ICNRG Interim Meeting Minutes 2015-10-3

Messages and Semantics

LW: NDN TR published 1.5 years ago with more details. on manifest. manifest embedding so you don't have to wait for round trip time to embed it
Chris Wood: This doesn't preclude that
LW: Does not show it
Chris: Im aware of that feature - this grammar doesn't preclude. Can still do embedding as Ilya suggests.
LW: It is an inadequate expression of what his work is.
Dave Oran: meta point - were talking not just about manifest data structure but about how you use it as well. We need to discuss in the process of writing a manifest stack - format and how to use. and if you don't write down manifest mechanisms for embedded RTTs and that is an omission in that spec. If you think that is important, it is an important design issue.
Chris -
LW: should agree to NDN TR - manifest get sequence number - consumers ask sequentially next data block. Manifest itself is a ? with a different data type. You should read the TR.
Chris; I will.

All in one stream version released in summer…

Q: Are you going through the history or is this current?
A: Going through history first

DO: Before we start doing a lot of custom info elimination from the manifest therefore making the parser potentially expensive and memory expanding, we should compare everything we do to a simple Gzip of the whole thing. If you can just construct the manifest with everything very plainly there, may have dups, with gzip and get about the same number of bits at the end - reducing the complexity - worth comparing.
Chris: we aren't wasting too many cycles focusing on this.
DO: every time you think about one of these optimizations, compare to gzip.

Ravi Ravinden: is it something for the end point or manifest supposed to be consumed just by end points or anyone?

chris: good question.
do - anyone who has the decrypt key?
chris - should this manifest be something the network knows and is aware of?
Cedric Westphal: if its end to end why would you have this specific …?
Marc Mosko - we don't want everything to have to implement their own parser, encoder … we want there to be a library
do: rather than every single app having to deal with this
GQ Wang; if this is on the end point functions, actually dash is doing the same thing
DO - this looks nothing like a dash manifest.
Nacho Solis: the base forwarder doesn't need to understand anything about manifests but it may be required for
Dirk Kutcher: if its useful will end up being de facto mandatory

JT: leaping ahead, manifest use these name by hash things which may include nameless objects and that would have implications on the forwarder. So to handle the manifest the forwarder would have to handle the nameless objects.

Chris will send around the the writeup for FLIC to mailing list (its on github now at: xxxx)

DO: If you're parsing one of these things and you get a T node, says you have to traverse the link to find out what the type of the node actually is (data or manifest)?
A; No - it's the node structure
marc: payload the is a field; pointer type tells what the pointer to object. so payload type should equal pointer type ???
DO: I don't understand what pointer type TNode means
A; will now be encoded in a manifest type.
Marc: T manifest means that the payload of the CO follow the manifest body definition; t node says it follows the node definition which means it can include data and external pioneers; t data means … I think its a little confusing too
DO: I would agree except for the word "little". If there are real semantics associated with this - it can't be called a node - only if there are no real semantics. If it has semantics need a better name.
A: its just redirection
DO - but there is all this other stuff there!
Nacho: Node basically just has pointers and payload

DO - whats in payload?
Marc - user and data
DO: only if its an application
Marc: this structure allows the app payload to be spread between leaf nodes and internal nodes
DO: I think my brain just exploded. How do you ensure loop freedom?
A: When you're parsing you maintain the hashes
Nacho - whats written on the board requires hash restrictions; requires hashes for everything
Chris: If you don't have a hash you can't do any kind of loop detection in the parser.

Ravi - is that a link or a pointer?
A: Its just a name - keyed restriction, hash restriction and type of what you point to
Ravi: name is just a content hash id
Nacho - thats nameless objects
ravi - is that something that is covered in the ccnx 1.0?
nacho: not in spec but will discuss in 1/2 hour.

chris; spec for this should include structure and use. We have a doc we need to circulate that describes potential use cases for the manifest. Good to circulate around group and getting some consensus as to how we should use manifest so we are all on the same page. I will circulate these documents and try to spark some discussion in the group.
ravi: is this the v2 doc to be shared?
A; work in progress - all very fluid. goes with nameless objects etc. Design not set in stone.
Q: presumably you're going to circulate v3. You're leaving us hanging!
DO: in order for the community to contribute to design in proactive mode rather than reactive. Like to suggest that discussion topics between v2 and 3 be posted to mailing group rather than discussed in smaller groups where others can't track easily. Because then if people find problems we have to wind back.

JT: is there something that says hash 256?
A: thats the default
JT:Is there forward compatibility if things change?
marc: has to be baked in to the forwarder?

JT:
marc: first address how a forwarder deals with diff hash types
DO: an observation for future… every time someone has proposed something that didn't have hash function, key agility, crypto function it has been shot down. It may be OK to defer now, but looking forward don't think it will go far without crypt algorithm and key agility
marc: need to figure out how to define it in a CO.
DO; if we have hash names , might be best to have a T in there that says type hash function sha 256 vs something else. not necessarily right away but won't go far without it.

DO: wondering how we might be able to disentangle progression of manifest work from progression on key and access control work so they don't get entangled too early. hate to have a type dependence between this and that other work. Lots of work going on in other area. This SDM there -at least annotate that that thing is not cooked or take it out for now and put in later. or leave a place holder for it.
A: It is not fully baked.
DO: also don't think it has had public discussion yet either.
A: the only requirement right now is that the SDM defines access control for everything in it - minimal coupling we have right now.

Q: there is now way where it says this is the certificate you're using. that is what you want to defer?
DO - I'm saying don't keep interdependence
A: metadata is just metadata about the clear text data. access control should be encoded in the SDM and they need to be separated.

What is the primary use of a manifest? Is it to encode a single thing? how you encode, parse, use it

Ravi: what does t mean you are asking for a chunk of content and you have a manifest in it? What is this use case where you are given content and manifest
A; now we got rid of the typed payload info, so if it has a manifest body its a manifest; if it has a regular payload its a CO carrying data.
Marc: manifest contains things like manifest info. so a small object e.g. that wants to have manifest info and payload. if payload could only be in a leaf node and manifest internal node you have to have two objects and squish them

together. or you could have one CO with manifest and some data and you're done.
Mark Stapp: issue is - i agree with DO - as someone who has been thinking about manifests for some time this is painful because its combining about 7 things - access control, crypto identity, etc too many things. This is the thing that turns into 150 page RFC that they refuse to review. manifests have some compelling use cases and then several wouldn't it be nice. They don't all have to be in at once. Couple of high value items: straightforward encoding of metadata; also I want to sign one thing, not every one of the 4k pieces. may be other things to do once you've got all the hashes. Write a spec on those two things first.

Marc: In version 3, there are a few decision points. one choice in v3 - the payload field is application payload. not a mix of manifest data and payload like in v2. Don't have blobs of app data being mixed in with manifest data. There is a new TLV thats Manifest data. Within manifest data there are sections for other manifest nodes, data nodes, and metadata like total length. hearing you, we could split those sections into different write ups.

GQ: Are manifests visible to forwarders?
DO: likely these manifests will be encrypted. Q1 - is it likely that the forwarder would have the key and would it care? Diff between forwarder and consumer. Any body can sit there and act like a consumer. even if you're an interim forwarder. Q is whether a forwarder or a consumer in the same box is the one doing it is kind gf an angel on the head of the pin question. I don't know a single node that is a router that isn't also a host. Really just a question about whether the decrepit key is on the box.

Manifests
LW: NDN GR published 1.5 years ago with more details. on manifest. manifest embedding so you don't have to wait for round trip time to embed it
Chris: This doesn't preclude that
LW: Does not show it
Chris: Im aware of that feature - this grammar doesn't preclude. Can still do embedding as Ilya suggests.
LW: It is an inadequate expression of what his work is.
DO: meta point - were talking not just about manifest data structure but about how you use it as well. We need to discuss in the process of writing a manifest

stack - format and how to use. and if you don't write down manifest mechanisms for embedded RTTs and that is an omission in that spec. If you think that is important, it is an important design issue.
Chris -
LW: should agree to NDN TR - manifest get sequence number - consumers ask sequentially next data block. Manifest itself is an object with a different data type. You should read the TR.
Chris; I will.

All in one stream version released in summer…

Q: Are you going through the history or is this current?
A: Going through history first

DO: Before we start doing a lot of custom info elimination from the manifest therefore making the parser potentially expensive and memory expanding, we should compare everything we do to a simple Gzip of the whole thing. If you can just construct the manifest with everything very plainly there, may have dups, with gzip and get about the same number of bits at the end - reducing the complexity - worth comparing.
chris: we aren't wasting too many cycles focusing on this.
DO: every time you think about one of these optimization, compare to gzip.

ravi: is it seething for the end point or? manifest supposed to be consumed just by end points or anyone
chris: good question.
do - anyone who has the decrypt key?
chris - should this manifest be something the network knows and is aware of?
cedric: if its end to end why would you have this specific …?
marc - we don't want everything to have to implement their own parser, encoder … we want there to be a library
do: rather than every single app having to deal with this
Q; if this is on the end point functions, actually dash is doing the same thing
DO - this looks nothing like a dash manifest.
nacho: the base forwarder doesn't need to understand anything about manifests but it may be required for
DK: if its useful will end up being de facto mandatory

JT: leaping ahead, manifest use these name by hash things which may include nameless objects and that would have implications on the forwarder. So to handle the manifest the forwarder would have to handle the nameless objects.

Chris will send around the the writeup for FLIC to mailing list (its on github now)

DO: If you're parsing one of these things and you get a T node, says you have to traverse the link to find out what the type of the node actually is (data or manifest)?
A; No - its the node structure
marc: payload the is a field; pointer type tells what the pointer to object. so payload type should equal pointer type ???
DO: I don't understand what pointer type TNode means
A; will now be encoded in a manifest type.
Marc: T manifest means that the payload of the CO follow the manifest body definition; t node says it follows the node definition which means it can include data and external pioneers; t data means … I think its a little confusing too
DO: I would agree except for the word "little". If there are real semantics associated with this - it can't be called a node - only if there are no real semantics. If it has semantics need a better name.
A: its just redirection
DO - but there is all this other stuff there!
Nacho: Node basically just has pointers and payload
DO - whats in payload?
Marc - user and data
DO: only if its an application
Marc: this structure allows the app payload to be spread between leaf nodes and internal nodes
DO: I think my brain just exploded. How do you ensure loop freedom?
A: When you're parsing you maintain the hashes
Nacho - whats written on the board requires hash restrictions; requires hashes for everything
Chris: If you don't have a hash you can't do any kind of loop detection in the parser.

Ravi - is that a link or a pointer?
A: Its just a name - keyed restriction, hash restriction and type of what you point to

Ravi: name is just a content hash id
Nacho - thats nameless objects
ravi - is that something that is covered in the ccnx 1.0?
nacho: not in spec but will discuss in 1/2 hour.

chris; spec for this should include structure and use. We have a doc we need to circulate that describes potential use cases for the manifest. Good to circulate around group and getting some consensus as to how we should use manifest so we are all on the same page. I will circulate these documents and try to spark some discussion in the group.
ravi: is this the v2 doc to be shared?
A; work in progress - all very fluid. goes with nameless objects etc. Design not set in stone.
Q: presumably you're going to circulate v3. You're leaving us hanging!
DO: in order for the community to contribute to design in proactive mode rather than reactive. Like to suggest that discussion topics between v2 and 3 be posted to mailing group rather than discussed in smaller groups where others can't track easily. Because then if people find problems we have to wind back.

JT: is there something that says hash 256?
A: thats the default
JT:Is there forward compatibility if things change?
marc: has to be baked in to the forwarder?
JT:
marc: first address how a forwarder deals with diff hash types
DO: an observation for future… every time someone has proposed something that didn't have hash function, key agility, crypto function it has been shot down. It may be OK to defer now, but looking forward don't think it will go far without crypt algorithm and key agility
marc: need to figure out how to define it in a CO.
DO; if we have hash names , might be best to have a T in there that says type hash function ha 256 vs something else. not necessarily right away but won't go far without it.

DO: wondering how we might be able to disentangle progression of manifest work from progression on key and access control work so they don't get entangled too early. hate to have a type dependence between this and that other work. Lots of work going on in other area. This SDM there -aat least annotate

that that thing is not cooked or take it out for now and put in later. or leave a place holder for it.
A: It is not fully baked.
DO: also don't think it has had public discussion yet either.
A: the only requirement right now is that the SDM defines access control for everything in it - minimal coupling we have right now.

Q: there is now way where it says this is the certificate you're using. that is what you want to defer?
DO - I'm saying don't keep interdependence
A: metadata is just metadata about the clear text data. access control should be encoded in the SDM and they need to be separated.

What is the primary use of a manifest? Is it to encode a single thing? how you encode, parse, use it

Ravi: what does t mean you are asking for a chunk of content and you have a manifest in it? What is this use case where you are given content and manifest
A; now we got rid of the typed payload info, so if it has a manifest body its a manifest; if it has a regular payload its a CO carrying data.
Marc: manifest contains thingsl ike manifest info. so a small object e.g. that wants to have manifest info and payload. if payload could only be in a leaf node and manifest internal node you have to have two objects and squish them together. or you could have one CO with manifest and some data and you're done.
MS: issue is - i agree with DO - as someone who has been thinking about manifests for some time this is painful because its combining about 7 things - access control, crypto identity, etc too many things. This is the thing that turns into 150 page rf. that they refuse to review. manifests have some compelling use cases and then several wouldn't it be nice. They don't all have to be in at once. Couple of high value items: straightforward encoding of metadata; also I want to sign one thing, not every one of the 4k pieces. may be other things to do once you've got all the hashes. Write a spec on those two things first.

Marc: In version 3, there are a few decision points. one choice in v3 - the payload field is application payload. not a mix of manifest data and payload like in v2. Don't have blobs of app data being mixed in with manifest data. There is a new TLV thats Manifest data. Within manifest data there are sections for other

manifest nodes, data nodes, and metadata like total length. hearing you, we could split those sections into different write ups.

GQ: Are manifests visible to forwarders?
DO: likely these manifests will be encrypted. Q1 - is it likely that the forwarder would have the key and would it care? Diff between forwarder and consumer. Any body can sit there and act like a consumer. even if you're an interim forwarder. Q is whether a forwarder or a consumer in the same box is the one doing it is kind gf an angel on the head of the pin question. I don't know a single node that is a router that isn't also a host. Really just a question about whether the decrepit key is on the box.

Generic name resolution
MS: And the argument is that this is more effect than IP?
Cedric: yes

MS: certainly software schemes out there that forward more than on the performance slide - that is not reflecting the current state of the art.

Cedric; I don't know
Marc: seemed like the idea was that you could put in a extra header with source and destination name - those are ccnx names?
Cedric; an attempt to answer some comment from last time asking about the practical use for having multiple name spaces. So this is not a proposal - just an example .
Marc: I have an app on one link that understand one name space and another - and I want a gateway that goes from A to B. how does this help? This sounds like both apps need to understand this new framing type.
Cedric: the app is not involved in there - just an app for routing and formatting
M: so this is for two gateways that are doing the translation?
Cedric; Yes.
M - so thats the intermediate form
Cedric - the angle is - you can do some kind of translation and must think about those issues. Going to have to find a way to go from one to the other. end goal is not to do ethernet forwarding

Ralph: let me see if I got it. 1) L2 forwarding between disparate heterogeneous

wire formats 2) name space translation
Cedric: draft is about multiple name space resolution. Idea up for discussion - should we consider this type of multiple namespaces - is that of interest?
Ralph: certainly of interest - should pull it out and disentangle and the translation needs lots of discussion - not clear if its a bug or feature.
Cedric: trying to show practical use case:
Ralph: forwarding part - diff wireless technologies you're using have different MAC address formats. different size addresses for blue tooth, mac layer, 802 …
Cedric: names are those MAC addresses and that whats translation is about .

DO: if the two name spaces don't have a bijective mapping - that seems a precondition for any scheme that works like this
Cedric: interesting Q. hat kind of namespace can you map from one to the other? Or do you have to rely on manifests for metadata? section in draft about manifest as well… Fits into the idea of transition - do you have multiple addresses coexisting?

DO: other thing worth thinking about - practical use cases - are there cases where the namespaces are diff but the objects and semantics are identical and the only diff is that they are mapped through a diff namespace. Bijective mapping and the to app data schemes must be identical. Otherwise you have to bud a real ALG of which name mapping is only one part.
Marc: if this is really talking about how to bridge together multiple link techs with diff addressing schemes - DARPA has been doing this for a long time, with and without reframing and transcoding diff info. lost of stuff has been done on that previously
Ralph: Thats why i was trying to disentangle those things. two diff problems. Nothing that says that bridging two techs means they have diff namespaces.

Christian - in my view we shouldn't have one gateway. In one namespace you can say this is my translator - and this goes in both directions.
DO - yes but you would need a bijective mapping
MS - but you can't get signing ?
CT: could have a translator that does everything for that namespace.
GQ: Could be another scenario in the IOT case.
Marc: if you're going to shift the original thing to shift your signatures then your app already understands that other thing whereas the goal here is to have native apps on the gateways in which case you need a trusted intermediary.

Ravi -Are you trying to bridge protocols or make an arch which can support
MS- don't make the argument about performance; show an example that shows something where you don't have a server that speaks both. If you want some per to peer app over L2 links - make the case for that somehow, but its not about forwarding performance.

Link protocol

ChristianTschudin
Q: meant to be a point to point link?
CT: yes

DO: Negotiation may not be one thing.
CT: we start where you have the possibility of exchanging datagrams. If the link is there then that part has already been dealt with.
DO: Only the parts of the negotiation that occur after the security is discussed.

DO: curious - if you thought of another design alternative - cast the link negotiation in terms of existing interest exchanges and make it one hop and make a name schema for it.
ACT I will come to that

Q: fragmentation (couldn't hear question)
A: In NDN fragmentation is part of the link protocol. If you have a full NDN packet cut in pieces it will travel (here). I hope that will reflect the philosophy

GQ: If we compare two options on slides - in middle one your link over N interfaces - every lower layer interface you have ? On right hand side do you want the same layer? need clarification
A: yes it would have the same layer.

"inner security" on hold.

Q: Are they speaking to each other using ICN messages or something else?
A: Other - will get to it.

DO: I think this discussion is confounding type demultiplexing with instance

demultiplexing
A: I want to do both with one mechanism
DO: I want to raise my objection to doing both with one

DO: Is the model here that the two directions of the link are dependent?
yes

DO: is it receiver driven or sender driven model?
A: receiver driven

DO: What are the barrier synchronization properties of this? I have a queue of regular messages and these messages. Which messages in the normal queue gets which state of the last LL set operation change
A; Haven't looked into that.
DO: I'll point out that the simple type demultiplexing scheme makes this much harder. Much easier to do if you have everything in one stream. I'm not arguing that that makes the other multiplexing design no good. I think we do need barrier synchronization by the way
A: ?
DO: If you use DTLS you don't have packet reordering I believe
A: security layer brings some advantages.

LW: minor clarification - NDN team is not aware of this piece of work.

Alex: Want to mention that NDN LP specifically designed to add anything needed at the link layer. The landscape picture would not be correct because secure channel would not be needed - some areas where would not need secure channel or fragmentation. n some cases you just need fragmentation and cases where you need both. Just point out - we have the specification written (15 revisions at least) and comments are welcome.
Chris: latest on redmine?
Alex: everything linked to redmine
Chris: The LP packet header is very flexible and a great feature. So Alex is right - the fragmentation . We are saying that we wanted every thing to be encrypted.

Alan at Cisco: are all of these links point to point? Are there broadcast in here?
A; All faces are point to point.
Alan: your other protocol

A: we were looking for a subset of work solutions . All that negotiation discovery

Ralph: so this really is dependent on UDP and some other thing has happened to get pairs of UDP address that people communicate across. That really seems to stick you with staying on top of UDP forever. Somewhere somehow someplace you've got to be able to find out what the two end points are. You have to find the other end point. It seems like we want to be able to do this without tunneling on UDP or IP eventually. Are we going to need to invent another thing that will give us the MAC addresses instead of UDP address? How will we eventually do a discovery mechanism
A: the assumption is that we would have to do the work you pointed out but we would be blocked if we couldn't work until we solved he discovery problem.
Ralph: Im working at the wireless radio level trying to do secure discovery all without UDP or IP. IPV6 is really ugly in that environment while ICN is beautiful but I need to do discovery. My concern is not that you have solved all these problems a priori but it feels like this is painting s into a corner of never being ablate do discover stuff
A: The discovery wok have to run publicly in some context, but having a language that says. Its a different discussion and we should have a different session

MS: Not an LLC advocate but interested in sensor networks. One of the things we've tried to do in our work is try to disentangle those things. If we want to collaborate and do experiments together across implementation how can we do that? in the world of laptops and mobile phones etc. IOT has different packet… the idea that we will have one set of semantics, keying perspectives etc doesn't seem reasonable right now but we shouldn't let it constrains it.

Ralph: Still have intuition that at some point we are going to want to not layer on top of IP, UDP…
MS: The point is to say if you want to set up a rendezvous - were trying to set up some rules there.

DO: at the meta level part of this discussion is whether adaptation to a type of lower layer linked the negotiation of how you want to run ICN on that hop could be coupled or could be decoupled and were not clear what parts should be either.

Ralph: Lets make sure that they could be decoupled.

Alan: I don't know if well be able to make this universal. Another question for the group as a whole - I can trying to figure out how I can get ICN all the way down. I see these solutions that say fine but you'll have to do all this extra stuff on other protocols.
MS: NDN people have DNS as well so its
ALAN: Are we intending to finish the scaffolding on everything so we can make progress?
MS: We want to connect our labs together so we can experiment.

nacho: Remind everyone that we have biweekly calls that people can join.

DO: From the point of ICNRG, those discussions don't happen until moved to the NDN mailing list

Ravi: The landscape picture: this allows you to not restrict yourself to point to point links. Good to clarify that its not a datagram layer

marc: Using ICN over a broadcast channel doesn't require that encryption be put at the network layer or higher transport layer in the link message. macs or encryption wifi work perfectly well at the ? layer.

CCNx over UDP
Chris Wood
DO: I'm stunned that you haven't considered web sockets (after the fact: DO meant WebRTC). Everything you need is there. Its very heavy weight but it seems like you're depending on a ton of infrastructure already so why not depend on more - its already widely deployed.
A: No reason - just haven't gotten there yet.

Will move the discussants the mailing list

Ralph: couple of observations. Have you considered using dans service discovery in the same way its described in the RFCs from Stuart Cheshire?
A: We will look at that.
Ralph: Will work over unicast dns or multicast DNS. If you were to use multicast DNS - you can configure it to work on the individual devices that are providing

the service without having to modify . modifying DHCP just as problematic as modifying DNS
A: Was discussed a lot at UCLA
MS: Just this matrix of trying to recognize different administrative domains.At UCLA Jeff owns it all so he can do things.
A: Still flushing out the matrix and come up with some recommendations.

Alex: Another piece of NDN that was old CCNx code. list of elements for local configuration. Includes some form of DHCP. Confused about the presented work meaning - a lot of things already done and tried out. Need to be more careful about presenting existing work.

Two separate things - the split between unicast DNS, multicastDNS … is triggered by the . suffix at the end. Yo can do regular service lookups over DNS just fine.

Marc: Would be very helpful we should document what the configuration variables are. What is it that you're trying to discover and decouple it from how you're trying to discover it. What are the parameters and then talk about using method x vs y to discover it. For the link control, again, just saying that these are the inputs that go into discovering the link control then you can do it in whatever you want.

A; good observation

DK: this line of work seems to consider point to point - if we went to broadcast or multi cast would you use different solutions?
A: probably, but for now just biting off what we can chew.

MS: I want to do a simple thing - collaborate and experiment with others. I need to be able to reach my lab's forwarder - thats all I want to do. I could hard wire it but that seems fragile. Even that is somewhat complicated. a valuable exercise while were trying to build up to this simple thing. Thats not trivial. The matrix of how to do that is dependent on the variables in your IT dept.

DO. I want to do more.I agree that we need mohave sufficient inter op for the simple case. But the latency in getting good designs is measured in months and years. this isa research grip - interesting research in algorithms and

architectures - broadcast links into an ICN architecture. Don't restrict yourself to taking one step 2) another piece of architectural disentanglement to be done - diff between managing adjacencies vs links. Agencies are inherently point to point. Routing protocols need to manage adjacencies.

Nacho; Specifically trying to solve the simple problem of finding forwarder - not to preclude solving the other big problems.

Alex: This is done - we did that already. we have this capability. We have all this mechanism implemented.
MS: The semantics of other things you require does not meet our needs.
Alex: Follow the instructions.
LW: What has been done is one thing, what people want to get is an ongoing discussion.

DO: We want to talk about because architecturally its entangled in the NDN world - dynamically creating neighbor adjacency based on the arrival of an Interest and trying to bind it to the next hop. As opposed to reestablished neighbor relationships. Am I wrong?
LW - not wrong but its inaccurate.
DO: are all of these next hops to adjacent neighbor reestablished? Ah - they are all preestablished. I had a misunderstanding
MS: casual remark: part of the web sockets issue is that …
DO: I meant web RTC

Nameless Objects (Mosko)
MarkStapp: some people are concerned that name expresses locality, but the routing might not actually grab data from multiple locations. Point: there exists a service that provides objects everywhere, and service
Mosko: Network redirects interests to nearby locations automatically based on name
Stapp: Complicated -- must know about every object (duplication)
Mosko: Alternative -- Name expresses some namespace wherein some close "responder" provides the data (based on hash rest.) Flickr might do this by encapsulating content with names under their own namespace. Alternatively, they just re-sign content so that people can know it came from (a) Flickr or (b) me originally. Equivalent to Flickr rebranding content under its own namespace

without changing the trust of the original manifest-encoded data.
Alan: But all Content is still signed by you?
Oran+Mosko: Manifest is signed only (root of tree), and is signed by original producer
Ravi: how to get the "wrapper" that does redirection to local content (the inner manifest, really)
Mosko: haven't got there yet
Borje: what about having publisher providing this service? Provides scalability
Mosko: yes -- that's one model we've considered
Paul: no implied trust relationship between SigA and SigB (wrapper and inner manifest)
Mosko: correct — there's no implied relationship — trust model is separate and obtained/established using something else
Stapp: if I trust SigA, and I trust sigB, why do I need SigA? When would I need both?
Oran: original name is the authoritative name of the publisher, and from there they obtain the CDN redirection info, and the sig. from the CDN is not important in verifying the original content (but it is important for protecting against intermediate MITM attacks, e.g.).
Mosko+Oran: The original signature is needed as a way of confirming the delegation.
Mosko: One model where routing finds nearest replica. Another model is where there's a separate mechanism for redirection that provides a name for a specific location
Solis: redirection mechanism is orthogonal to nameless object concepts
Alex: this is only good for static content — wouldn't work for dynamic content (which CDNs provide). Are you trying to reinvent how CDN's operate today with DNS?
Stapp: No, that's not it. No need for dynamic routing information that is done with CDNs+DNS today
Oran: CDN selection is not part of this — the choice must still be made
Stapp: Topological information must be part of CDN routing
Alex: aren't we supposed to not be dependent on CDNs? Not everyone can do that.
Oran: If that's the case then this is a non-problem.
Mosko: For smaller devices, look to bit-torrent p2p model for redirection/routing/locating
Alex: not universal solution

Oran: this is not being offered as a universal solution
Solis: this is getting off topic — we're not talking about nameless objects any more. A lot (most?) of traffic is static, like short-lived web pages, so this technique still applies, and this covers most of the traffic on the Internet today
Mosko: sufficient condition for nameless object: requesting by hash and interests
JeffT: A name is absolutely required!
Mosko: yes, that's condition #1. However, one must index into the PIT without a name (when the Content Object is returned)
Solis: satisfying from the CS may not be a requirement
Ravi: change of variables needed?
Solis: maybe.
Mosko: interests have locators and identifiers, and maybe other things
Ravi: yes, and that affects forwarder behavior
Solis: Ravi wants to call the name a locator
Mosko: name is probably just a routable prefix, maybe additional components for service MUXing The name is not really related to the original name
Oran: precondition: must be able to match a a PIT entry independently of what name was used int he interest
Solis: yes
JeffT+Oran: different names with same hash map to same PIT entry?
Solis: no
Oran: Semantics are important — does the above case yield one or two PIT entries?
Mosko: I'm only talking about matching the PIT entry upon return of the Content Object
Oran: okay, I agree that it should be 2 entries
Alan: are hashes globally unique?
Oran: yes, if not we have bigger problems
***Oran: this needs more open design discussion. And I'm uneasy about matching PIT entries based on hashes alone.
Mosko: need to be able to do PIT matching without the name (i.e., only on the hash)
Oran: single hash matching 17 PIT entry hashes will satisfy all of them?
Borje: isn't this a departure from NDN/CCN?
Mosko+Solis: Not really (only CCNx 1.0 because of exact name matching)
Stapp: should be *all PIT entries*, not "any", when checking PIT entries for the corresponding entries
Ravi: is name a hop-by-hop header now?

Solis: no, it just carries a name like usual, for routing. Name and hash must be stored because an interest alone does not tell you if the response will be a content object with or without a name.
Ravi: how does redirection happen?
Solis: we don't handle this here — this scheme treats interest just as before. The new stuff is matching based on hashes.
Mosko: must be able to index PIT without name (by hash), and therefore can't index until the entire content object is received
Gibson: is aggregating based on hash only okay?
Oran+Mosko: No, it can lead to DoS.
ArizonaProf: are interests signed?
Solis: no
ArizonaProf: can't routers drop things without names?
Many: yes, that can happen with or without interest signatures
Mosko: nameless object can only match an interest with a hash restriction
Gibson: what would happen if content object has a hash and a name (that maybe didn't match the interest name)?
<missed>
Stapp: Private communication removes the need to do any field checking/ processing
Mosko: consumer will ask by name, name+keyid, name+hash, or all three….
Solis: cooperating attackers can poison caches for 3rd parties
Mosko: nameless object must be nameless, else injection is possible by foreign names
ArizonProf: documentation available?
Oran: yes, white paper is online, and it needs a lot design/work
Mosko: nameless objects are a way to position objects on many replicas without needing to resign, rename, etc.
Oran: you can do that now, and this is not necessary for achieving these semantics, but we just might cache the same thing twice. There may be two things combined here, but they may have practical downsides.
Dirk: like the idea, since we tend to overload the name for organizational info/ structure, etc., and removing the name gives us kind of a flat name structure, and it blends well with CCN/NDN architecture. Wants to explore further.
GQ: saves a lot over wireless interfaces (because the bits aren't there)
Christian: how does content validation work without signatures?
Mosko: based on the hash restriction, and as long as hash-based names(locators) are used. It doesn't matter where that information comes from,

be it a (signed) manifest or other stuff.
JeffT: one could also fetch the manifest by name.

NDN Protocol Development (Alex)
Oran: What's a hub?
Alex: other node (the forwarder you're connecting to in the testbed)
ArizonaProf: is it a gateway?
Alex: yeah
Oran: used for two forwarders to bring up a link?
Alex: could be used for that too — hub is basically a gateway (local gateway to the testbed, or remote gateway to the testbed) — just a forwarder
Oran: is this for pairs of forwarders (inside the testbed) to bring up links amongst themselves?
Alex: yes, this is a different protocol
Oran: would be interesting to explain why different protocols are needed for edge connections and two internal node adjacency/link protocols — is it the same and just not implemented as such yet?
Alex: currently for stub nodes to connect to the testbed
Lixia: we will document why this is a separate protocol
Ralph: (r.e. NDNLPv2) given NDN semantics, why can't you build this protocol in NDN? Why does it have to be a separate protocol? Is this due to a limitation in NDN? Why something separate than Interest/Data?
Alex: we needed fragmentation
Ralph: if you added fragmentation, could you build this in NDN?
Lixia: this is just a wrapper
Oran: why design a new state machine?
Lixia: this is just an encapsulation of the interest/data state machine processing
Alex: there may or may not be a state machine for processing NDNLPv2 packets
Oran: how do NACKs propagate beyond one hop to consumer?
Lixia: it's just one hop
Oran: I understand the difference between link failure NACK and either (a) end-to-end NACK nor (b) routing NACK
Lixia: nodes need to determine how to handle single-hop NACKs (i.e., reroute interests if no FIB entry existed upstream)
Oran: Interesting design decisions to be made, need to find or agree upon scope of network hop-by-hop NACKs
Alex: processing happens on hop-by-hop basis, and error code may change based on stateful processing of routers

Oran: there is not a clear difference between L2 and L3 errors (since it's baked into NDNLPv2) (?) — this puts these types of errors together. I can see arguments for keeping them together and separating them.
Oran: if interest needs to be re-forwarded due to a link error, this NDNLPv2 coupling means that interest must be decapsulated and then encapsulated
Ravi: What fields are in LpHeaderField?
Alex: they're online
JeffT: list of types of header fields is missing
GIbson: LINK is not part of NDNLPv2, right?
Alex: right, I'm moving on
Oran: LINKs are cacheable?
Alex: yes, but different interests from different LINKs lead to duplicated cache entries
Oran: unidirectional in the sense that HTTP links are unidirectional, and not bidirectional like Nelson links?
Alex: yes, unidirectional, but may not fully understand the question
Oran: delegation in one way, not that the delegatee agrees to the delegation
Alex: yes, right
Oran: we may want to consider bidirectional LINKs
Solis: interests issued from LINKs carry the LINKs themselves (so that an interest is forwarded based on name and LINK object delegation, if name cannot be routed upon)
Alex: at a high level, yes.
Solis: does each hop annotate a routing prefix?
Alex: any hop can annotate an interest about what has been picked as the next hop, so interests are annotated on a hop-by-hop basis
Ravi: what if you are able to route on both names
Alex: pick one
Ravi: what if it brings you to the wrong dest?
Alex: pick another one. The forwarding strategy (i.e., when does a forwarder choose to use the LINK to forward) is a bit complex and discussed on the red mine.
Solis: LINKs serve as hints to forward aside from the actual name, and eventually you get to something that can handle the original name. When content is sent back, do you annotate content based on how it was matched?
Lixia: No, content object is not matched.
Oran: LINKs seem like secure authorized routing hint.
Solis: Poisoning is possible because content object does not annotate the path

after forwarding based on LINK.
Lixia: mitigated by retrying interests.
Solis: protocol is broken if you don't handle poisoning.
Alex: still under discussion.
Mosko: this is not about trust delegation, but about routing delegation
Alex: yes, we need to use a different term
Ravi: what kind of mobility?
Alex: one of the use cases is publishing data while moving
Ravi: it's network-level mobility?
Oran: not network level mobility, it just handles handoff
Mosko: who verifies the LINK signature?
Alex: consumer, but forwarder could also verify the signature.
Mosko: what about trust?
Alex: still under discussion, but schematized trust models can help limit scope
Solis: you have a new cert. format?
Alex: yes, we treat content objects with public keys and signatures as certificates. We are trying to clearly define the security elements as NDN elements.
Solis: can you comment on current state of selectors and excludes?
Lixia+Alex: work in progress — it's application driven work.

Controlled Sharing of Sensitive Content (Yingdi)
GQ: content key is symmetric key?
Yingdi: Yes
Mosko: is there just one entity that's encrypting under a given encryption key? or is it multi producer with the same encryption key?
Yingdi: multiple producers will have multiple keys, but these content keys will be encrypted using the same group key
Mosko: I believe the content because it's signed with a key that's separate from the encryption key?
Yingdi: yes, we use separate keys for signatures and encryption. Content names and key names are under different namespaces.
Oran: is "C-KEY" uppercase because it's a constant? Or is it just a random string?
Yingdi: it's a constant string that indicates it's a key used for content encryption.
Oran: This is an example where naming conventions trigger semantics, as opposed to other architecture which use typed name components to do this
Yingdi: Yes

Service centric networking architecture for challenged networks (Wang)
<no content questions>
Oran: can you send me your slides?
Wang: yes

Privacy discussion (Dirk and Mark)
Dirk: IAB meeting last week, discussed impact of encryption on mobile services/ nodes. One of the insights was that everyone is betting on TLS for privacy. This needs to be discussed in the context of ICN. It's an important issue that is being ignored.
Cisco: Privacy is not the same as confidentiality
Stapp: Privacy means what TLS means
Oran: ICN does good for integrity, and a good job for confidentiality, but it does not do a good job for privacy. It would be useful if we defined privacy better.
Oran: encryption doesn't preclude temporal caching (for retransmissions), but it does for cross-user caching
Mosko: if you were the research advisor, what do you think should be solved first?
Stapp: important bodies that own platform (NDN and CCN) should forbid papers and presentations that depend on in-clear information/exposure.