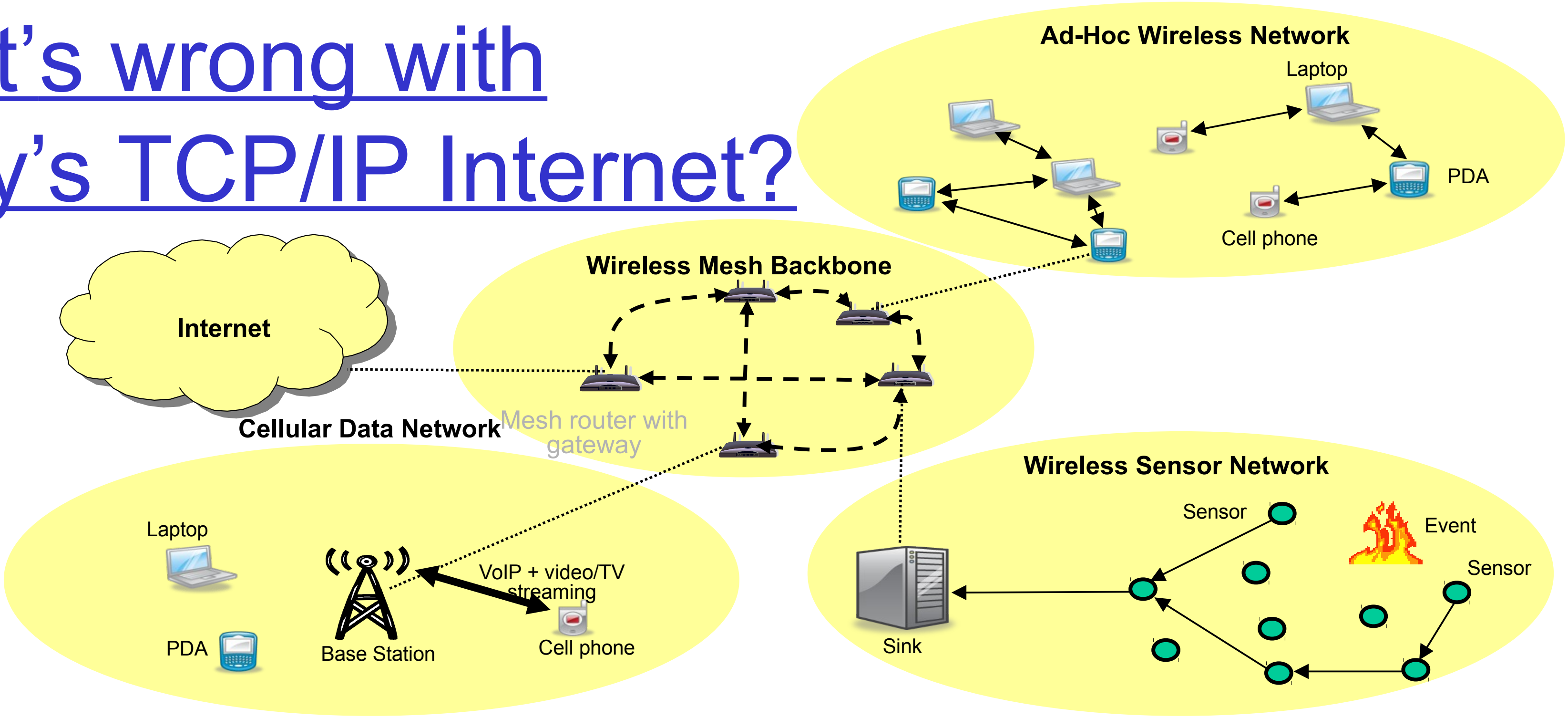
Assessing the Security of a Clean-Slate Internet Architecture

Security as Byproduct of Decoupling Different Concerns

Gowtham Boddapati John Day
Ibrahim Matta Lou Chitkushev
Boston University

Seventh Workshop on Secure Network Protocols (NPSec), October 30, 2012, Austin, Texas

What's wrong with today's TCP/IP Internet?



- The new brave world
  - Larger scale, more diverse technologies
  - New requirements: **security**, mobility, competition, …
- Custom point-solutions: No or little "science"
- Lots of problems: **security attacks**, bad performance, hard to manage, …

Questions?

- Is the Internet’s architecture fundamentally broken that we need to “clean slate”?

    Yes

- Can we find a new architecture that is complete, yet minimal? If so, what is it?

    RINA (Recursive InterNetwork Architecture)?

    Based on a Distributed Inter-Process Communication (IPC) model

    With NO consideration for security!

R
I
N
A Better Network Structure

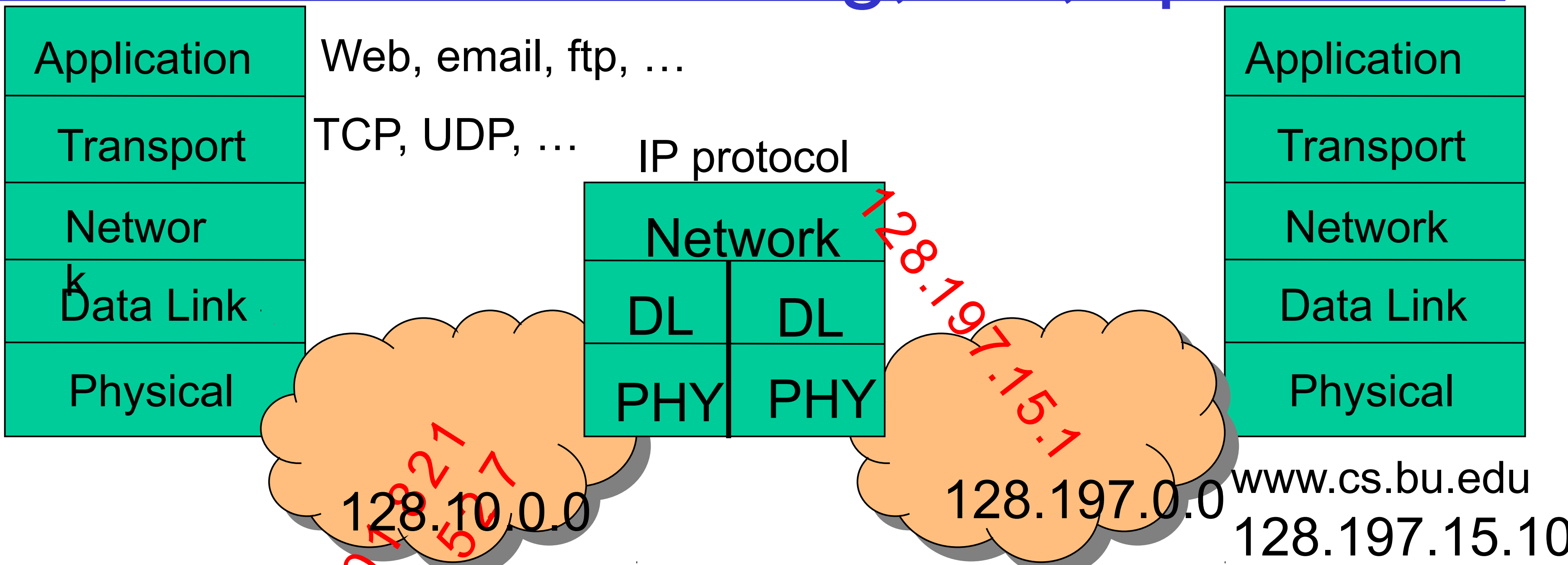
- Can we transition to it without requiring everyone to adopt it?

    Yes

We show in this paper…

- Without crypto support, RINA can resist security transport-level attacks faced by TCP/IP
- RINA decouples *authentication* from *connection management*
  - Limiting attacks to "insider" attacks
- RINA decouples *port allocation and access control* from *data synchronization and transfer*
  - Making attacks harder to mount
- We analyze how hard it is to compromise RINA given typical field lengths in packets
- *Good security as byproduct of good design!*

Internet's view: one big, flat, open net



- There's no building block
- The "hour-glass" model imposed a least common denominator
- Either didn't name what was needed or named the wrong things (i.e., interfaces)
- We exposed addresses to applications
- We hacked in "middleboxes"
- …

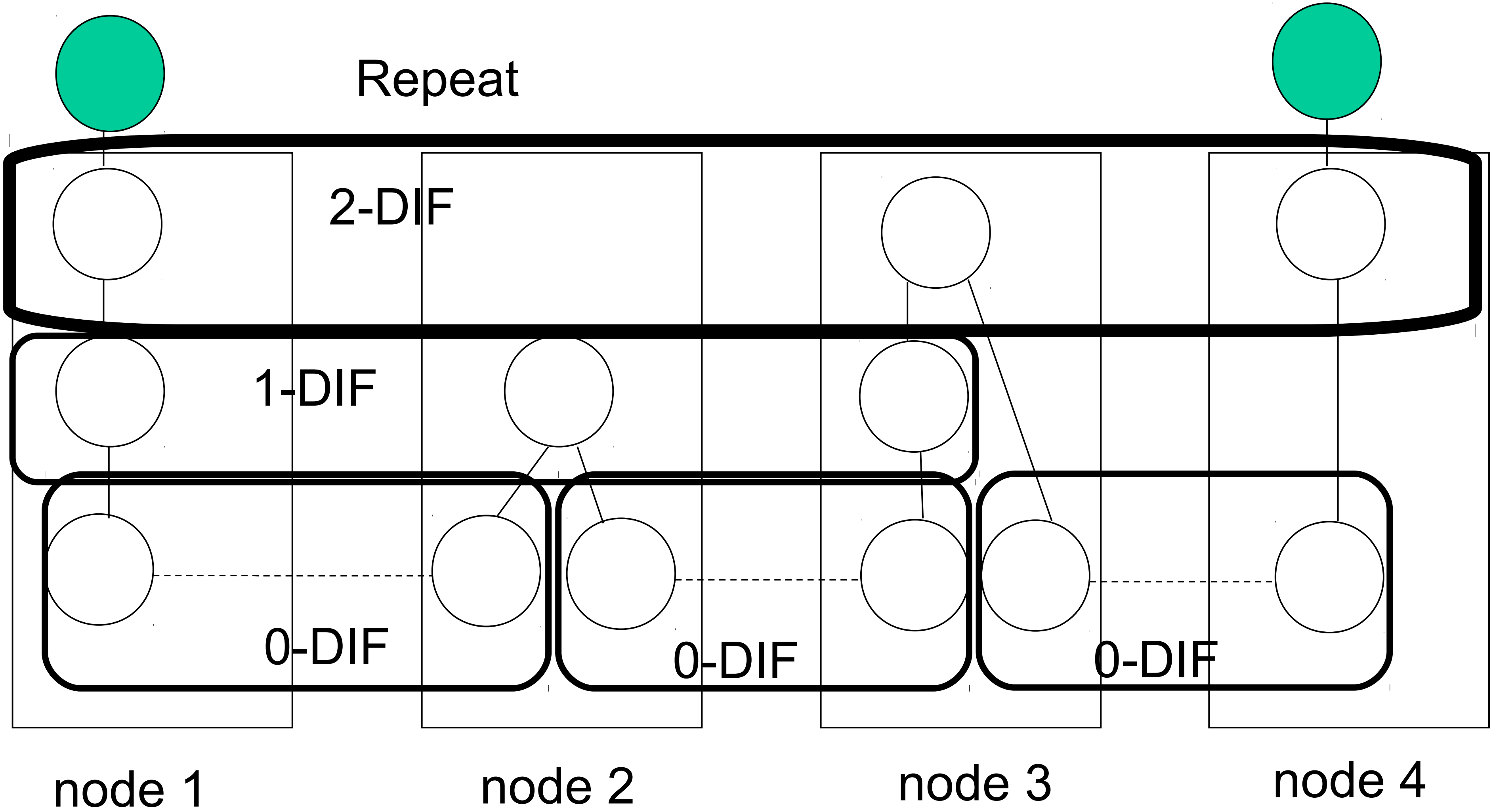
Our Solution: divide-and-conquer

- ❑ Based on going back to fundamentals
- ❑ Application processes communicate over Distributed IPC Facility (DIF)

    A distributed application that does IPC
- ❑ DIF management is hidden ➔ better security
- ❑ IPC processes are application processes of DIF's
- ❑ Recurse as needed

    ➔ better manageability & scalability
- ❑ Well-defined interfaces ➔ predictable service

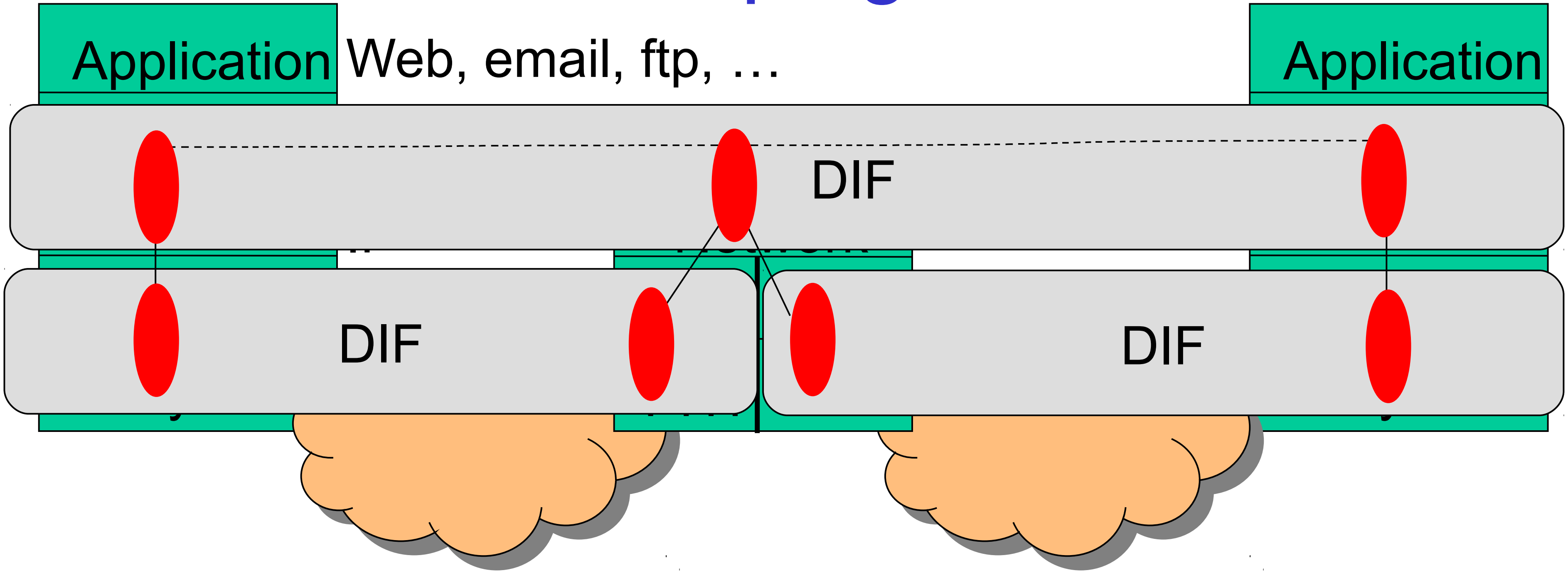
Recursive Architecture based on IPC



DIF = Distributed IPC Facility (locus of shared state=scope)

Policies are tailored to scope of DIF

RINA allows scoping of services



- The DIF is the building block and can be composed

    A DIF has all what is needed to manage a "private" network, i.e. it integrates routing, transport and management

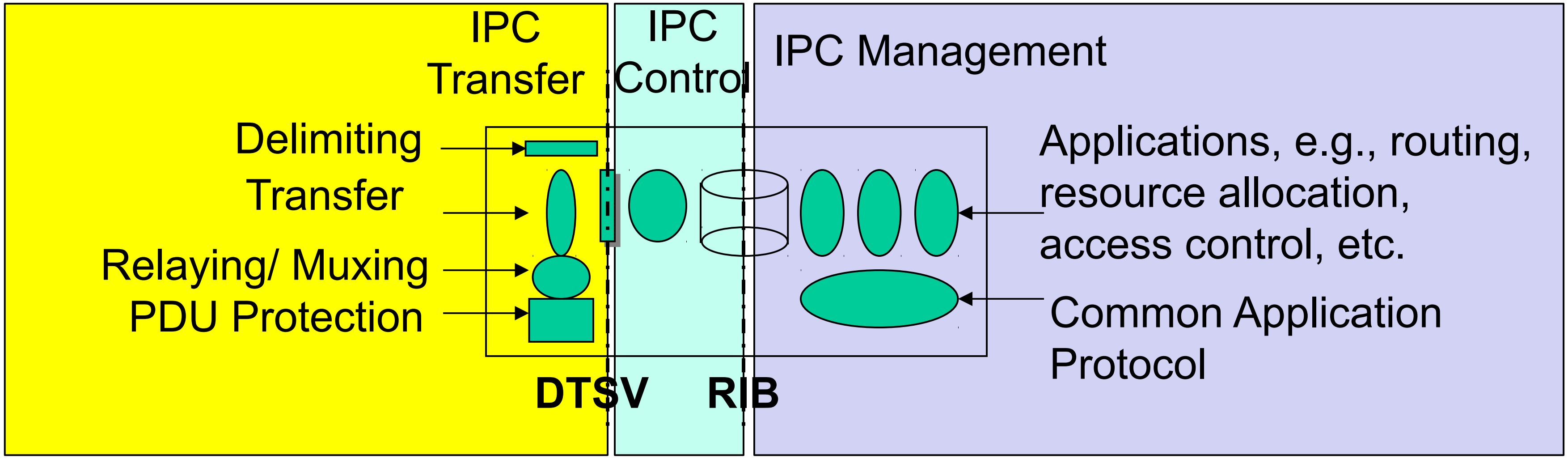
- E2E (end-to-end principle) is not relevant

    Each DIF layer provides (transport) service / QoS over its scope

- IPv6 is/was a waste of time! A single ubiquitous address space is unnecessary

    We can have many layers / levels without too many addresses per DIF layer

What goes into a DIF?


IPC Transfer
IPC Control
IPC Management
Delimiting
Transfer
Relaying/ Muxing
PDU Protection
Applications, e.g., routing, resource allocation, access control, etc.
Common Application Protocol
DTSV
RIB

- Processing at 3 timescales, decoupled by either a State Vector or a Resource Information Base

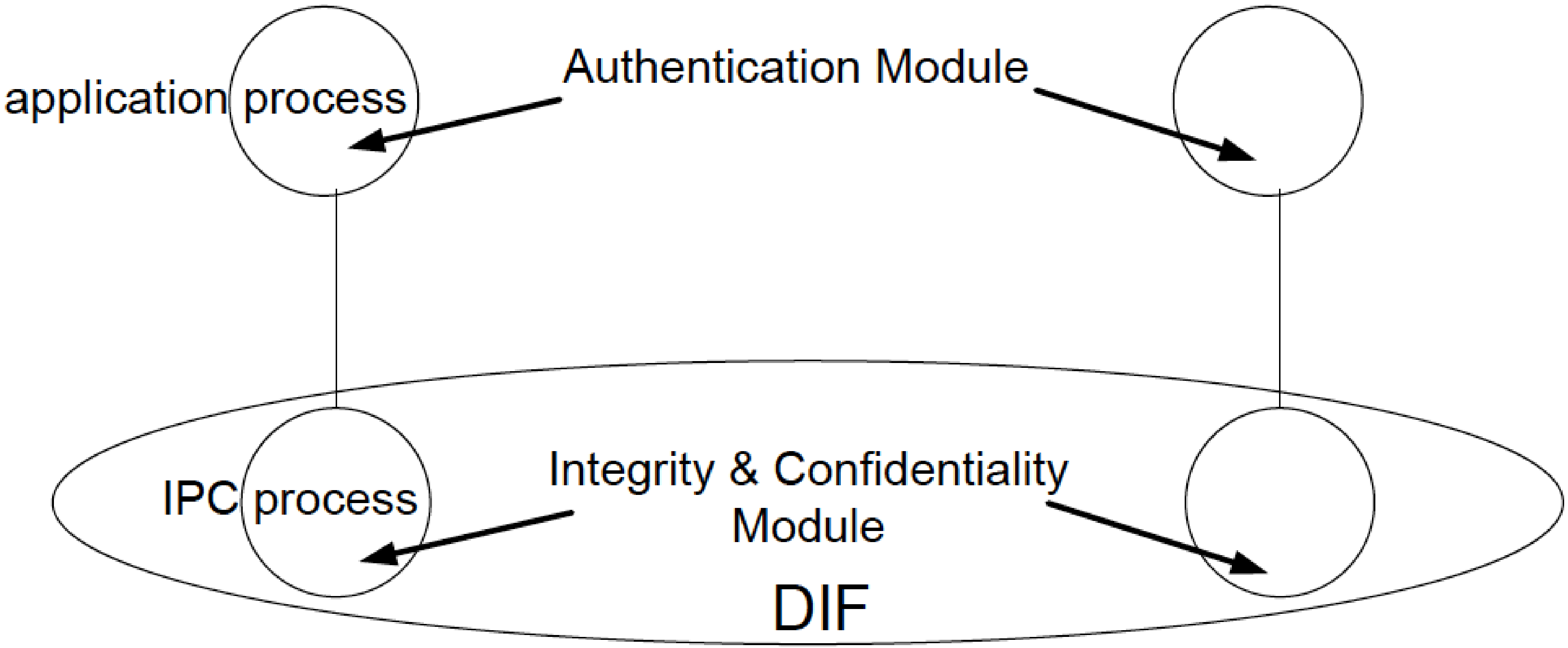
    IPC Transfer actually moves the data

    IPC Control (optional) for error, flow control, etc.

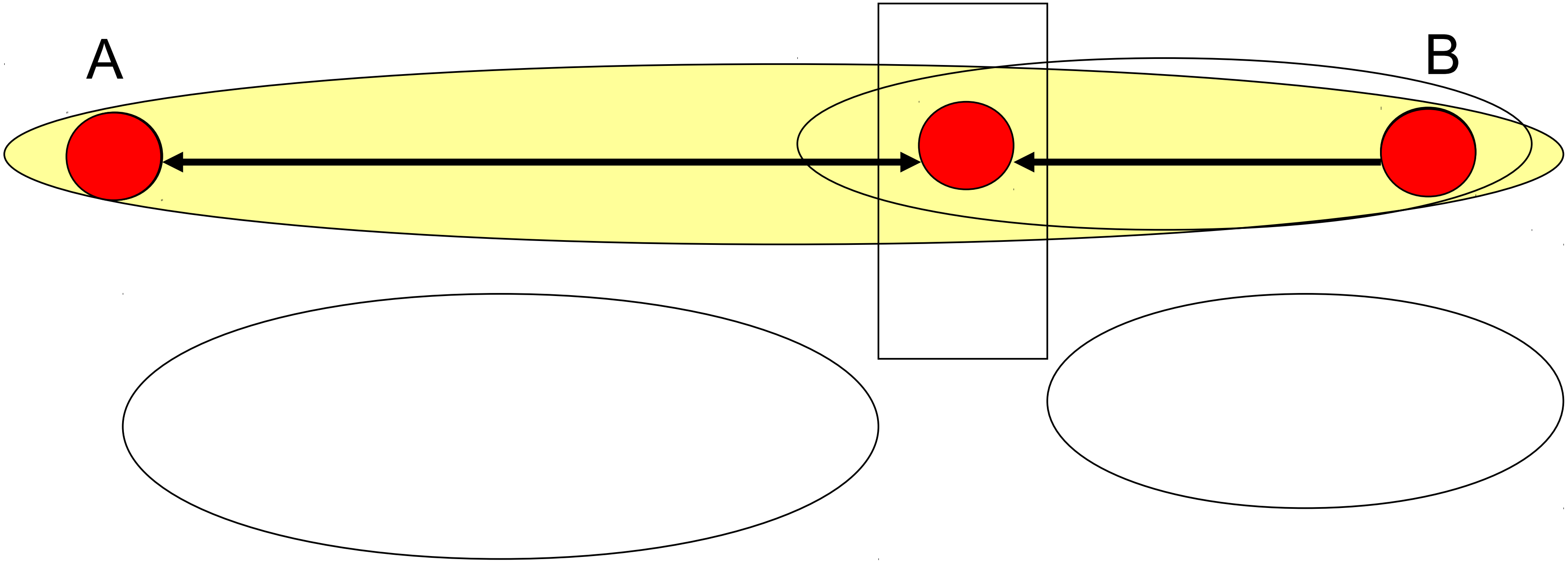
    IPC Management for routing, resource allocation, locating applications, access control, monitoring lower layer, etc.

Where security goes …

- Authentication and encryption are applied recursively – no “shim” sublayers

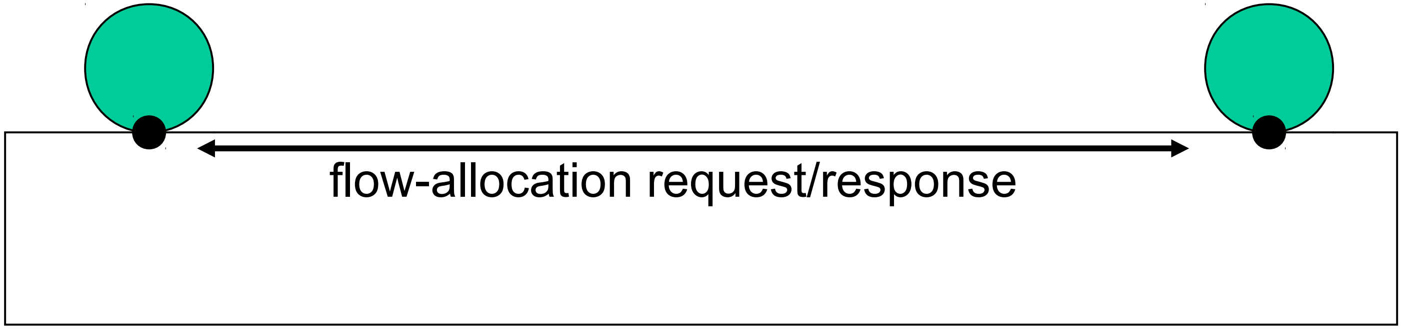


DIF is a secure container

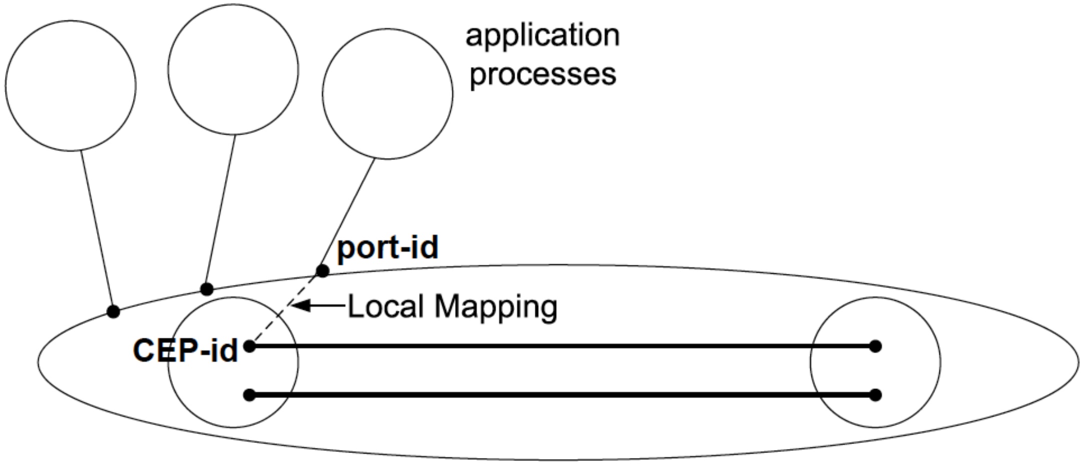


- Nothing more than applications establishing communication
  - Authenticating that A is a valid member of the DIF
  - Initializing it with current DIF information
  - Assigning it an internal address for use in coordinating IPC
  - This is enrollment, i.e. explicit negotiation to join DIF (access control)
  - RINA decouples *authentication* from *connection management and integrity/confidentiality*

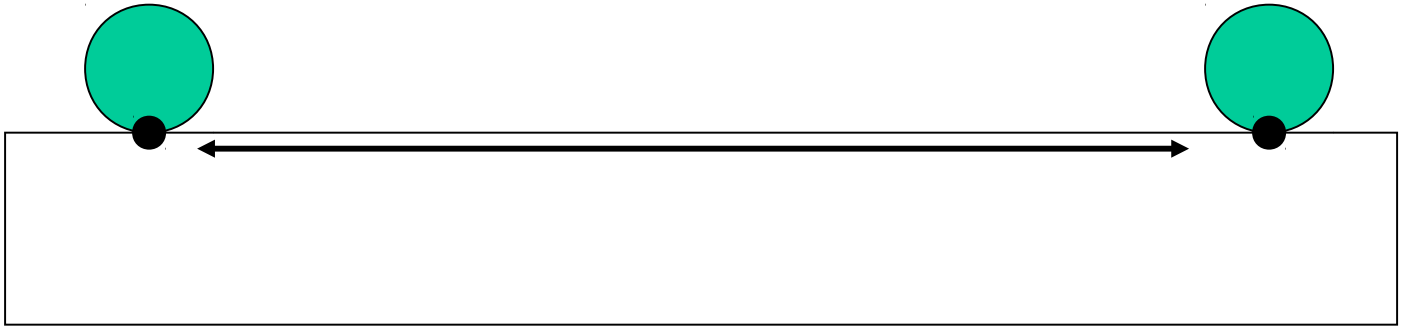
Only one Data Transfer Protocol



- In RINA, service is accessed by its application name
- Port allocation and access control decoupled from data transfer
- At each end, port and conn ID are allocated dynamically and bound to each other by management, in a hard-state fashion



Only one Data Transfer Protocol (2)

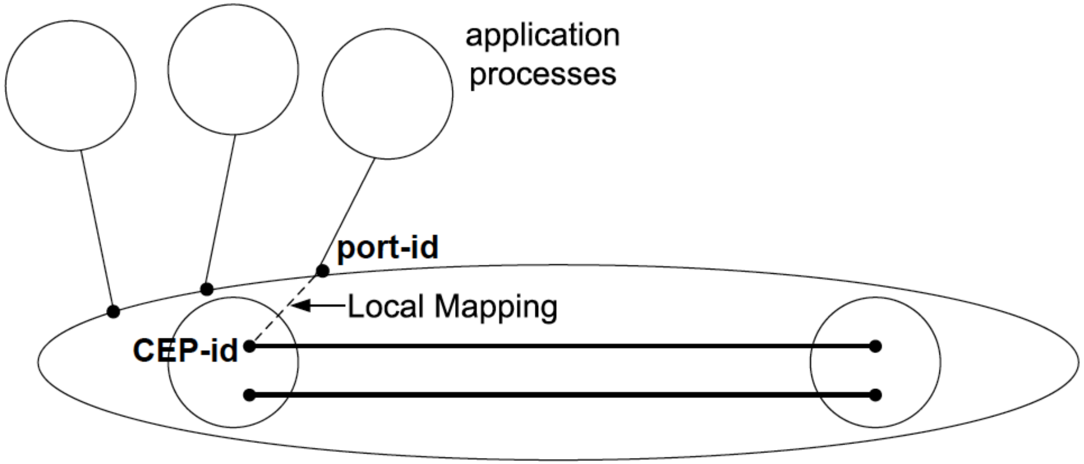


- Once allocated, Data Transfer can start following Delta-t [Watson'81], a soft-state protocol

    Flows without data transfer control are UDP-like. Flows without reliability requirement do not ACK. Different policies support different requirements

    If there is a long idle period, conn state is discarded, but ports remain

    Conn IDs can be changed during data transfer and bound to same ports

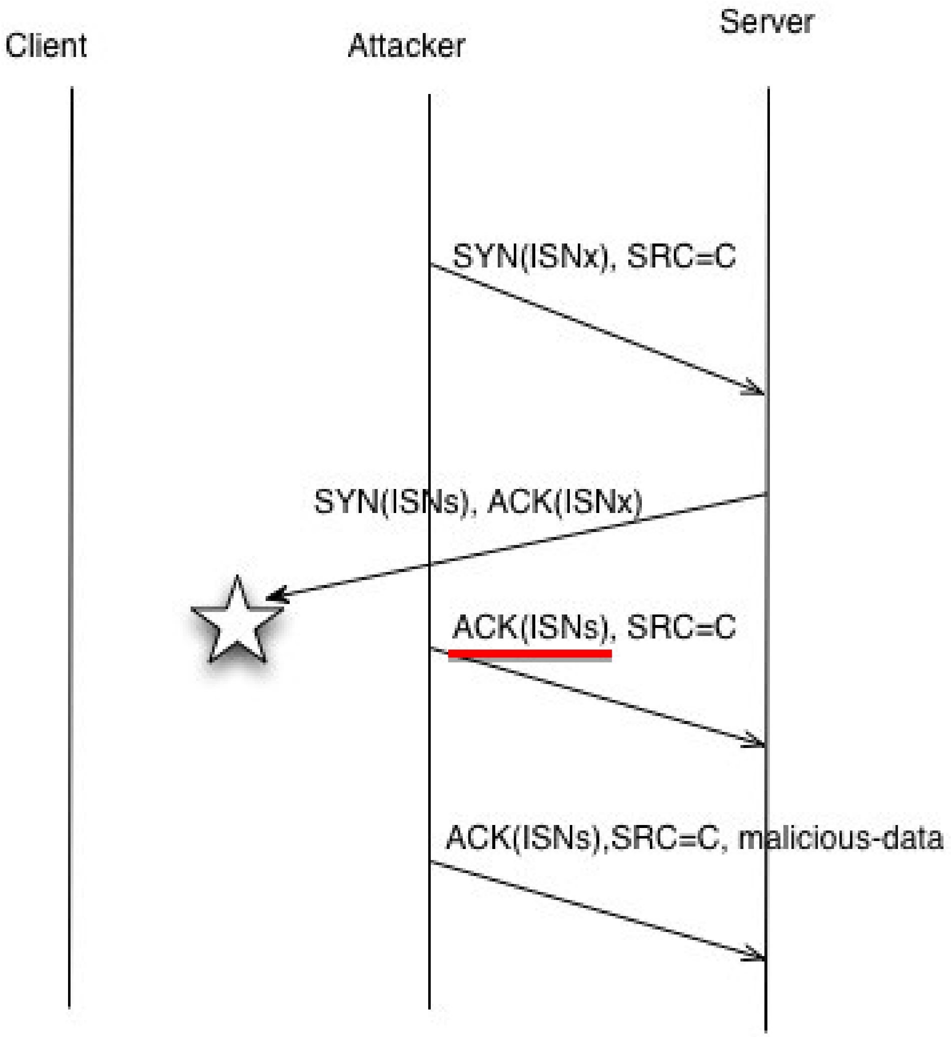


Port Scanning Attacks

- Goal: first step for an attack, explore “open” ports
- In RINA, requesting applications never see addresses nor conn IDs
  - No well-known ports
  - Ports, dynamically allocated, are not part of conn IDs
  - Service requested by application name
- Traditional port scanning attacks not possible
- Scanning application names is much more difficult
- Attacker has to join the DIF too
  - For the sake of comparison, we assume the attacker overcame this hurdle!

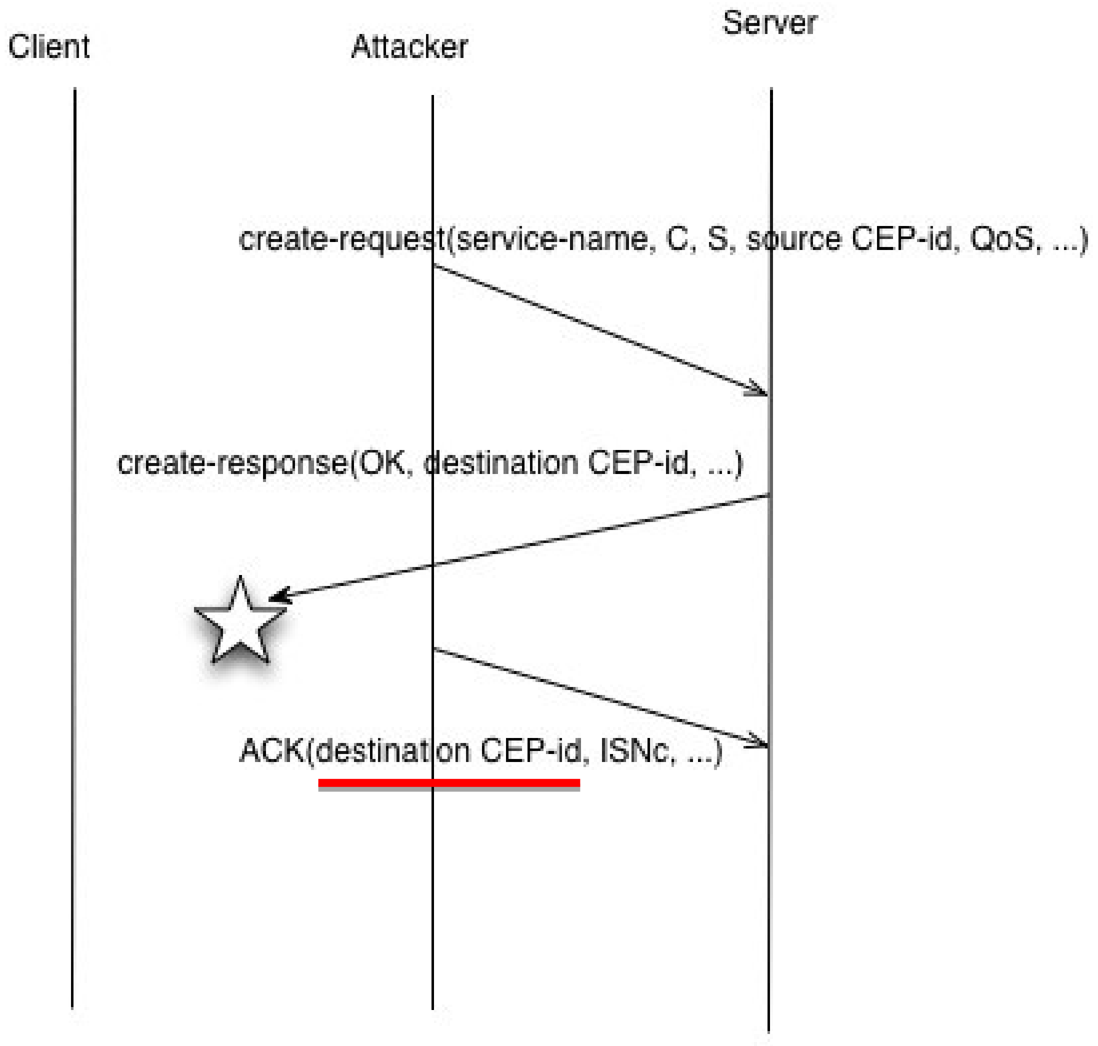
Connection Opening Attacks: TCP/IP

- □ Attacker has to guess server's Initial Sequence Number (ISN)
- □ Given 32-bit sequence number, [redacted] 2^{32} possibilities



Connection Opening Attacks: RINA

- Attacker has to guess destination CEP-id
- Given 16-bit CEP-ids, 2^{16} possibilities
- Akin to port-scanning attacks, which raise more suspicion
- Client can use any ISN


Client
Attacker
Server
create-request(service-name, C, S, source CEP-id, QoS, ...)
create-response(OK, destination CEP-id, ...)
ACK(destination CEP-id, ISNc, ...)

Data Transfer Attacks

- Goal is to inject a legitimate packet, *e.g.* TCP “reset”
- Attacker has to guess source port and SN within transmission window
- Given 16-bit port numbers and 16-bit max window, $2^{16} * 2^{(32-19)=13} = 2^{29}$ guesses

RINA

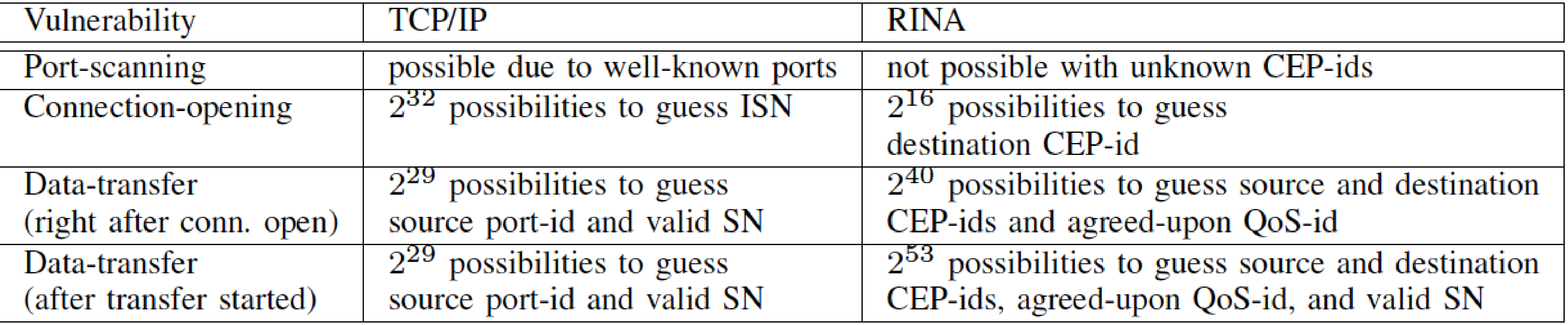
- **Right before data transfer starts**
- Attacker has to guess conn IDs and QoS ID
- Given 8-bit QoS ID, $2^{(16+16+8)} = 2^{40}$ guesses
- **During data transfer**
- Attacker has to also guess SN, so $2^{(40+13)} = 2^{53}$ guesses
- Note: RINA can change conn IDs on the fly

Attacking the reassembly of TCP segment

- Attack by inserting malicious data into IP fragment carrying part of TCP payload

- Not possible in RINA
- Transport and relaying are integrated in each DIF layer
- Fragmentation/reassembly is done once as data enters/leaves the DIF layer

Good Design leads to Better Security



Vulnerability	TCP/IP	RINA
Port-scanning	possible due to well-known ports	not possible with unknown CEP-ids
Connection-opening	2^{32} possibilities to guess ISN	2^{16} possibilities to guess destination CEP-id
Data-transfer (right after conn. open)	2^{29} possibilities to guess source port-id and valid SN	2^{40} possibilities to guess source and destination CEP-ids and agreed-upon QoS-id
Data-transfer (after transfer started)	2^{29} possibilities to guess source port-id and valid SN	2^{53} possibilities to guess source and destination CEP-ids, agreed-upon QoS-id, and valid SN

- ❑ For comparison sake, we assume RINA has been compromised, and a rogue member joined the network
  - ➢ a hurdle that is not present in TCP/IP networks

Good Design leads to Better Security (2)

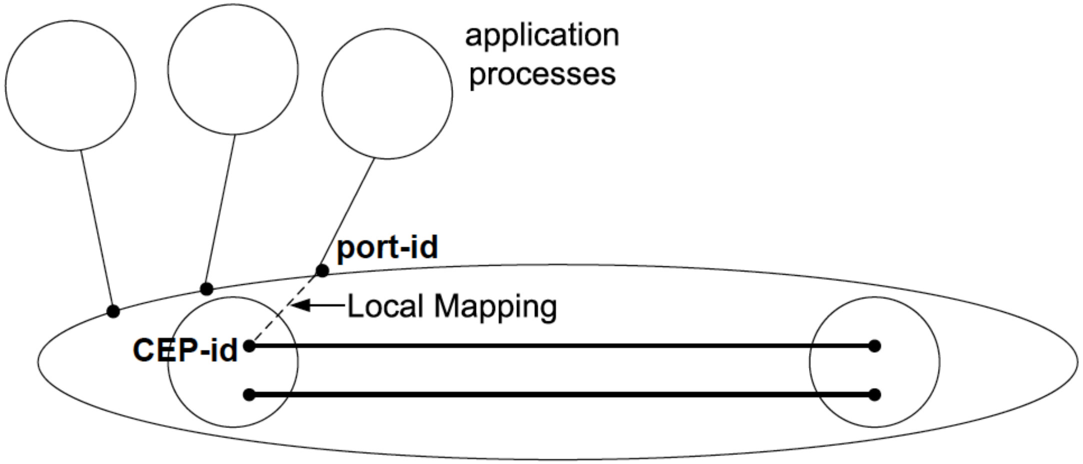
- ❑ In RINA, requesting apps never see addresses nor conn IDs
  - ➔ traditional port scanning attacks not possible

- ❑ Underlying IPC processes must be authenticated to join DIF
  - ➔ only “insider” attacks possible

Good Design leads to Better Security (3)

- Conn IDs are allocated dynamically, so they are hard to guess
- State of data transfer is soft, so there aren't explicit control messages to fabricate

  Note: Delta-t was developed in the 80's with NO consideration for security!



RINA

A Better Network Structure

More @
http://csr.bu.edu/rina