RESTful Information-Centric Networking

Dirk Kutscher and Dave Oran

ICNRG @ IETF-115
2022-11-08
8:00 - 9:00 JST: Panel 2
Session Chair: Alexander Afanasyev (Florida International University)
Panel: ICN and the Metaverse – Challenges and Opportunities

Statement: RESTful Information-Centric Networking

Dirk Kutscher  
Hong Kong University of Science and Technology  
Guangzhou, Guangdong, China  
dku@ust.hk

David Oran  
Network Systems Research & Design  
Cambridge, MA, USA  
daveoran@orandom.net

Statement: As TCP/IP is to the Web, ICN is to the...?

Jeff Burke  
jburke@remap.ucla.edu  
UCLA REMAP  
Los Angeles, California, USA
More Background
Internet Protocols for Efficient RPC Communication

Systems Approach

QUIC Is Not a TCP Replacement

Bruce Davie  
Sep 26

The publication of a new, definitive specification for TCP (RFC 9293) is enough of a big deal in our world that we couldn’t resist a second post on the topic. In particular, we were intrigued by the discussion that compared QUIC to TCP, which inspired this week’s newsletter.

https://systemsapproach.substack.com/p/quic-is-not-a-tcp-replacement
Representational State Transfer

Theory: Stateless Requests

Verb

Resource Identifier

GET /resource/A
; including all request & content parameters

Response
; representation of server state

MODIFY /resource/A
; including all request & content parameters

Response
; representation of server state
Representational State Transfer

Reality: Not So Stateless Requests (Cookies)

Verb
Resource Identifier

GET /resource/A
; including all request & content parameters

Response
; representation of server state

MODIFY /resource/A
; including all request & content parameters

Response
; representation of server state
RESTful Reality

HTTP3

- Connections, security contexts, channels
- Request parameters, cookies

QUIC

TCP-like congestion control
loss recovery

UDP

IP

HTTP3

TLS-1.3

QRIC

• Connections, security contexts, channels
• Request parameters, cookies
Information-Centric REST?

• ICN-idiomatic RESTful communications as a building block for applications
  • Clients and servers in sessions
  • Common understanding of state evolution
  • Suitable for a broad range of applications
  • At least HTTP/TLS’s security and privacy features
• Can we do this better than state of the art (HTTP3/QUIC/TLS-1.3)?
  • Simpler protocol machinery
  • Less overhead on the wire
  • Leveraging typical ICN benefits
Naïve ICN Approach

Interests as Vehicles for Requests

Verb

Resource

Identifier

Interest /resource/A/get
; including all request & content parameters

Data (Response)
; representation of server state

Interest /resource/A/modify
; including all request & content parameters

Data (Response)
; representation of server state
Naïve ICN Approach
Interests as Vehicles for Requests

• Flow balance
  • Request parameters can require a lot of bytes (often more than the state representation in the response)
  • Interests are intended to regulate Data packets

• Computational overload attacks on server

• Application layer processing time vs. network layer timeouts

• Secure sessions and name confidentiality
Naïve ICN Approach
Interests as Vehicles for Requests

• Flow balance
  • Request parameters can require a lot of bytes (often more than the state representation in the response)
  • Interests are intended to regulate Data packets

• Computational overload attacks on server

• Application layer processing time vs. network layer timeouts

• Secure sessions and name confidentiality
RESTful ICN Design
Data-oriented REST Sessions

• Enable client/server communication
  • With a series of request/response interactions in a session context

• Employ Reflexive Forwarding for RPC communication
  • Allow for robust ICN-idiomatic client/server communication with client parameter passing
    • For both key exchange and actual RESTFul communication

• Enable secure RESTful communication using standard ICN mechanisms
  • Content Object encryption and signatures
    • Without forcing all interactions into TLS-like tunnels
RESTful ICN Design 2

Efficiency

• Supporting a series of requests (in a session)
  • Avoid setting up context state for every request and the corresponding protocol interactions

• Establish and maintain shared "session" state
  • Using identifiers of keys and associated security context negotiated by setup phase
  • Reflexive Forwarding Parameter passing machinery for clients to refer to previously created application state
  • Emulating HTTP cookies
State Management

- Secure referent state held on a particular server (through key-ids) and a referent to application state through parameters secured through those keys
- Basis for enabling key features of today’s session based RESTful protocols
  - Application state caching on clients to allow server agility
  - Securing application state exchanged through pair-wise session keys with particular server
  - Rapid setup of these keys using TLS 1.3-compliant key exchange protocol
  - Efficient state evolution (minimizing round-trips and state representation overhead)
  - RESTful semantics for multiple interactions with the application through the same server
- Caveat
  - Have to make sure that client talks to the same server over multiple requests
  - Or that there is some server-side state synchronization machinery
CCNx Key Exchange

Mosko, Ersin, Wood: draft-wood-icnrg-ccnxkeyexchange

- TLS-1.3-like key exchange protocol between two peers
  - For establishing a shared, forward-secure key for secure and confidential communication
- Wraps "inner" ICN communication (Interest/Data) into "outer", TLS-style secured Interest/Data exchanges
  - Orthogonal to reliability and congestion control
- Designed for client/server scenarios
  - Protection against computational overload attacks
  - Can use different infrastructure for security and service functions
RESTful ICN
Session Setup

• Integrating CCNx-style kex exchanges in Reflexive Forwarding framework
  • Same semantics
  • Less data in unsolicited Interests
  • A few more roundtrips
• Coupling session state and keying
  • Key revocation => session termination

Round 1
(HELLO – HELLO-REJECT)

Round 2
(Full-HELLO)

Round 3
(App-Data)
RESTful ICN
Requests and Responses

- **Reflexive Forwarding**
  - ClientContextHandle in initial Interest
  - Contains necessary SessionID and key-id for the security context
  - Plus encrypted name for application state representation

- **Responses**
  - Request results
  - Encrypted name for new session state representation

- **Not using tunnel-like encryption**
  - Encrypting content objects with symmetric key
Conclusions

• Time to think about web over ICN: basic Interest/Data not enough
• Key idea here: Integrating key exchange with reflexive forwarding
  • Provide required context handles in initial Initial interest
  • Use negotiated keys for symmetric content object encryption
• Approximate capabilities of current state of the art (HTTP3/QUIC or TCP)
  • Overcoming complexities of 3 layer approach with isolated implementations and protocol machinery
  • Potentially easier to implement
  • Still enjoying the usual ICN greatness
• Future work
  • Name privacy
  • Build it