Efficient Cross-Layer Negotiation

Bryan Ford
MPI-SWS and
Yale University

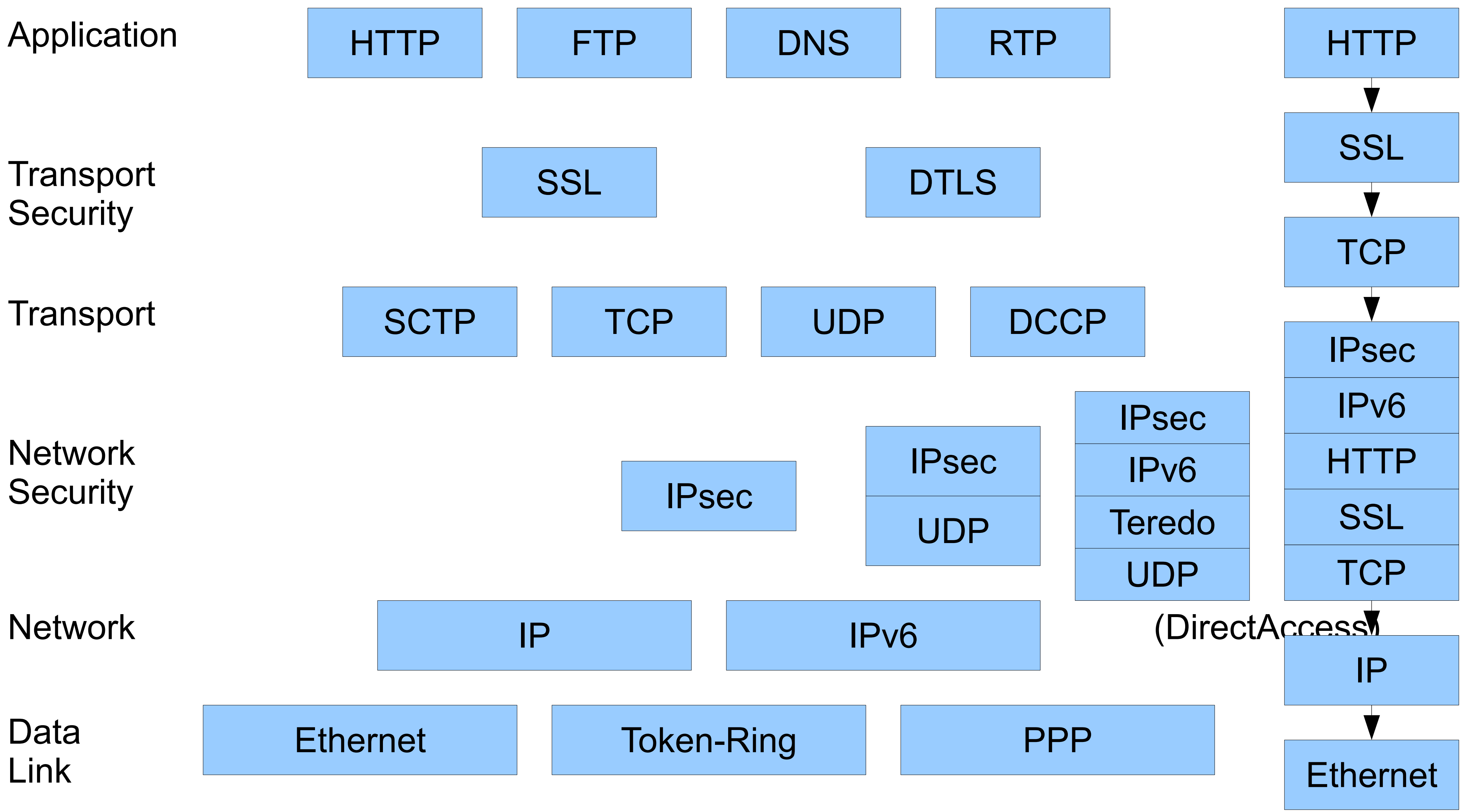
Janardhan Iyengar
Franklin & Marshall
College

Presented at HotNets-VIII, October 22, 2009

“Tng: Transport Next Generation” Project

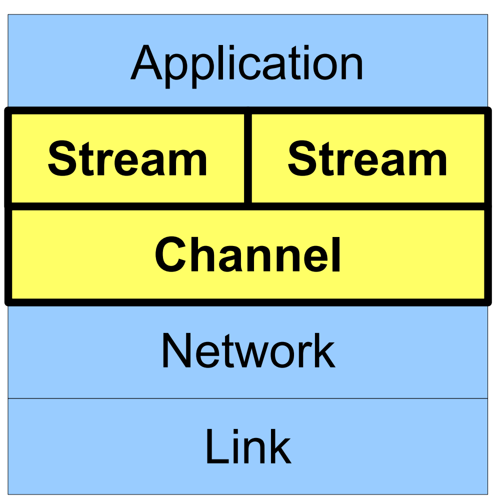
Support: NSF FIND grants CNS-0916413 and CNS-0916678

A Proliferation of Layers and Layer Combinations

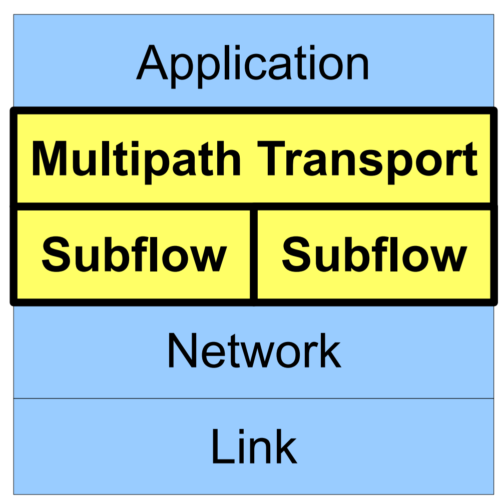


Future: Ever More Layers/Combinations?

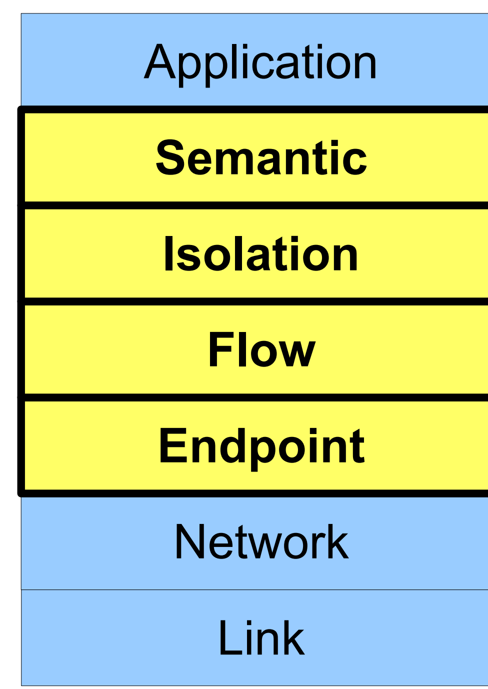
Multi-Streaming Transports
SCTP [rfc4960],
SST [SIGCOMM'07]



Multipath Transports
SCTP [rfc4960],
MPTCP [WIP]

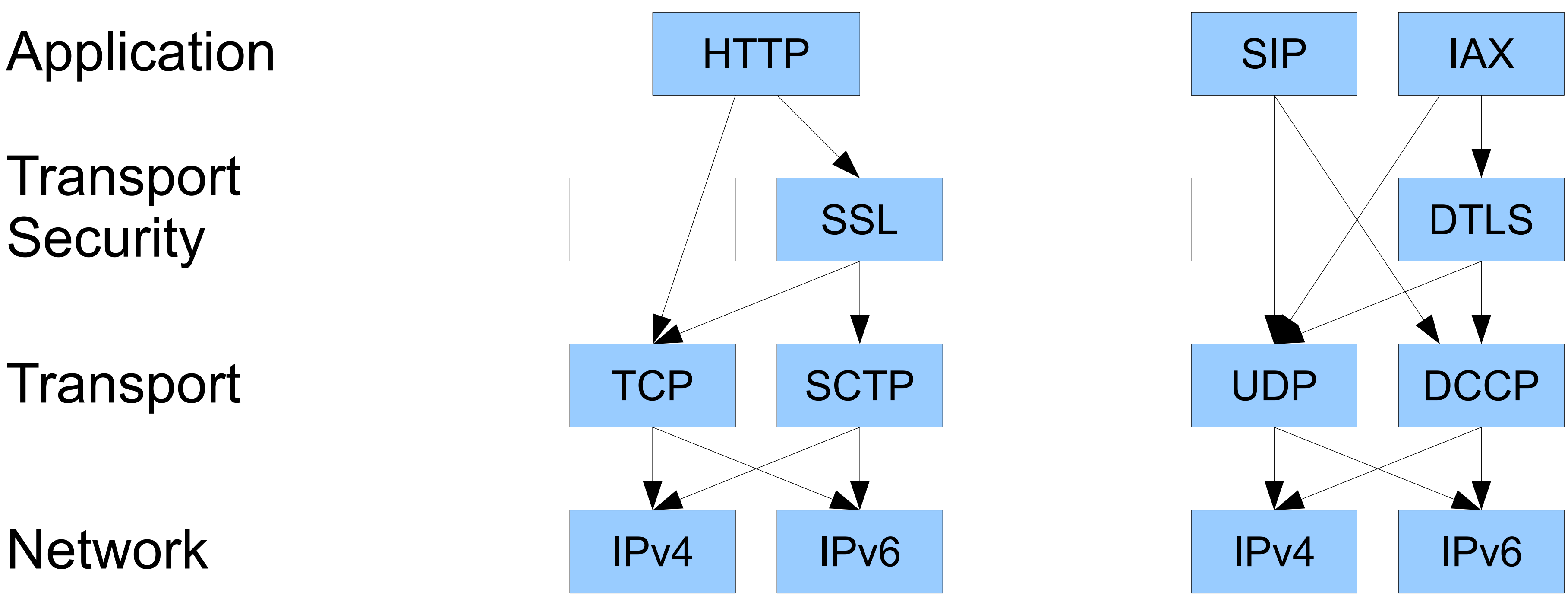


Further Decomposition
[“Breaking Up the Transport Logjam”, HotNets'08]



The Negotiation Problem

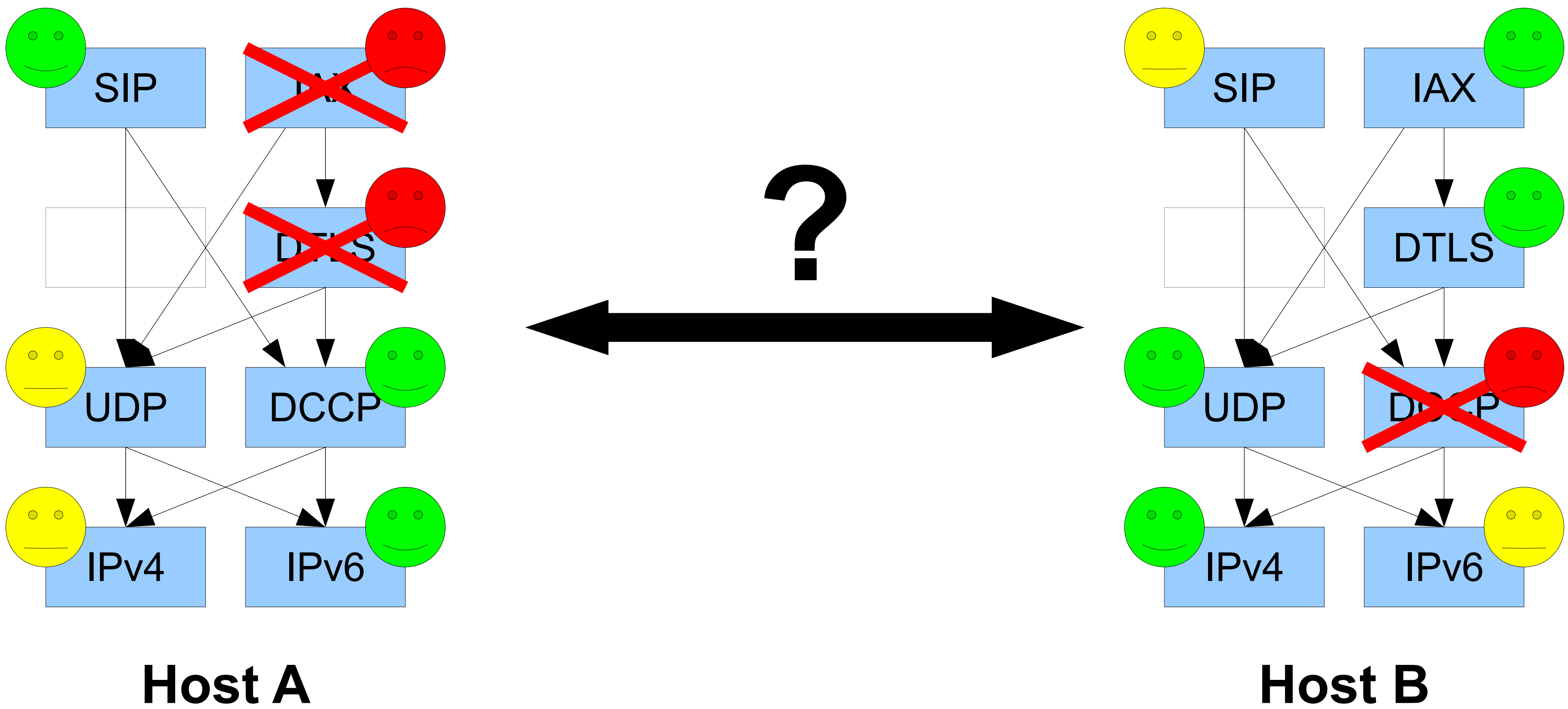
Decisions, decisions!



Compatibility and Preference

Which combinations do *both* endpoints support?

Which combinations do they *prefer*?



Talk Outline

- Background and Alternatives
- A Model for Negotiation
- Negotiation Transport Protocol
- Discussion, Conclusion

Background and Alternatives

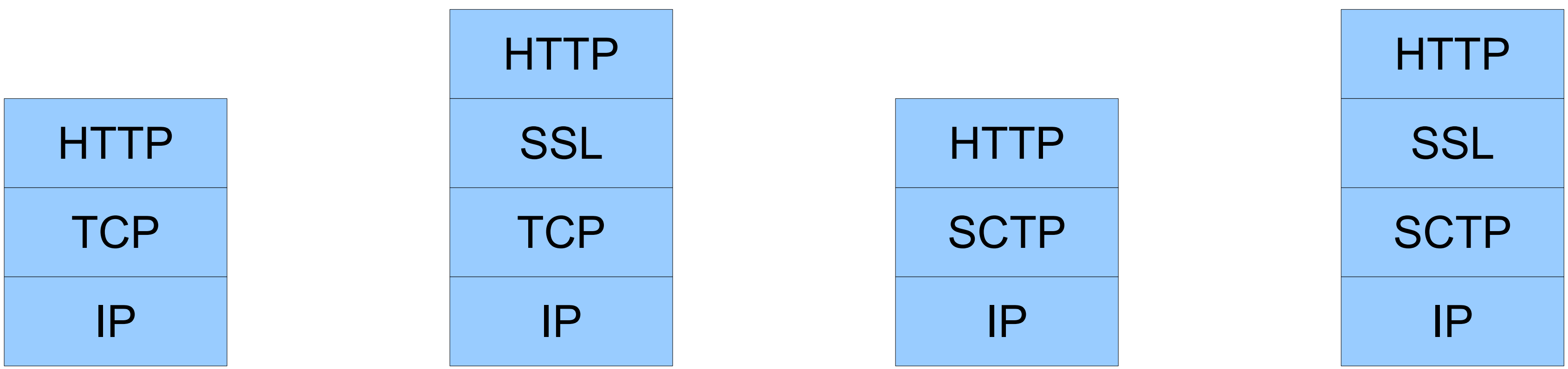
Approach 0: Name Encoding

http:// *means:*

https:// *means:*

http++sctp:// *means:*

http++ssl++sctp:// *means:*



`[rfc2616]`

`[rfc2818]`

`[draft-wood-tae-specifying-uri-transports]`

?

Disadvantages of Name Encoding

Loss of Transparency

- User cares about *application*, not underlying stack... but is *forced* to see and care about underlying stack
- When underlying stack changes, URLs change/break
  - redirectors proliferate between **http://** and **https://** spaces

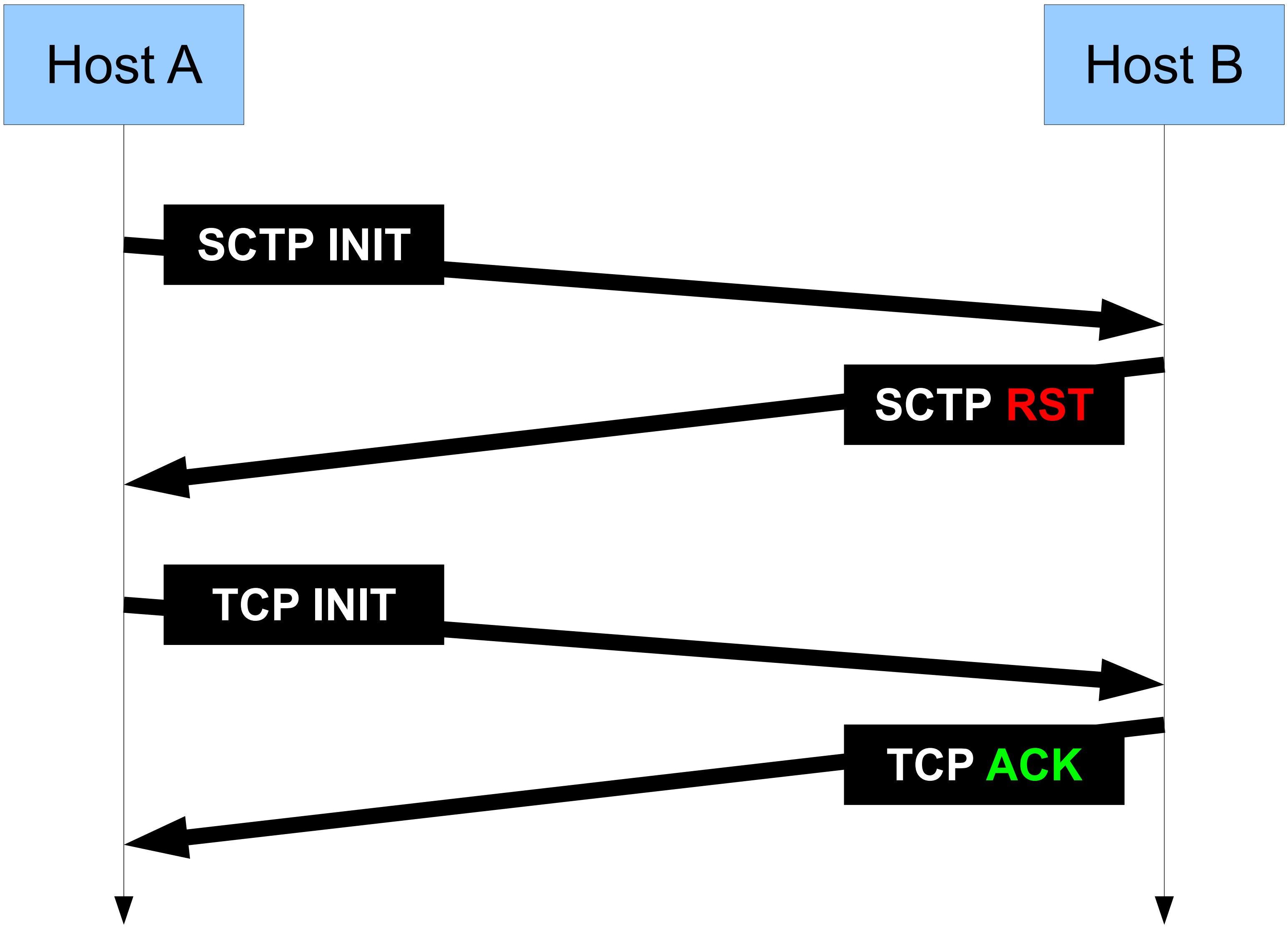
Loss of Compatibility

- If user puts "**http++sctp://...**" link on a web page, legacy browsers break; *cannot* fall back to TCP

Where Do You Stop?

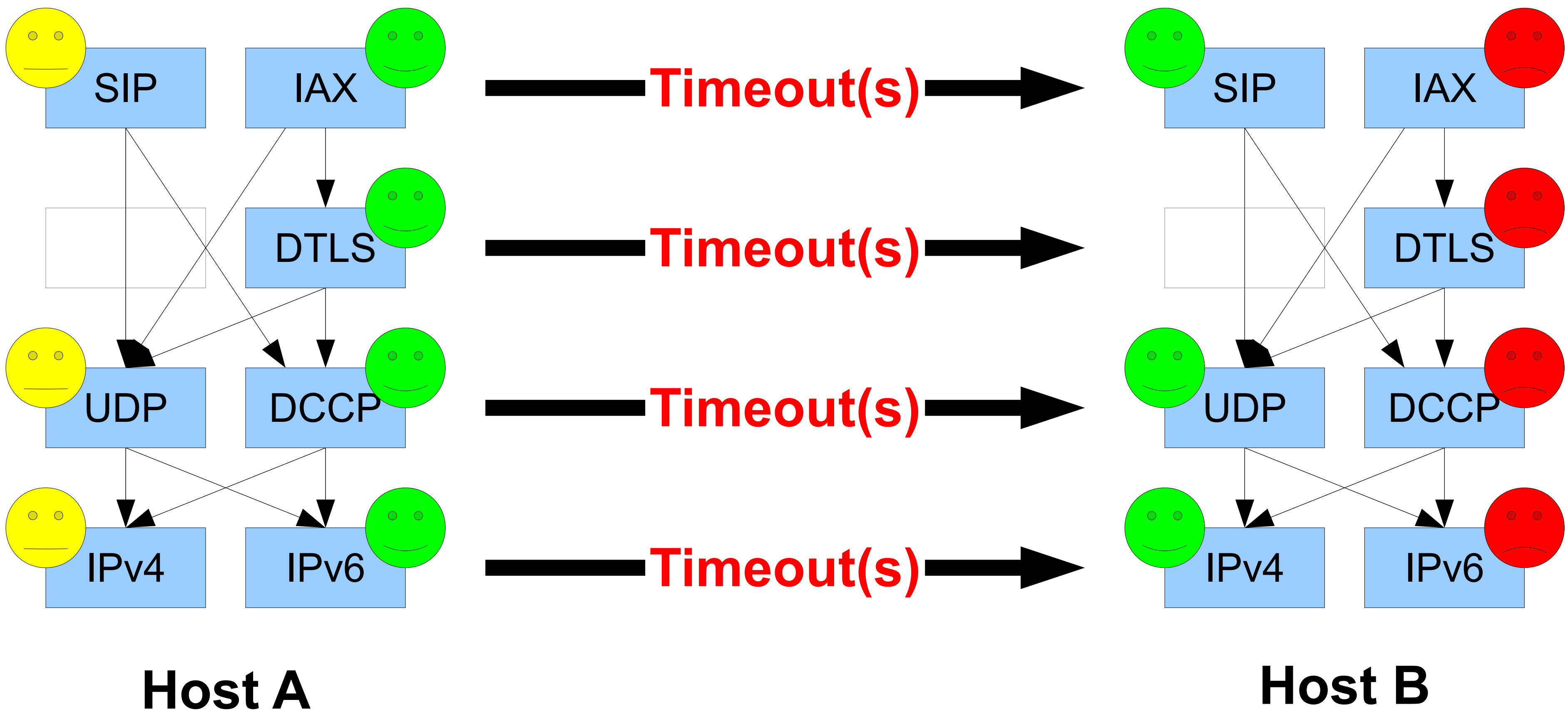
- "**http++tls++tcp++ipv6++ethernet**" ???

Approach 1: Try and Fall Back

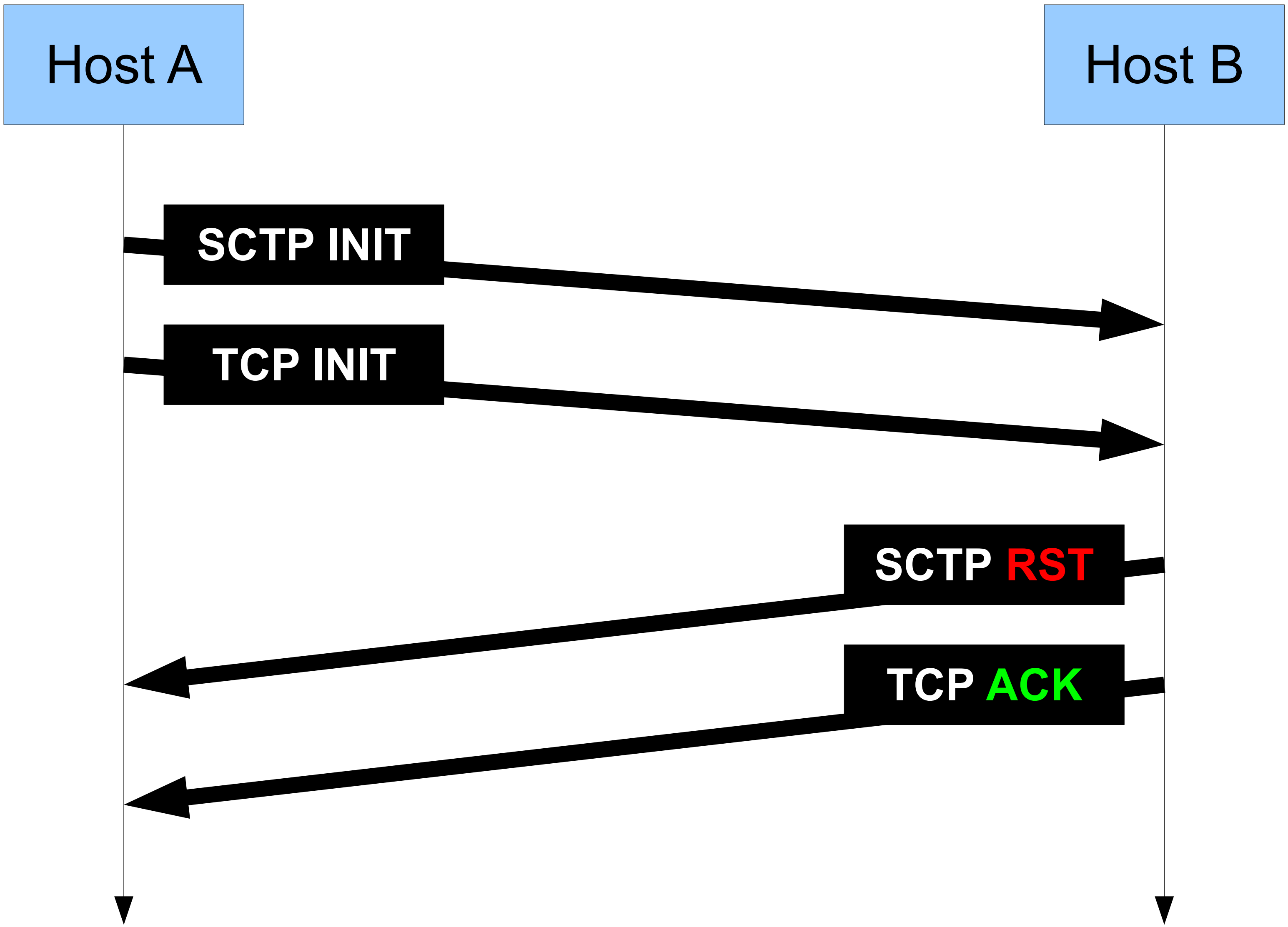


Challenge 1: Controlling Delay

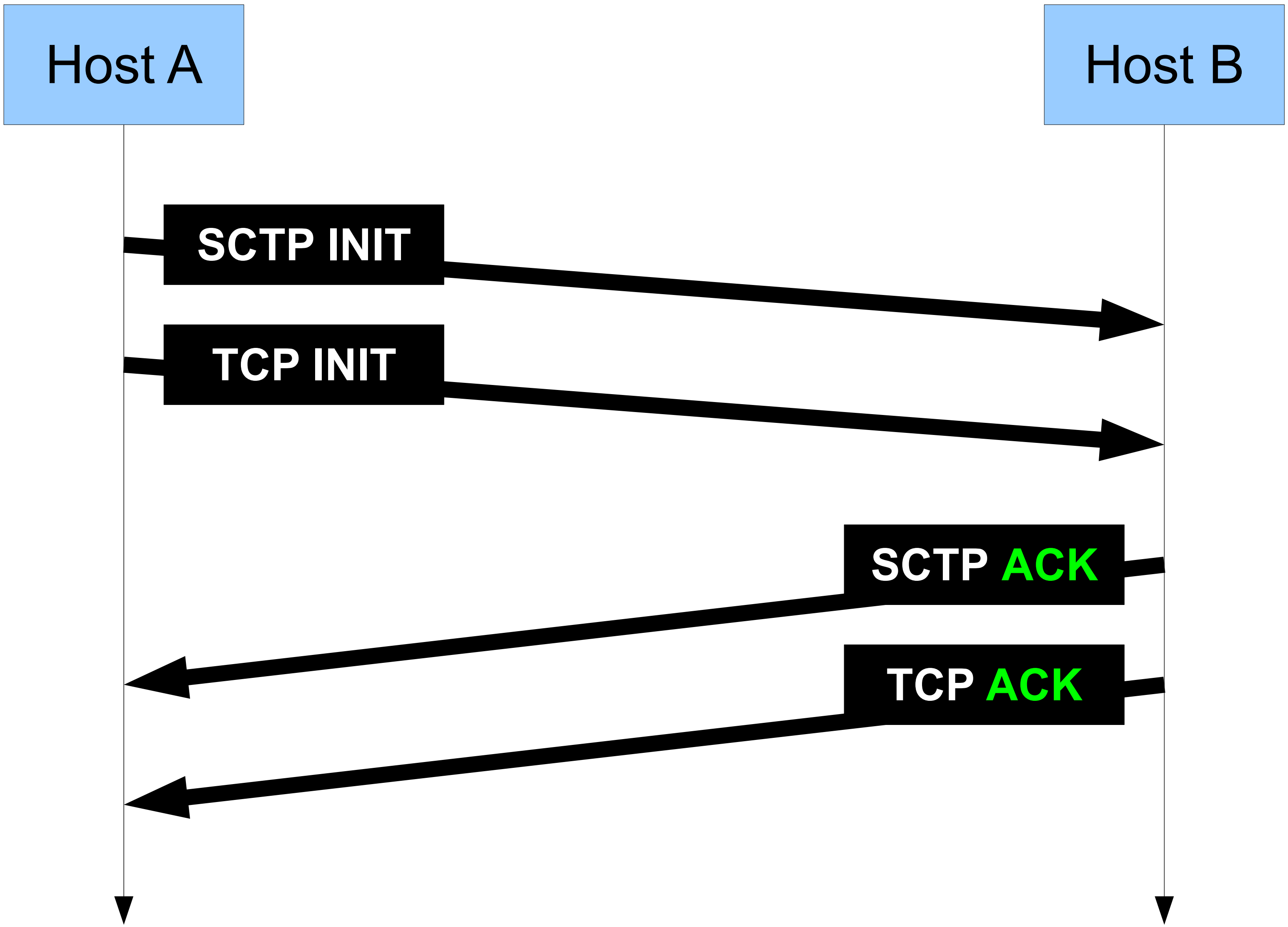
- Failures can incur *timeouts* (e.g., due to NATs)
- ...potentially *compounded* by layering



Approach 2: Try in Parallel

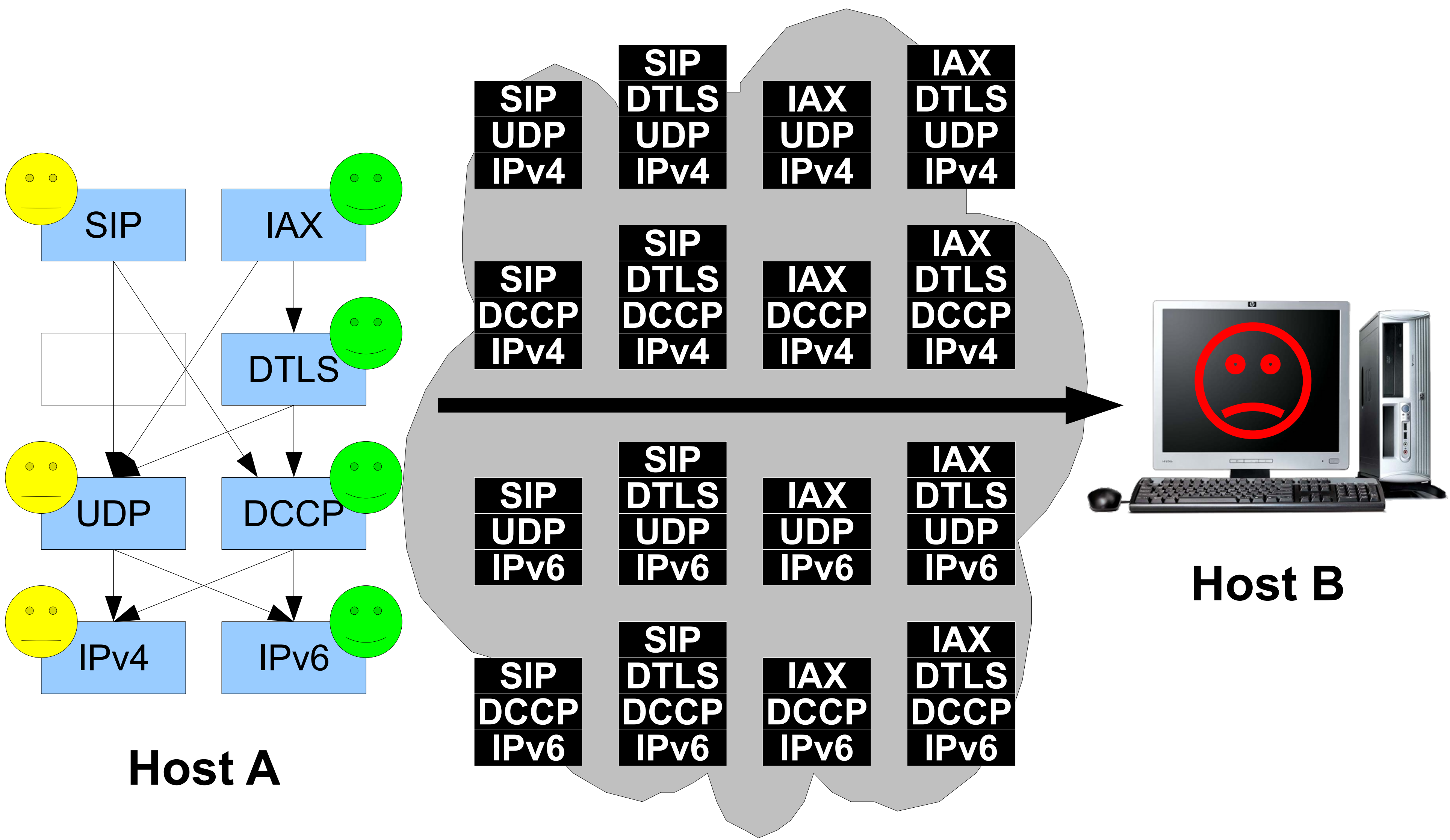


Challenge 2a: Redundant State

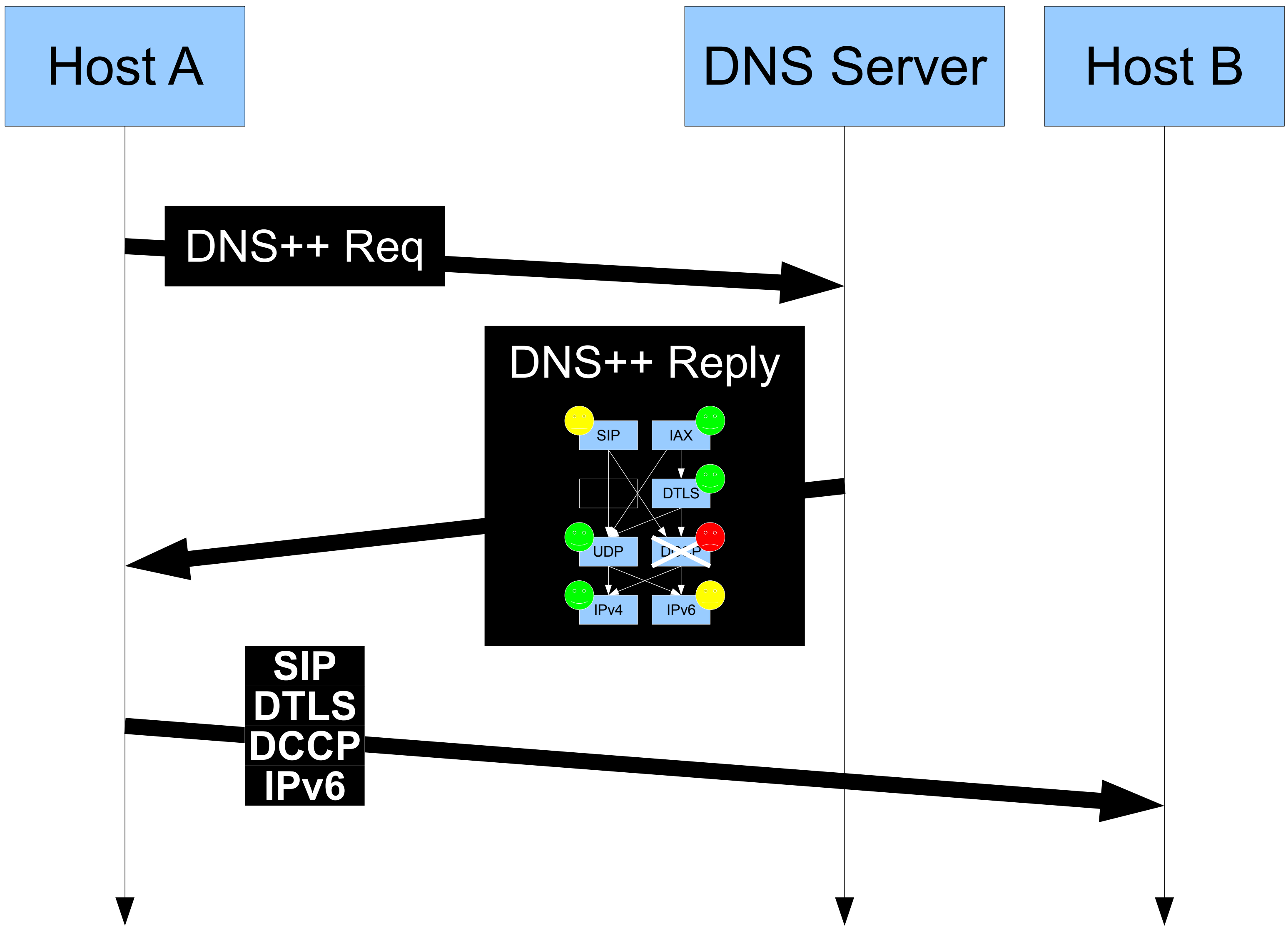


Challenge 2b: Combinations

Layering can lead to explosion of choices



Approach 3:
Out-of-Band Information

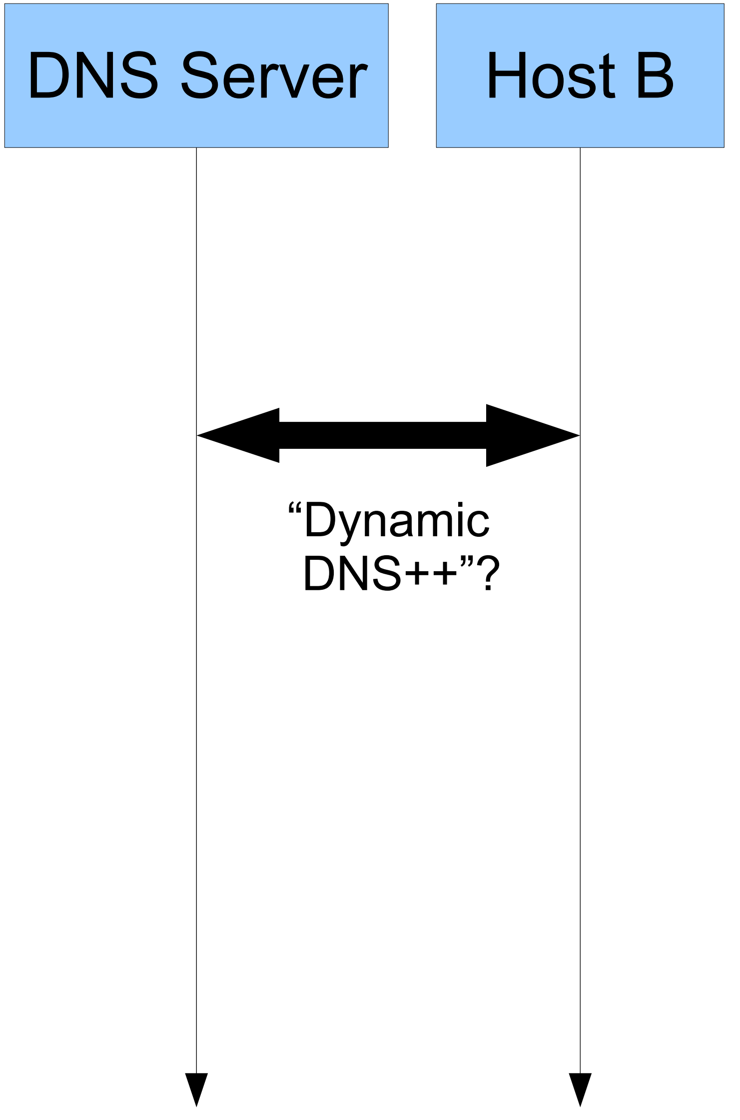


Challenge 3a: Administration

DNS server must know:

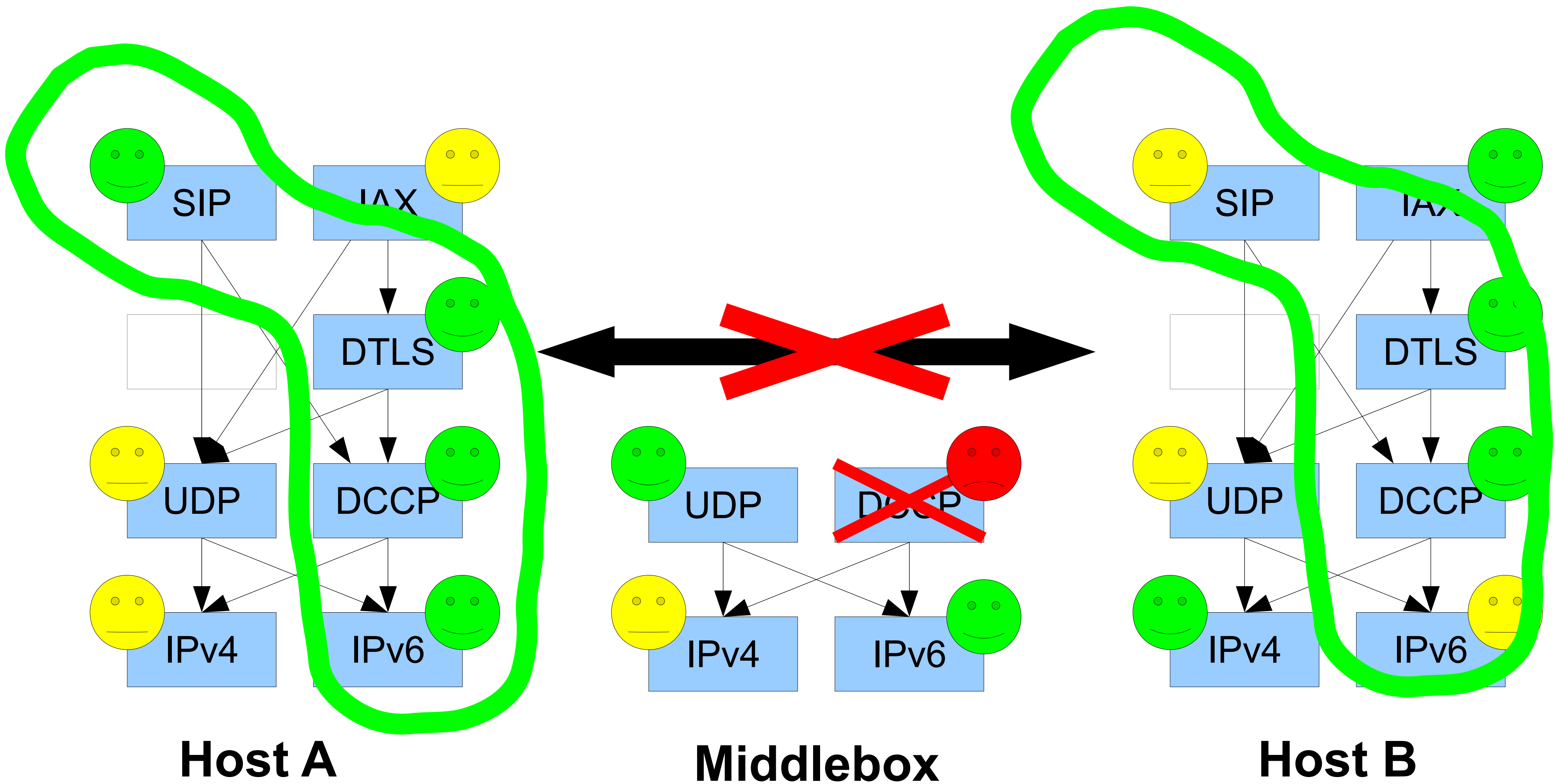
- Name→IP mapping (as before)
- Entire protocol stack supported by Host B
- Protocol options...?

⇒ ***Synchronization Nightmare?***



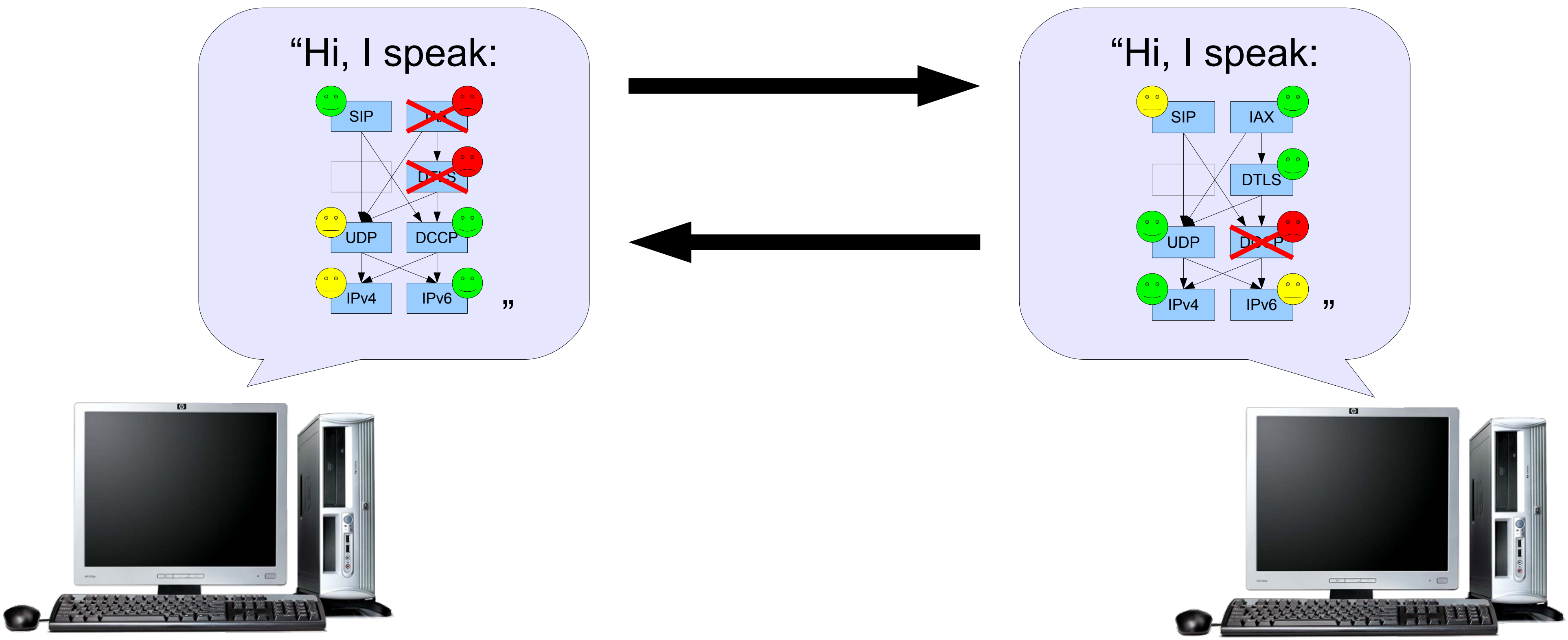
Challenge 3b: E2E Robustness

If endpoints agree on configuration X, *will it work?*



Our Solution: ***Negotiation***

- Hosts explicitly describe possible configurations during initial “meta-communication” exchange, *before* actual communication commences



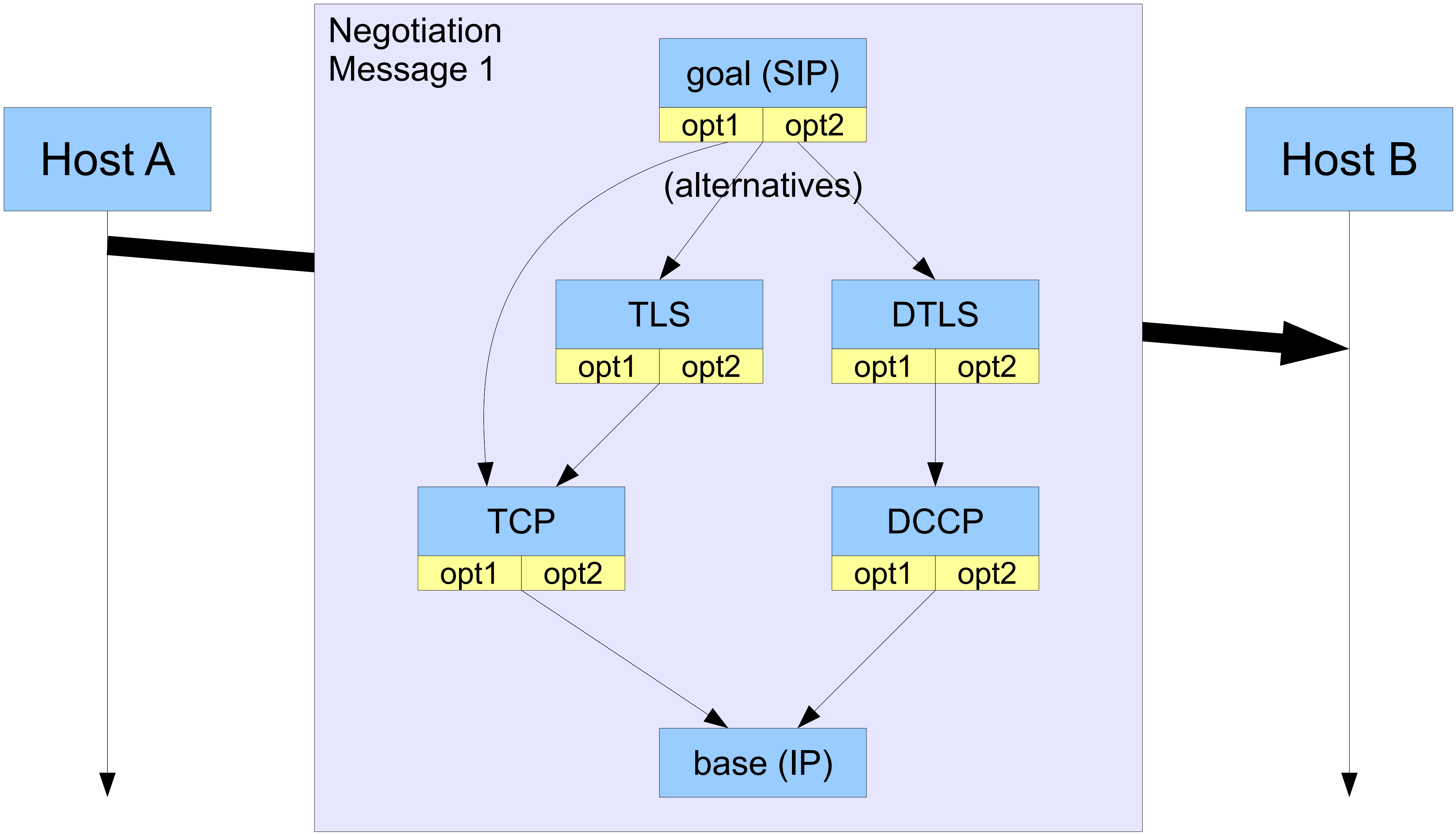
Host A **Host B**

A Model
for Negotiation

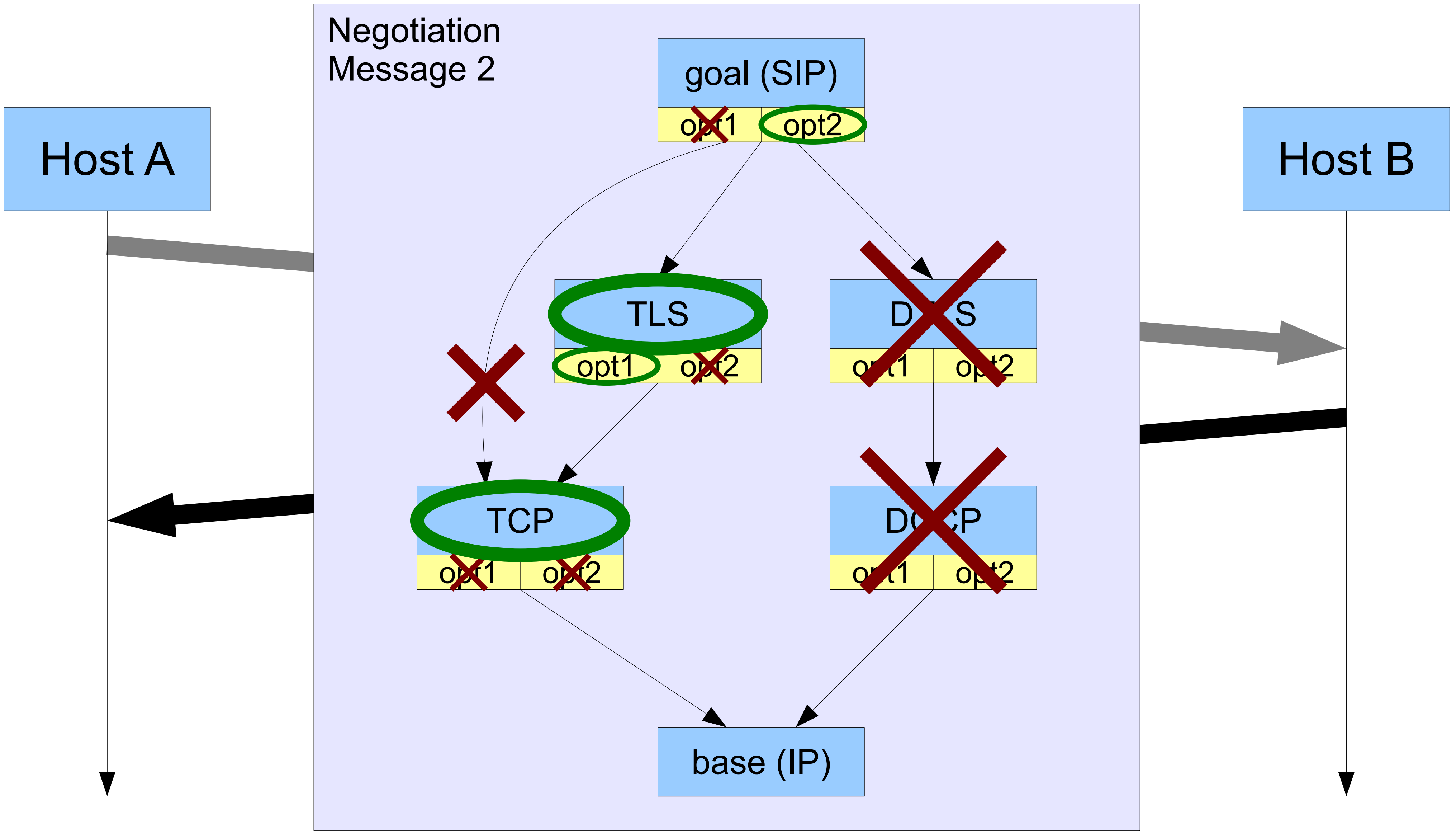
Negotiation Model Overview

1. Initiator sends a **Protocol Graph Proposal**
2. Responder returns **Revised Protocol Graph**
3. (Optional) further protocol graph revision steps
4. Peers commit, **Acknowledge Protocol Graph**
5. Peers communicate via negotiated protocols

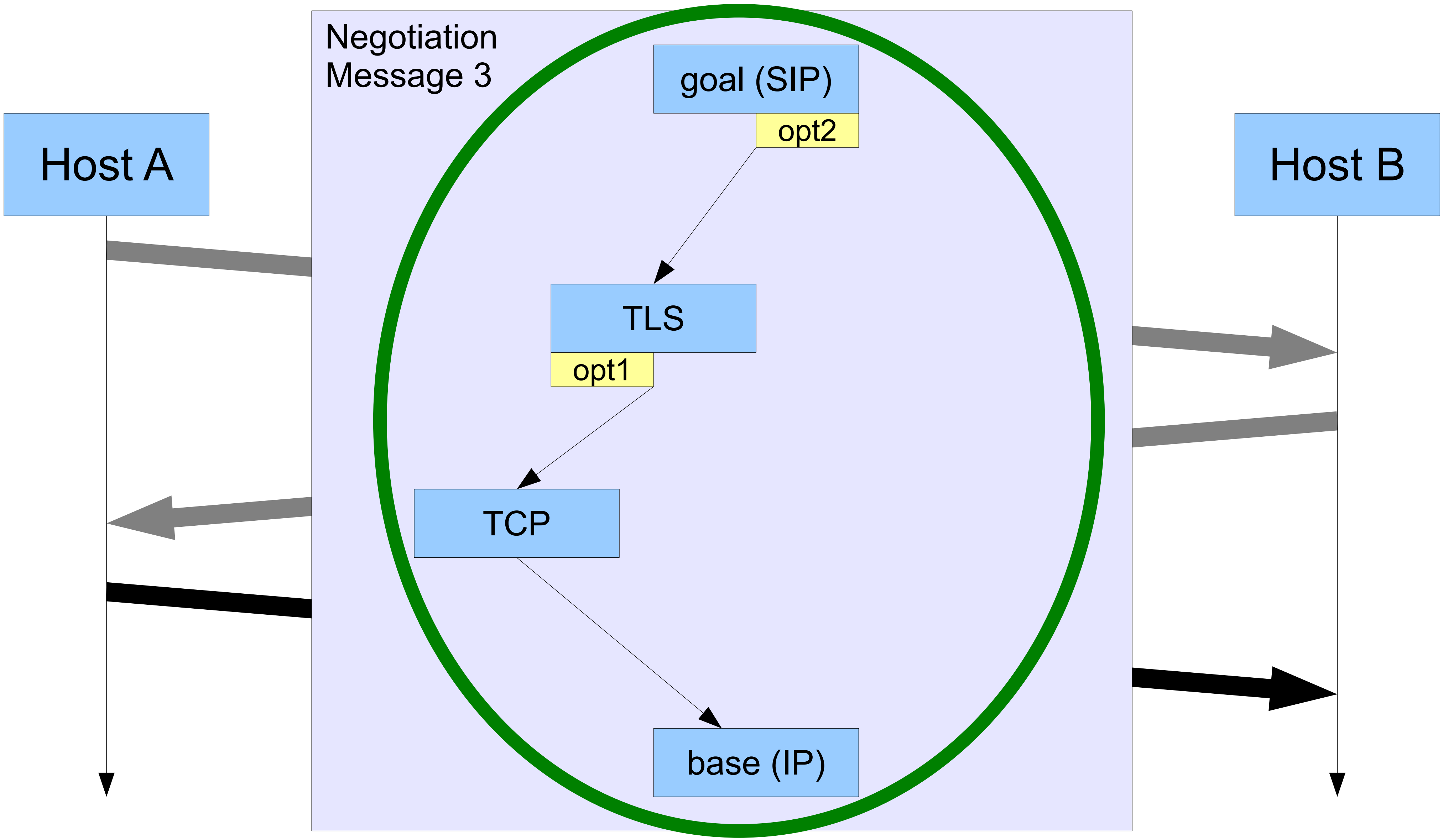
Message 1: Initiator → Responder: **Propose Protocol Graph**



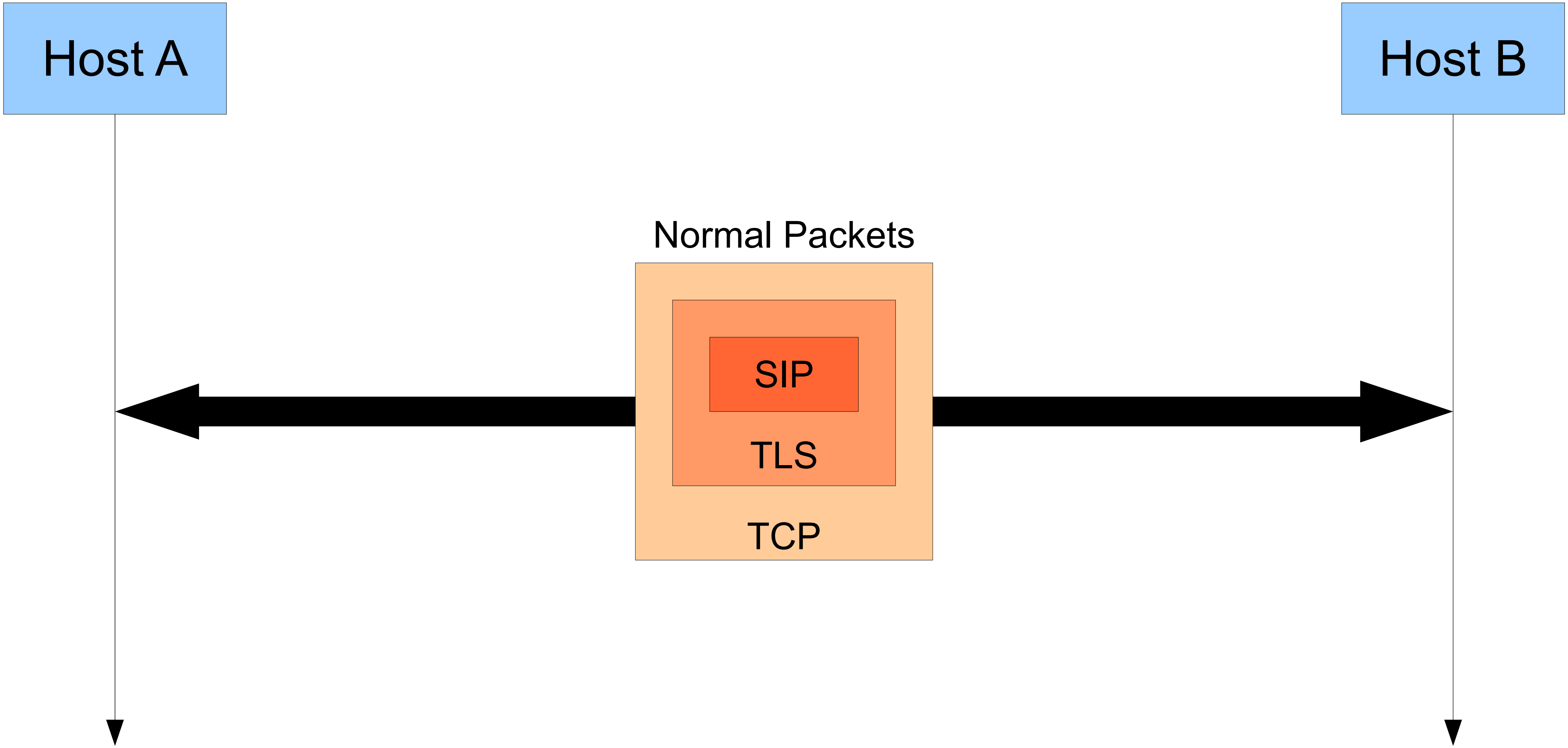
Message 2: Responder → Initiator:
Revise Protocol Graph



Message 3: Initiator → Responder:
Acknowledge Protocol Graph



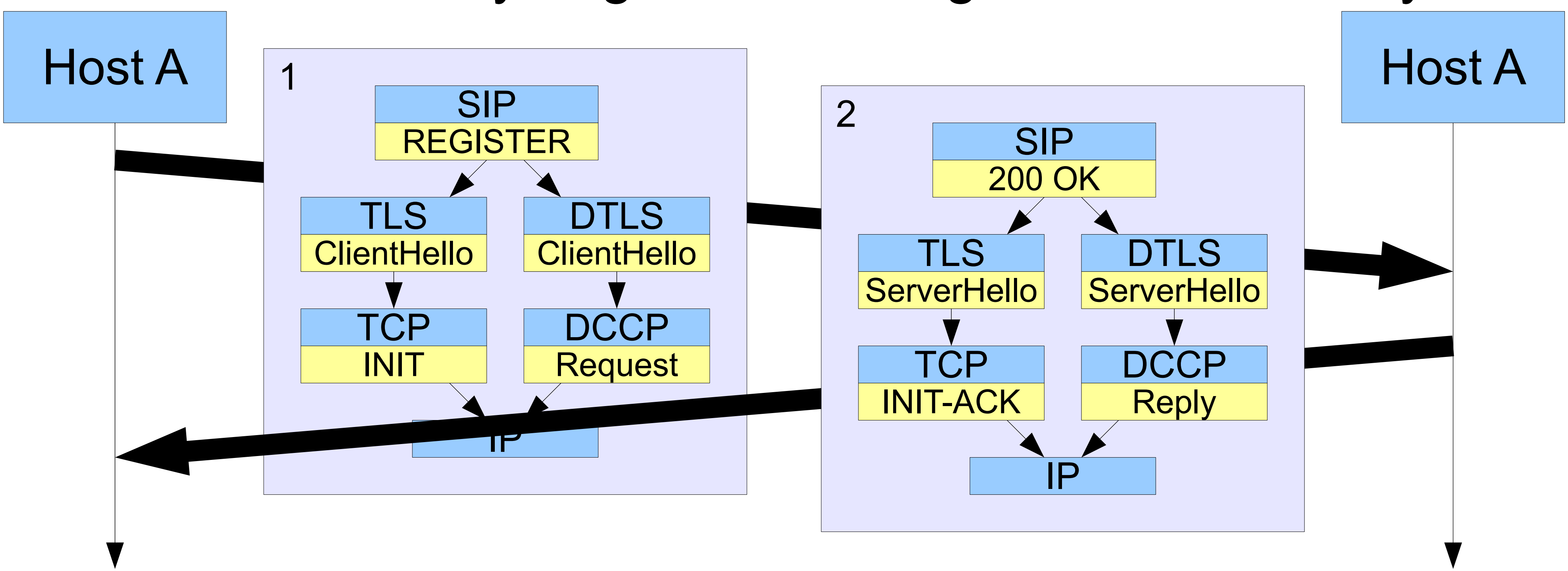
Message 4+:
According to Negotiated Stack



Concurrent Protocol Initialization

Whenever feasible:

- ***embed*** protocol-specific handshake info into graph
- ***run handshakes concurrently*** while negotiating
- **commit** only negotiated configuration atomically



Key Benefits of Negotiation Model

- Supports backward-compatible evolution
  - New smart nodes can fall back on old dumb protocols
- Happens strictly between nodes concerned
  - Users don't have to care (e.g., between http: & https:)
  - Name server administrators don't have to care
- Protocol graph representation scales to handle:
  - Arbitrarily deep protocol stacks
  - Many alternatives per layer
- Setup whole “layer cakes” in minimal # of RTTs
  - Regardless of protocol stack depth

Further Challenges & Extensions

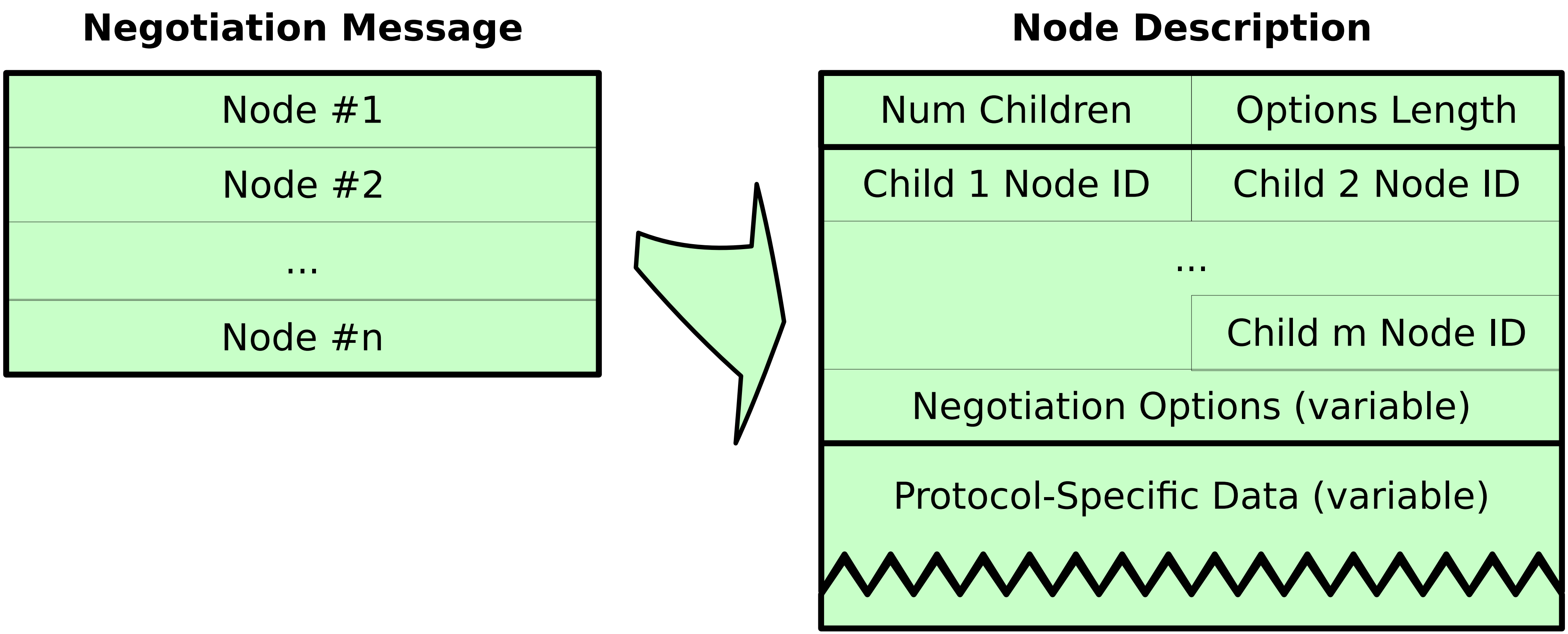
(see paper)

- Multi-Round Negotiation
  - due to dependencies, hiding of alternatives, graph size
- Negotiation Across Multiple Contexts
  - IPv4 vs IPv6, new protocol vs legacy, UDP encapsulation
- Recursive Negotiation
  - negotiate “crypto wrapper” and “contents” concurrently
- Peer-to-Peer Negotiation
  - symmetric peers must converge on a configuration

Negotiation Transport Protocol

How to Express Protocol Graphs?

Negotiation Message Structure:



How to Convey Protocol Graphs?

Negotiation messages might be big:

- Many layers × many alternatives for each to describe
- Embedded protocol-specific data: crypto keys, etc.

Individual graph nodes may be large or small

- Segment large nodes, aggregate small ones into packets

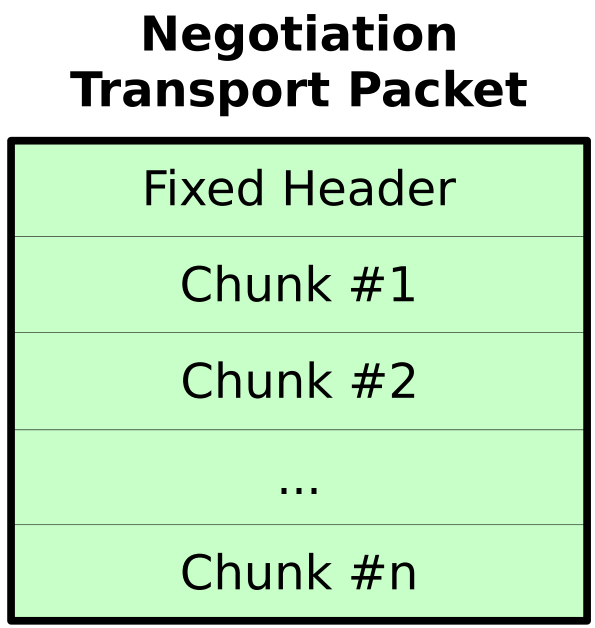
Receiver probably wants only specific nodes

- Efficiently ignore/drop anything it doesn't understand

⇒ Specialized **Negotiation Transport Protocol**

Negotiation Transport: Packet Structure

Fixed header + multiple *chunks* [SCTP]
each describing different graph node



Negotiation Transport

Negotiation packet sequencing permits individual packet ack/retransmit [SST]

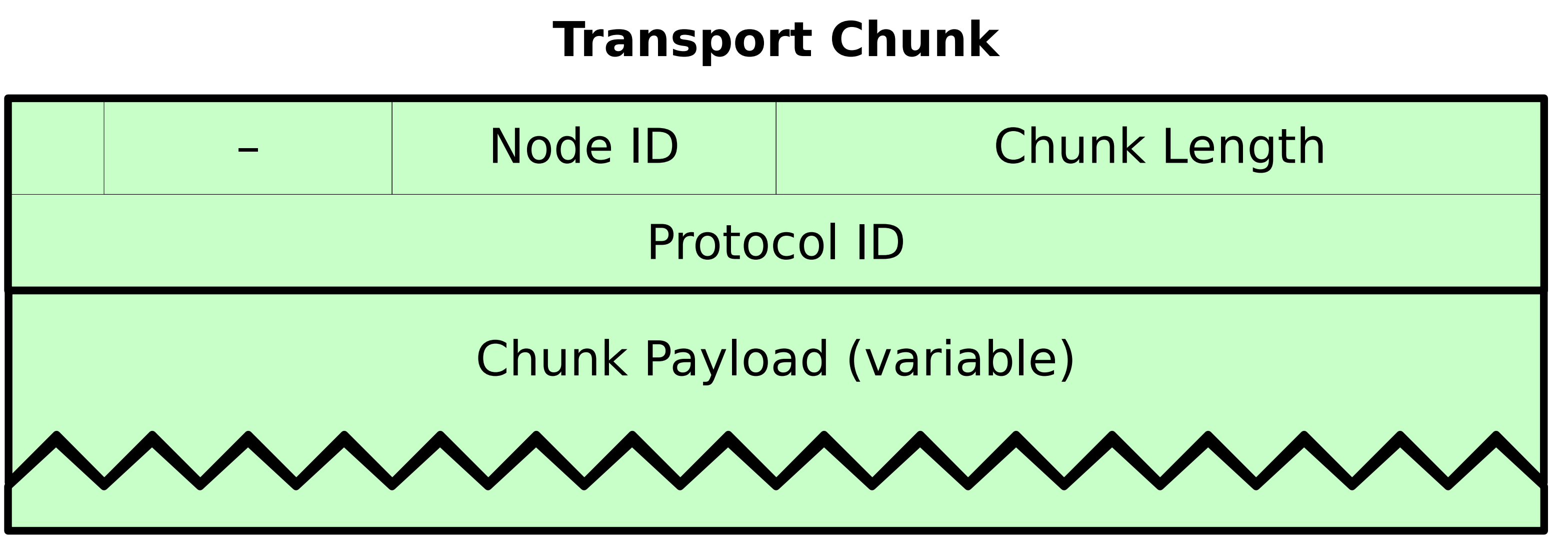
Transport Header

<table>
<tr><td></td><td>Msg Type</td><td colspan="2">Step Number</td><td>Transmit Seq</td></tr>
<tr><td colspan="5">Negotiation Protocol Magic Cookie</td></tr>
<tr><td colspan="5">Negotiation Transaction ID</td></tr>
<tr><td colspan="3">–</td><td>AckCt</td><td>Ack Seq</td></tr>
</table>

Negotiation Transport

Each chunk describes [part of] a graph node

- Receiver can ack & discard *all* chunks for unknown protocols without storing *any*



Not needed

Let negotiated protocol worry about:

- Connection state machines
- Application-friendly semantics (e.g., streams)
- Flow control
- Congestion control (beyond slow-start)
- ...

Discussion, Conclusion

What Doesn't (Really) Work

- Encoding protocol stacks in names
  - Non-transparent to user; compatibility hell
- Try alternatives serially & fall back
  - Delay/timeout hell
- Probe alternatives in parallel
  - Redundant protocol instances; combinatorial hell
- Encode alternatives in DNS responses
  - Not end-to-end robust; administrative hell

What *Might* Work

Explicit In-Band Negotiation:

- Get user & third parties out of the loop
- Describe alternatives in compact protocol graphs
- Handshake deep layer cakes concurrently
- Receiver stores only what he understands & wants

"T*ng*: Transport Next Generation" project

`http://bford.info/tng/`

Support: NSF FIND grants CNS-0916678 and CNS-0916413