ICN4BLOCKCHAIN

EFFICIENT BLOCKCHAIN ACCESS VIA INFORMATION-CENTRIC NETWORKING

TNO innovation for life

Dr. ir. Niels van Adrichem - niels.vanadrichem@tno.nl
Jeffrey Panneman MSc - jeffrey.panneman@tno.nl
Dr. Oskar van Deventer - oskar.vandeventer@tno.nl
20-12-2018 - https://blockchain.tno.nl

SCALABILITY ISSUES OF BLOCKCHAIN

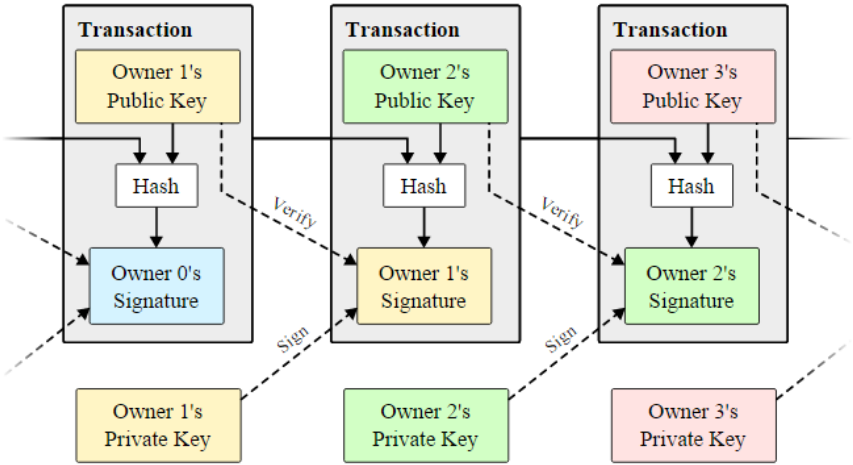
- Recently, Blockchain Technology (BCT) has gained much traction
  - Cryptocurrencies, smart contracts, decentralized registers, international banking, notary agreements
  - https://blockchain.tno.nl/blog/the-blockchain-business-case/

- BCT, however, suffers from a large scalability issue, endangering its global adoption
  - Both transaction verification and update distribution induce a large data overhead
  - The overlay peer-to-peer networks are inefficient and unaware of the underlying network, and hence cannot solve this problem independently

- We show that a specialized Content Distribution Network set up through Information-Centric Networking, in particular Named Data Networking, significantly improves this bottleneck

BLOCKCHAIN TECHNOLOGY (BCT)

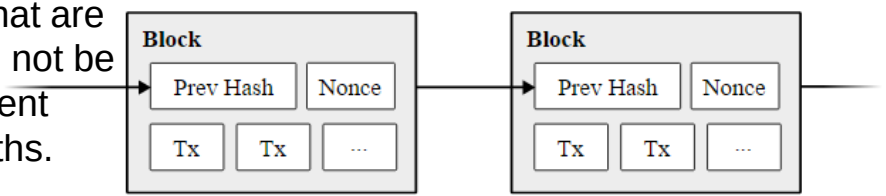
- Peer-to-peer cryptography realizing a fully-distributed architecture reliably storing and processing
  - (Crypto) currencies
  - Transactions (value or property transfer)
  - Registrations
  - Smart contracts
- Everybody can verify all transactions
- Miners validate the set of past transactions since the last block in a new block and add the new block to the public ledger (or Blockchain)



An example of transferring a crypto-coin from owner 0 to 1, 1 to 2, and 2 to 3
[Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).]

BLOCKCHAIN TECHNOLOGY (BCT)

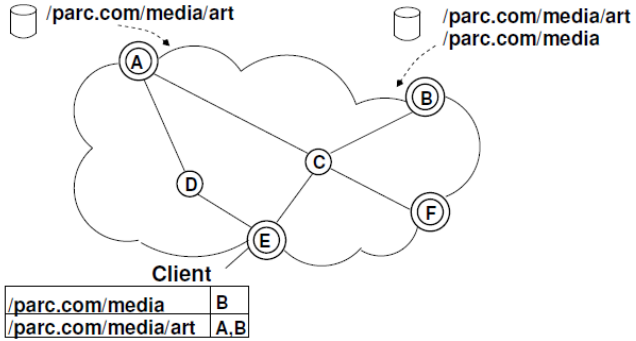
- Although considered reliable, Blockchain Technology knows several scalability issues:
- *Data overhead*: A lot of data has to be exchanged to insert transactions, acquire updates and determine consistency and authenticity of the Blockchain.
  - The Bitcoin and Ethereum blockchains are currently respectively 184 GB and > 1 TB large, a required download for an initializing node
- *Communication complexity:* Blockchains use peer-to-peer technology to interact, which is inherently inefficient.
  - P2P networks are overlay on top of IP, without taking into account the actual underlying network properties. Hence, two nodes that are considered peers in P2P, most likely will not be so in the IP underlay and data is often sent redundantly over the same links and paths.

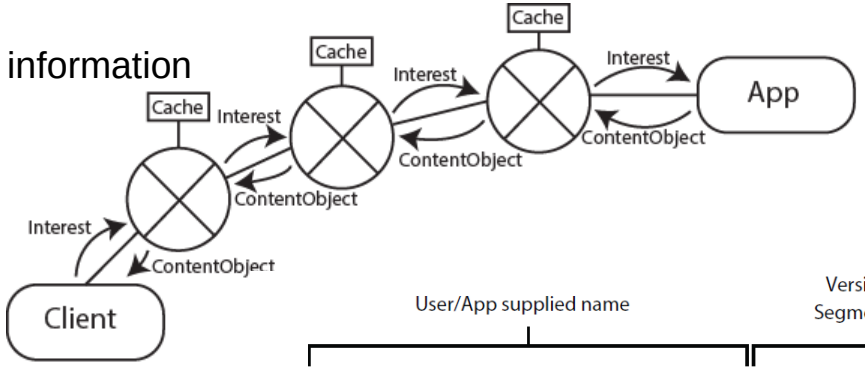


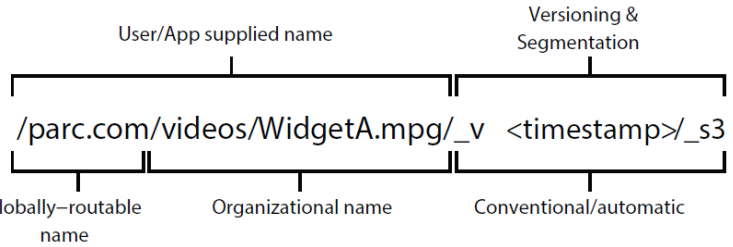
Chain of Blocks verifying all past Transactions
[Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).]

INFORMATION CENTRIC NETWORKING (ICN)

- ICN offers technology optimizing the Internet for content distribution.
- Route-and-cache by name principle through Layer 3 request for information (Interests)
- Rely on next hop to either
  - Deliver Data (from cache)
  - Send out Interest to a node closer to the information







[V. Jacobson, D. K. Smetters, J. Thornton, M. F. Plass, N. Briggs, and R. Braynard, “Networking Named Content,” *CoNEXT 2009*, 2009.]

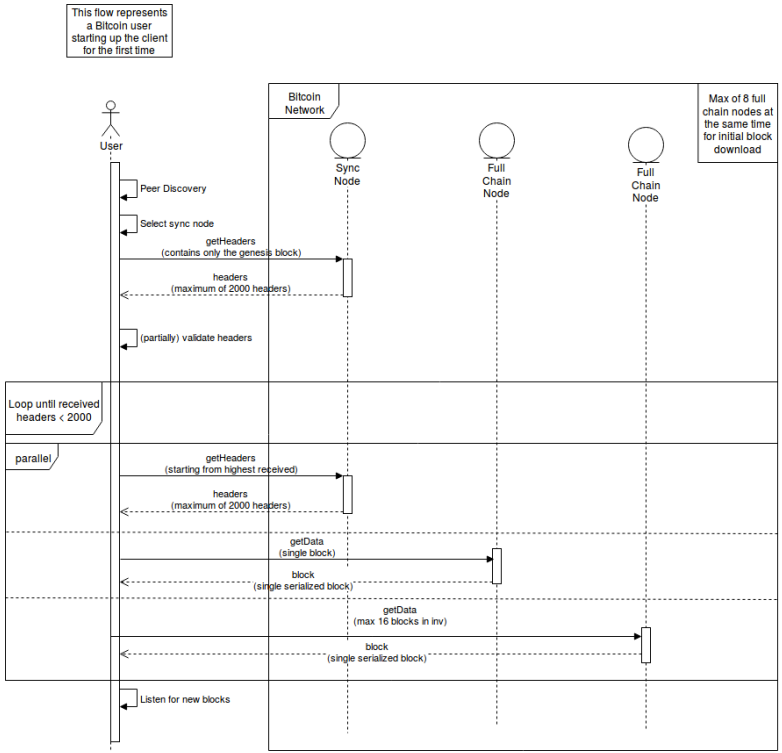
THE ICN4BLOCKCHAIN SOLUTION

- Blockchain's data overhead is essential to its reliability/authenticity determination
  Hence, it is difficult to improve without effecting Blockchain's reliability
- The derived communication overhead, however, is a typical content distribution problem
- Since ICN technology is designed to improve content distribution, we foresee this to be a natural solution to relieve this bottleneck

- With ICN4Blockchain, we have successfully
  - Designed an architecture describing the essential ICN and Blockchain interoperability functions
  - Implemented Proof-of-Concept software verifying its functionality
  - Evaluated through quantitative experimentation

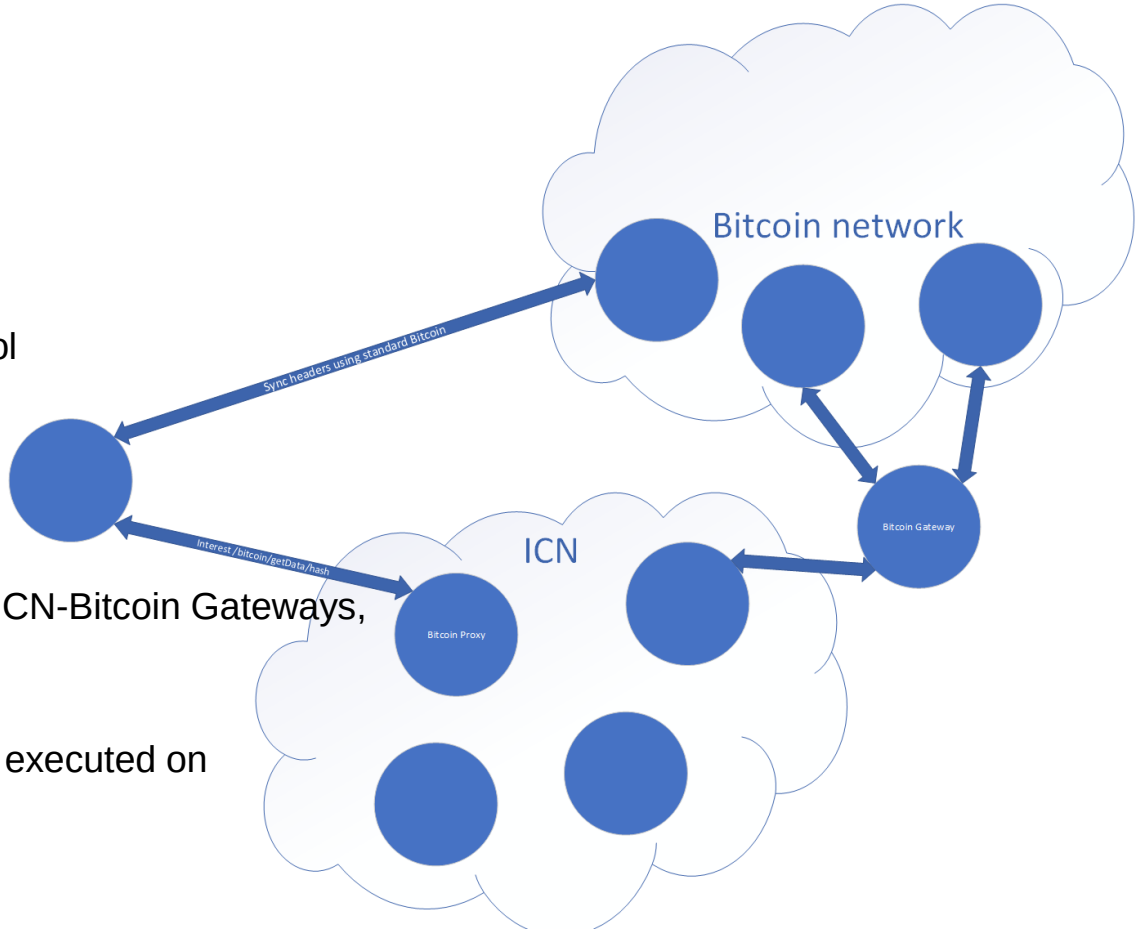
BCT SYNCHRONIZATION

- Due to its widespread use, use private Bitcoin as PoC
- Initial block download ~185 GB
- Online transaction updates core part of Bitcoin code
  - Initialization more easily modifiable
- Online transactions should also be ICNified
  - PoC can already prove our point on just the sync



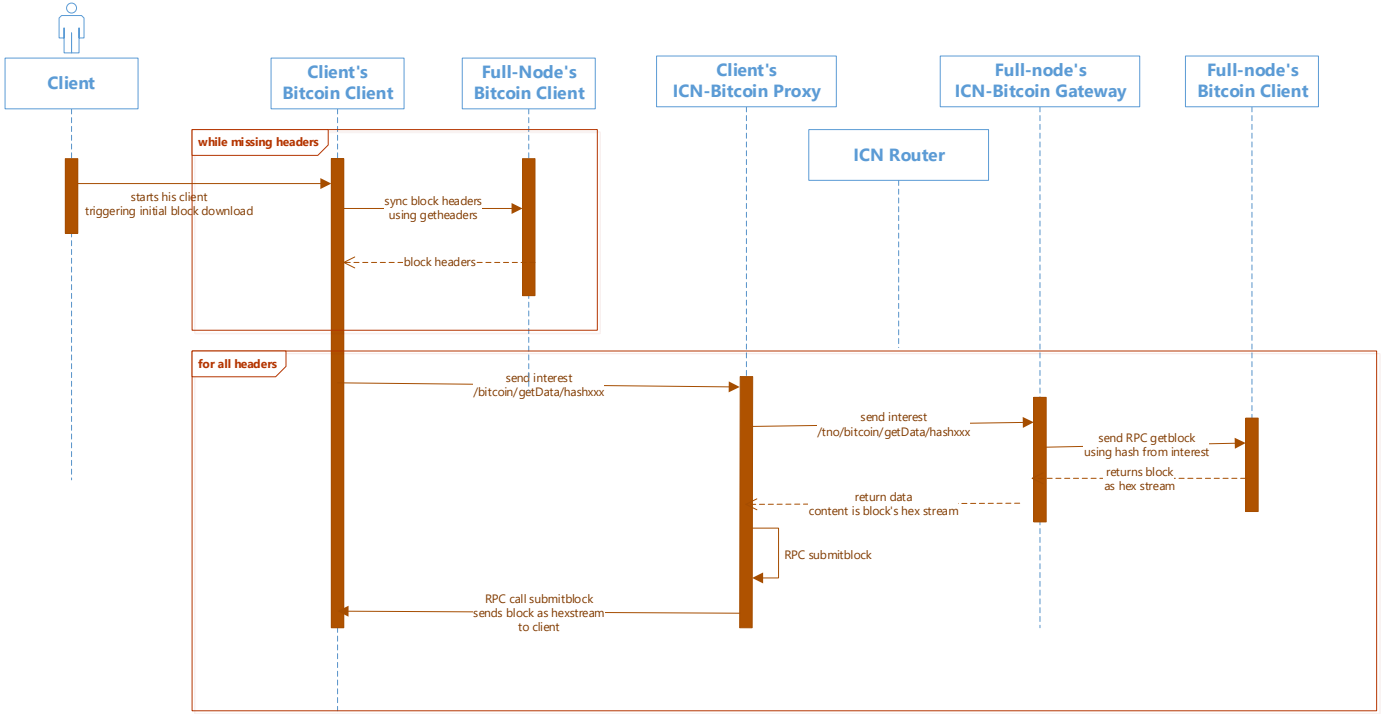
IMPLEMENTED POC

- Adapted Bitcoin+ICN client
- Download headers using Bitcoin protocol
- Download Blocks through ICN interface, connected to ICN-Bitcoin Proxy
- ICN Bitcoin Proxy forwards Interests to ICN-Bitcoin Gateways, which connect to Bitcoin Full Nodes
- Implementation efforts and experiments executed on TNO's Private Hi5 Research Cloud

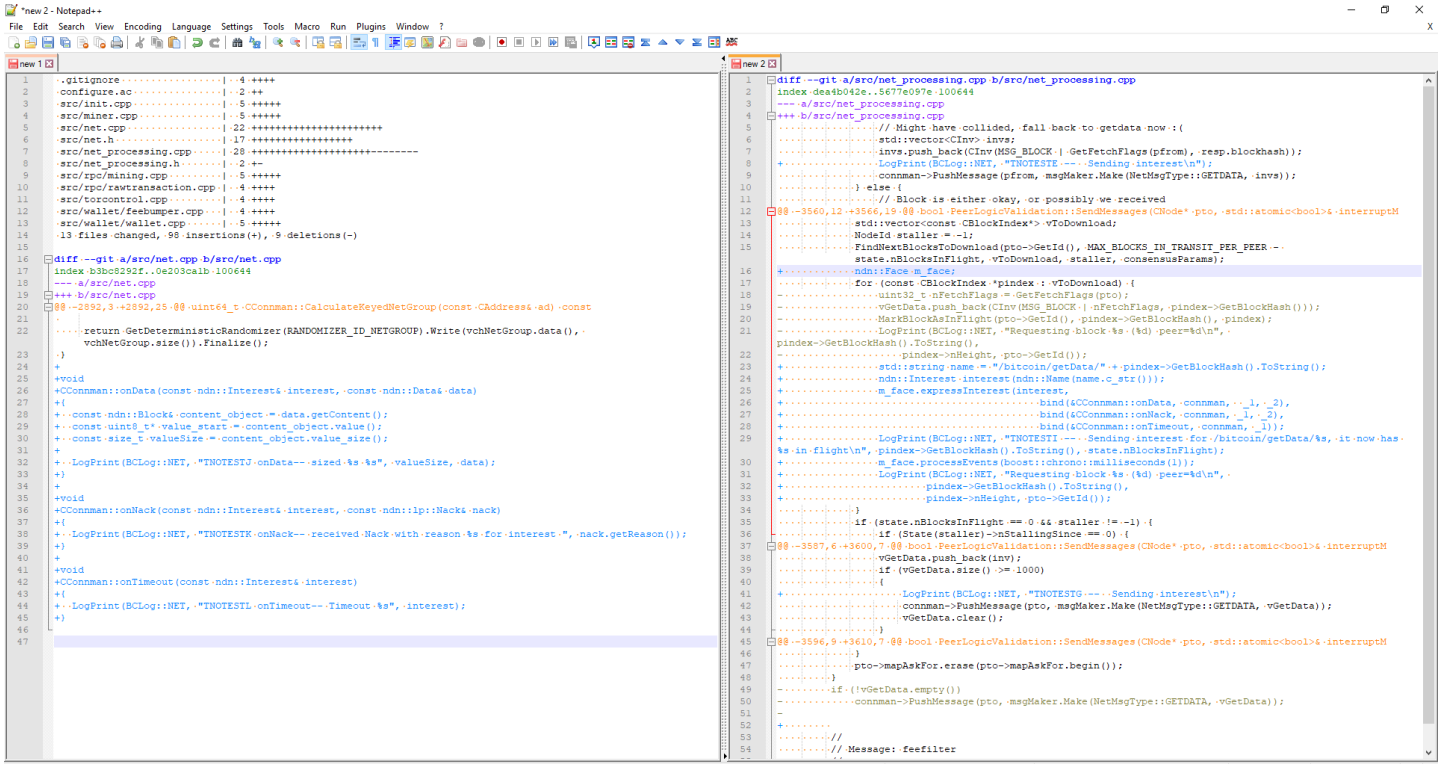


ICN SEQUENCE

- Changed the sequence of Bitcoin to download actual Blocks through Named-Data Networking

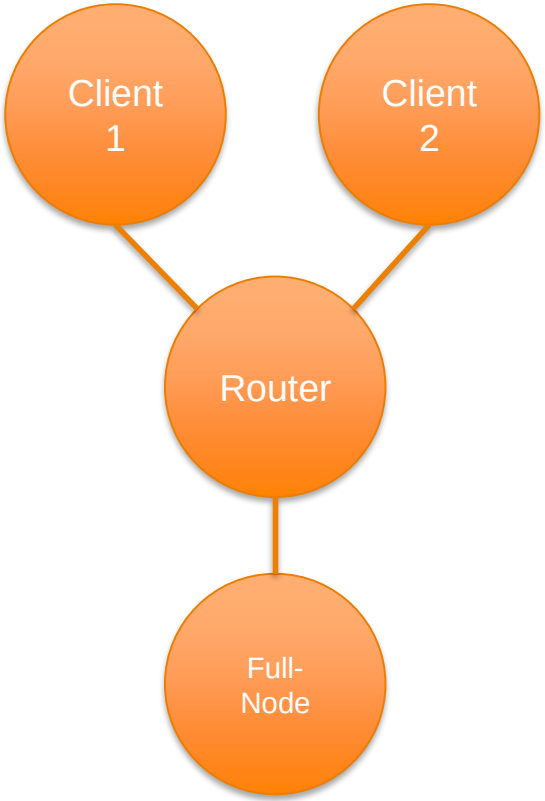


CODE POC


*new 2 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1
new 2
.gitignore | 4 ++++
configure.ac | 2 ++
src/init.cpp | 5 +++++
src/miner.cpp | 5 +++++
src/net.cpp | 22 ++++++++++++++++++++++
src/net.h | 17 ++++++++++++++++
src/net_processing.cpp | 28 ++++++++++++++++++++++--------
src/net_processing.h | 2 ++
src/rpc/mining.cpp | 5 +++++
src/rpc/rawtransaction.cpp | 4 ++++
src/torcontrol.cpp | 4 ++++
src/wallet/feebumper.cpp | 4 ++++
src/wallet/wallet.cpp | 5 +++++
13 files changed, 98 insertions(+), 9 deletions(-)
diff --git a/src/net.cpp b/src/net.cpp
index b3bc8292f..0e203ca1b 100644
--- a/src/net.cpp
+++ b/src/net.cpp
@@ -2892,3 +2892,25 @@ uint64_t CConnman::CalculateKeyedNetGroup(const CAddress& ad) const
return GetDeterministicRandomizer(RANDOMIZER_ID_NETGROUP).Write(vchNetGroup.data(), vchNetGroup.size()).Finalize();
}
+void CConnman::onData(const ndn::Interest& interest, const ndn::Data& data)
+{
+ const ndn::Block& content_object = data.getContent();
+ const uint8_t* value_start = content_object.value();
+ const size_t valueSize = content_object.value_size();
+
+ LogPrint(BCLog::NET, "TNOTESTJ onData-- sized %s %s", valueSize, data);
+}
+void CConnman::onNack(const ndn::Interest& interest, const ndn::lp::Nack& nack)
+{
+ LogPrint(BCLog::NET, "TNOTESTK onNack-- received Nack with reason %s for interest ", nack.getReason());
+}
+void CConnman::onTimeout(const ndn::Interest& interest)
+{
+ LogPrint(BCLog::NET, "TNOTESTL onTimeout-- Timeout %s", interest);
+}
diff --git a/src/net_processing.cpp b/src/net_processing.cpp
index dea4b042e..5677e097e 100644
--- a/src/net_processing.cpp
+++ b/src/net_processing.cpp
// Might have collided, fall back to getdata now :(
std::vector<CInv> invs;
invs.push_back(CInv(MSG_BLOCK | GetFetchFlags(pfrom), resp.blockhash));
LogPrint(BCLog::NET, "TNOTESTE ---- Sending interest\n");
connman->PushMessage(pfrom, msgMaker.Make(NetMsgType::GETDATA, invs));
} else {
// Block is either okay, or possibly we received
@@ -3560,12 +3566,19 @@ bool PeerLogicValidation::SendMessages(CNode* pto, std::atomic<bool>& interruptM
std::vector<const CBlockIndex*> vToDownload;
NodeId staller = -1;
FindNextBlocksToDownload(pto->GetId(), MAX_BLOCKS_IN_TRANSIT_PER_PEER -
state.nBlocksInFlight, vToDownload, staller, consensusParams);
ndn::Face m_face;
for (const CBlockIndex *pindex : vToDownload) {
uint32_t nFetchFlags = GetFetchFlags(pto);
vGetData.push_back(CInv(MSG_BLOCK | nFetchFlags, pindex->GetBlockHash()));
MarkBlocksAsInFlight(pto->GetId(), pindex->GetBlockHash(), pindex);
LogPrint(BCLog::NET, "Requesting block %s (%d) peer=%d\n",
pindex->GetBlockHash().ToString(),
pindex->nHeight, pto->GetId());
std::string name = "/bitcoin/getData/" + pindex->GetBlockHash().ToString();
ndn::Interest interest(ndn::Name(name.c_str()));
m_face.expressInterest(interest,
bind(&CConnman::onData, connman, _1, _2),
bind(&CConnman::onNack, connman, _1, _2),
bind(&CConnman::onTimeout, connman, _1));
LogPrint(BCLog::NET, "TNOTESTI ---- Sending interest for /bitcoin/getData/%s, it now has %s in flight\n", pindex->GetBlockHash().ToString(), state.nBlocksInFlight);
m_face.processEvents(boost::chrono::milliseconds(1));
LogPrint(BCLog::NET, "Requesting block %s (%d) peer=%d\n",
pindex->GetBlockHash().ToString(),
pindex->nHeight, pto->GetId());
}
if (state.nBlocksInFlight == 0 && staller != -1) {
if (State(staller)->nStallingSince == 0) {
@@ -3587,6 +3600,7 @@ bool PeerLogicValidation::SendMessages(CNode* pto, std::atomic<bool>& interruptM
vGetData.push_back(inv);
if (vGetData.size() >= 1000)
{
LogPrint(BCLog::NET, "TNOTESTG ---- Sending interest\n");
connman->PushMessage(pto, msgMaker.Make(NetMsgType::GETDATA, vGetData));
vGetData.clear();
}
@@ -3596,9 +3610,7 @@ bool PeerLogicValidation::SendMessages(CNode* pto, std::atomic<bool>& interruptM
}
pto->mapAskFor.erase(pto->mapAskFor.begin());
}
if (!vGetData.empty())
connman->PushMessage(pto, msgMaker.Make(NetMsgType::GETDATA, vGetData));
//
// Message: feefilter
Diff file
length : 3.369 lines : 56
Ln : 16 Col : 31 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

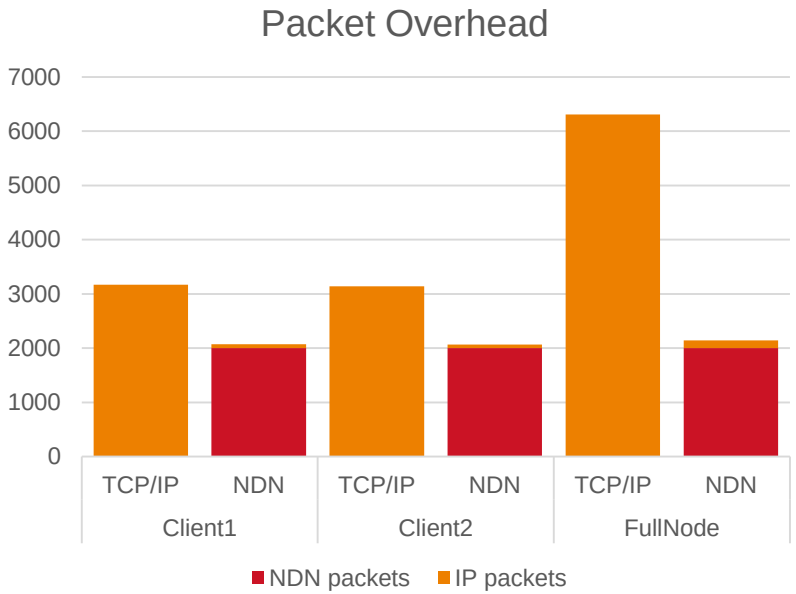
EVALUATION POC

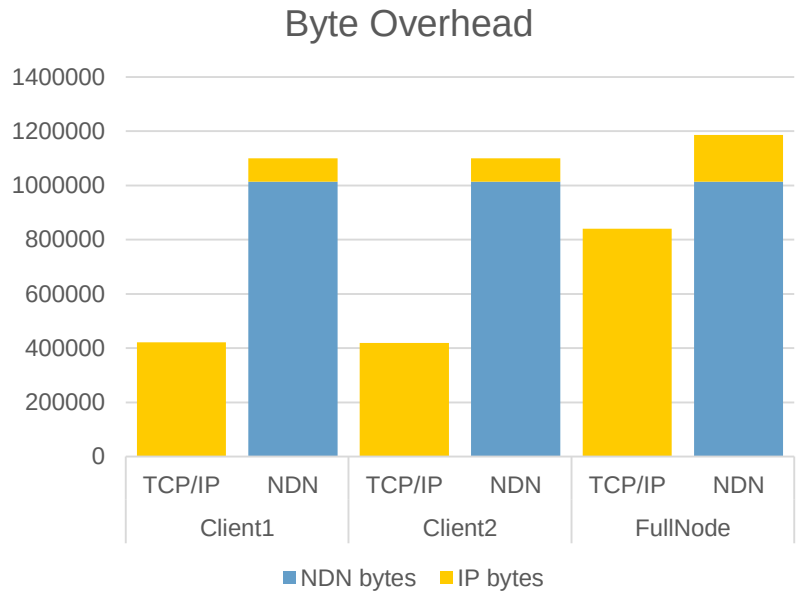
- Classic Bitcoin with TCP/IP scenario
  - Clients 1 and 2 run regular Bitcoin clients
  - Router performs IP routing
  - Full-node is regular Bitcoin full node
- Bitcoin with ICN scenario
  - Clients 1 and 2 run ICN-Bitcoin clients and local ICN-Bitcoin Proxies
  - Router performs ICN caching and forwarding
  - Full-node runs ICN-Bitcoin Gateway and regular Bitcoin full node
  - ICN implies using Named Data Networking (NDN) over Ethernet
- 100 iterations of Initial Block Download
  - Measure OSI Layer 3 (both IP and NDN) communication overhead



INITIAL MEASUREMENTS

- Measured IP and NDN packets transmitted and received on different hosts
- Some IP overhead in Bitcoin with NDN exists due to Bitcoin header exchange



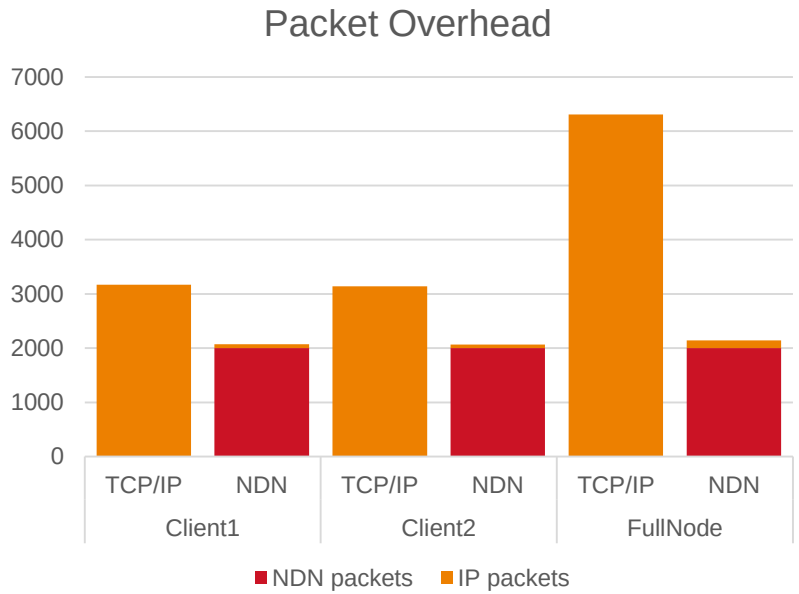


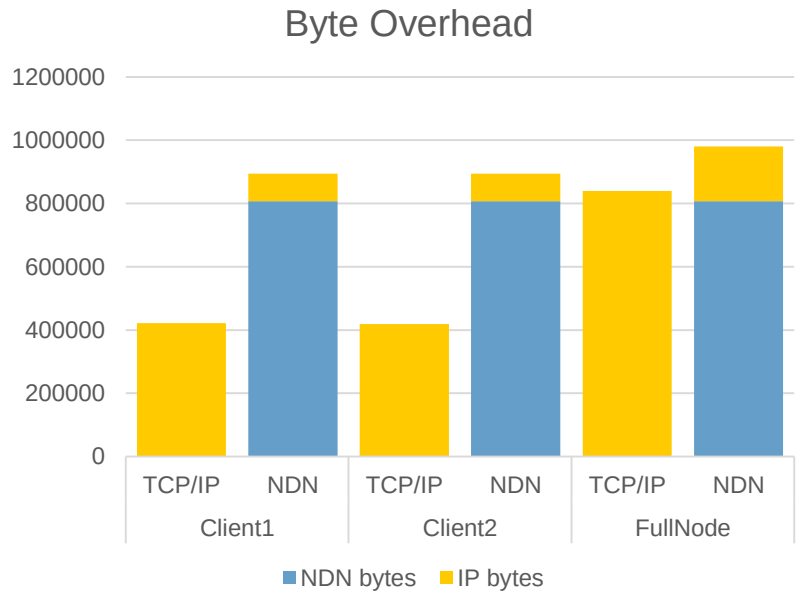
INITIAL MEASUREMENTS

- Great improvement in number of transmitted packets
  - Switch performance bottleneck principally upper bound and measured by Millions of Packets Per Second (MPPS)
- However, total transmitted bytes increased
  - Additional analysis shows plenty room for further improvements

IMPROVEMENT 1: BLOCK LINE ENCODING

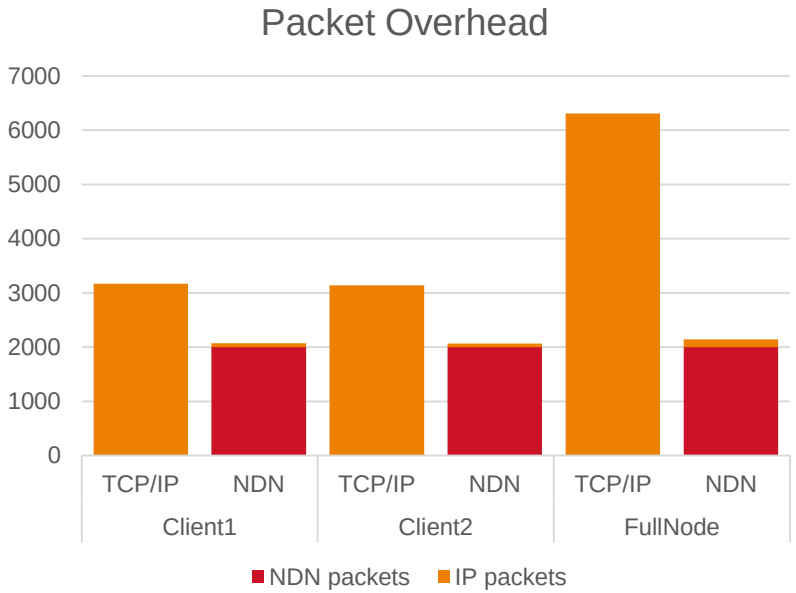
- Initially, we used the encoding of blocks into Bitcoin's RAM, instead of the line encoding of blocks which is more compact

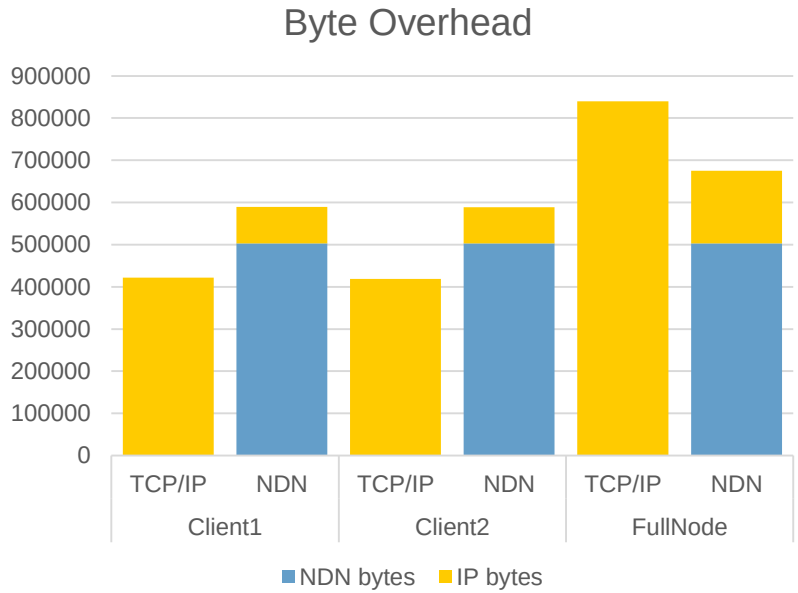




IMPROVEMENT 2: REMOVE REDUNDANT SIGNATURES

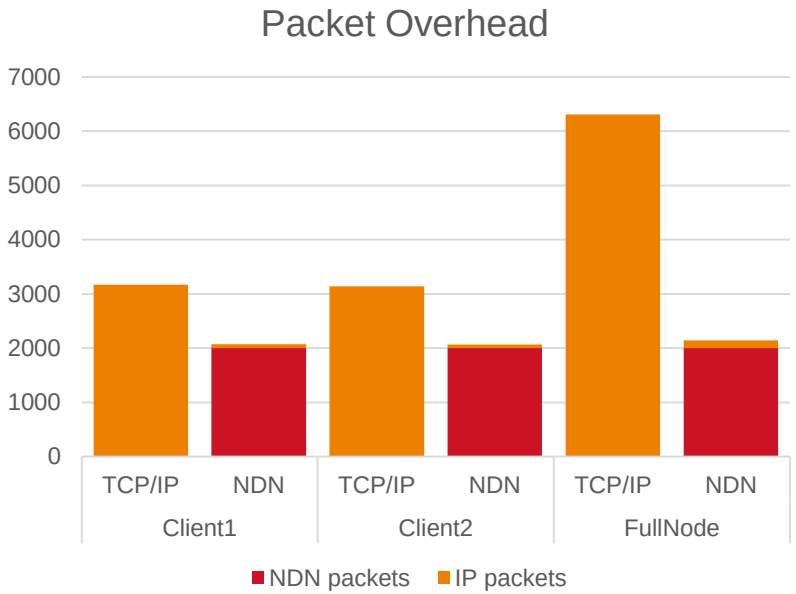
- Block identifier is self-authenticating hash, hence, we don't need the NDN signature
  - Transmitted bytes already improved for Full-Node at only 2 clients

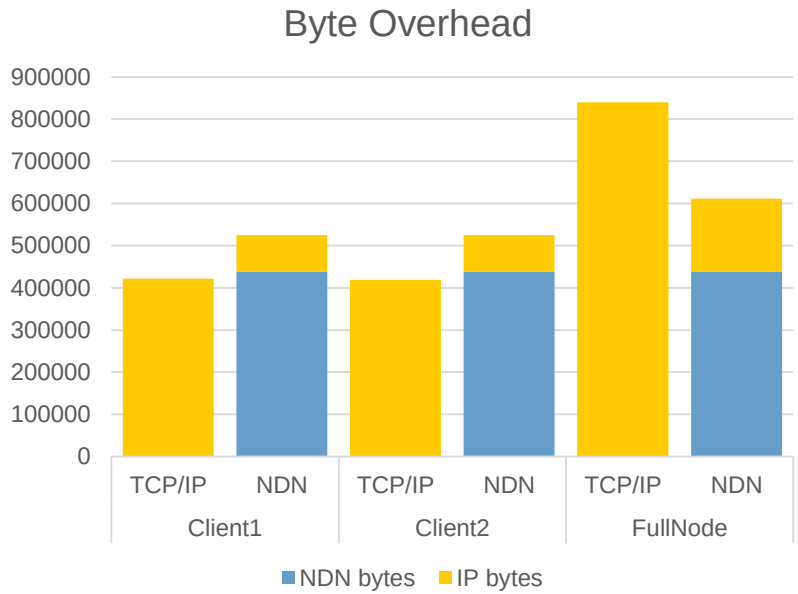




IMPROVEMENT 3: ENCODING OF BLOCK ID

- Using the 32-byte hexadecimal encoding of a Block Id instead of its character string representation in NDN names, further improves byte overhead with 32 bytes per NDN packet





ADDITIONAL IMPROVEMENTS

- We used very small block sizes of the minimum of 250 bytes because we lack actual transactions
  - However, the maximum and current 90-day average block size in public Bitcoin equal respectively 1 MB and 750 KB
  - Larger block sizes imply a much lower relative NDN packet overhead, hence we expect higher performance using Blockchains which contain actual transactions

- Where we retrieve 1 block with each request, Classic Bitcoin over TCP/IP retrieves up to 2000
  - Retrieving multiple blocks with each Interest will further improve Bitcoin over NDN

CONCLUSIONS

- Already in a simple 2-Client and 1-FullNode evaluation the number of packets significantly decreases for the Initial Block Download on all nodes.
  - This dimension is the bottle neck for switching and router logic
- Although Bitcoin Clients experience a small penalty on transmitted bytes, load on Full Nodes significantly reduces
- NDN is suitable to relieve the communication overhead of Blockchain Technology

FUTURE WORK

- Include online transaction updates and verification
- Extend testbed to a larger network of clients and full nodes
- Include other Blockchain Technologies
- Use (part of) public Blockchain(s) to improve data representability
- Find optimal transmission of multiple blocks for a single getData request to improve header overhead
- Disseminate results further in standardization (IETF, ICNRG), market (Techruption, Dutch Blockchain Coalition)

GLOSSARY

- BCT: Blockchain Technology
- Bitcoin: A specific BCT implementation (OSI Layer 5 to 7)
- Ethereum: Another specific BCT implementation (OSI Layer 5 to 7)
- Ethernet: Networking technology used in Local Area Networks (OSI Layer 2)
- ICN: Information-Centric Networking
- IP: Internet Protocol, the networking layer of the Internet (OSI Layer 3)
- NDN: Named Data Networking, a specific ICN architecture (OSI Layer 3)
- OSI: Open Systems Interconnection model, defining the conceptual layers of communication functions in telecommunication networks
- P2P: Peer-to-peer network, a computing or networking distributed application architecture
- PoC: Proof of Concept
- TCP: Transmission Control Protocol, a transport protocol offering reliable pseudo-connections over IP (OSI Layer 4)