Transactions in ICN

Dirk KUTSCHER

2024-03-19
ICNRG @ IETF-119
ICN and Distributed Computing

- Group of nodes evolving a common state
- State: could be seen as shared namespace and its Named Data Object (NDO) hierarchy
- NDOs immutable and 'uniquely' named
- State evolution through NDO publication events
- Protocols such as PSync, SVS.
Transactions – ACID properties

• Atomicity: operations as single entity
• Consistency: maintaining consistent state before and after transactions – easier in client-server settings...
• Isolation from other transactions
• Durability: transactions make persistent changes to data
Different Ways to Realize Transactions

1. ICN as a network layer
   - Client-server communication between two nodes
   - Implement transaction semantics on top of an ICN messaging service

2. Recording state changes in shared data structures
   - Shared namespace, potentially functioning as a transaction ledger
   - Still need to think about atomicity etc.
1. Transactions as Messaging over ICN Network

- Client-server communication between two nodes
- Implement transaction semantics on top of an ICN messaging service
- Different approaches
  - A: **Traditional layering**: Using NDN-like systems as a messaging layer
    - Assign prefixes to client & servers
    - Send messages back and forth, and implement reliability and transactions semantics on top
  - B: **ICN-native communication**: Use Interest-Data as request-response abstraction for transactions
    - Mapping transaction communication and state evolution more directly to ICN, e.g., Interest-Data in NDN
    - Collapsing traditional network, transport, application layer functions
Today

1. **B: ICN-native communication: Use Interest-Data as request-response abstraction for transactions**
   - Based on RICE and Reflexive Forwarding

2. **Recording state changes in shared data structures**
   - Based on manifests
Secure Web Objects and Transactions

Dirk KUTSCHER

2024-03-19
ICNRG @ IETF-119
Original Web Idea

• Hypermedia system with document composability and application interoperability
  • Characterized as an Information Management approach
  • Different types of link semantic
  • Different types of applications (not only static object access)

• Universal Naming Concept (URIs)
  • Independent of server locations/addresses
Web Reality

• **E-commerce-driven security architecture**
  - TLS-based server authentication for bootstrapping secure communication between clients and servers
  - Names (DNS) for servers mostly

• **Elaborate distribution of app logic, rendering, UX**
  - Server- and client-side frameworks: React, Vue, Node.js etc.
  - Virtual secure pipe between browsers and servers – "single server applications"
Data-Oriented Web

- Enable Web applications to create and exchange Web objects
  - Accessed by URI-like names
  - Without relying on the communication channel for security
- (Re-)enable an array of different interaction styles,
  - Individual Web object access across applications
  - Scalable multi-destination distribution
- Secure web objects (SWOs)
  - Application-data units – usual ICN properties
  - Trust beyond Web-PKI – ICN-intrinsic certs and key access
  - Amenable to both infrastructure-connected as well as local information access: local-first communication
- SWOs beyond Named Data Objects
  - Meta data concept to support computing on and on-demand transformation of objects
  - Provenance
  - Effective group access control and key distribution

Information systems start small and grow. They also start isolated and then merge. A new system must allow existing systems to be linked together without requiring any central control or coordination. – Tim Berners-Lee
ICN Components – What do We Need?

- **WebRTC-style interactive media**
  - Protocol-wise, we know how to do this
  - Opportunity in creating Zoom-scale conferencing systems with in-network mixers and transcoders
  - Unifying scalable media distribution and low-latency streaming / interactive communication à la Media-Over-QUIC (MOQ)

- **Linking and aggregation**
  - File-like ICN Collections (FLIC) – be great to have one way of doing this and usable implementations
  - Collections for dynamic content: more work to do

- **Loosely-coupled coordination / consensus**
  - Sync – ready for web-scale deployment?
  - Higher-layer consensus: mapping app semantics to eventual consistency service: CRDTs – but have to do better than just layering CRDT libraries on top of Sync

- **Transactions**
  - Not everything is about accessing information objects
  - Robust client/server state evolution still needed: "online banking" use
Transactions

• Robust request-response scheme in ICN
  • Interest/Data not enough

• Secure session establishment and state evolution
  • Security context between client and server
  • Authenticated clients
  • Session concept for a series of transactions

• Data-oriented instead of connection-oriented
  • Standard encryption and signing ICN objects
  • Potential sharing of "response objects"

• Unique potential to do multi-party web interactions more efficiently
  • Individual sessions – but shared responses
Transactions with Reflexive Forwarding

• Utilize forwarder state established by Interest sent from consumer to producer
  • Allow for not just a returning Data message, but a Reflexive Interest to flow from producer to the unique consumer who sent the original Interest

• Define a scheme for Reflexive Name Prefixes
  • Can only be seen and understood by already established consumer/producer pairing
  • Do not reveal consumer identity (temporary names within the RI context)

• Provide forwarder mechanism for routing these back to consumer from producer

• Couple state of the original Interest/Data exchange with the reflexive exchange(s)
  • Ensure state gets mapped correctly by both consumer and producer
  • And unwound properly at forwarders when Data message responding to original Interest is sent back
High-Level Protocol Overview

Consumer

I1[P=P1,RNP=X1]

PIT [P=P1]

Forwarder

RI[P=X1]

PIT [P=X1]

Producer

I1 State [RNP=X1]

DR[P=X1]

D1
Previous Approach (version 01)

Consumer

Forwarder

Producer

I1[P=P1,RNP=X1]

PIT [P=P1]

RFIB [RNP=X1]

I1 State [RNP=X1]

RI[P=X1]

PIT [P=X1]

DR[P=X1]

D1
Conclusion

- **Transactions: how to realize in information-centric systems?**
  - Not about mapping existing protocols, but about actual distributed computing semantics

- **Transactions with ICN-native communication**
  - Hard to provide with basic Interest/Data
  - Reflexive Forwarding + CCNx Key Exchange + transaction semantics
  - Reflexive Forwarding needs extensions to forwarders
  - What other extensions to current Interest/Data service are needed?

- **Transactions with shared data structures**
  - Next presentation