Towards More Expressive Transport-Layer Interfaces

Lars Eggert (NEC)
&
Wesley Eddy (Verizon/NASA GRC)

ACM/IEEE MobiArch 2007
Friday, December 1, 2006

Motivation

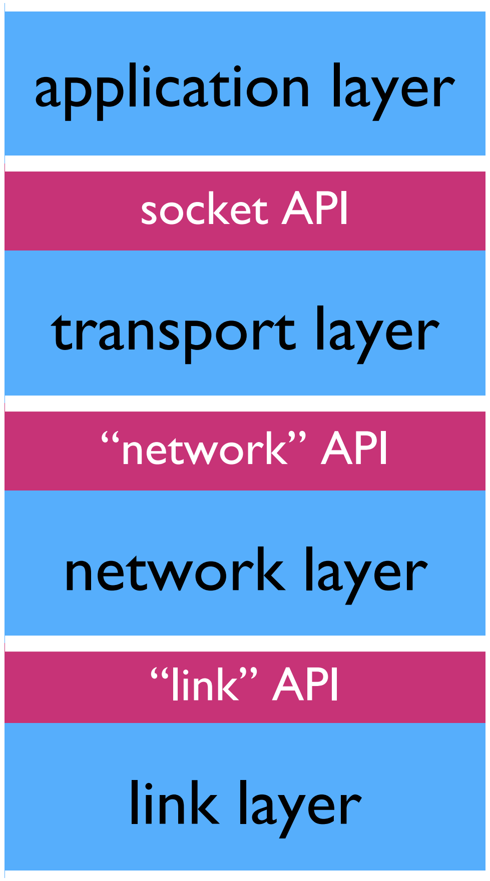
- Internet transport protocols can operate inefficiently or even fail
- in many scenarios that are becoming common
  - mobile and multi-homed nodes and networks
  - links and paths with quickly changing characteristics
  - nomadic use

Why?

- network evolution
- Internet is becoming much more dynamic and heterogeneous than when its protocols were designed
- claim: the original abstractions have started to limit performance and operation of the Internet

Abstractions

- layers in the network stack can be seen as virtual machines
- expose well-defined set of operations & information through an API
- hide intricacies of a layer (& layers below) to its users



application layer

socket API

transport layer

“network” API

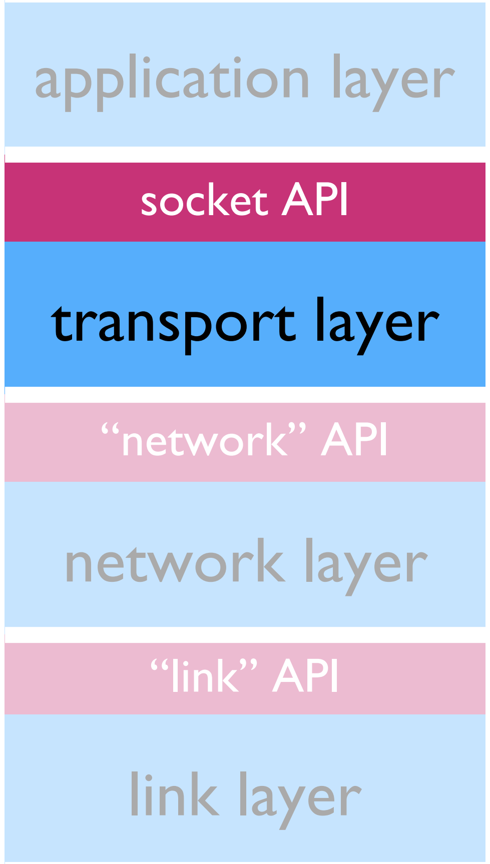
network layer

“link” API

link layer

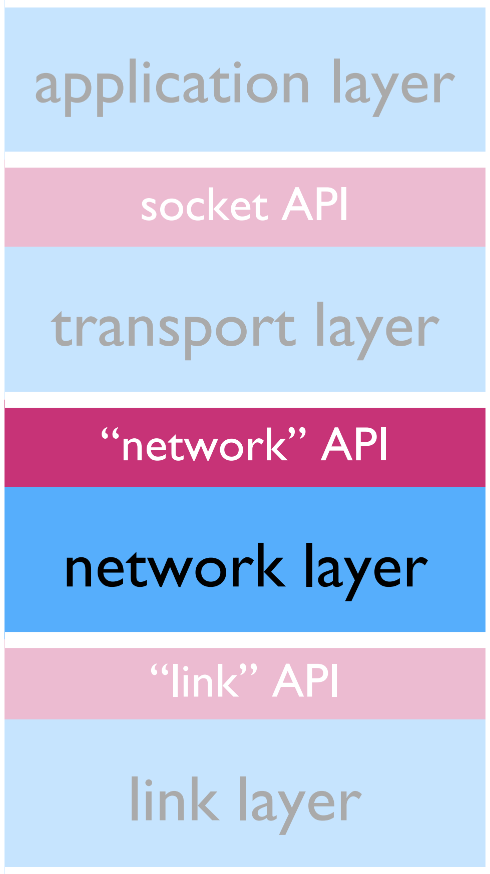
Example: Transport Layer

- abstraction offers operations to manage and use communication channels with various characteristics
  - TCP, UDP, SCTP, DCCP, etc.
- hides other functionality, such as
  - flow control
  - congestion control
  - reliability



Example: Network Layer

- abstraction is something like
  - will deliver your packets in some order
  - may deliver multiple copies of some packets
  - may not deliver some others
- hides other functionality, such as
  - fragmentation/reassembly
  - route computation and forwarding



But in Practice

- users of the network-layer abstraction have made additional assumptions about it
  - and in the past, they have been true
- these assumptions are the basis of many key transport-layer mechanisms, such as
  - congestion control
  - flow control
  - reliability mechanisms

Some Assumptions

- hosts remain at the network port identified by an IP address for long times
- packets between the same source and destination addresses mostly follow the same path
- paths change on time scales that are orders of magnitude greater than the RTT
- path characteristics change on similarly large time scales
- connectivity along a path is very rarely disrupted

Reality Check

- many of these assumptions are no longer generally true
- especially with recent/proposed network layer extensions
  - MIP, HIP, SHIM6, NEMO, MANET, etc.
- but also simply because recent link technologies are different
  - link-level mobility
  - link-layer retransmissions
  - non-congestion packet loss

Consequence

- traditional transport mechanisms are not performing well in these scenarios
- this is not news: gazillion of "optimize transport protocol X for scenario Y" research papers
  - where X is mostly TCP
  - and Y = satellites, 802.11, GSM, 3G, ad hoc net, high bit-error links, etc.
- but vast majority of these are band aids
  - specific fixes for limited scenarios
  - not appropriate for a general-purpose Internet

Why Not Appropriate?

- we don’t want transport protocol flavors optimized for different MANET routing protocols, link technologies, IP mobility management schemes, etc.
  - if a MANET node talks across the Internet to a host on UMTS - what TCP flavor do I use?
- transport protocols are end-to-end!

What Is Appropriate?

- extend the “virtual machine” abstraction that the network layer provides
- but do it in a way that is generic
  - independent of network-layer extensions
  - independent of specific link technologies

Strawman Design

1. provide additional pieces of information or notifications about network-layer events
   - should be advisory and optional: transports shouldn't depend on them
2. design transport-layer response mechanisms that act on (1) to improve transport operation and performance

Not a New Idea

- other proposals are already enhancing the network-layer “virtual machine” abstraction in this way
  - ECN (“I’m about to start dropping these packets”)
  - Quick-Start (“you may send me packets at rate n”)
  - TRIGTRAN/ALIAS (but this is arguably broader)
  - IEEE 802.21 (similar idea one layer down - easier?)
- and don’t forget about ancient stuff like ICMP
  - unreachables: “this host/network is not here”
  - original source quench: “stop sending so fast”

Why Is This Hard?

- it's easy to optimize for one particular lower layer ("TCP over 802.11" hacks)
- it's hard to identify a small (minimal?) set of generic pieces of information or signals that:
  - can be provided by different underlying technologies (in different ways)
  - are expressive enough to allow significant performance improvements for many uses
- security - if the source of the information isn't local, how do you know you can trust it?

The Plea

- further incremental improvements to point solutions are of little practical use
  - we don’t need yet another “TCP over 802.11” proposal or similar optimizations
  - the potential benefits of cross-layer schemes are well documented already
- when designing network-layer mechanisms, think about what will run on top of them
  - great PDR doesn’t mean great TCP performance

The Pitch

- a cross-layer approach that is generally useful in many scenarios, so it can improve people's Internet experience
    - notify transports about lower-layer events, so they can act appropriately
    - in a technology-independent way
- this is not (yet) an engineering problem - there is some research left to be done