TRIGTRAN
Strawperson Framework

draft-dawkins-trigtran-framework-00.txt

Spencer Dawkins
spencer_dawkins@yahoo.com
Carl E. Williams
carlw@mcsr-labs.org
Alper E. Yegin
Alper@docomolabs-usa.com

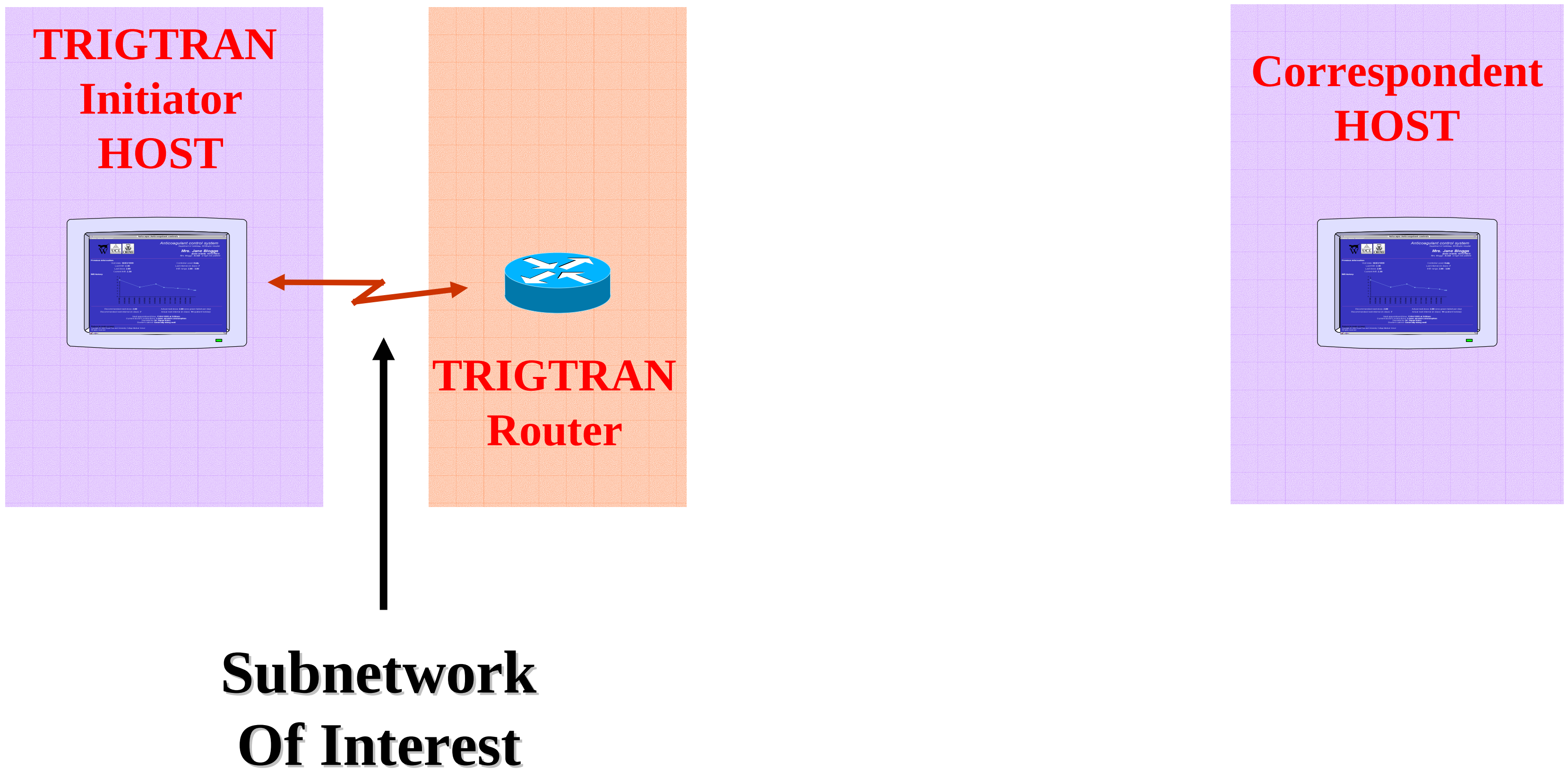
So, you already have a Framework?

- No.
- We’re exploring an approach
- … because we’re looking for fatal flaws
- … like “can we actually generate triggers?”
- … and “can we actually send them?”
- This approach helped us ask these questions
- … but “Connectivity Restored” doesn’t need it
- … so Framework should be on hold for now

Framework Basics

- Accommodate multiple transports
  - Focus on TCP, don't break SCTP – others?
- Initiator/Correspondent model
  - Focus on access links
  - Focus on single-homed Initiators
- Protocol flow
- Canonical triggers?
- Canonical responses?
- Notification protocol mechanisms?
- Canonical security considerations?

Minimal TRIGTRAN Architecture



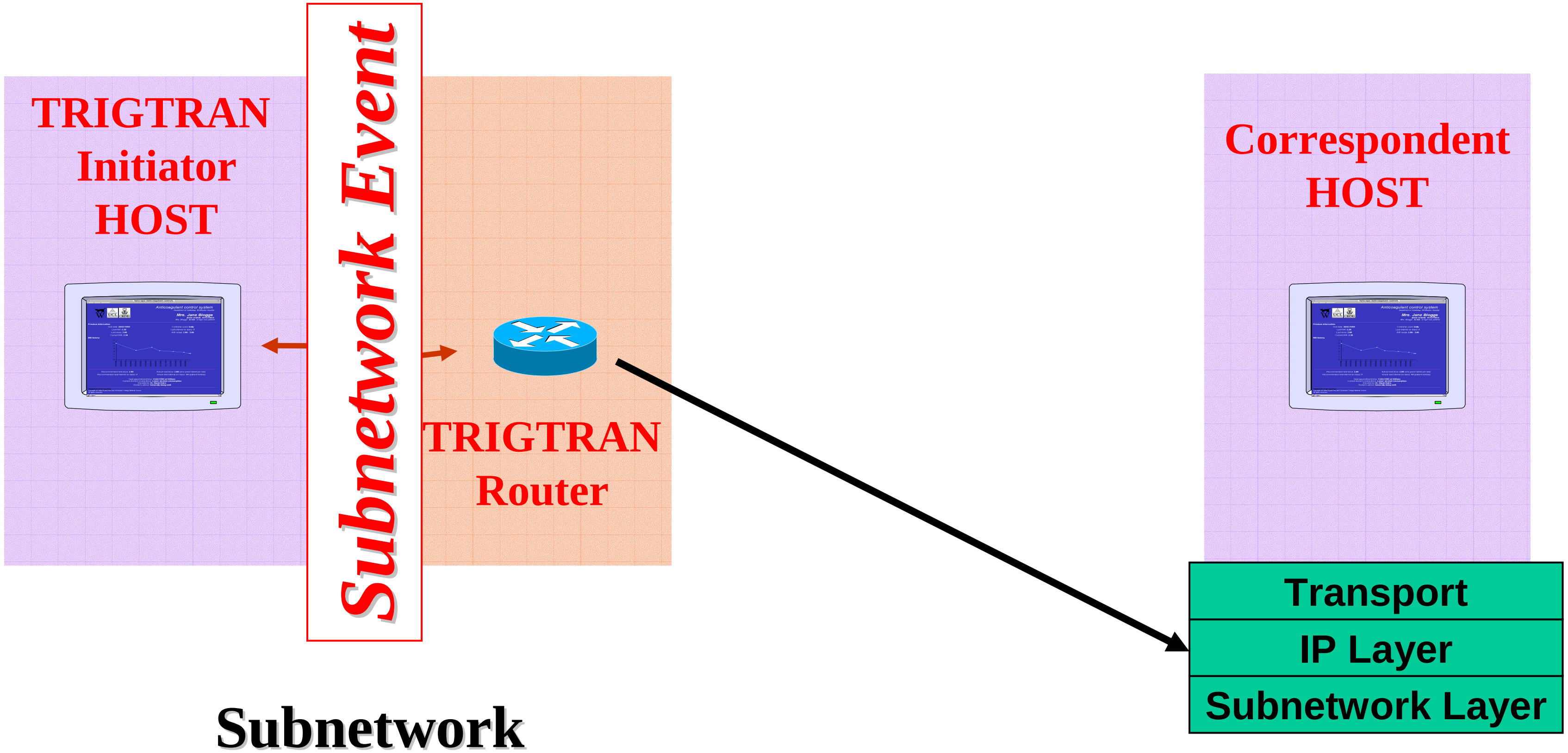
TRIGTRAN Initiator HOST

TRIGTRAN Router

Correspondent HOST

Subnetwork Of Interest

Minimal TRIGTRAN Functionality



TRIGTRAN Initiator HOST

Subnetwork Event

TRIGTRAN Router

Correspondent HOST

Transport

IP Layer

Subnetwork Layer

Subnetwork Event Here

Notify Transport Here

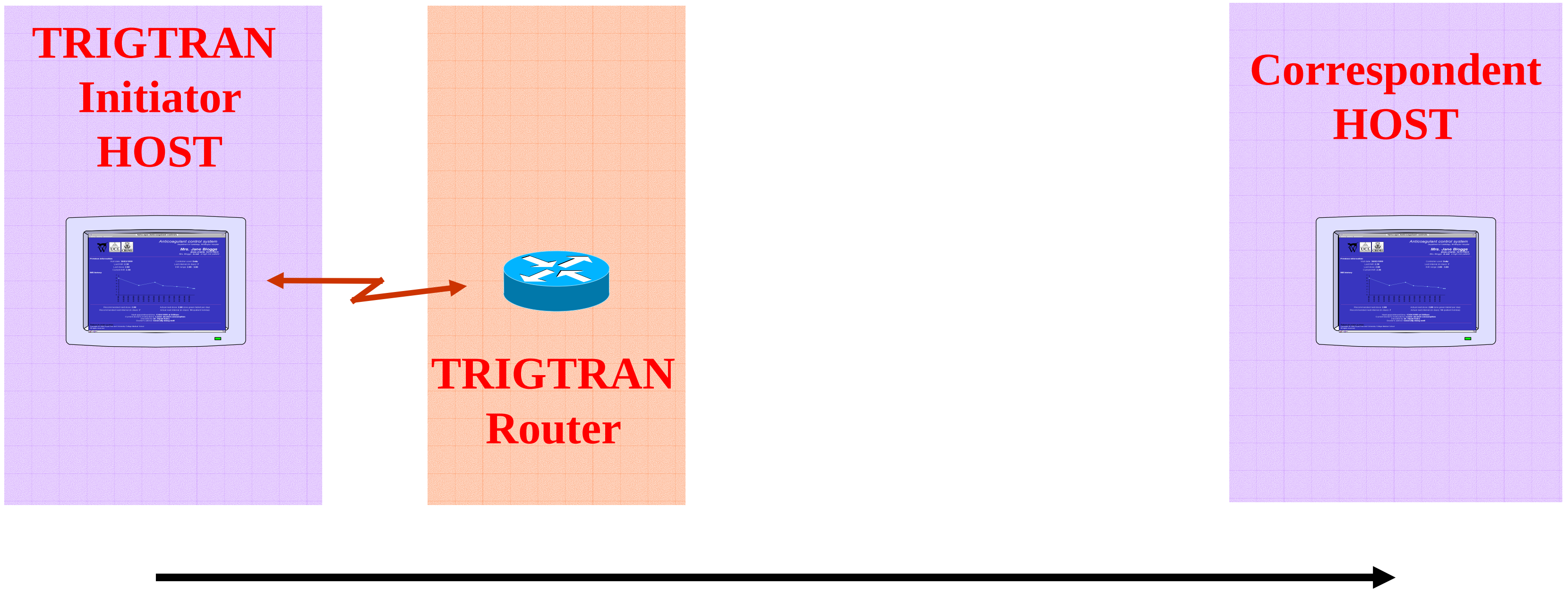
Focus on Access Links

- Many problematic links are access links
- Can’t guarantee core routers see all packets
- Core network will reroute anyway
- Avoid core network scaling problem
- Access network may have incentive to deploy
- Core network does not have this incentive

Focus on Single-homed Initiators

- Maps to one class of problematic subnetworks
  - Wide-Area Wireless Networks
- Avoid “fan-in” problem at correspondent host
- Unambiguous notifications are most valuable
- New interface -> new bandwidth anyway

Protocol Flow - Initiation



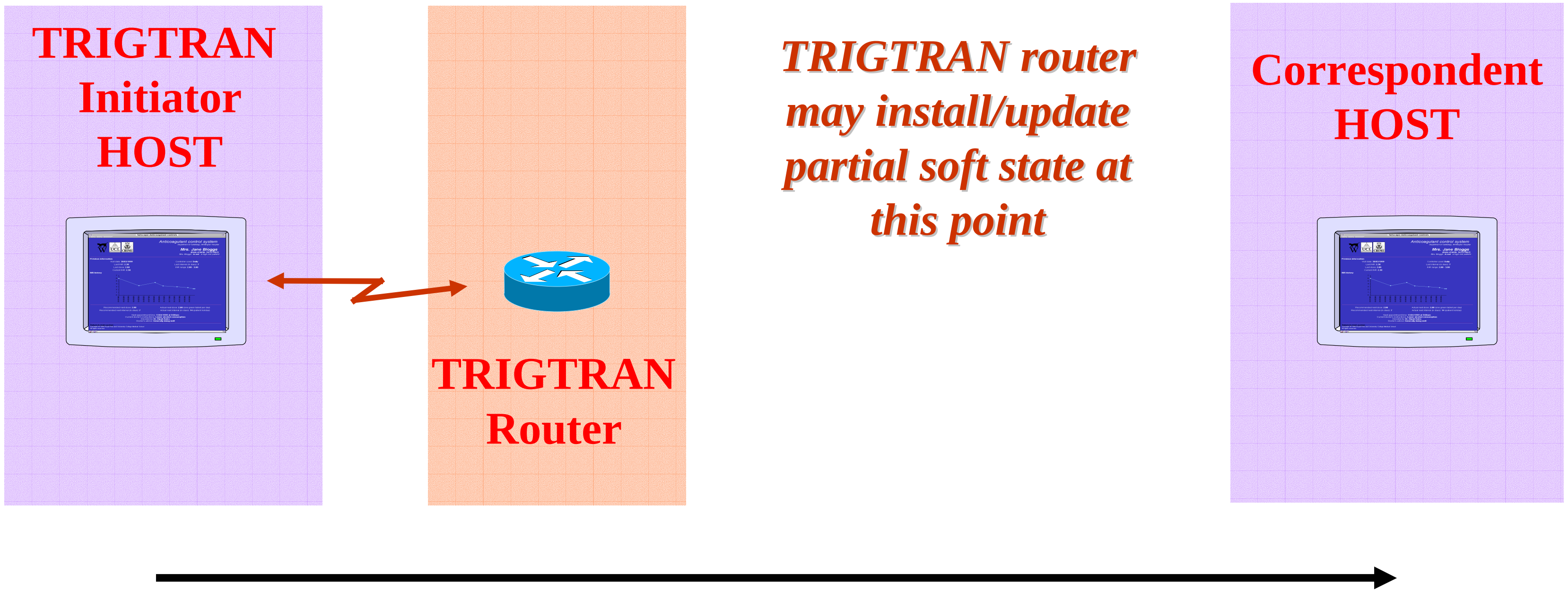
TRIGTRAN Initiator HOST

TRIGTRAN Router

Correspondent HOST

Arbitrary packet with TRIGTRAN Initiate bit set

Router Action - Initiation



TRIGTRAN Initiator HOST

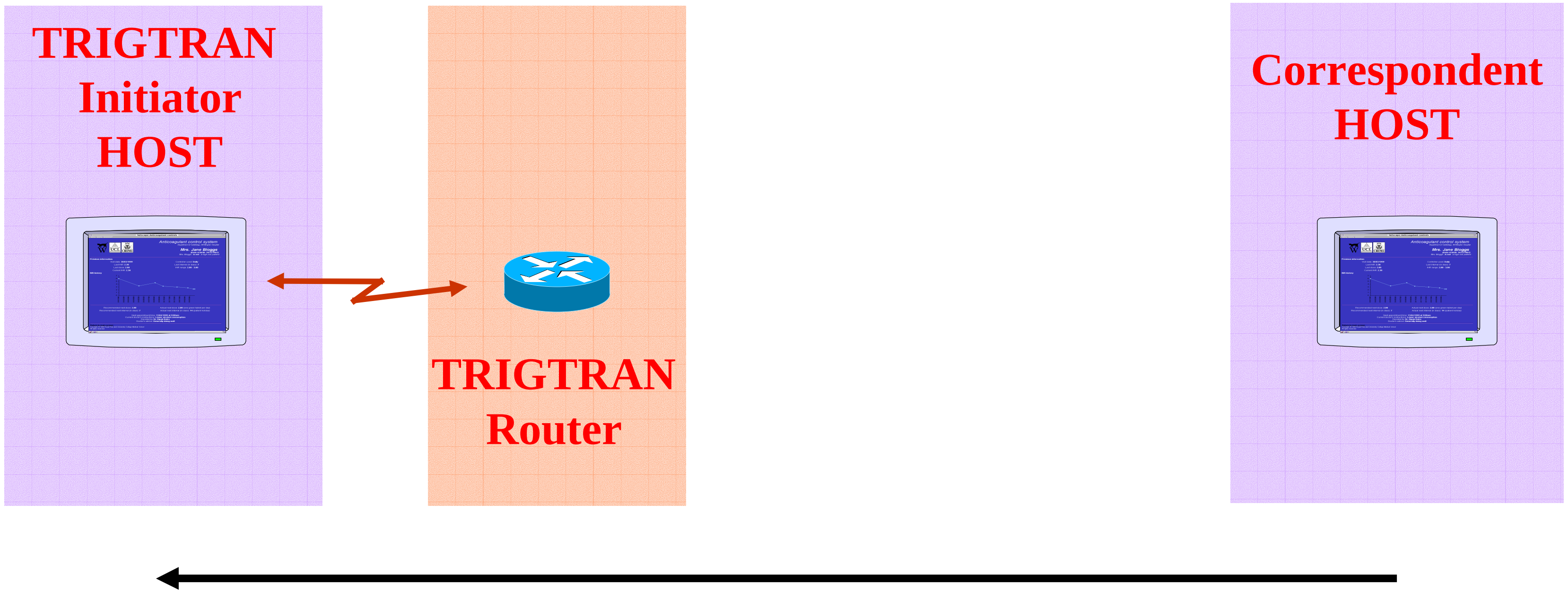
TRIGTRAN Router

TRIGTRAN router may install/update partial soft state at this point

Correspondent HOST

Arbitrary packet with TRIGTRAN Initiate bit set

Protocol Flow - Request



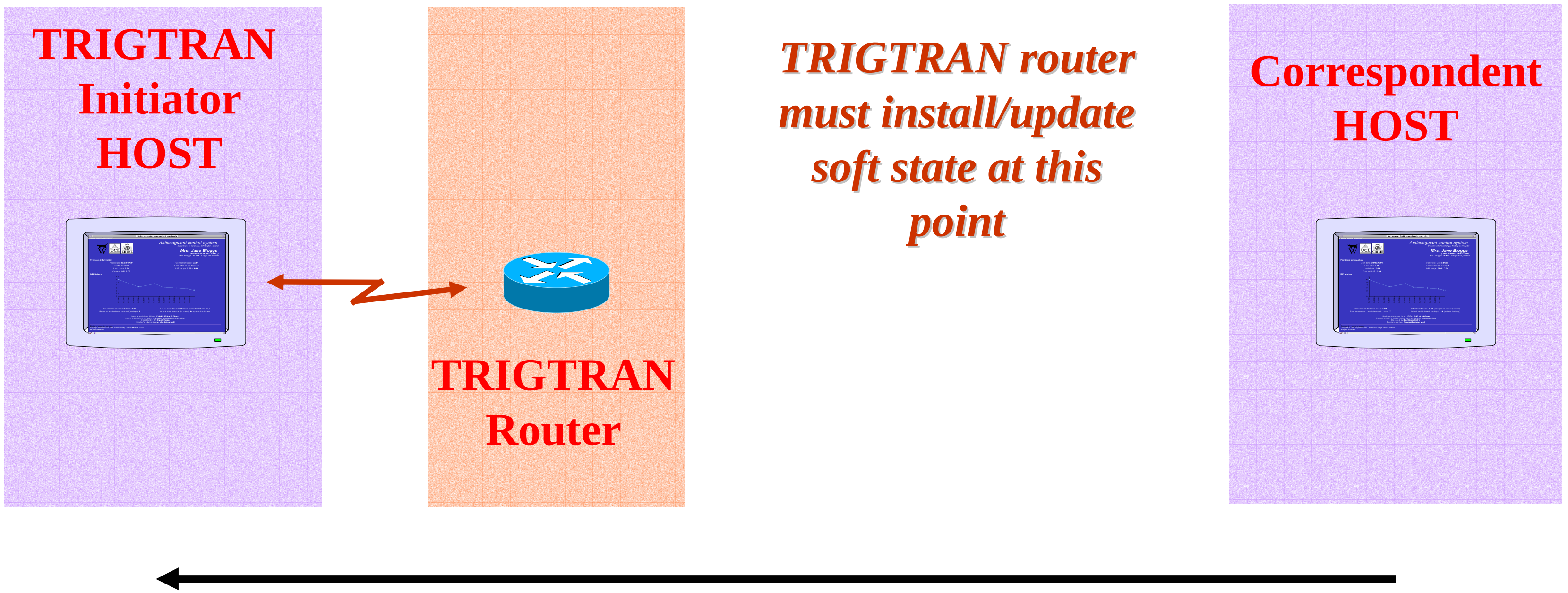
TRIGTRAN Initiator HOST

TRIGTRAN Router

Correspondent HOST

Arbitrary packet
with TRIGTRAN
Initiate and Request bits set

Router Action – Request



TRIGTRAN Initiator HOST

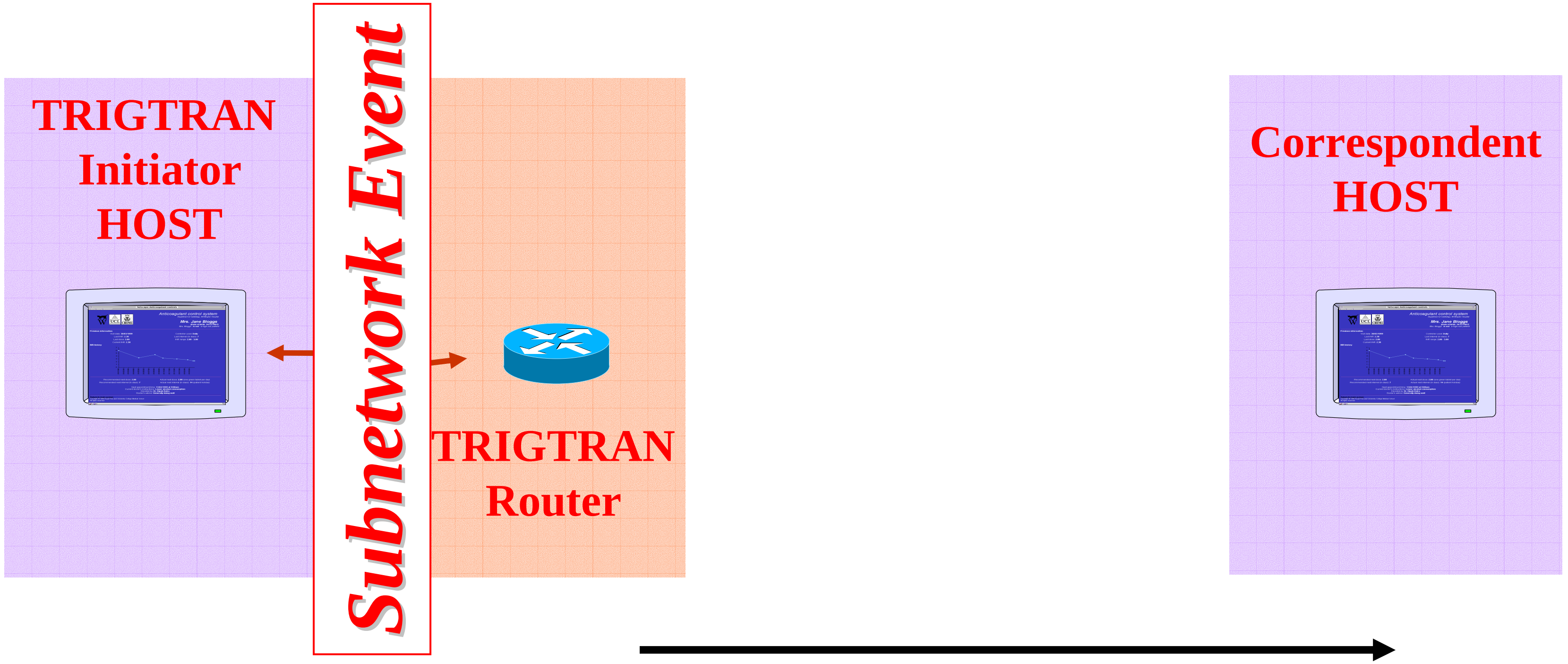
TRIGTRAN Router

TRIGTRAN router must install/update soft state at this point

Correspondent HOST

Arbitrary packet
with TRIGTRAN
Initiate and Request bits set

Protocol Flow - Notification



TRIGTRAN Initiator HOST

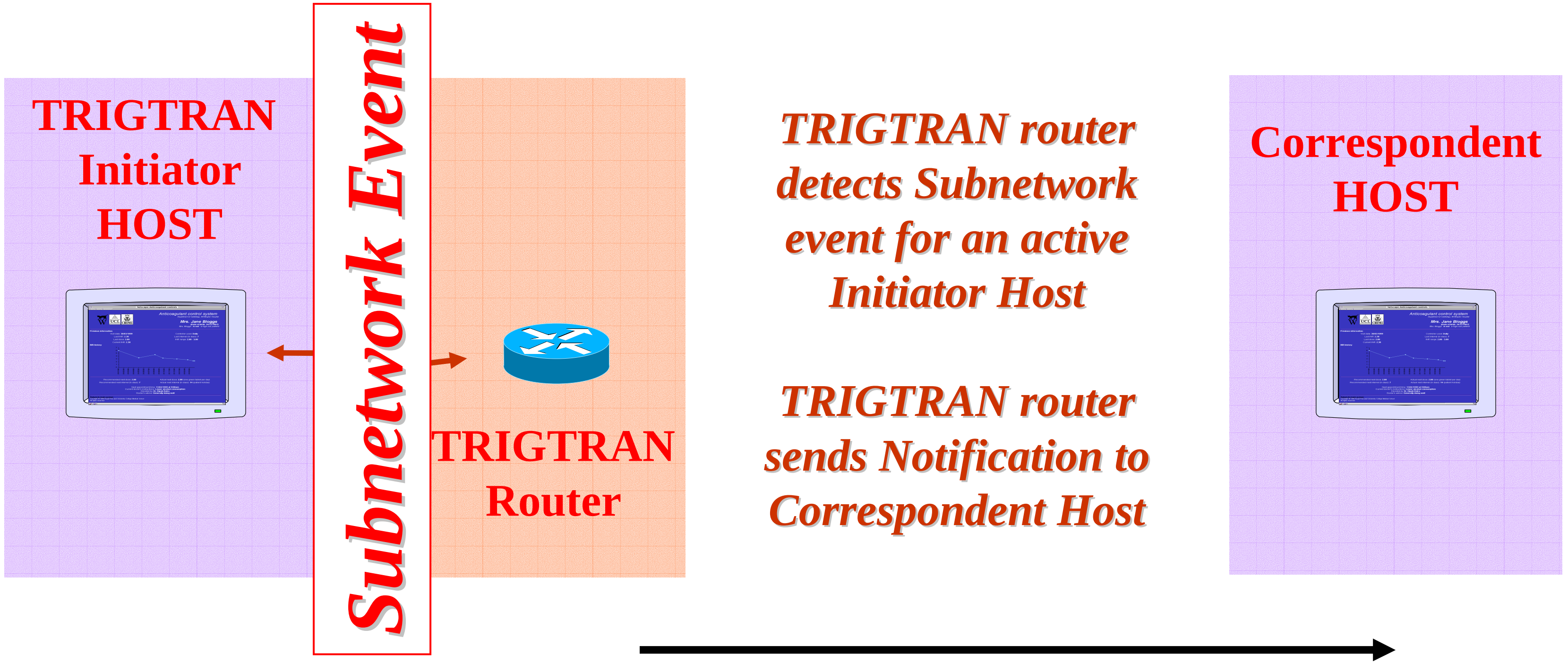
Subnetwork Event

TRIGTRAN Router

Correspondent HOST

TRIGTRAN Notification
from router to Correspondent Host

Router Action - Notification



TRIGTRAN router detects Subnetwork event for an active Initiator Host

TRIGTRAN router sends Notification to Correspondent Host

TRIGTRAN Notification from router to Correspondent Host

Canonical Triggers?

- One proposal for minimal set of events:
  - Connectivity Interrupted
  - Connectivity Restored
  - Packets Discarded by subnetwork, not due to congestion
- More ambitious ("research") events:
  - Sub-network path changes ("horizontal handoff")
  - Packet corruption loss
  - Non-congestion loss
  - Nominal sub-network bandwidth change
- *Current Framework does not include "ambitious" events*

Notification Protocol Mechanisms?

- We’re dealing with a huge issue here
- ICMP message is right answer conceptually
  - A less ambiguous/more flexible Source Quench?
- But is it deployable?
  - Old implementations, NATs, Firewalls, etc.
- Is a new UDP message likely to be better?
- DCCP flows too heavyweight?
  - Number of flows for an access router?
  - Not a connection, but still need per-flow state
- TCP is right for end-to-end TCP Kickstart…

Canonical Security Considerations?

- Non-starter
  - Assume security association between TRIGTRAN access router and arbitrary correspondent host somewhere on the Internet
- First attempt at solving this problem
  - Limit TRIGTRAN to advisory role
  - If you have notifications and ACKs, believe ACKs!
  - No new transport behavior
- Alternative choice?
  - Explore Purpose-Built Keys framework
  - No identity component – only spoof-resistance
  - MIGHT allow different different class of responses

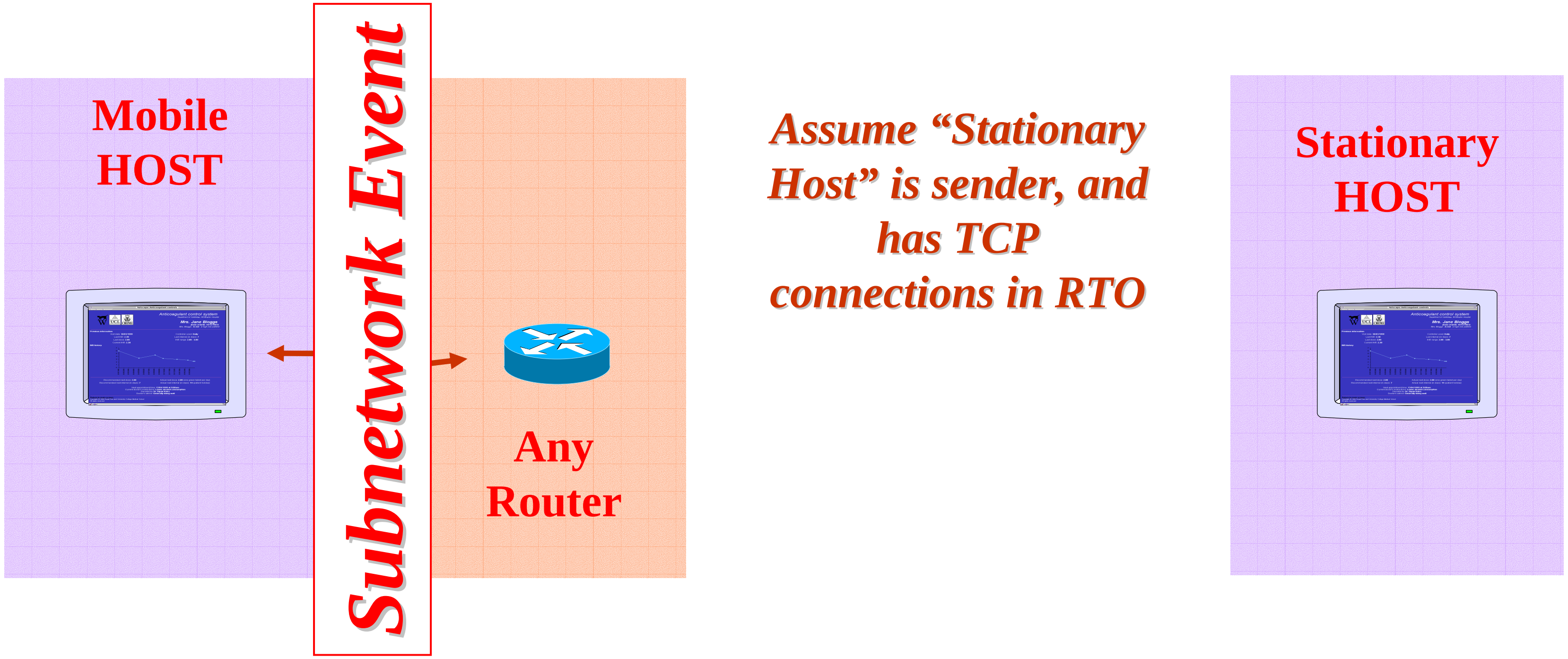
Canonical DOS Considerations?

- Assuming strawperson security considerations proposal (advisory)
- Clearing Initiate/Request bits not interesting
  - Gives current transport behavior
- Setting Initiate/Request bits not very interesting
  - Requires attacker on both sides of router to install state in router
- Forged Connectivity Interrupted not interesting
  - Believe end-to-end ACKs if they come
- Forged Connectivity Restored not interesting
  - Probe once during Connectivity Interrupted, then normal loss processing
- Forged Packets Discarded not interesting
  - Resend packets once during loss event, then normal loss processing
- DOS flooding of TRIGTRAN routers not interesting
  - No worse than any Router Alert flooding attack
  - Reverts to current transport behavior during flooding attacks - but who cares?

Feedback in the halls so far

- “Trigger” name still seems to give the wrong message
- Need to be clear about timeframes – think “five years”
- Out-of-band notifications are very problematic
  - ICMP blocks, UDP blocks, firewalls, NATs, ALGs, etc.
- “Packets Discarded” ambiguous – looks like “handoff”
- “Connectivity Interrupted” response isn’t clear
  - Transports that retry more persistently? Or give up sooner?
- Even “Connectivity Restored” requires TCP change
- Sending notifications all the time is simpler
  - No bits, no “initiator/requestor”, no decisions
  - And, if we’re headed for general deployment, maybe right idea
- Need to be clear about topology aspects of DoS attacks

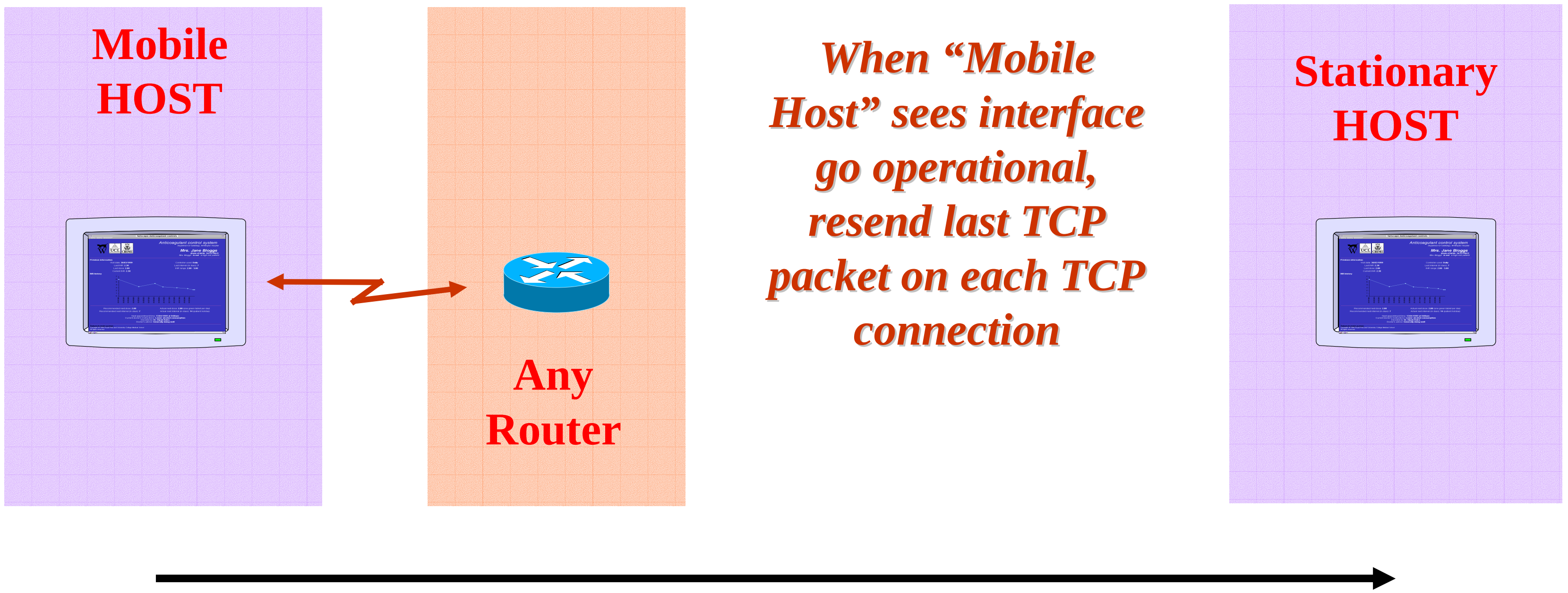
Kicking TCP



Assume "Stationary Host" is sender, and has TCP connections in RTO

Phil Karn, "Kicking TCP", March 2000 PILC list posting

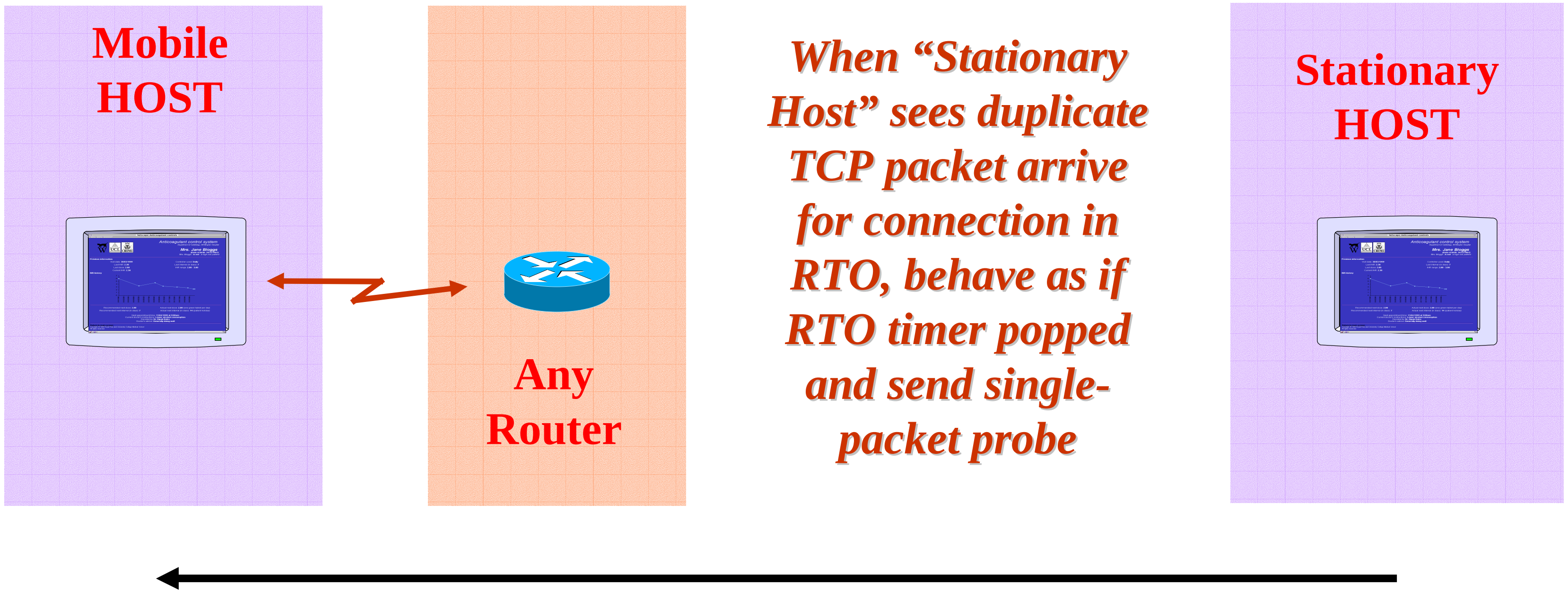
Kicking TCP



When "Mobile Host" sees interface go operational, resend last TCP packet on each TCP connection

Last TCP packet previously sent on this connection

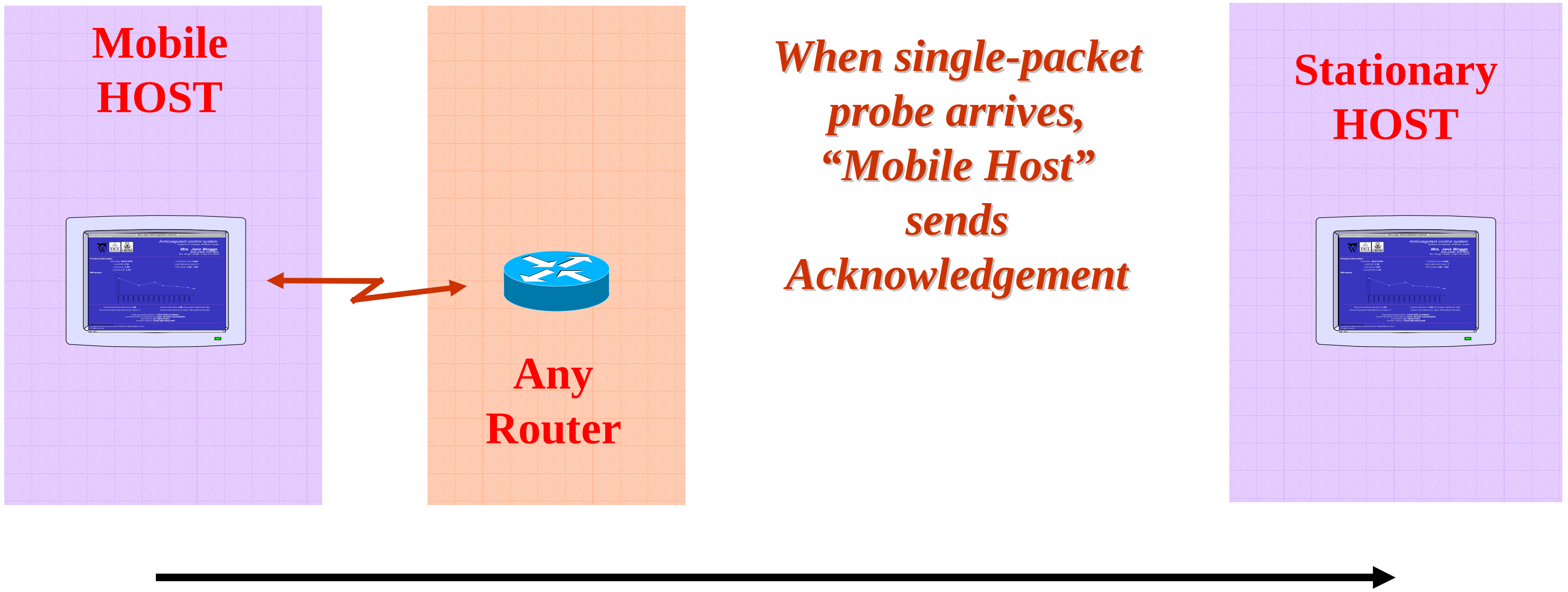
Kicking TCP



When "Stationary Host" sees duplicate TCP packet arrive for connection in RTO, behave as if RTO timer popped and send single-packet probe

Single-packet probe on this connection

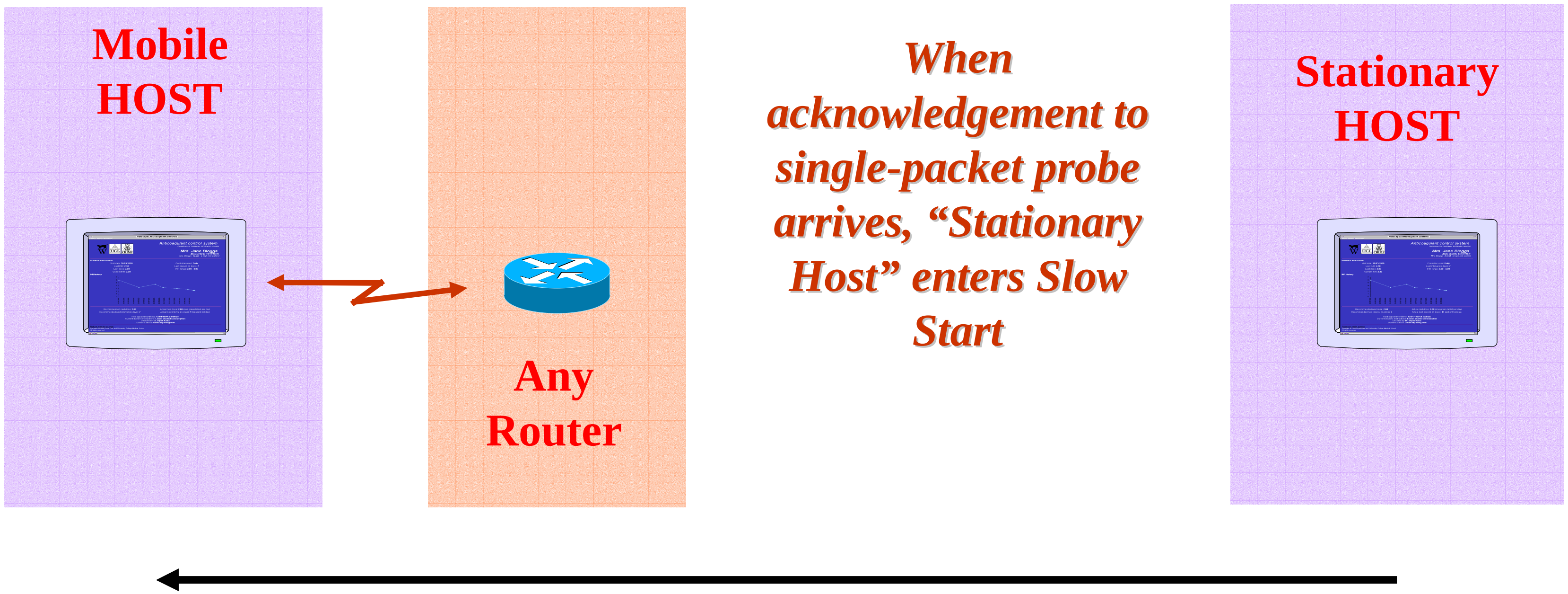
Kicking TCP



When single-packet probe arrives, “Mobile Host” sends Acknowledgement

Acknowledgement for Single-packet probe on this connection

Kicking TCP



When acknowledgement to single-packet probe arrives, “Stationary Host” enters Slow Start

Normal transmission resumes with ACK clocking

If We Really “Kick TCP”

- Need a small change to TCP for duplicate packets received on RTO connections
- Don’t need modifications to routers
- No router per-connection state
- “Last packet”goes anywhere TCP was going
  - No (more) NAT, firewall, ALG considerations
- Safe (no response to probe is no-op)
- Recovers RTOed TCP sooner
  - Could be up to 30 seconds sooner, with a human in the loop
- Need to define similar facility for other transports?
- Can’t reuse this mechanism for any other trigger
  - Likely would require explicit notification, maybe edge-to-end