Identifying NDN/CCNx1.x Commonalties and Differences: a high level summary

1. Common Starting Point
	1. Encoding
	2. Naming
	3. Data fetching
	4. Name discovery
	5. Caching
	6. Interest aggregation
	7. Misc
2. NDN
	1. Preserved
		1. Naming
		2. Data Fetching
		3. Name discovery
		4. Caching
		5. Interest aggregation\*
		6. Misc
	2. Updated / Evolved
		1. New Encoding
		2. Introduction of Link/Network Adaptation
		3. Add name-based scoping
		4. Add typed name components
		5. Add LINK object
		6. Add HopLimit (under development)
		7. Progress with Data-Centric Security
		8. Progress with Multi-Party Sync
3. CCNx 1.x
	1. Preserved
		1. Naming\*
		2. Caching\*
		3. Interest Aggreagation\*
	2. Updated / Evolved
		1. New Encoding: Packets carrying Messages
		2. Restricted Data Fetching
		3. Changed name discovery (in-network -> app)
		4. Add typed name components
		5. Add interest payload
		6. Misc

\* includes changes

Where we all came from: Van Jacobson’s invention of content centric networking.

# CCNx 0.8 as a common starting point

* Binary XML format
* Packet Naming
	+ Full name : “/foo/bar” + implicit digest
	+ Exact name : “/foo/bar”, 0 components after
	+ Prefix name : “/foo/\*”, 0 or more components afterwards
* Initial set of naming convention to carry semantics of the name component contents
* Allow Data fetching by prefix
* In-network name discovery
	+ with Selectors support
	+ data packet carrying “FreshnessSecond”
		- relative time, not assuming sync’ed clock
* Opportunistic in-network caching
	+ Each data packet can be cached with forwarded-defined policies
	+ “Fresh”/”stale” semantics for the cached data
* Similar Interest aggregation
* Nonce in Interest packets to detect and prevent Interest packet looping

# NDN

Since fall 2012 (Jacobson resignation from PARC), NDN continues application-driven architecture development direction, applying NDN to address different application scenarios.

## NDN protocol preserves:

* Naming
	+ Full, exact, prefix name
	+ In general, a Data packet is uniquely identified by the exact name
	+ Full name as last resort under unexpected conditions
* Allow Data fetching by prefix <I thought this is part of name discovery?>
* In-network name discovery
	+ With Selector support
		- As a temporary mechanism to implement in-network name discovery
		- Open research for the adequate replacement
	+ “FreshnessPeriod” in Data packets as a relative time to treat Data “fresh” for discovery purposes
* Opportunistic in-network caching
* Similar interest aggregation
	+ With addition of new exponential-back off interval to allow interest retransmission
* Nonce

## NDN protocol changes:

* Binary XML to TLV
	+ Variable length encoding of T and L to allow flexibility for diverse network environments
* Introduction of network- and link-specific Network and Link Adaptation layers
	+ One instantiation of these is NDNLPv2
	+ Carry network-specific information Hop-by-Hop or within several network hops (e.g., transmitter geo coordinates in vehicular networking segment)
* Nama-based scoping
	+ “Scope” selector replaced with naming conventions “/localhost”, and “/localhop” to limit interest propagation within a single host or a single hop.
* Typed name components
* LINK object to implement optional indirection for Interest forwarding
* HopLimit as part of the network adaptation layer
	+ (spec under development)
* Progress with data-centric security efforts
	+ Exploring signature formats: RSA, ECDSA, HMAC
	+ Command (Signed) Interests
	+ Trust schema
	+ Name based access control
* Exploration of new design patterns (e.g., multi-party synchronization as a new transport paradigm)

# CCNx 1.x

PARC Protocol Preserves:

* Packet naming in general
	+ Full name combines of “Name” and “digest” that is not logically tied to the name
* Opportunistic in-network caching
	+ With exception of changed “fresh/stale” semantics to “alive/dead” semantics
* Similar interest aggregation
	+ With addition of “re-expressed” Interest detection

## CCNx 1.x Protocol Changes

* Binary XML to fixed-header plus TLV
	+ Fixed header for end-to-end network layer with optional TLVs that can be added/modified hop-by-hop
	+ Followed by TLVs that describe ICN packet (“message”)
		- TLV with fixed length field
* Naming
	+ Concepts of full, exact, prefix name are preserved, but only full name is expected to uniquely identify Data
	+ Introduces intentional use of the same exact name for different Data (protocol needs)
* Allow data fetching using exact or full names only
	+ Resulted implication: intentional use of the same exact names for different Data
	+ CS semantics changed from fresh/stale Data packets (CS can keep stale) to alive/dead (CS must remove dead)
	+ (current spec only) CS alive/dead decision requires absolute time synchronization within required discovery resolution
* In-network name discovery replaced with app-defined name discovery:
	+ Manifests for static data
	+ Encoding Selectors as part of the Interest name
* Typed name components
	+ As a replacement for naming conventions
* Removes Nonce from Interest packets
* Adds HopLimit in Interest packets
	+ Replaces Scoping
* Adds Interest Payload
* Special treatment of Data packets whose full name contains only the implicit digest
	+ Data is matched against “restriction” field; name is completely ignored---can be used as Interest forwarding indirection