What is In-Network Computing (INC) in the light of and alongside E2E?

Continued

Jeffrey He, Ike Kunze, Dirk Trossen

This slide deck condenses thoughts collected during IETF 115 and the ensuing discussion on the COINRG mailing list. You can find a list of original contributors on the last slide.

Nov. 2023

Intention

- This slide deck tries to condense the discussions at IETF 115 and on the mailing list and intends to inspire thoughts and discussions on the INC / E2E topic

- The immediate goal is to find out which statements see common agreement and where opinions start to differ

- The ultimate goal is to make a judgement whether it is reasonable to bring an INC vision to the real world

- Common abbreviations
  - INC: In-Network Commputing
  - E2E [p|a]: End-to-end [principle|argument]

Opinions: Mental models for INC

- INC is about creating "some kind of 'wire with **bumps** in it' with packet transfer emulating a wire”.
  - Still use the notion of "**e2e session**" but extend the function towards execution in network nodes
  - Analogy: multicast (“function” = copy in branching points)
  - ***BUT: can function be more than the network’s core role, namely to forward packets, i.e., more app-specific?***

- INC is about pushing innovation into the network
  - …without coupling network innovation cycles tightly with application innovation

- INC is an e2e session stitched together from multiple end-to-intermediate or intermediate-to-intermediate connections.
  - Still use the overall notion of "e2e session”, but split it up

Opinions: What is INC? – (Functionality View)

- Possible **groups of functions**
  - (1) execution environment in the network that can execute nearly any function
  - (2) "atomic" computing functions inside the network that can be used by many applications
  - (3) application-specific functionality into the network.

- Is (1) a realization of (2) and (3)? Is (3) a more specific version of (2)?

- Should INC aim for supporting functions that are less application-specific (rather (2))?
  - Where to draw the line?
    - Only “core networking tasks”?
    - Where does application logic start?

- WHAT and HOW many functions f_1, …, f_n MAY, SHOULD, or MUST a INC network have?

Opinions: What is INC? – (Layer View)

- Are operations on transport layer and above even possible with increasing e2e encryption?
  - INC as a network layer technology?
  - Or are in-network functions just **application (sub-)functions**?

- Implications of having INC reside on a specific layer at the example of routing
  - INC L4 elements: overlay-based routing might take the computational load of the COIN elements into account – no changes to L3 routing.
  - INC L3 elements: L3 routing probably needs to take the load into account, which also complicates the whole system as there might be complicated trade-offs that are better solvable with application knowledge.

Opinions: What is INC? – (Statefulness View)

- Another core principle/argument of the Internet: *fate-sharing*
  - a packet conveying enough information can be forwarded without requiring state inside the intermediate systems
- Distinguish between
  - requiring application-specific functionality
  - statefulness/statelessness
- Does INC allow for statelessness inside the data or control plane during forwarding?
  - If not, fate-sharing might not hold anymore
- Categories regarding required state
  - Non-stateful generic functionality = best?
  - Non-stateful, application-specific functionality = somewhat okay?
  - Stateful generic functionality = somewhat okay?
  - Stateful, application-specific functionality = bad?

Designing an INC System

A New Problem or a Rehash of Known Thinking?

Harking back ~40 Years in Internet History

[E2E] states:

The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)

-> the original Internet design argumentation foresaw (even suggested) COIN as a method for realizing communication systems & their applications!

[E2E] End-to-End Arguments in System Design, J.H. Saltzer, D.P. Reed and D.D. Clark, ACM Transactions in Computer Systems 2, 4, November, 1984, pages 277-288., http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf

Opinions: Intention of the E2E [P|A] (1/2)

- The E2E [p|a] provides **guidance** regarding the relationship between applications and the network.

- The E2E [p|a] is a design principle that follows the simplicity principle (KISS) by suggesting to avoid placing application-specific functionality into the network yet **not disallowing** it either!

- However, the E2E [p|a] **equally** provides guidance that (part of) application functions may well reside in the network
  - The E2E [p|a] does not explicitly include **E2E transparency** when placing app functions into the network.

- The E2E [p|a] is more relevant for networks conveying many applications than for single-purpose / mission-specific networks.
  - Decoupling of network from application innovation to allow for **permission innovation** to happen

- The E2E [p|a] is not an absolute law, but has been enshrined as part of regulations worldwide

Opinions: Intention of the E2E [P|A] (2/2)

- The E2E [p|a] is heavily inspired by the perception that changing the hardware in the network infrastructure is **slow and difficult**.

- Adhering to the E2E [p|a] **improves robustness** by not relying on network functionality to exist.

- The E2E [p|a] guides the formation of ecosystems of network and application providers, where collaborations between those with specific ROI intentions are required for the resulting system to perform as intended.

- The gain for applications using functionality added to the network should outweigh the cost for applications not using it

Opinions: What is Missing in E2E to Make INC Work?

*The key issue is how can we **transparently** and **trustworthily** insert as well as instruct the execution of a partial application function in-between application endpoints, i.e., in (various) network locations?*

- In other words, if we assumed possible sub-functions $f_1,\ldots, f_n$ in the network, how do I build E2E applications in such environment?
  - **Simplicity** and **transparency** may be good guidance points
  - **Transparency**: you ought to know that function f_i is being executed in the network for the endpoints to perform their functions properly

    -> with transparency comes **explicitness** over choice of time of execution as well as location of and relation to execution point
  - **Simplicity**: pushes you for carefully considering your number *n* for the in-network functions you may want.
- What could be an execution environment aiming for such explicitness and transparency?
  - Semantic networking vision https://conferences.sigcomm.org/sigcomm/2022/files/workshop-fira/FIRA-Slot2+Arch-Keynote-Luis.pdf outlines an architectural framework

Asking the Right Questions to Move Forward

Solicit Input

- **Research problems, for instance**
    - (1) define the functions implemented in the programmable network and justify them
    - (2) design a solution to overcome the obstacle from the end2end encryption.
    - (3) design a solution to handle the network layer and transport layer appropriately: routing, reliability, congestion resolution/rate control
- **Questions to be asked**

Original contributors

- Roland Bless
- Toerless Eckert
- Jeffrey He
- Ike Kunze
- Dirk Kutscher
- Erik Nordmark
- David Oran
- Andy Reid
- Dirk Trossen