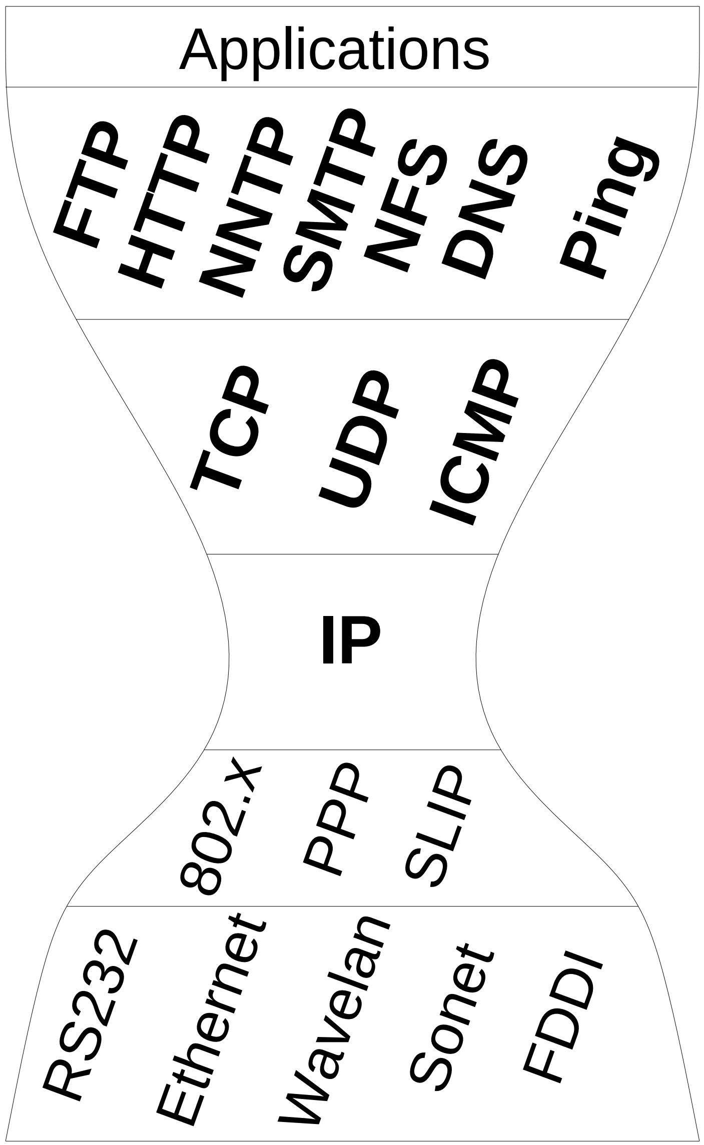
On Inter-layer Assumptions
(A View from the Transport Area)

Mark Handley
ACIRI/ICSI
mjh@aciri.org

The Internet Hourglass



IP is the unifier

- **Transport protocols only have to deal with IP**
  - Don't care about different link layers

- **Link layers only have to support IP**
  - Don't care about applications

IP is the unifier

- **Transport protocols only have to deal with IP**
  - Don't care about different link layers
- **Link layers only have to support IP**
  - Don't care about applications
- **At least that's the theory.**
- **In practice:**
  - There are implicit assumptions that transport protocols make about IP that are affected by the link layer.
  - To effectively support IP, a link layer must also support common transport protocols.

Assumptions and Standards

- **Changes in technology tend to reveal what these assumptions really are.**
  - Wireless technologies are just such a change.
  - When you violate the assumptions, things break.
- **Not writing these assumptions down in advance is good.**
  - Specify the minimum required for interoperation and safe network behavior.
  - Otherwise we can't be flexible.
- **At what stage do we make implicit assumptions explicit?**
  - Do we add inter-layer "hints" to retain flexibility?
  - In which cases do we modify Internet protocols to change their assumptions?

End-system IP-level assumptions:

- **Routing pre-computes viable routes to all reachable destinations.**
- **An IP source sends a datagram which is delivered to a destination.**
  - There are no guarantees about when or if it arrives.
  - (NATs violate this assumption)
- **The destination address should be reachable.**
  - Usually via pre-computed routing tables in routers.
- **What do we assume about the source address?**
  - Does it have to be the same host?
  - Does it have to be the same network?
  - Do routers check it?

End-system IP-level assumptions:

- **What do we assume about the source address?**
  - Does it have to be the same host?
  - Does it have to be the same network?
  - Do routers check it?
- **As of 15th Feb 2000:**
  - RFC 2267 "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing" is "Best Current Practice"
- **What's the implication for Mobile IP?**

TCP: Assumptions about IP

- **Endpoint addresses are static**
  - Connection can't survive renumbering
- **Packet loss is caused by congestion**
  - Halve transmission rate.
- **Corrupted packets should be dropped**

  Packet reordering in the network is small scale
  - less than 3 packets out-of-order (or 3 DUP ACKs imply loss).
- **Delay is predictable**
  - less than ***SRTT + 4*RTT_var*** **or treated as loss.**

TCP: Assumptions about IP

- **Packet loss is caused by congestion**
- **For congestion, correct behavior is:**
  - Halve congection window,
  - Or exponentially backoff of retransmit timeout
- **What about fading, corruption, or link-layer initiational delays?**
- **The temptation is to design link-layer specific protocols or extensions.**
  - This is bad.
  - TCP/IP works end-to-end across many concatenated link layers.

TCP: Packet loss = Congestion

- **Without admission control, an IP network will always (in some cases) have to drop packets to cope with congestion.**

- **Explicit Congestion Notification (ECN):**
  - mark packets at times of mild congestion
  - drop packets at times of severe congestion because the buffer is full.
- **ECN will greatly decrease the number of losses due to congestion, but cannot change the basic assumption that loss implies congestion.**

TCP: Packet loss = Congestion

- **Inter-layer hints to disambiguate non-congestive loss are perhaps reasonable?**
  - "Explicit Corruption Notification" hint
  - "Destination Now Reachable" hint

- **Loss of a hint only results in more conservative behavior**

TCP Header Compression: Loss = Congestion

- **TCP/IP header compression (RFC 1144) works by not sending fields that change in a predictable way.**
- **Only intended for single hop links:**
  - Congestive loss of compressed packets cannot happen because compression takes place on the output from the queue.
- **Assumes the link itself is negligably lossy.**
  - If not, context is lost.
  - Bad assumption with a Metricom modem!
- **draft-jonsson-robust-hc-03.txt is a possible solution**

TCP: Packet reordering is small scale

- **3 DUP ACKs imply to TCP that the packet was lost.**
  - => retransmit and halve the congestion window.
- **Why 3?**
  - Tradeoff between reacting fast to loss and reacting spuriously to reordering.
  - Link-layer ARQ might confuse this (probably not)
  - Wireless handoffs can change routing and delay.
  - Diversity routing in multi-hop wireless.
- **TCP-Sack (draft-floyd-sack-00.txt) allows spurious reordering to be detected and the DUP-ACK threshold to be adaptive.**

Delay is Predicable

- **Delay is less than:**
  - ***RTO = SRTT + 4*RTT_var***
- **Or retransmission occurs, the congestion window is halved, and slowstart occurs.**
- **TCP-Sack (draft-floyd-sack-00.txt) allows spurious retransmission to be detected.**
  - How to adapt is an open question.

TCP: Delay is Predicable

- **Link-layer ARQ can cause interesting delays:**

```
64 bytes from 192.150.187.20: icmp_seq=2 ttl=237 time=430.150 ms
64 bytes from 192.150.187.20: icmp_seq=3 ttl=237 time=420.148 ms
64 bytes from 192.150.187.20: icmp_seq=4 ttl=237 time=400.201 ms
64 bytes from 192.150.187.20: icmp_seq=5 ttl=237 time=420.174 ms
64 bytes from 192.150.187.20: icmp_seq=6 ttl=237 time=420.180 ms
64 bytes from 192.150.187.20: icmp_seq=7 ttl=237 time=820.171 ms
64 bytes from 192.150.187.20: icmp_seq=8 ttl=237 time=510.240 ms
64 bytes from 192.150.187.20: icmp_seq=9 ttl=237 time=538.432 ms
64 bytes from 192.150.187.20: icmp_seq=0 ttl=237 time=480.157 ms
64 bytes from 192.150.187.20: icmp_seq=1 ttl=237 time=470.189 ms
64 bytes from 192.150.187.20: icmp_seq=2 ttl=237 time=440.208 ms
64 bytes from 192.150.187.20: icmp_seq=3 ttl=237 time=410.193 ms
64 bytes from 192.150.187.20: icmp_seq=4 ttl=237 time=410.224 ms
64 bytes from 192.150.187.20: icmp_seq=5 ttl=237 time=430.184 ms
```

- **Metricom modem, lightly loaded path.**

Assumptions of Non-TCP Apps

- **SCTP**
  - Congestion Control mechanisms make similar assumptions to TCP.
- **RTP**
  - Predicable delay (for adaptive playout buffer)
- **NTP**
  - Symmetric delay
- **Reliable Multicast**
  - SRM: Predictable delay (for feedback suppression)

Link-layer assumptions about IP

- **Delay/loss tradeoff:**
  - "Best-effort IP makes no guarantees about delay or loss"
  - How much delay is reasonable?

- **Packets are independent?**
  - Reordering doesn't matter?

- **It's all TCP?**

Interesting Delays:

```
64 bytes from 204.179.128.49: icmp_seq=174 ttl=243 time=28097.003 ms
64 bytes from 204.179.128.49: icmp_seq=177 ttl=243 time=29893.651 ms
64 bytes from 204.179.128.49: icmp_seq=180 ttl=243 time=28236.982 ms
64 bytes from 204.179.128.49: icmp_seq=185 ttl=243 time=28051.881 ms
```

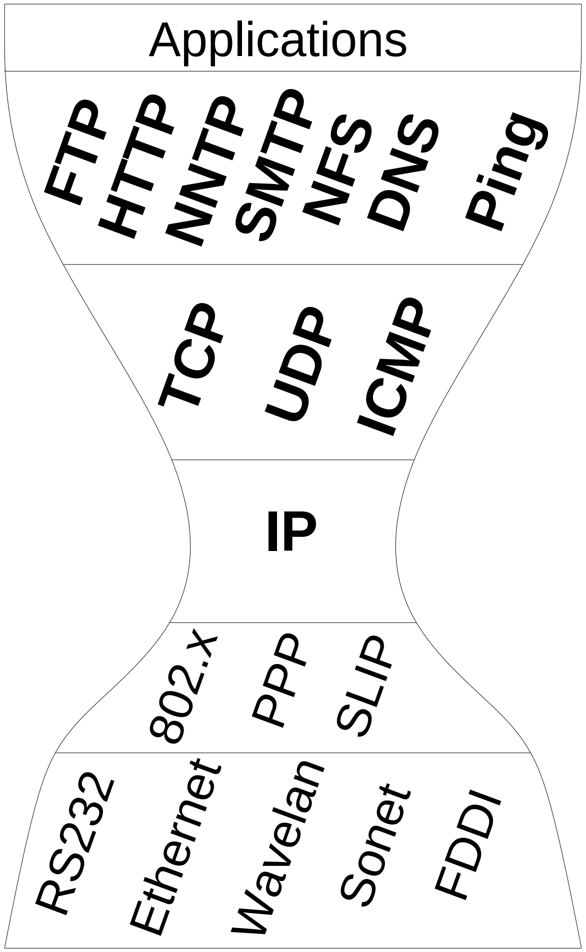
- **Metricom modem, loaded with an incoming 16Kb/s UDP stream (loss rate is 40%).**
- **These delays won’t happen with TCP...**
  - Bad to design a network assuming TCP.

Miscellaneous Issues for Wireless IP

- **Multicast**
  - Can receive anywhere, but...
  - Reverse-Path Forwarding check on source address means cannot send using home source address without relaying through home agent.
- **DDoS Attacks**
  - Unicast RPF may be desirable.
  - May be at odds with Mobility.
- **Middle-boxes**
  - E.g., Akamai, etc
  - More implicit assumptions about location.
- **Mobile clients vs Mobile Servers?**

Conclusions

Layering is a simple design principle that means each protocol designer only has to deal with two interfaces: one to the layer below and one to the layer above.



Conclusions

Layering is a simple design principle that means each protocol designer only has to deal with two interfaces: one to the layer below and one to the layer above.

- **If you believe this, you are designing for the lowest common denominator service.**
- **Good performance means:**
  - Taking into account the assumptions of all other layers, whether written down or not.
  - Making protocols more adaptive so they have fewer rigid assumptions.
  - Making the tradeoffs more explicit in the form of hints.
- **But don't design transport protocols to assume a particular link-layer.**