem that has significant problems.

- 6) Page 4, Middle first column: So after talking about creating lots of namespaces, you create a namespace of namespaces (UUIDs), which effectively means you have created a single namespace. Wasn't this what at the outset of the paper you were claiming you weren't doing?

This concept is equivalent to the registration authorities of registration authorities and the object-id approach developed 25 years ago and still widely used. So we are back where we started. Why not use it?

- 7) Page 4, 3rd paragraph. You state that the “. . . node address comes from the MAC address . . .” How can you use an interface address for a node address?

At the end of that paragraph, there is talk of pre-configuring applications with UUIDs. Won't there be problems with this approach if the list has to be updated with new UUIDs? The paper does not seem to be clear about what can be assumed when, i.e. when Watson's bounds can be assumed and when they can't. This needs to be made clearer.

- 8) Last Paragraph first column of page 5. A batch of names is referred to as a sub-namespace. They are not the same concept. A batch of names is simply that just a batch of names still operating under the same rules. A sub-namespace presumably has a different set of rules. At the very least who the new owner may give them to.

Same paragraph top of second column: Aren't UUIDs the top-level HPoNs?

- 9) Please get your facts correct. ASN.1 Packed Encoding Rules are about as bandwidth efficient as can be done. Compression of them often comes out larger than before compression. They are certainly much more efficient (and richer) than JSON. It is true that the Internet use of ASN.1 was flawed, like most of the work it does. The reason ASN.1 is called an abstract syntax notation is because it is. Multiple concrete rules may be used with it. A property that JSON does not have. We have found Google Protocol Buffers superior to JSON, but we don't have (for now) the bandwidth constraints that that you are designing for. We have however left the door open for other concrete syntaxes.
- 10) Section 6.1. Doesn't this assume that revocations will propagate faster than the bad guys use of a compromised certificate and given the nature of SCFs may not propagate at all? This would seem to be a major weakness.
- 11) Section 6.1 paragraph 2. “. . . it is difficult to synchronize state across SCF . . .” The definition that this is not a network. Clearly modeling this as a distributed application with transient network layers would clarify a lot. Understandably, the Internet context you are using makes this difficult.
- 12) The biggest problem I see with revocation here is the time to notify, which could easily be infinite. In fact given the sophistication of attackers these days, I would fully expect them to take advantage of this property.
- 13) This whole structure seems to continue the flaw of the Internet architecture of assuming that addresses are exposed to applications and by implication that the communication ends at the layer boundary. Significant problems originate from this as does additional complexity.

- 14) Page 10, first paragraph. It is far from clear what this paragraph is saying. From reading, it appears to be the common naïve approach that will not solve the multihoming problem, notwithstanding the comment about taking care “not to use the locator as and [sic] identifier.” As state earlier, this distinction is not only a false distinction but the wrong distinction.

As a final comment, to reiterate the point I started with it is far from clear that there are not pathological interactions between the requirements of authentication and not being able to meet Watson’s conditions. Clearly, modeling this as a distributed application doing some sort of remote storage/distributed database function with transient support by network layers that do meet Watson’s constraints would go along way to making these things clearer. I would suggest that thinking about this as a network tends to bring in unconscious assumptions about what is the case that do not hold in this environment. Furthermore, the Internet architecture provides no help in this regard, since it really says nothing about the nature of applications nor even provides the barest minimum of hooks. We have found that the operation of these sorts of distributed applications in domains where Watson’s conditions are met, the resulting systems are more secure and far less complex where the structures of the PNA IPC model are adhered to. The fact that a DIF is a securable container (work in progress) has implications for distributed applications as well. In addition, the architecture for distributed application implied by the IPC Model provides considerable leverage to understanding these problems. We would invite you to join us in working on them.