IPoC: IP over CCN for seamless mobility

Greg White, CableLabs

ICNRG Interim Meeting

Nov. 13, 2016

Background

- ICN seems attractive for mobile networking
  - Elegant consumer mobility via stateful forwarding
  - Multipath connectivity managed by the mobile device
  - In-network caching and processing
- How do we get there?
  - Network slicing? – and run two networks in parallel?
  - ICN over IP? - and lose the benefits above?

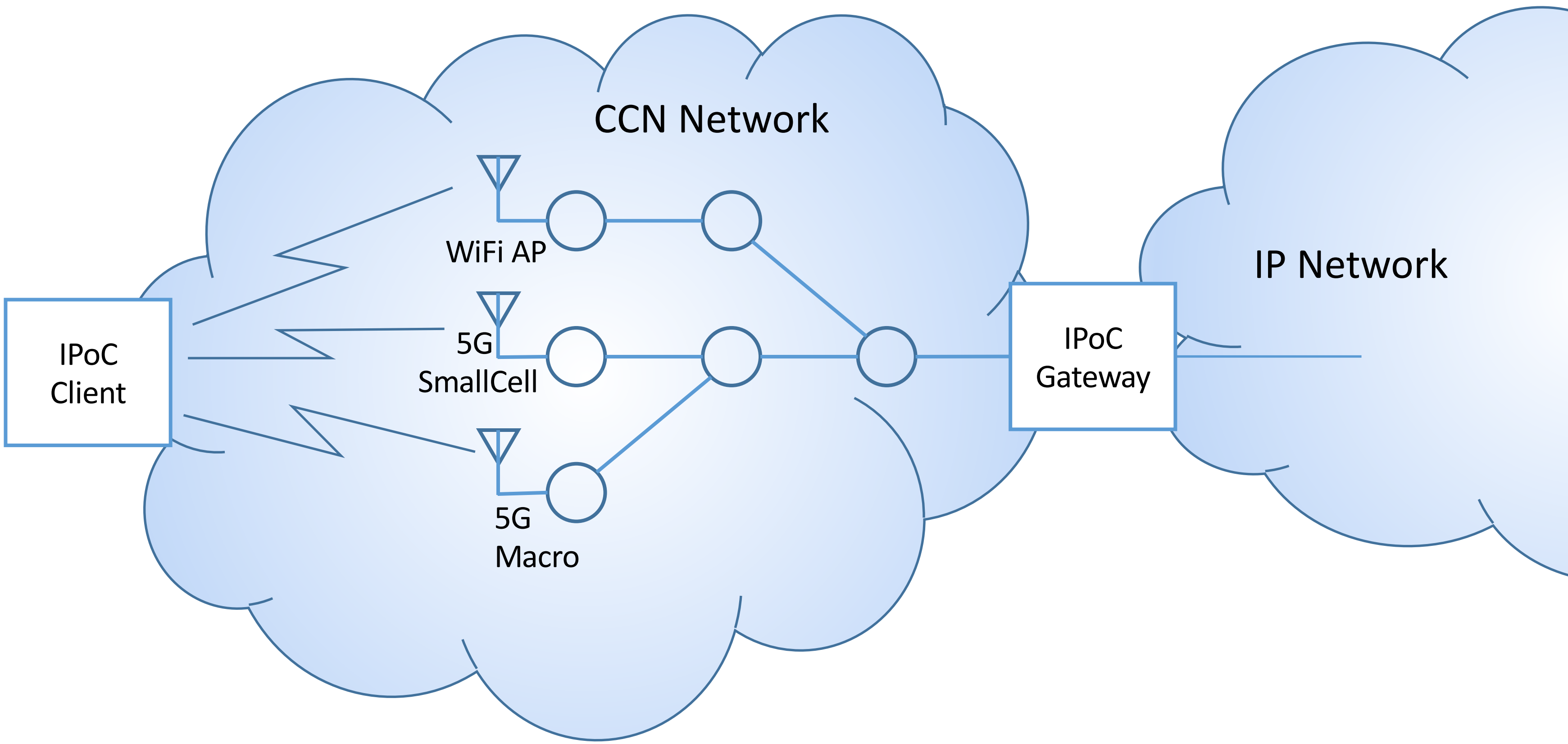
Concept

- Explore the idea of using CCN as THE forwarding plane for 5G
- Support existing IP services via an “IP over CCN” protocol – replacing LTE-EPC (GTP Tunnels) for IP Mobility
- Enable deployment of native CCN applications, preserving the benefits

IP over CCN (IPoC) Goals

- Support all existing IP applications & transports without modification
  - Incl. TCP, UDP, SCTP, DCTCP, QUIC, BBR, etc.
  - …maybe not IP multicast.
- No change to IP stacks
- Leverage consumer mobility of CCN
- Support multipath connectivity
- High performance
- Low overhead
- Be a compelling replacement for EPC

Architecture



Leverage consumer mobility

- IPoC Client only sends Interest messages
  - “upstream” IP packets carried as Interest payloads

- IPoC Gateway responds with Content Objects containing “downstream” IP packets

IPoC Naming Convention

- ccnx:/ipoc/<hex_ipaddr>/<b64_seq>
- `hex_ipaddr:` Client IP address
  - for IPv4: four name segments each encoding (in hex) an octet of the IP address.
    - 192.168.1.100 -> “`c0/a8/1/64`”
  - for IPv6: RFC2737-sec.2.2,para.1 encoding, with colons replaced with name segment delimiters
    - 3ffe:1900:4545:3::fe21:67cf -> "`3ffe/1900/4545/3/0/0/fe21/67cf`"
- `b64_seq:` Interest Sequence Number
  - base64-encoded, monotonically increasing (with rollover)

CCN Routing

- Each IPoC Gateway on the CCN network supports connectivity and address assignment for one or more IP subnets.
- Each IPoC Gateway advertises routes within the CCN network for:
  - `ccnx:/ipoc/<ip4prefix>`
  - `ccnx:/ipoc/<ip6prefix>`
  - `ccnx:/ipoc/init`

Multiple prefixes can be advertised

IP Address Assignment

- Client sends Interest for:
  `ccnx:/ipoc/init/<nonce>`

- CCN network routes Interest to nearest IPoC Gateway

- Gateway responds with Content Object containing IP address configuration information (i.e. the DHCPv4 / DHCPv6 information)

“Upstream” IP packet handling

- Client: Upon receipt of one or more IP packets from the local stack:
  - Send an Interest message
    - Name formed by client’s IP address and next sequence number
    - Body contains entire IP packet(s)
- Gateway: Upon receipt of an Interest message
  - De-encapsulate IP packet(s) and add to resequencer for forwarding to IP network
    - Resequencer ensures in-order delivery
  - Add Sequence Number to the “Client Interest Table”

Client Interest Table (CIT)

- The CIT is a FIFO queue maintained by the gateway
- CIT contains received Interest Sequence Number and Arrival Time tuples
- One CIT per active client IP address

“Downstream” IP packet handling

- Gateway:
  - Arriving IP packets are queued on a per-client-IP basis*
  - Queues are serviced in a round-robin manner
  - Queue blocks when its CIT is empty
  - Packet(s) are dequeued to form a Content Object
  - CIT entry is dequeued to form CO name
  - CO includes a CO Sequence Number (monotonically increasing, with rollover)
    - CO Sequence Number space is independent of Interest Sequence Number space
- Client: Upon receipt of a Content Object
  - De-encapsulate IP packet(s) and add to resequencer for delivery to IP stack

*more sophisticated queuing, e.g. fq_codel could also be used

Maintaining the CIT

- To avoid introducing downstream latency, the CIT needs to always contain at least one sequence number, ideally more than one in order to support a burst of downstream traffic

- Gateway has a target number of CIT entries that it seeks to maintain during idle conditions: min_CIT (e.g. 10)
  - This could be dynamically adjusted based on traffic expectations

- Client maintains an Interest Deficit Count
  - Upon CO arrival, Client increments IDC
  - Upon Interest transmission, Client decrements IDC
  - If IDC > 0, Client sends an “empty” Interest (no payload) – paced
  - IDC is bounded as: -IDC_limit <= IDC <= IDC_limit (e.g. 5)

Managing In-flight Count

- Gateway can trigger an adjustment of the number of in-flight Interest Sequence Numbers (and hence CIT size)
  - Interest Deficit Report (IDR) included in Content Object
  - Allowed IDR values: -1, 0, 1
  - Client adds IDR value to its Interest Deficit Count

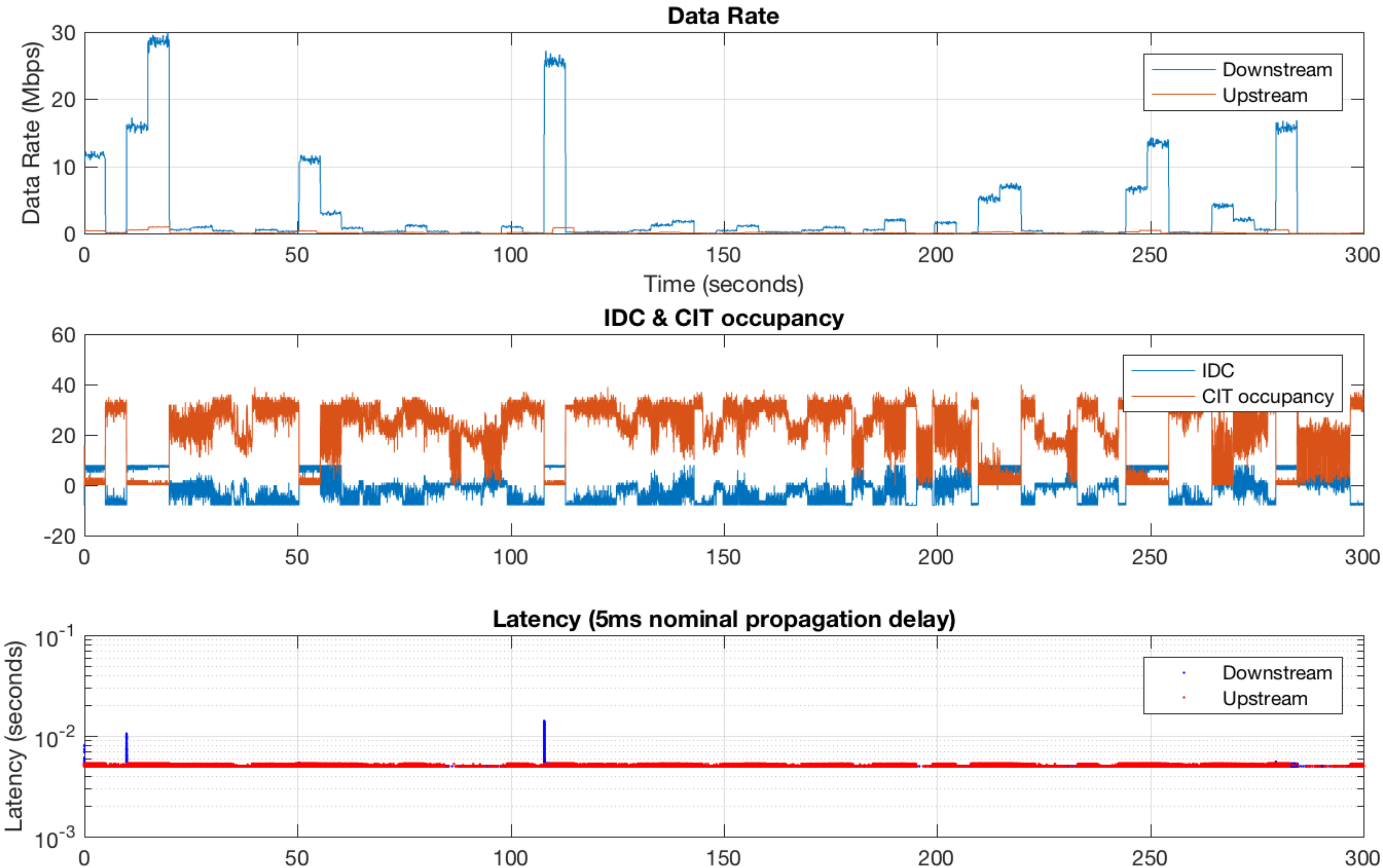
- IDR rules
  - Upon receipt of an Interest when the corresponding CIT is full
    - Gateway dequeues the head of CIT and sends CO with IDR = -1
  - When transmitting a CO, if the CIT size < min_CIT:
    Send IDR = 1

PIT Entry Lifetimes

- It is expected that PIT entries in intermediate nodes will have finite lifetimes, e.g. 300ms
- Gateway calculates a CIT Lifetime after which it considers a CIT entry to be stale
- When head-of-queue CIT entry times out, Gateway sends an “empty” Content Object
  - If CIT size < min_CIT: IDR = 1
  - If CIT size == min_CIT: IDR = 0
  - If CIT size > min_CIT: IDR = -1

If no traffic, this drains the CIT down to ~min_CIT entries

Matlab Simulation



Linux implementation

- Implemented using Metis forwarder, using 32K max message size
- IPoC Client and Gateway processes installed as “tun” devices (similar to a VPN interface)
- Tested in small lab network configuration (1 client, minimal RTT)
- Tested using Netperf and speedtest.net
- IPoC params:
  - Max CIT = 63 entries
  - min_CIT = 10
  - IDC_limit = 6
- Efficiency (total IP bytes sent / total bytes sent):
  - Downstream: 98.9%
  - Upstream: 99.4%
- 99.9th Percentile Latency
  - Downstream: 8.4ms
  - Upstream: 14.9ms

Areas for Future Work

- NS3 simulation
- Evaluate performance impact of parameter values
- Security considerations (signed Interest payload?)