Implementing Packet Dynamic Awareness in Argus

Carter Bullard
QoSient, LLC
carter@qosient.com

John Gerth
Stanford University
gerth@stanford.edu

FloCon 2012
Austin, Texas
Jan 10, 2012

Guha, Kidwell, Barthur, Cleveland, Gerth and Bullard: A Streaming Statistical Algorithm for Detection of SSH Keystroke Packets in TCP Connection

ICS-2011 - 12th INFORMS Computing Society, Monterey, pp. 73-91

ISBN 978-0-9843378-1-1 DOI 10.1287/ics.2011.0036

- New ideas are needed to improve computer network defense with regard to scalable attack attribution (AA) and situational understanding (SU)
- Packet Dynamics are a new set of flow-level variables, including, but not limited to, inter-packet arrival, protocol state transition, one-way and round trip latency, transaction duration and session arrival times.
- Packet Dynamic metrics can provide new additional awareness needed to deliver flow based anomaly detection for a number of difficult issues.
- We will discuss its implementation in Argus, and how we use Packet Dynamics in near-realtime cyber-situational awareness systems.

Packet Dynamics

- Dynamics generally refer to property change over time
- Packet dynamics (PD) have been described as connection-level properties such as packet shaping, ordering, loss and delay, and how they change over time.

V. Paxon, End-to-end internet packet dynamics. IEEE/ACM Trans. Netw. 7, 3 (June 1999).

- Packet dynamics include many properties such as inter-packet arrival times, packet burst behavior, protocol state transition times, latency, and packet size frequency.
- New understanding of packet dynamics can provide additional awareness needed for successful network path assurance, man-in-the-middle detection, stepping stone detection, replay and attribution.

Packet Dynamics

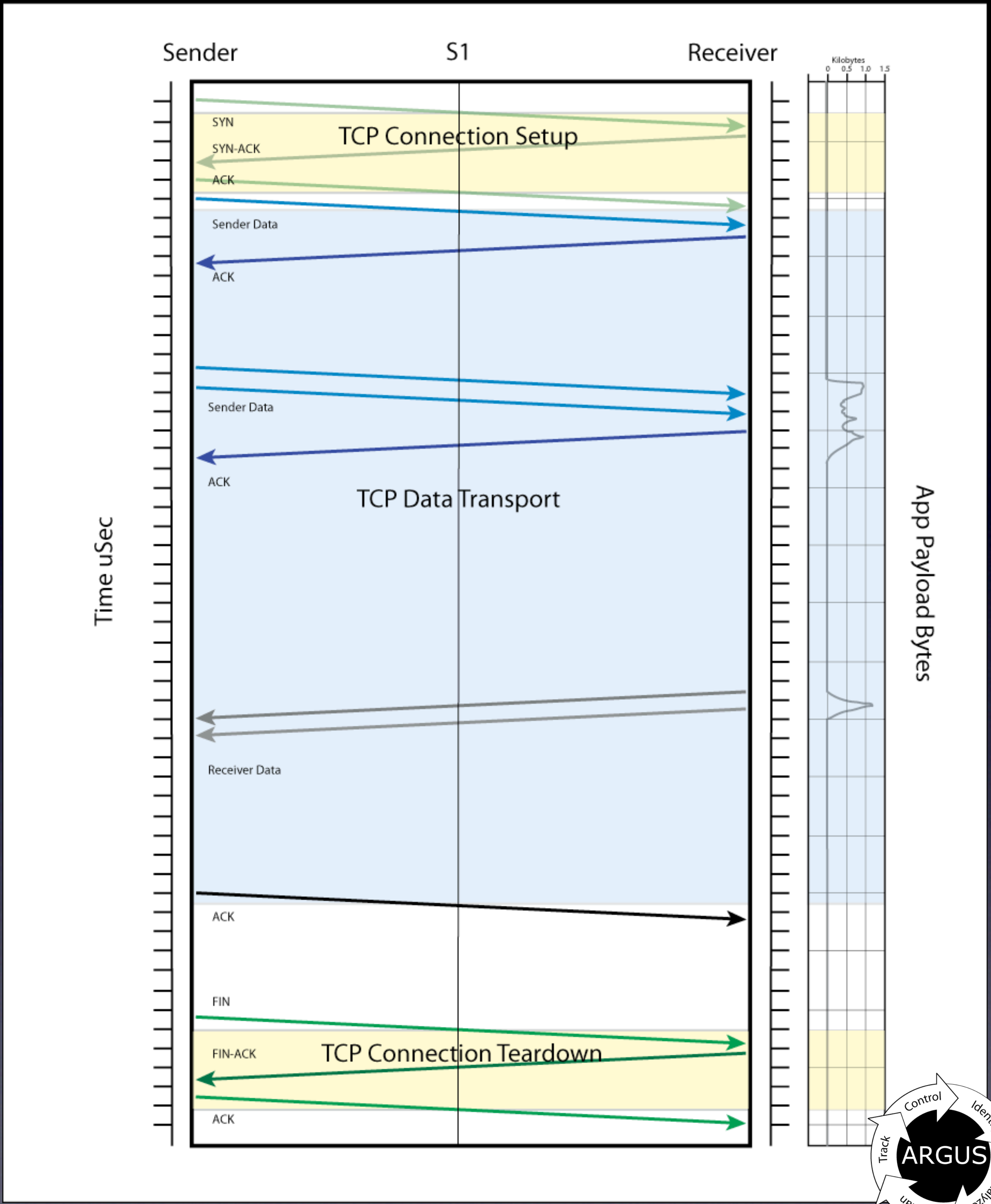
- Replay attack detection
  - Bi-Directional Protocol Time Uncoupling
- Stepping stone detection
  - Two completely independent flows, that share the same instantaneous burst behavior and packet size frequency distribution (shifted for encapsulations)
- Man vs Machine detection
  - Interactive vs Non-Interactive Session Detection
  - Packet, transaction and session jitter analysis
- Man-in-the-middle detection
  - Pass Thru - Detectable one-way latency, hop count, path resource modifications
  - Proxy - Connection setup time modifications, header attribute changes
- Performance as an Asset that needs Protection
  - Path Availability, Bandwidth, Latency, Jitter, MTU, ...
  - Continuous One-Way latency determinations

TCP Replay Attacks

- Predicting a machines next TCP base sequence number, enables elementary trusted source attacks.
  - Attack a remote target, such as a border router, by simply streaming a hand crafted TCP connection to the target machine, masquerading as an internal trusted source
  - Don't have to be in the data streams path (blind transmission).
  - Very common and successful strategy against older TCP/IP stacks
  - Detection is very difficult with simple uni-directional flow data
- Awareness of the packet dynamics of the attacking TCP connection, can lead to simple and immediate detection.

TCP Normal Connection

- State Development Time
  - TCP Setup Time
  - Data Presentation Delay
  - Data Transport Time
  - TCP Teardown Time
- Host Dynamics
  - Data Burst Rate
  - Data Ack Delay
  - Host Processing Time
- Network Dynamics
  - One-way Delay (shaping)
  - Round Trip Time
  - Loss Rate



TCP Replay Sender

- Attacker generally sends entire TCP connection all at once.

  Sophisticated attempts inject pseudo packet delay

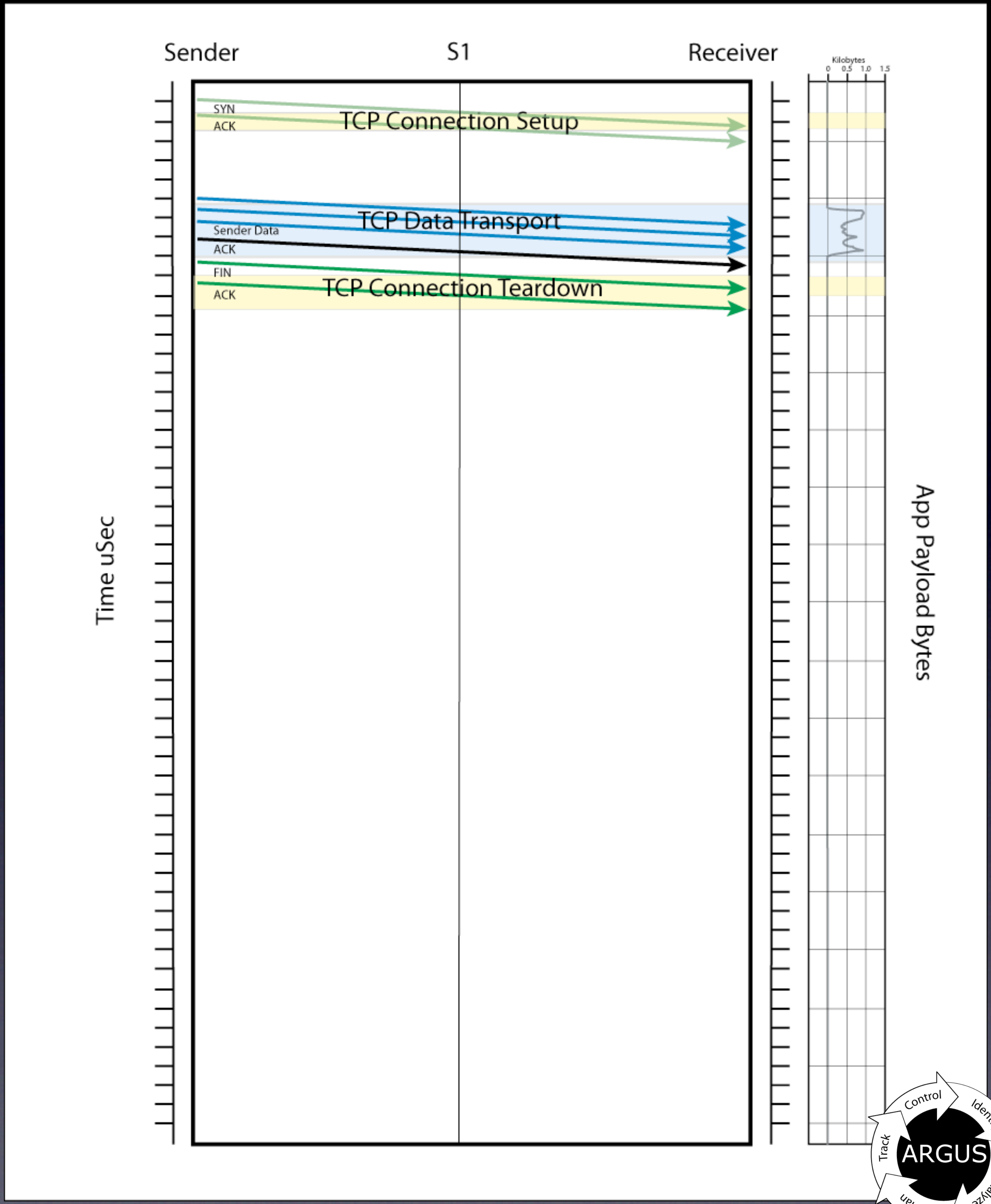
- State dynamics ignored

  RTT Insensitive

    TCP Setup Time

    Data Transport Time

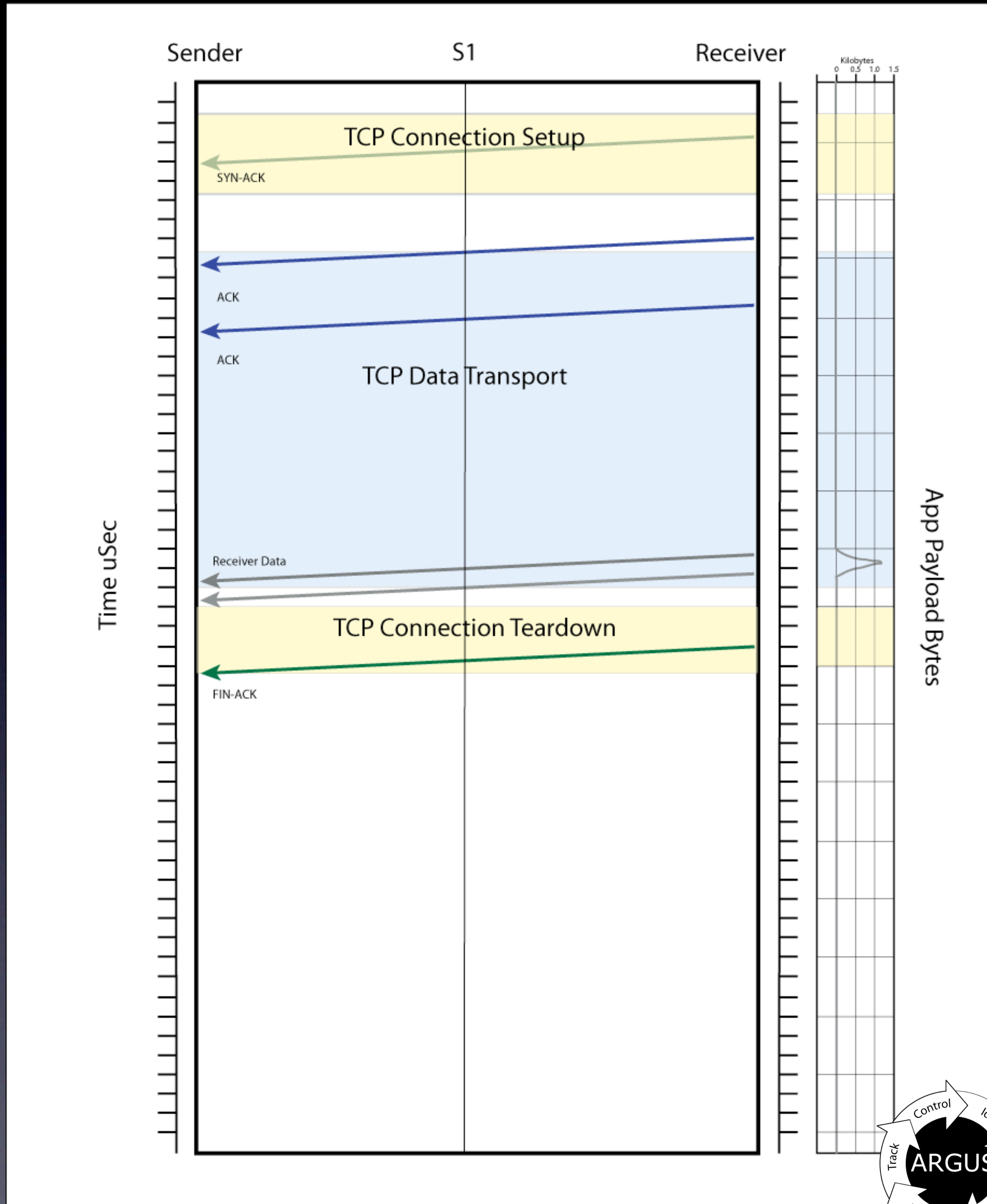
    TCP Teardown Time



TCP Replay Receiver

- Receiver responds in a conventional manner

Dynamics, however, shifted toward being driven by processing delay, rather than network delay.



TCP Replay Connection

- Overall connection is contracted

  All data transport appears to be at line rate

- Protocol state violated

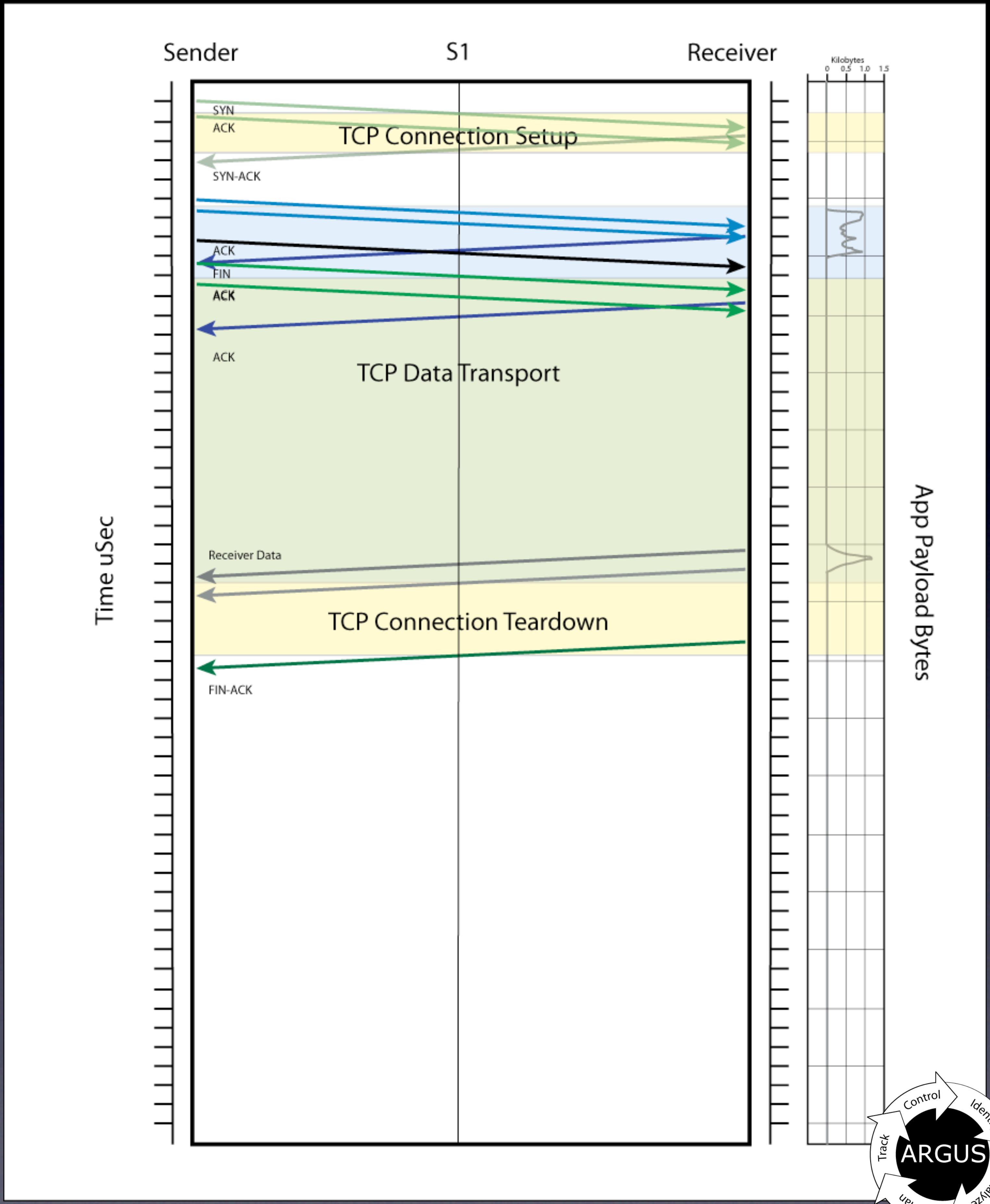
  Response seen before Request

  ACKs before Data

  Impossible TCP Setup Time

  Packets out of phase but not out of order

  Massive TCP Teardown Time



Argus Based Packet Dynamics

- Bi-Directional Protocol State Monitoring
  - Protocol State Tracking / Durations
    - TCP connection setup duration
      - SYN -> SYN_ACK and SYN_ACK -> ACK time
    - TCP connection teardown duration
      - FIN -> FIN_ACK and/or FIN -> RESET time
    - TCP state transition reporting
  - Transport performance
    - RTP / TCP / UDT loss reporting
    - TCP window load with size advertisements (tcpmax)
- Bi-Directional inter-packet arrival times
  - Mean, max, min, variance, frequency distribution
- Request / Response Duration Times
- Packet Size / State Correlations

SSH Keystroke Detection

- Discriminating between interactive and non-interactive network activity
  - Threat model is different between automation and human interactive sessions
- Discovering SSH keystroke traffic behavior in arbitrary packet streams (UDP, IP, ICMP, ethernet) provides reliable detection
- Discerning behavior without regard to contents provides suitable continuous monitoring strategy

Algorithm Goals

- Detect SSH client keystroke packets in arbitrary traffic connections (TCP as candidate test protocol).
  - Detect SSH flows in any connection, on any port.
  - No packet content inspection, header only (privacy)
- Deployable in production networks at traditional observation points (performance)
- Suitable for near realtime detection and reporting (alarm / alerting)

Algorithm Strategy

1. Identify human typing patterns in offered stream
2. Track candidate single and small clustered character transmissions
3. Detect character echo as strong correlate
4. Exploit TCP protocol characteristics
5. Exploit SSH protocol specific properties

Algorithm Development

- Exhaustive statistical analysis of commodity and scripted packet traces
  - Traces from Univ. of Leipzig and Purdue Univ.
  - Represented large session variability
    - Logon failure, single command, subsystem requests
- Initial exploratory, unstructured study
  - Based on visualizations and numeric methods
    - Focus on arrival times, packet sizes, flags, seq numbers
- Followed by formal statistical testing
  - Multi-response fractional-factorial statistical experiment

Algorithm Details

SSH Protocol Rules

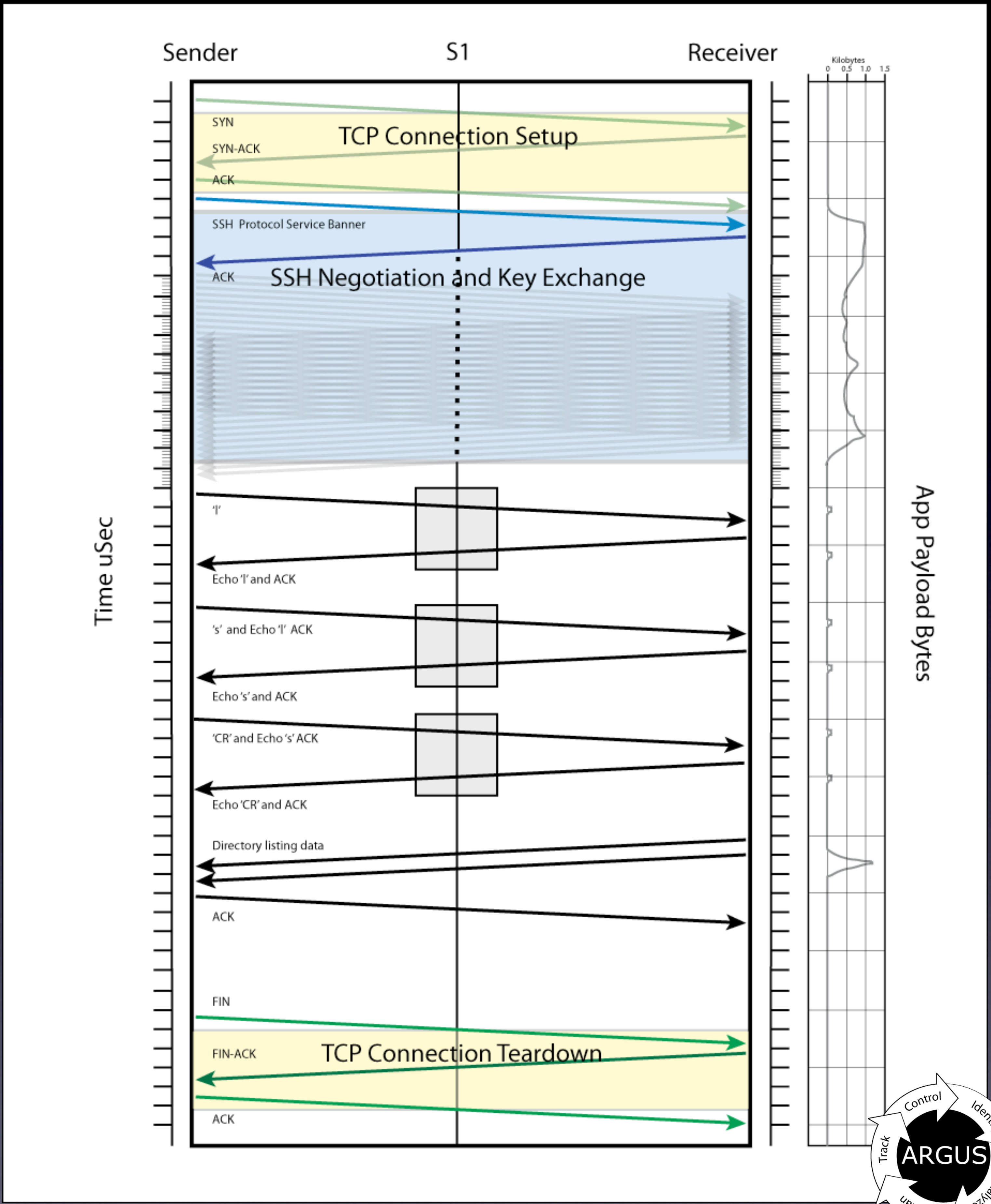
I. SSH Startup Handshake Detection (22 pkts)
II. SSH Packet Size (mod 4 = 0)

Packet Dynamic Rules

1. Minimum client data size (48 bytes)
2. Maximum client data size (128 bytes)
3. Maximum server echo gap size (3 pkts)
4. Minimum server echo data size (24 bytes)
5. Maximum server echo data size (256 bytes)
6. Minimum client inter-arrival time (50 mSec)
7. Maximum absolute log inter-arrival ratio (1.122)
8. Maximum previous-current gap (3 pkts)

SSH Keystroke Detection

- All data packets of the correct size (% 4 = 0)
- SSH protocol in data transfer state (> 22 pkts)
- Previous packet arrival within tolerances (human typing).
- Bi-directional sequential data packet sizes within tolerances
- Data and Echo Packets RTT within tolerances



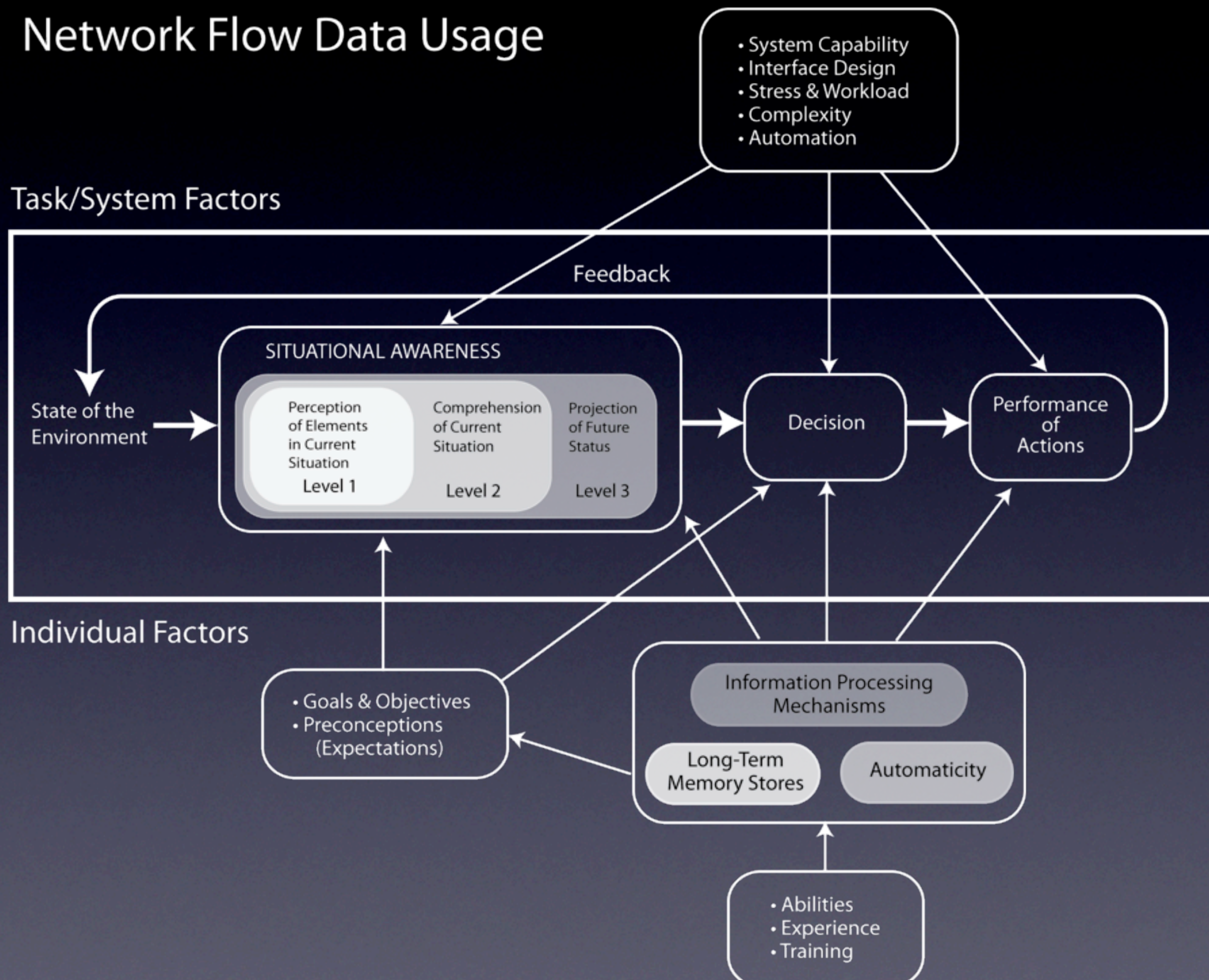
Keystroke Detected

Reporting Metrics

- Argus Behavioral DSR
  - Algorithm Family Identifier (DSR subtype 8 bits)
  - Algorithm Identifier (DSR qualifier 8 bits)
  - Algorithm Specific Metrics (< 32K bytes)
- Presence of the DSR indicates that the algorithm was applied to the flow during the status interval.
- Presence of algorithm metrics indicate that there were analytic results during the status interval
- Algorithm generates keystroke counts for both directions
- DSR meets all Argus DSR operational requirements
  - ra keyword " nstroke " (number of keystrokes)
  - printable, graphable, filterable, aggregatable, etc.....

Model of Situational Awareness in Dynamic Decision Making

Network Flow Data Usage



Alarming Conditions

- Keystrokes are expected in most ssh connections
  - For many SSH connections the absence of keystrokes is the actionable condition
  - Keystrokes should appear in various phases of an SSH.
    - Human initiated connections should have keystrokes at the beginning and end of an SSH connection.
    - Absence of keystrokes at closure, may indicate timeout conditions.
  - Historical behavioral baselining will identify connections where keystroke monitoring is helpful
- Almost always (> 99.9%) , the originator of the SSH connection will be doing the typing
  - Anytime the destination originates keystrokes, you probably have a really serious problem !!!!!

Live Demonstration

from Presentation Laptop

ra and ratop screens with me ssh'ing to the qosient.com

hopefully we can have that going on a separate

projector during the talk. need to tune the parameters

for end system sensing.

Supporting Slides

Classification Variables

- Whether a packet is from the client or server
- Whether there are more than 22 packets in the connection
- Whether there are client packets after packet 22 or not
- Data size for client packets
- Data size for server packets
- The number of packets between a client packet with data and the next acknowledging server packet with data
- Inter-arrival time of two successive client packets with data
- Inter-arrival time of two successive echoing server packets
- The ratio of a client inter-arrival time and the inter-arrival time of the corresponding echoing server packets
- The number of packets between two successive client packets with data.

Situational Awareness

Level 1 SA - Perception

- The perception of elements in the environment within a volume of time and space
- Involves timely sensing, data generation, distribution, collection, combination, filtering, enhancement, processing, storage, retention and access.

Level 2 SA - Comprehension

- Understanding significance of perceived elements in relation to relevant goals and objectives.
- Involves integration, correlation, knowledge generation.

Level 3 SA - Projection of Future Status

Endsley, M. R. (1995b). Toward a theory of situation awareness in dynamic systems. Human Factors 37(1), 32-64.