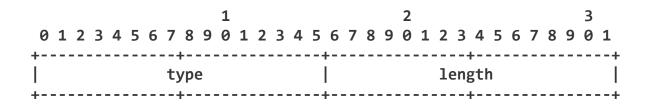


CCNx 1.0 Wire Format

Computer Science Laboratory Networking & Distributed Systems

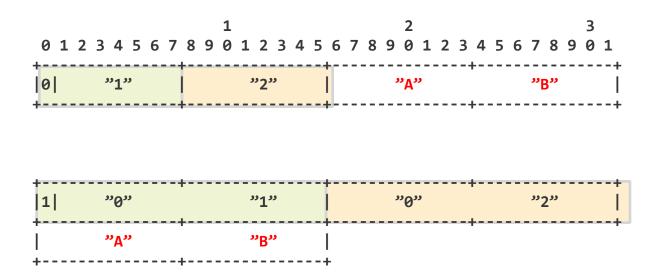
March 2014





New wire format uses Type-Length-Value (TLV) 2 byte Type 2 byte Length Up to 64KB value



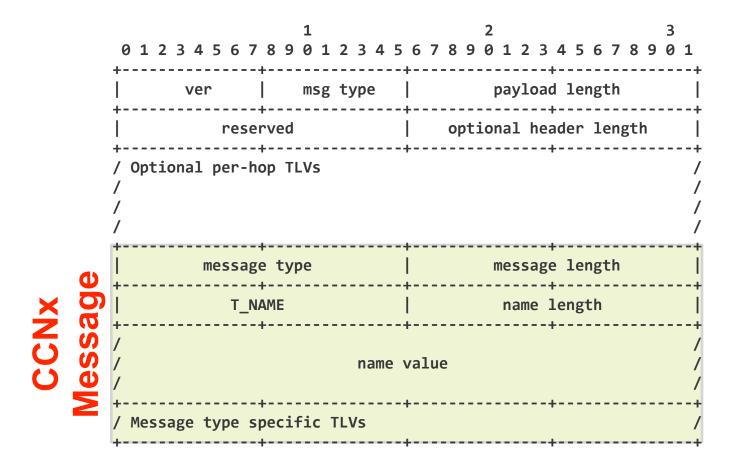


Variable TL alternative — and problems
How to normalize aliases?
Is type "01" same as "1"?
Is length "02" same as "2"?
If they are the same, how will a hash detect that?



- o ver: the version of the packet.
- o optional header length: The length of optional per-hops headers. The minimum value is "0".
- o msg type: 0 = content object, 1 = interest
- o payload length: Total octets following the headers (fixed header plus optional headers).







/utf8=foo/binary=0x656060/serial=24/ segment=3 Name path segments have types Binary **Application Specific** Nonce Keyld Metadata **Content Object Hash** Content Object Segment **Version Timestamp** Version Serial Number etc



0 1 2 3 4 5 6 7 8 9 0 1 2 3							
F	ixed H	leader	•				İ
/ Per-hop TLVs	•						/
T_INTEREST							
T_NAME							
/ / / name value /							/
/ Optional Interest KeyId TLV						/	
/ Optional ContentObjectHash TLV					/		
/ Optional Scope TLV							/
/ Optional AllowedResponseType TLV					/		
+/ / Optional Interest Lifetime TLV +						/	



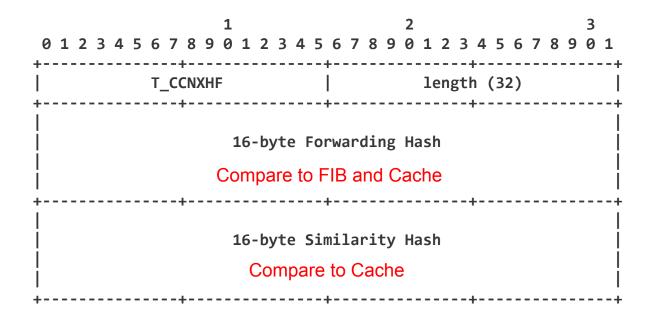
	1		2			3			
	2 3 4 5 6 7 8 9 0 1 2 3 4 !								
+++									
+		.+	+				¦		
/ Pe	r-hop TLVs						/		
Ī		Ì	message	length	1		ĺ		
		1	name 1	ength					
/ / / name value /							/		
/ Ma	+/ / Mandatory Name Authenticator TLV								
/ Optional Protocol Information TLV						/			
/ Op	/ Optional Contents TLV +						/ +		
	Mandatory Signature Block TLV								



Interes Fragment Stream ID (random) | X X | FragCnt | X X | FragNum Path Minimum MTU T_OBJFRAG Fragment Stream ID (content object hash) Object Maximum MTU | X X | FragCnt | X X | FragNum **Interest Stream ID** (matches)



Forwarding based on pre-computed values in Per-Hop Headers CCNx Hash Forwarding (CCNxHF)





CCNx 1.0 Protocol Roadmap

Labeled URIs

Peer-to-Peer Discovery

Sync Based Discovery

Directory Based Discovery

Selector Discovery

TimeVersion

SerialVersion

Segmentation

Core Protocol Equals, ComputeHash

Hash Forwarding

Fragmentation

TLV Wire Format



Documents

- 1. CCNx 1.0 Protocol Specification Roadmap
- 2. CCNx Semantics
- 3. TLV Packet Format
- 4. CCNx Messages in TLV Format
- 5. Labeled Segment URIs
- 6. Labeled Content Information URIs for CCNx
- 7. CCNx Content Object Caching
- 8. CCNx End-to-end Fragmentation
- 9. CCNx Content Object Segmentation
- 10.CCNx Publisher Clock Time Versioning
- 11.CCNx Publisher Serial Versioning
- 12.CCNx Selector Based Discovery
- 13.CCNx Hash Forwarding

