Architectural Considerations for a New Generation of Protocols

—

presented by Xiaowei Yang

Overview

- Two Architectural Principles
  - ILP (Integrated Layer Processing)
    * Layering is a design concept
    * And may not be the most effective modularity for implementation.
  - ALF (Application Level Framing)
    * Get data to applications as soon as possible, in a manner the applications can cope with.

Background

- The paper was written 10 years ago. Back then
  - The fate of ATM and OSI were unclear
  - Authors were trying to figure out how to unite IP network and ATM network
  - We didn't know how to write networking code efficiently

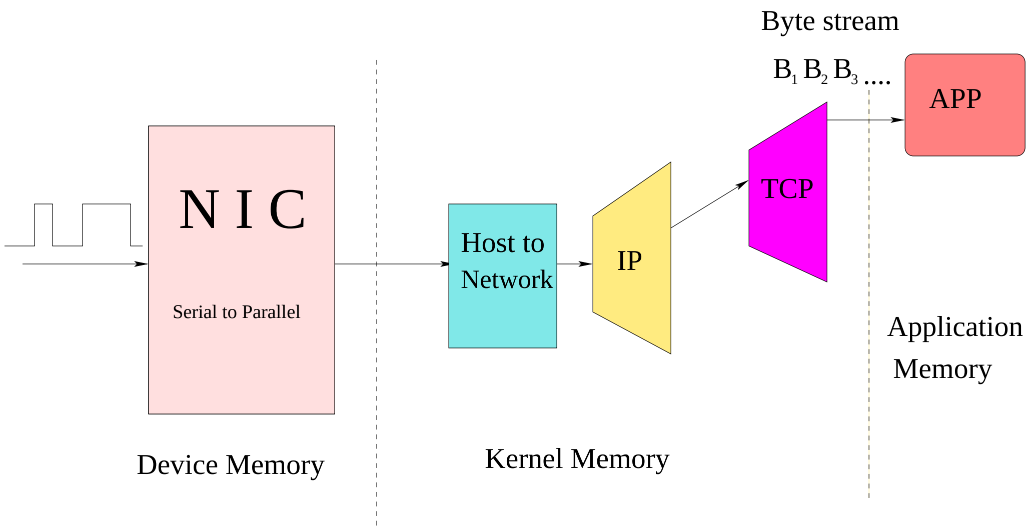
Structuring Principle of Protocol Design

- OSI's 7-layer architecture
  - Physical, data-link, network, transport, session, presentation, application
- Internet's architecture
  - host-to-network, IP, transport, application
- A design choice to decompose complex protocol into functional modules
- Should not constrain efficient implementations

Protocol Functions

- What are protocols for?
  - Transfer application information among machines
- Multiple Data Manipulation Steps

- Moving to/from net
- Error Detection
- Buffering for retransmission
- Encryption
- Moving to/from application address space
- Presentation formatting



Integrated Layer Processing

- Multiple data touches are expensive
  - gap between processor/memory speed
- Example: Copy + CheckSum
  -
$$\frac{1}{(\frac{1}{130}+\frac{1}{115})} = \frac{1}{0.00769+0.00869} = \frac{1}{0.164} = 61$$
  - Combing the two together get 90Mbps
- Solution: Reduce multiple data touches.
  - Do it in one loop if possible.

ILP: Today's View

- Network is usually the bottleneck.

- Application is the bottleneck: presentation conversion (next slide)

- Automatically generating ILP code is hard.

    * Many approaches: compiler support, formal languages.

    * None of them really worked.

- ILP leverages special coding techniques such as hand-coded unrolled loops.

    * Loss of generality.

    * Code is difficult to understand and maintain.

Application Level Framing: Original Motivation

- Presentation conversion is the bottleneck
  - ASN.1 Integer to ASCII : 28Mb/s.
  - Copy: 130Mb/s; Checksum: 115Mb/s
- 97% of the overhead was attributable to the presentation conversion
- Solution
  - Eliminate presentation conversion: ASCII protocols
  - Optimize

Application Level Framing: the Problem

- TCP's reliable in-order byte-stream interface prohibits the out of order data delivery to application.

- Application is prevented from performing presentation conversion as data arrives.

- Since presentation conversion is the bottleneck, it will fall behind forever.

$\rightarrow$ Allow data manipulation to happen in the presence of mis-ordered and lost packets

- Out of order data manipulation improves performance even when presentation conversion is absent.

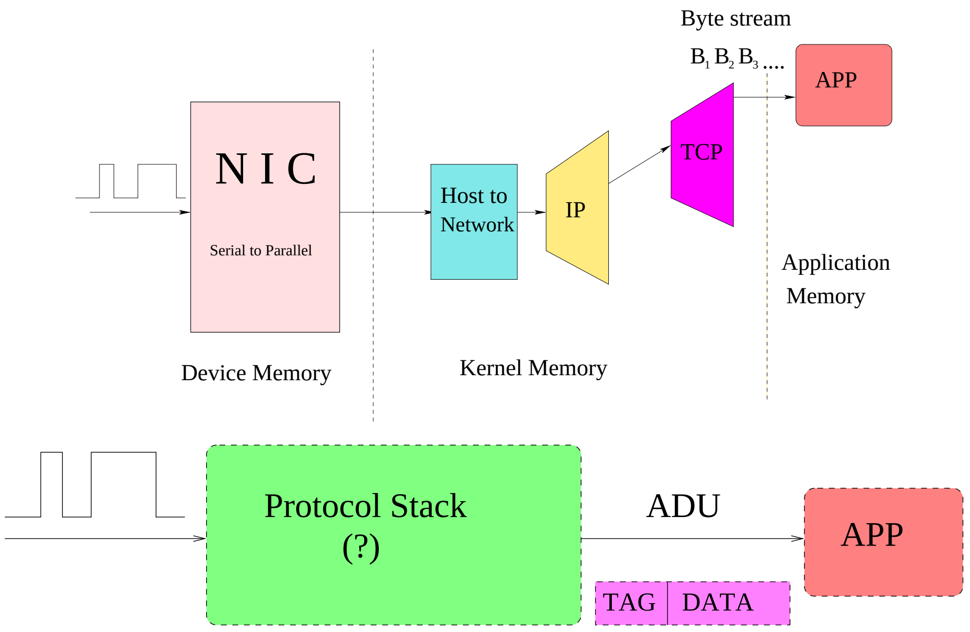
Application Level Framing: Why

- General requirements for out of order processing:
    * “synchronization points” in data streams
- Example: Checksums are computed on per packet basis. Packet boundary serves as synchronization points.
- Synchronization points have to make sense to applications.
    * TCP numbers the bytes in the data stream, which has no meaning to applications.
    * Presentation changes the application data format and does not preserve the size.

Application Level Framing: What

- ALF (Application Level Framing)
  - Lower layers deal with data in units the application specifies.
  - Applications are encouraged to deal with data loss and data recovery in their preferred fashion.
    * selective reliability, out of order processing
- ADU (Application Data Unit)
  - the smallest data unit that an application can process out of order

Application Level Framing: What (continued)



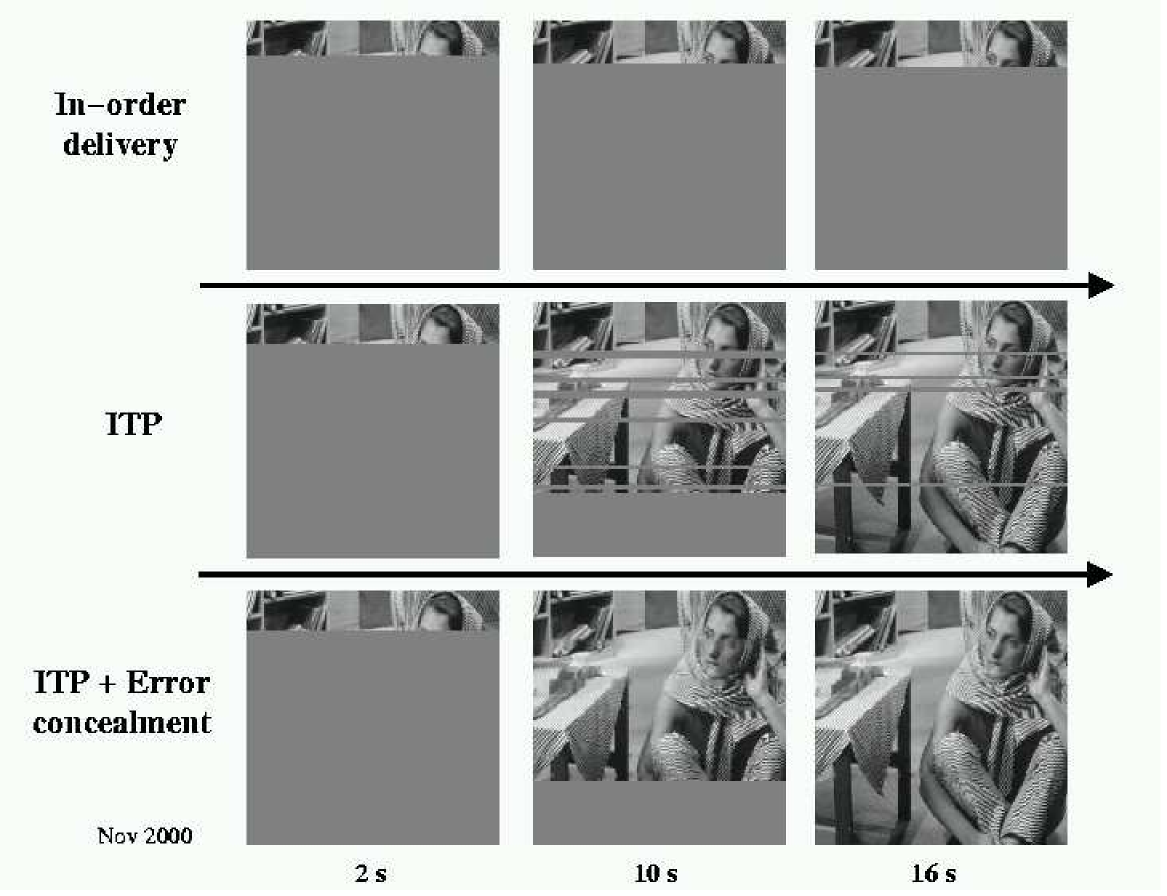
Application Level Framing: How

- Receiver needs to understand where to put ADUs and what to do with them
- Sender can compute a name for each ADU: a meta data that tags the ADU
- The name permits the receiver to understand its place in the sequence of ADUs

Example I: Image Transport Protocol (ITP)

- Problem
  - Images account for much of today's Internet traffic
  - Image transport is over HTTP/TCP
  - TCP's in order delivery results in poor latency in lossy networks
- Solution
  - Image data is structured
  - Frame data into macro blocks (ADUs)
  - Deliver and process ADUs out of order
  - Interpolate missing ADUs

ITP Performance



Example II: ALF in Reliable Multicasting

- Difficulties in achieving Scalable Reliable Multicasting: ACK implosion
- Scalable Reliable Multicasting (SRM)
  - Senders computes meta-data that summarizes all available data
  - Receivers request the retransmission of any desired data triggered by meta-data using multicast damping

Scalable Data Naming to Express Semantics

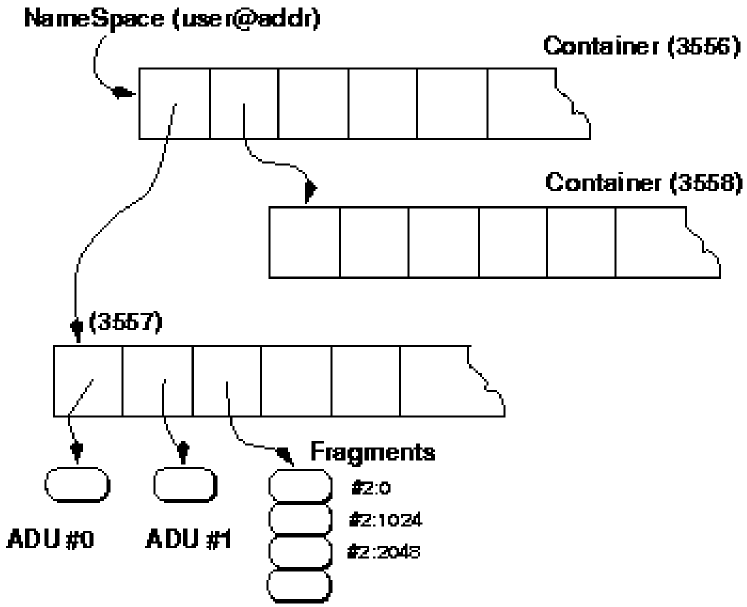
- Problem:
  - Traditional reliable protocols number data units sequentially to detect losses
  - Transport-level sequence numbers do not express applications' reliability semantics
    - *wb*: sequence number 5000 is associated with page 10
  - Receiver-driven reliability is cumbersome to achieve
- Solution
  - A data naming scheme to expose the structure of application data to transport layer
  - A Receiver is able to express its reliability semantics to the transport layer.

Scalable Naming and Accouncement Protocol: Hierarchical Data Naming

- Allow senders to transmit differet objects independently
- Allow receivers to easily specify the data it requires
- The meta-data is scalable even when the data set is large

Example: An ADU from *wb*

- The 5th drawing operation on page 2 from source 9

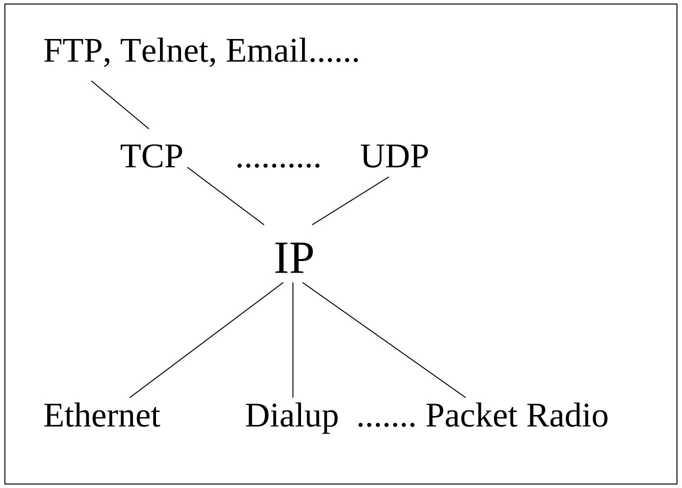


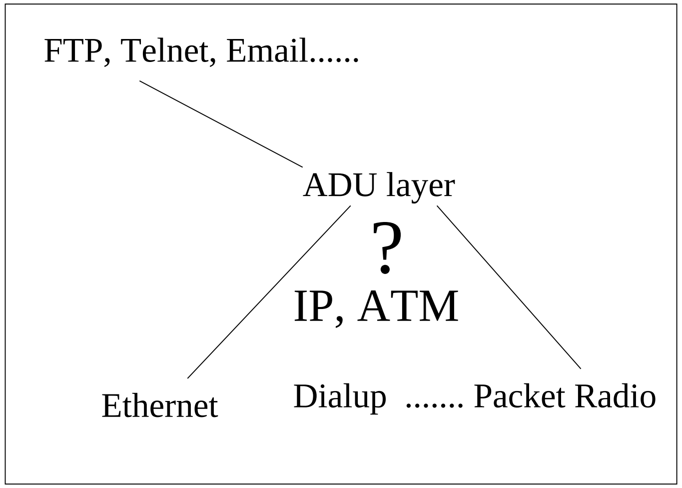
Comments on ALF

- Good for interactive applications, where user perceivable performance matters.
- Good for graphical applications, where data are inherently multi-dimensional.

The Paper's Influence

- Inspired three trends of research
  - A new protocol stack : a debatable issue
    * ALF == UDP + application specific protocols?





The Paper's Influence

- Inspired three trends of research
  - A new protocol stack : a debatable issue
  - Protocol implementation : unsuccessful
    * Micro protocol design
    * Specialized protocol implementation (*e.g.* TCP for telnet)
    * Lessons: taking into account Moore's Law for performance optimization. :)
  - ALF based applications and protocols : the most successful branch
    * ITP, wb, reliable multicasting