Transport Performance Metrics MIB

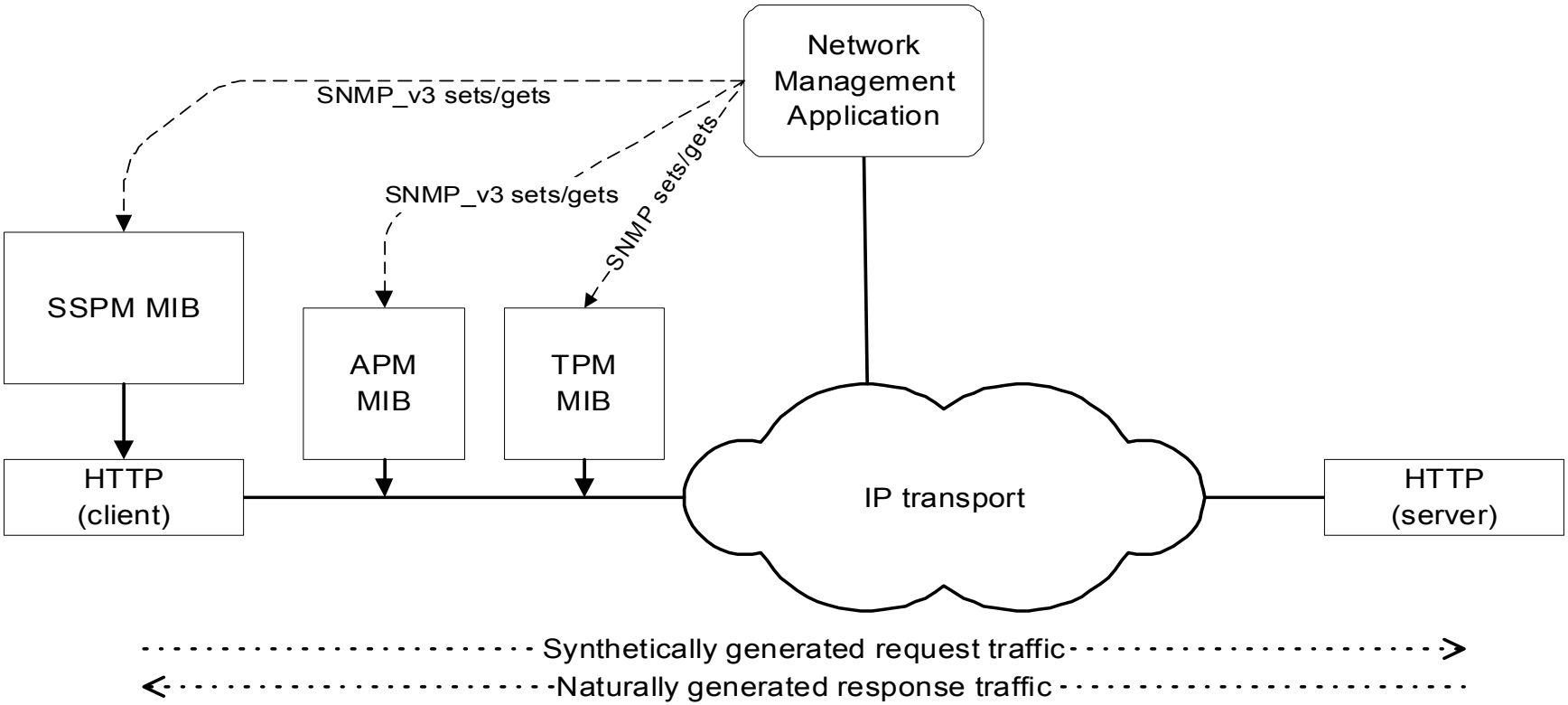
draft-ietf-rmonmib-tpm-mib-06.txt

Robert Cole, Russell Dietz
rgcole@att.com, rdietz@hifn.com

RMONMIB Working Group

54th IETF –
Yokohama, Kanagawa-Ken, Japan

Synthetic, Complex Application, e.g., Capital WEB



SSPM Functions
- Control source and sink on common platform (scheduling,end-point config)
- Config's may include dest ipaddr, http header information, TOS bits, timeouts, etc.

HTTP Client Functions
- Build http_get request packet
- Issue request
- Parse html for embedded objects
- Request objects

TPM Functions
- Drill down from complex application measurements in APM
- Metric definition (pointers to multiple IPPM standards, i.e., IPPM-round-trip delay metric)
- Raw measurement collection (History Table)
- Statistical aggregation (Report Tables)

APM Functions
- Availability and Responsiveness Reporting
- Aggregation of reporting information
- Thresholding and Notification

Transport Performance Metrics (TPM-MIB) - Objectives

Objectives (as stated in draft):

- Provide a drill-down capability to compliment the user-perceived monitoring defined within the APM MIB [APM].
- Provide additional performance metrics and related statistics.
- Support standards based metrics and associated statistical aggregation by defining methods to reference those standards.
- Provide (as an option) a history table storing the raw measurements of the metrics and protocols in question.

Transport Performance Metrics (TPM-MIB) - Modifications

Modifications (since draft 05):

Added:

```
IMPORT
        ClientID, DataSourceOrZero,
        TransactionAggregationType    FROM APM-MIB
        ClockSource, MicroSeconds     FROM SSPM-MIB
```

Added:

```
TransactionMetricIndex ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "An index used to uniquely identify an entry in the
        tpmTransactionMetricDir table.  Each such entry defines
        the protocol transaction and metric instance to be
        monitored for a specific application."
    SYNTAX       Integer32 (1..65535)
```

Transport Performance Metrics (TPM-MIB) - Modifications

Modifications (since draft 05):

Added:

```
MetricDefID ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "An index that identifies through reference to a
  specific
        performance metric.  The metrics are referenced
        through their type (connect, delay, loss, etc.), their
        directional characteristics (one-way, round trip, etc.),
        their name, their reference to a documented definition."
    SYNTAX       Unsigned32 (1..2147483647)
```

Transport Performance Metrics (TPM-MIB) - Modifications

Modifications (since draft 05):

Added: (to allow for APM MIB control)

```
tpmApmMIBCompliance MODULE-COMPLIANCE
STATUS  current
DESCRIPTION
        " Describes the requirements on the APM MIB
        for conformance to the TPM MIB operating in
        drill-down mode in support of APM MIB
        measurements"
MODULE  -- this module
MANDATORY-GROUPS
 { tpmApmReportControlGroup,
   tpmApmTransactionControlGroup,
   tpmApmExceptionControlGroup  }
::= { tpmCompliances 2 }
```

Transport Performance Metrics (TPM-MIB) - Modifications

Changed:

- tpmExcpReports objects changed back to revision 04 tmpExcpReports - as discussed at last meeting.
- Removed tpmExcpReportCntlTable - exception control through APM MIB, tpmExcpReportCntlTable reports drill-down extensions.
- Changed SYNTAX of objects to be consistent with APM MIB definitions.
- Cleaned up text in introductory sections.
- Cleaned up objects' DESCRIPTION.
- Ran spell check and smilint and cleaned up errors based upon output.

Transport Performance Metrics (TPM-MIB) - Issues

Issues:

Issue #1 - Steve Waldbusser

- The TPM MIB is currently formatted with words broken by '-' to fill the lines. I'm pretty sure that the RFC guidelines don't allow this. Maybe the ID guidelines too.

  Action: Only in text, not in Definitions' section. Will make change.

- Also, section "5.3.4. The tpmExceptionReportsGroup", is inconsistent with the previous ones in that it doesn't end with a declaration of what tables the group consists of.

  Action: Will add description of tables in tpmExceptionReportsGroup

Transport Performance Metrics (TPM-MIB) - Issues

In closing…

- We believe we have addressed all of the issues from the last meeting, expect for Steve's issues which we'll clean up.
- We believe this ready for last call!

Transport Performance Metrics MIB

draft-ietf-rmonmib-tpm-mib-06.txt

Robert Cole, Russell Dietz
rgcole@att.com, rdietz@hifn.com

RMONMIB Working Group

54th IETF –
Yokohama, Kanagawa-Ken, Japan

Is the tpmMetricsDefType (51) and the tpmMetricsDefDirType (52) needed?

```
tpmMetricsDefType   OBJECT-TYPE
        SYNTAX      INTEGER  {
                        other(1),
                        connectMetric(2),
                        delayMetric(3),
                        lossMetric(4)
                    }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The basic type of metric indicated by this entry.
                 The value 'other(1)' indicates that this metric cannot be
                 characterized by any of the remaining enumerations
                 specified for this object.
                The value 'connectMetric(2)' indicates that this metric
                measures connectivity characteristics.
                The value 'delayMetric(3)' indicates that this metric
                measures delay characteristics.
                The value 'lossMetric(4)' indicates that this metric
                measures loss characteristics."
        ::= { pmMetricsDefEntry 2 }
```

Is the tpmMetricsDefType (51) and the tpmMetricsDefDirType (52) needed?

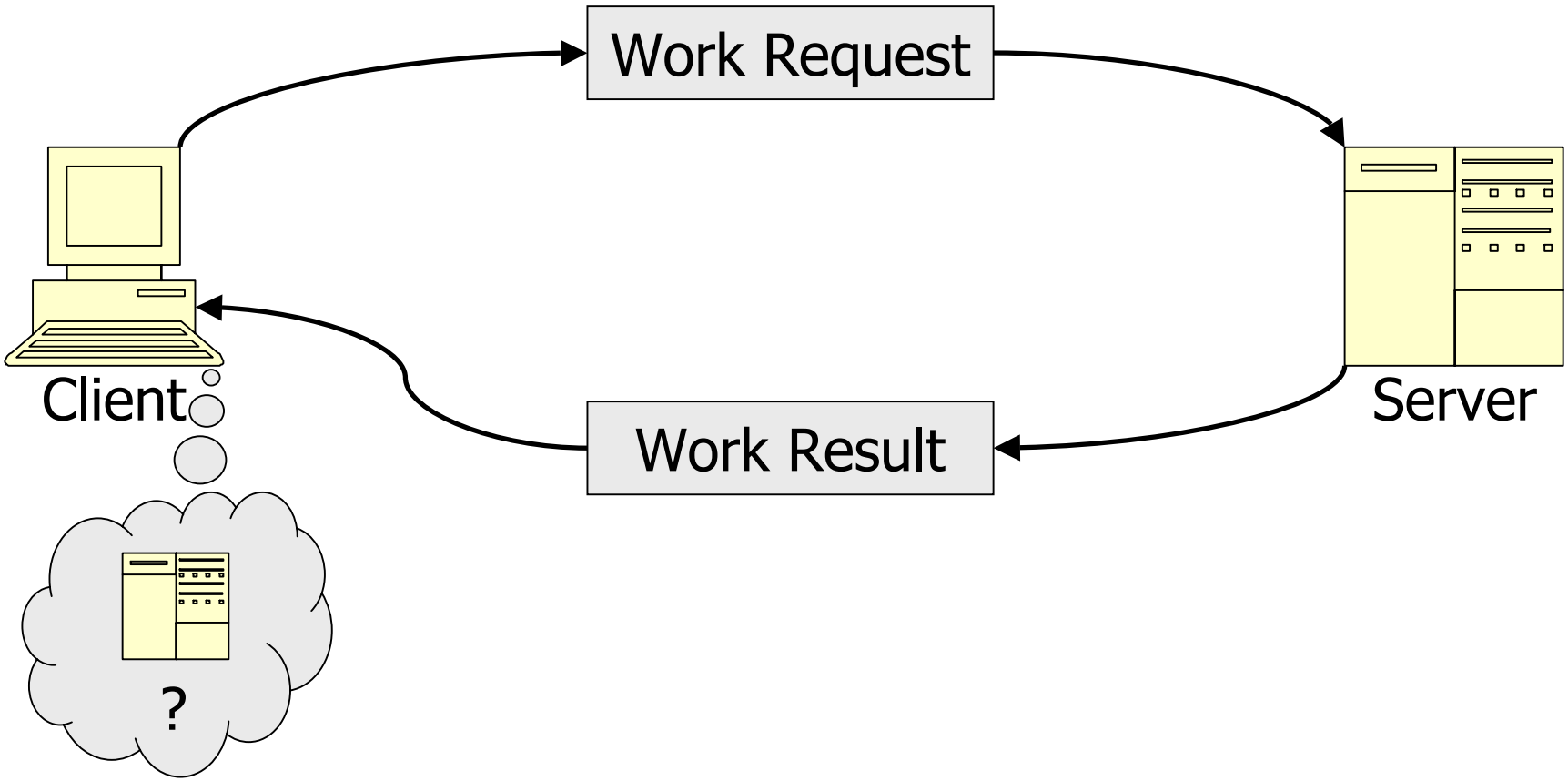
```
tpmMetricsDefDirType   OBJECT-TYPE
        SYNTAX      INTEGER  {
                            oneWay(1),
                            twoWay(2),
                            multiWay(3)
                        }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                "The directional characteristics of the this metric.
                The value 'oneWay(1)' indicates that this metric is measured
                with some sort of uni-directional test.
                The value 'twoWay(2)' indicates that this metric is measured
                with some sort of bi-directional test.
                The value 'multiWay(3)' indicates that this metric is
                measured with some combination of uni-directional and/or bi-
                directional tests."
        ::= { tpmMetricsDefEntry 3 }
```

Transport Performance Metrics (TPM-MIB) – Issues (continued)

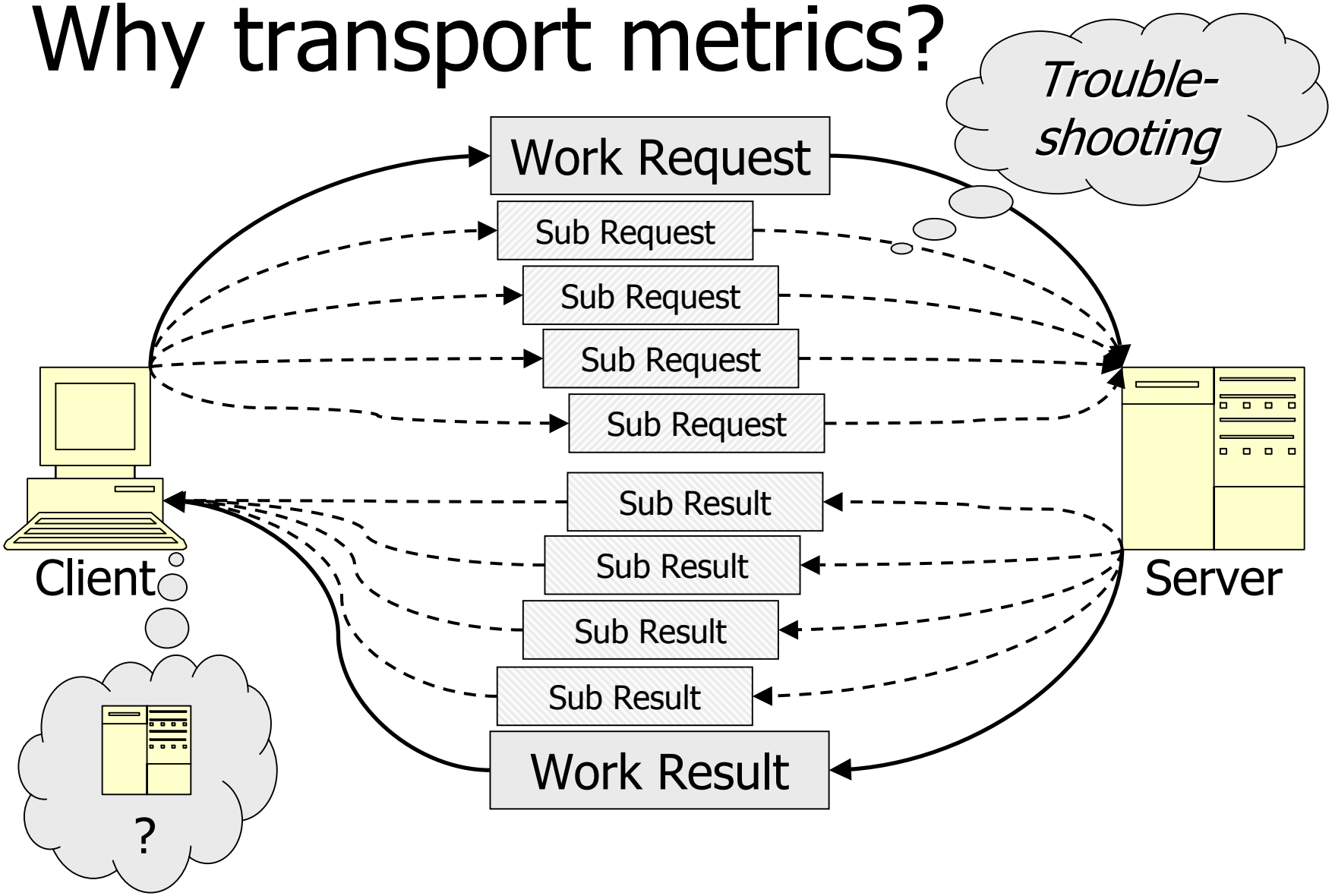
Issues (continued):

- When cross referencing the APMControlIndex, should certain entries in the tpmReportControlTable and the tpmMetricTable be automatically populated? E.g., right now most of the entries have access type "read-create", while when cross indexed to the APMControlIndex these entries coincide with APMControlTable entries.

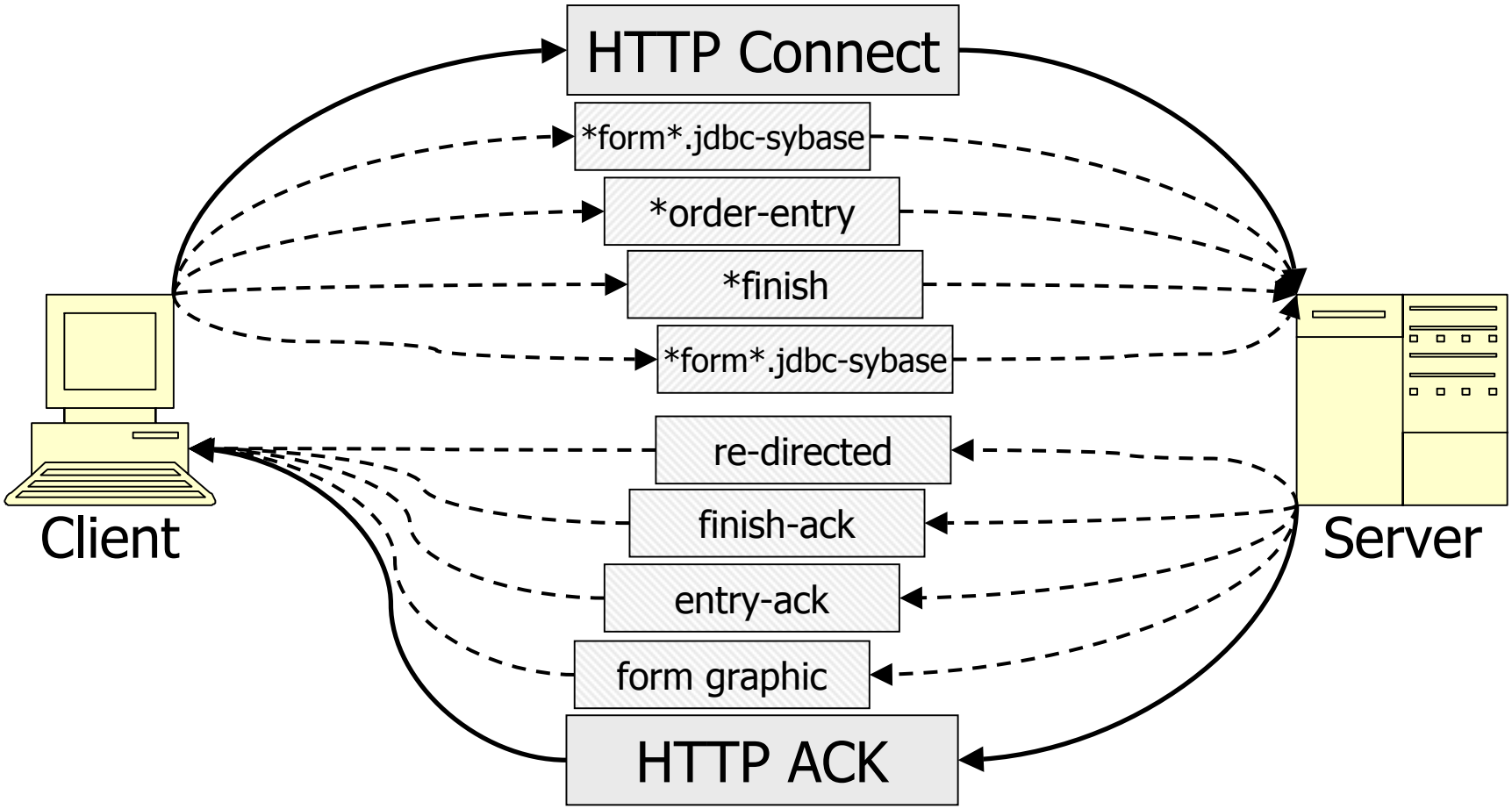
Application Performance…



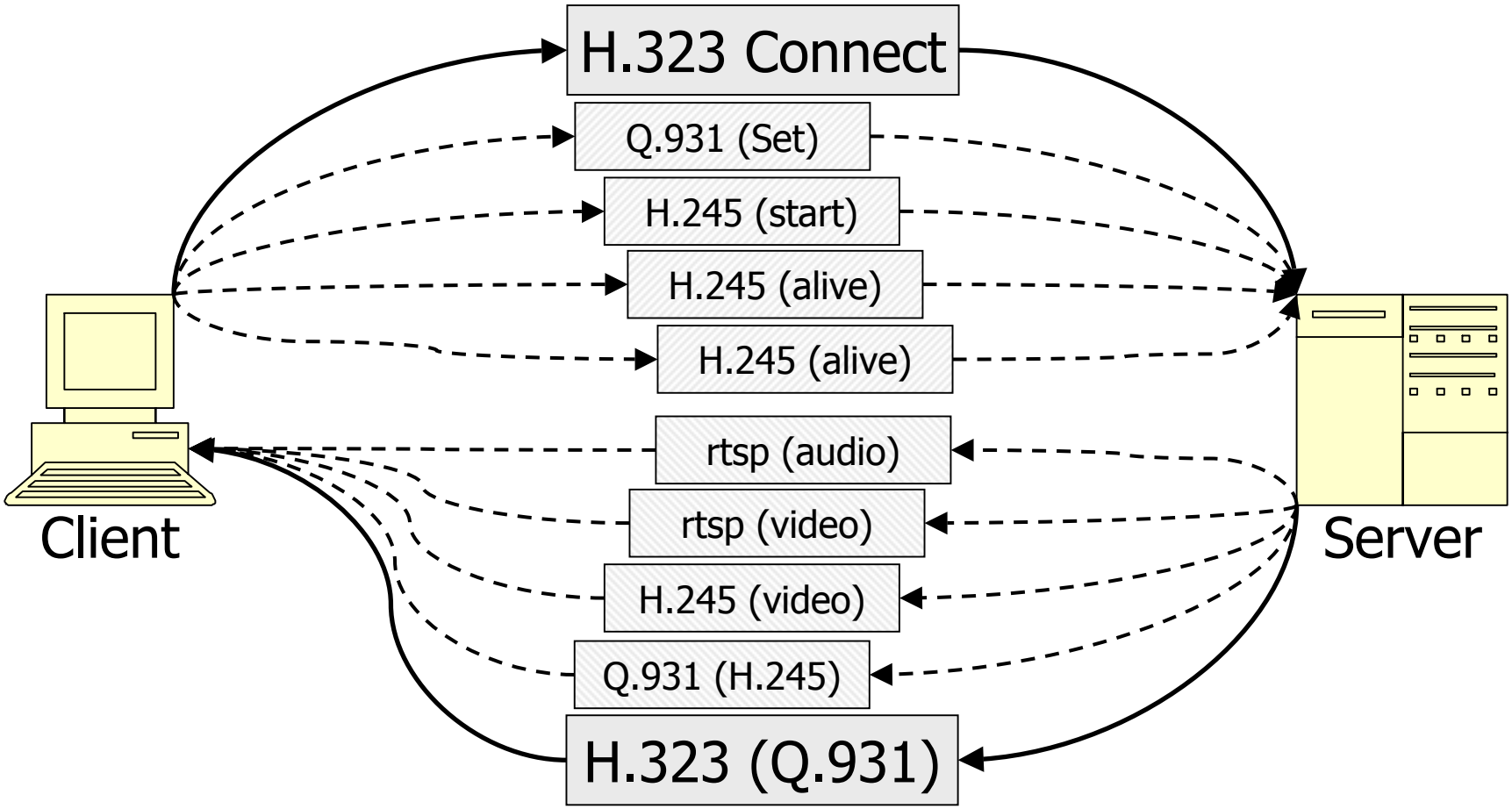
Why transport metrics?



Web (HTTP) Example



NetMeeting (H.323) Example



Categories of Metrics

- Jitter
- Exchange Response
- Application Response
- Connection
- Connection Sequence
- Connection Window
- Routing

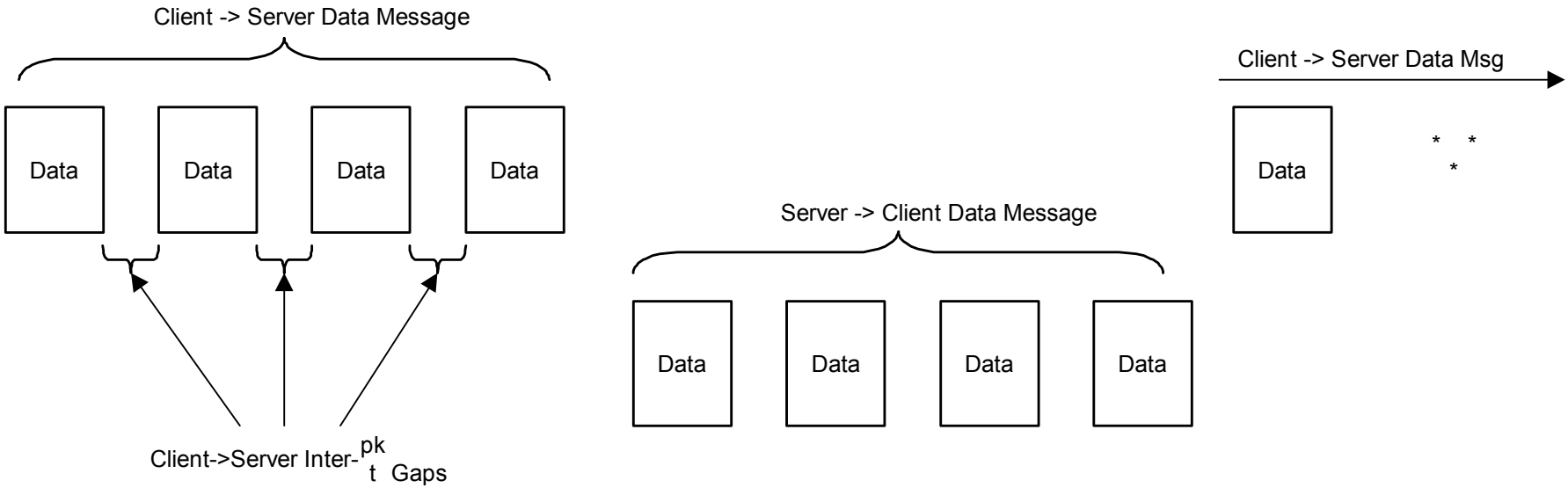
Jitter Metrics

- Message Jitter…
  - starts with the abstraction of a communicated message… a sequence of adjacent data packets transferred in a given direction, concluded by the transfer of one or more adjacent data packets in the other direction.
  - measures the inter-packet gaps only for data packets **within** messages. It does not measure the gaps **between** the messages themselves.

Message Jitter…

- example, an application that is downloading a series of large graphics. The client first requests a graphic file download from the server. The server then sends the graphic to the client. The client then requests the next graphic, and so on. Within each download, the observed inter-packet gaps are of interest. The inter-packet gap from the last packet of one download and the first packet of the next is of less interest These later gaps are excluded.

Message Jitter



Metric	Units	Description
N	Inter-Packet Gaps	Count of the **# of Inter-Packet Gaps** measured for Data from the Client to the Server
$\Sigma(x)$	uSec	Sum total of the **Delta Times** in these Inter-Packet Gaps
$\Sigma(X^2)$	$uSec^2$	Sum total of the **Delta Times** squared in these Inter-Packet Gaps
X_{max}	uSec	The maximum **Delta Time** of Inter-Packet Gaps measured
X_{min}	uSec	The minimum **Delta Time** of Inter-Packet Gaps measured.
$\Sigma(IX)$		Sum total of the data point # times the **Delta Times** in these Inter-Packet Gaps

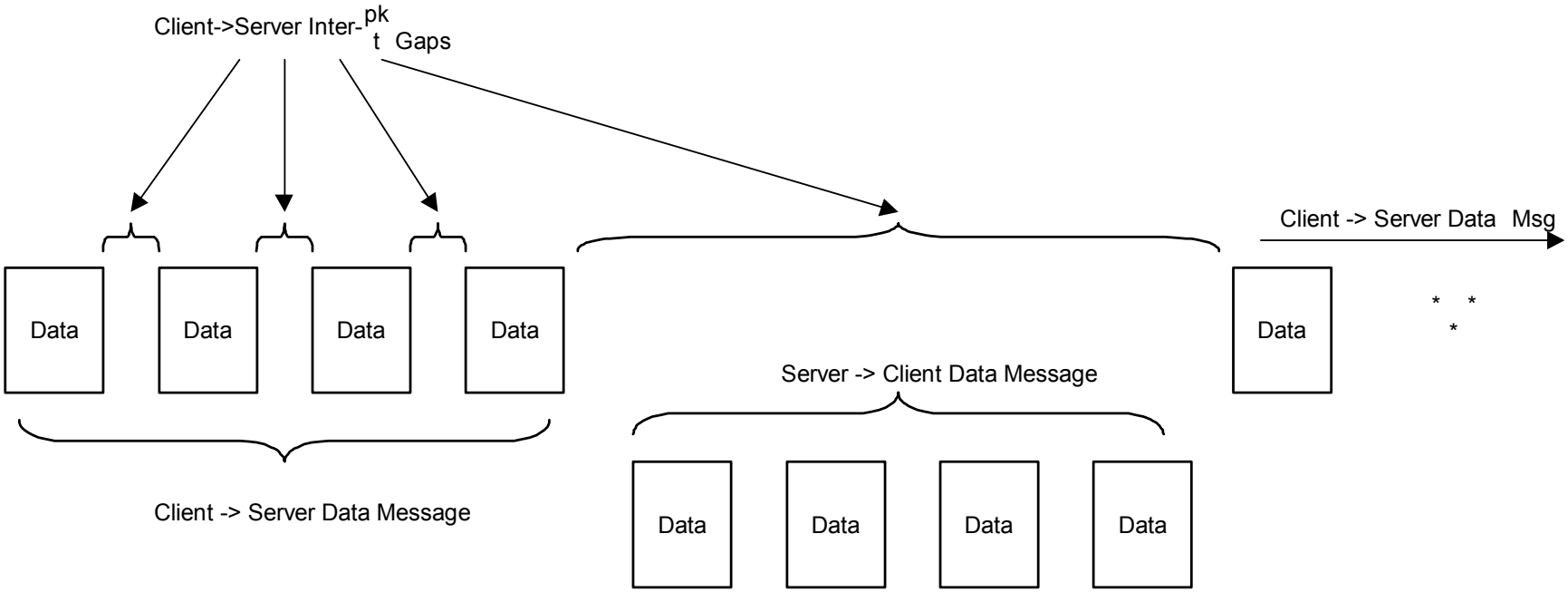
Jitter Metrics

- Stream Jitter…
  - does not include *message* considerations. Rather, measures the inter-packet gaps for **all packets** of the data stream for a given transfer direction.

Stream Jitter…

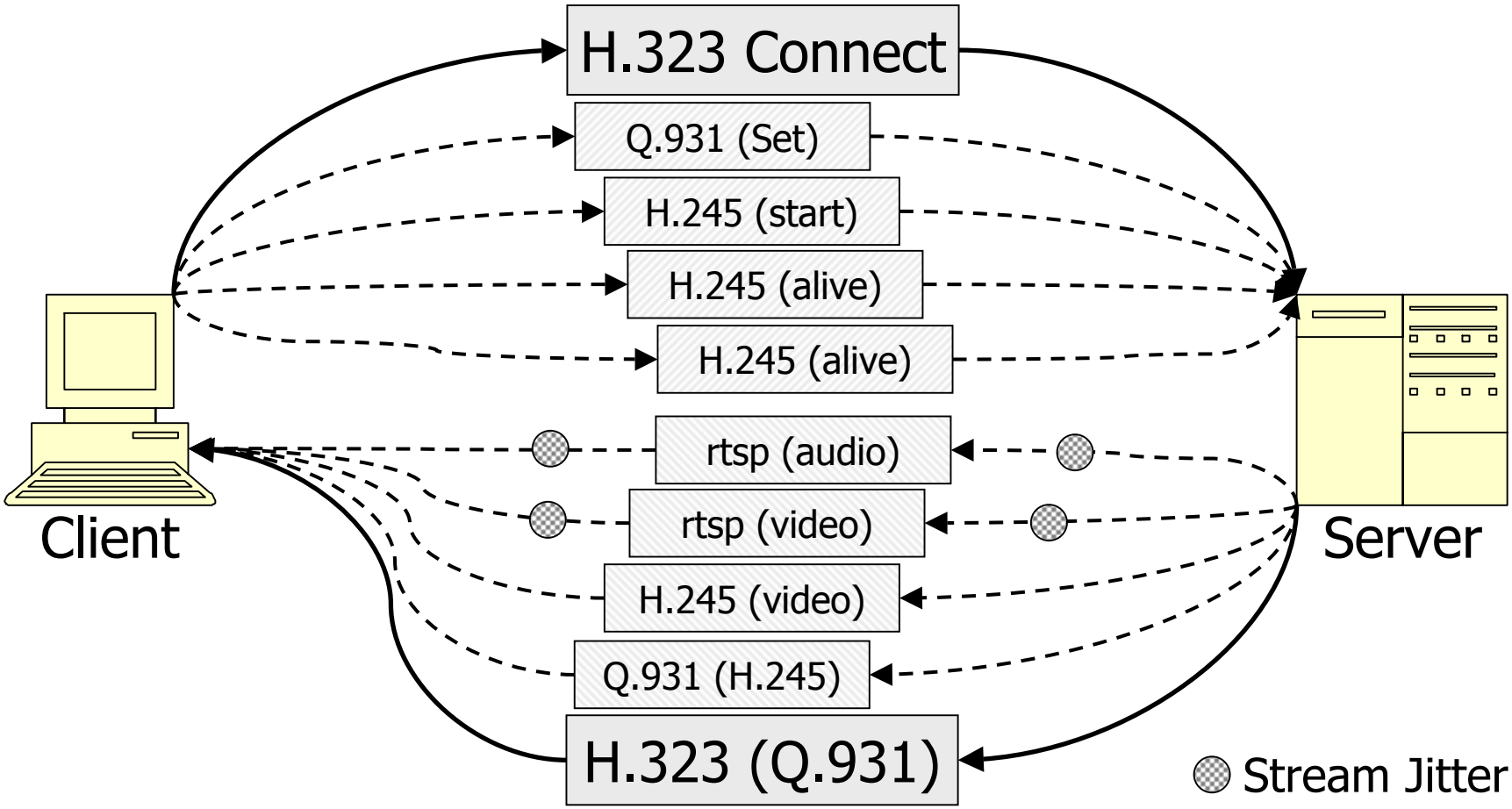
- example is where an H.323 Voice-over-IP (VoIP) data-stream is transferring an audio data-stream over RTP from one end-point to another. In this case, all inter-packets gaps observed would be of interest.

Stream Jitter



Metric	Units	Description
N	Inter-Packet Gaps	Count of the **# of Inter-Packet Gaps** measured for Data from the Client to the Server
$\Sigma(x)$	uSec	Sum total of the **Delta Times** in these Inter-Packet Gaps
$\Sigma(X^2)$	$uSec^2$	Sum total of the **Delta Times** squared in these Inter-Packet Gaps
X_{max}	uSec	The maximum **Delta Time** of Inter-Packet Gaps measured
X_{min}	uSec	The minimum **Delta Time** of Inter-Packet Gaps measured.
$\Sigma(IX)$		Sum total of the data point # times the **Delta Times** in these Inter-Packet Gaps

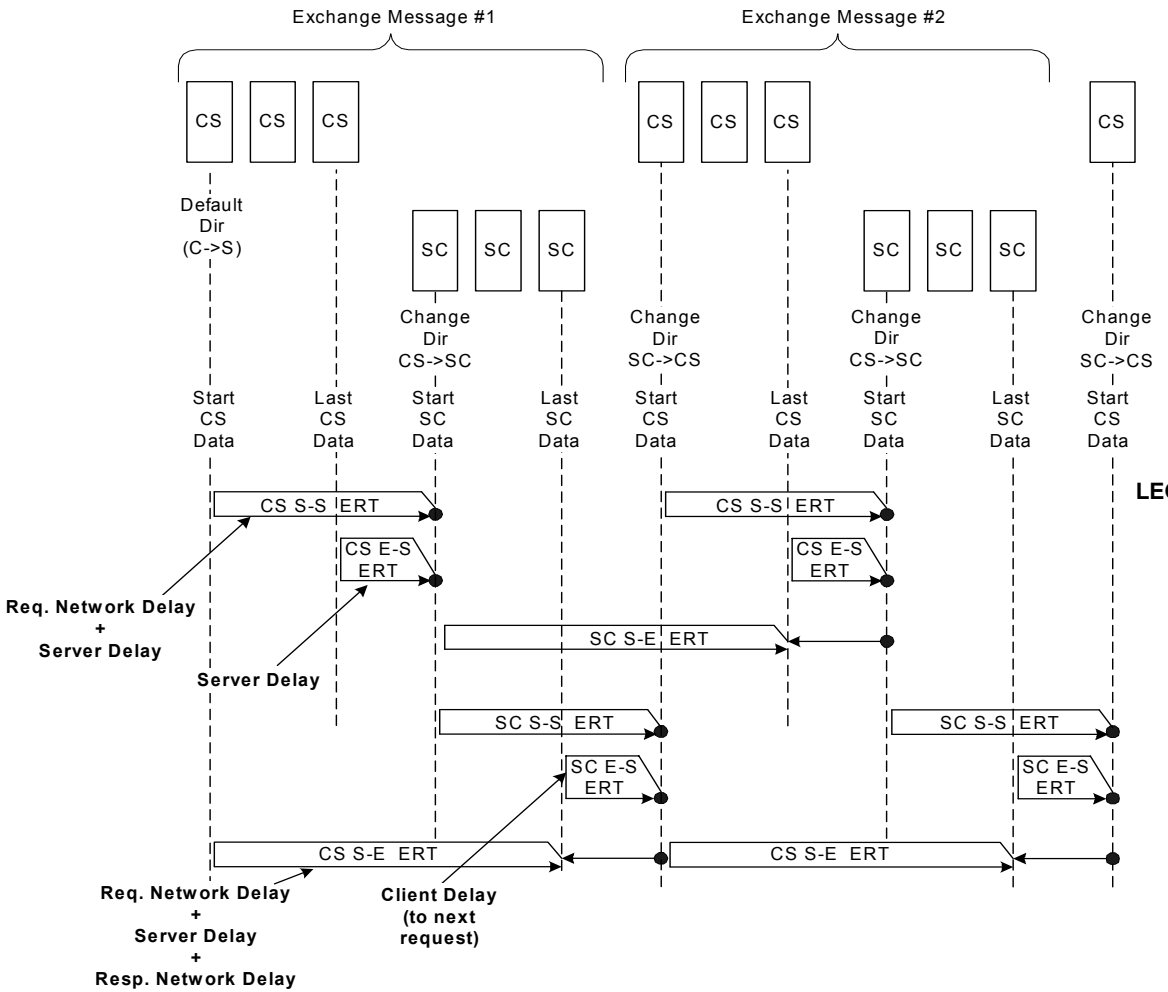
H.323 – Stream Jitter



Response Metrics

- Exchange Responses...
  - starts with the abstraction of a communicated *exchange-message*. An *exchange-message* is considered to start with a series of adjacent data packets transferred in a given direction (*exchange message request*). The end of such a message is defined to be the transfer of one or more adjacent data packets in the other direction (*exchange message response*). Matters of sequencing or retransmission (duplicate) detection are **excluded from consideration** in the analysis of data packets when observing *exchange-message* related packets.

Exchange Response Example



LEGEND

CS = Client->Server Data Packet
SC = Server->Client Data Packet

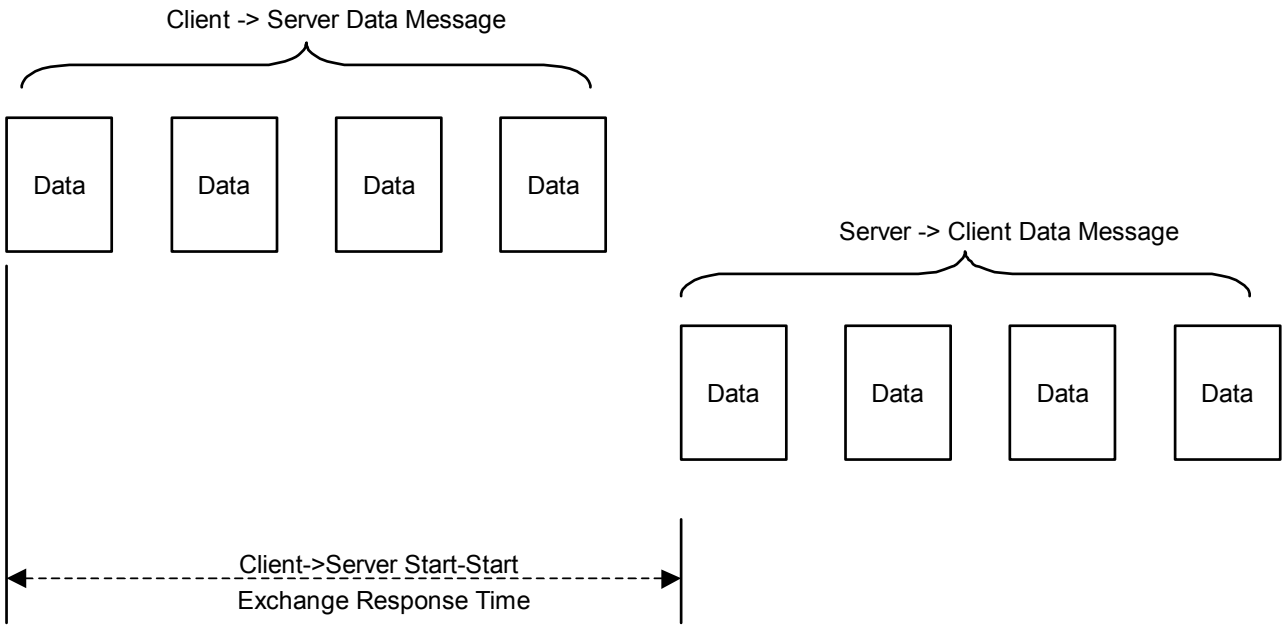
CS S-S ERT = Client->Server Start-to-Start Exchange Response Time
CS E-S ERT = Client->Server End-to-Start Exchange Response Time
CS S-E ERT = Client->Server Start-to-End Exchange Response Time
SC S-S ERT = Server->Client Start-to-Start Exchange Response Time
SC E-S ERT = Server->Client End-to-Start Exchange Response Time
SC S-E ERT = Server->Client Start-to-End Exchange Response Time

Req. Network Delay = SUM(CS S-S RT) - SUM(CS E-S RT);
Resp. Network Delay = SUM(CS S-E RT) - SUM(CS S-S RT);

Server Overhead = SUM(CS E-S RT);

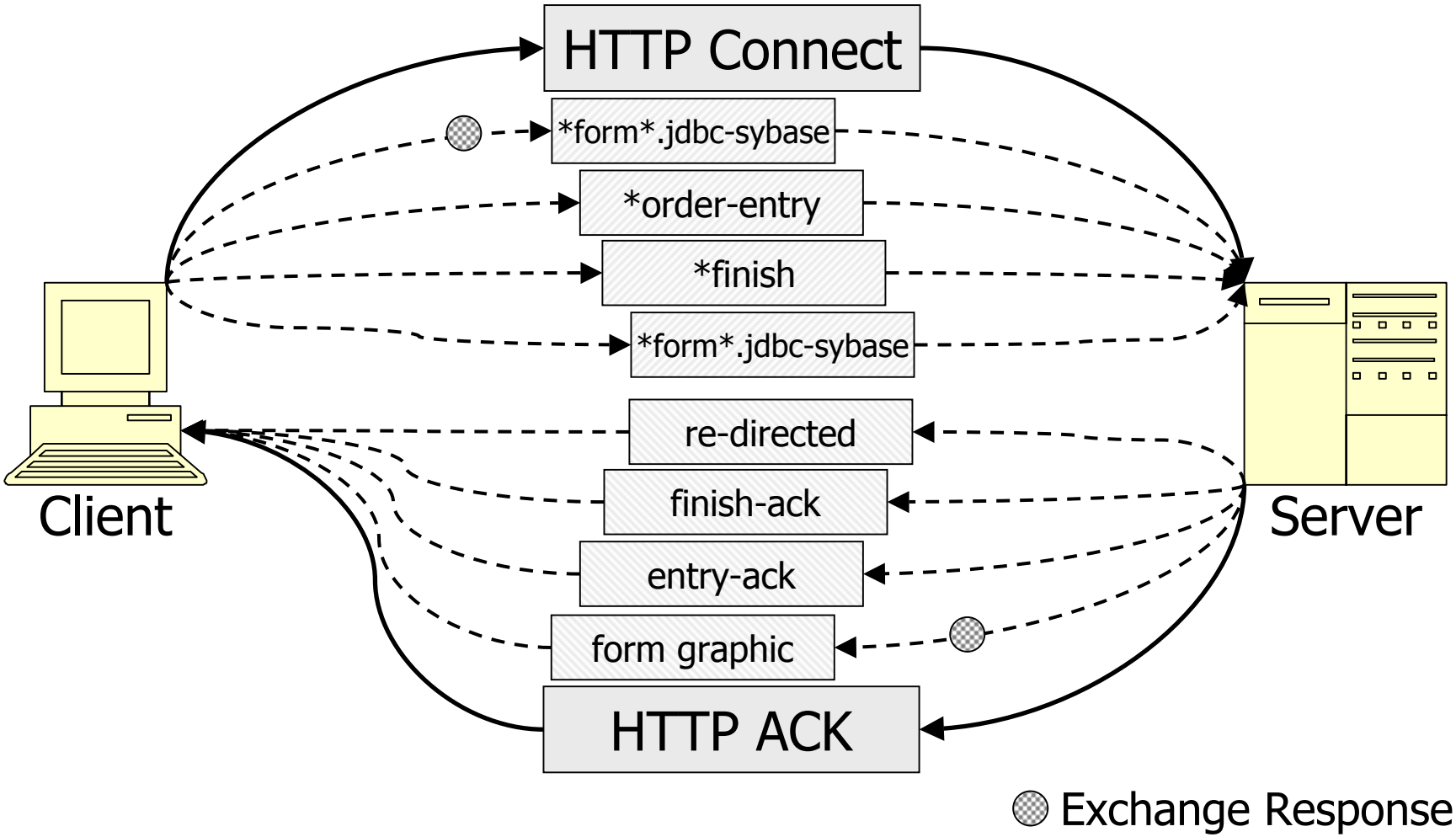
Client Overhead = SUM(SC E-S RT);

Exchange Response
Start to Start

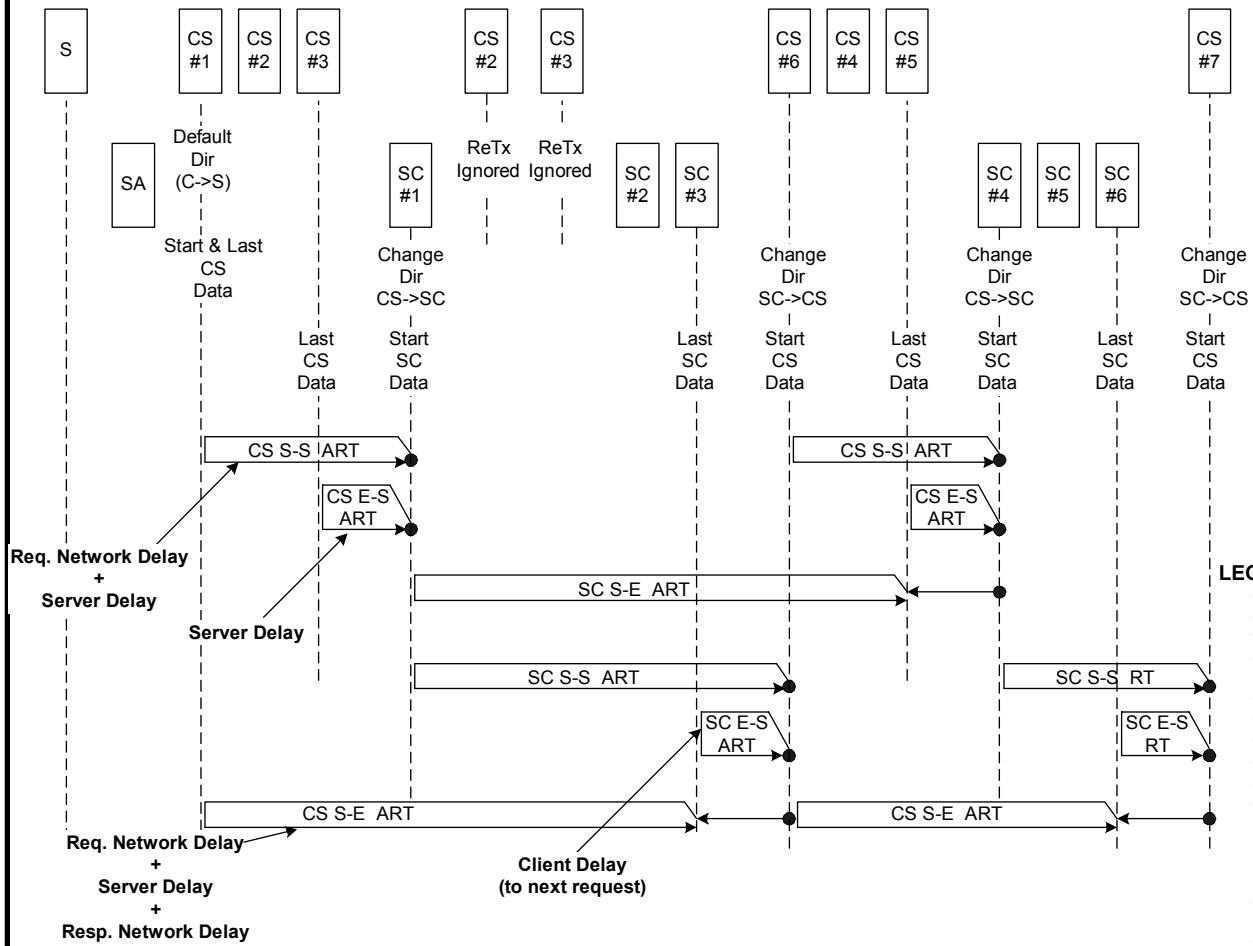


Metric	Units	Description
N	Client-> Svr Exchange Msg Requests	Count of **# Client<->Server request/response pairs** measured for *exchange messages* from the Client to the Server
Σ	mSec	Sum total of the **Start-to-Start Delta Times** in these Exchange Response Times
$\Sigma(x^2)$	$mSec^2$	Sum squared total of the **Start-to-Start Delta Times** in these Exchange Response Times
X_{max}	mSec	The maximum **Start-to-Start Delta Time** of these Exchange Response Times
X_{min}	mSec	The minimum **Start-to-Start Delta Time** of these Exchange Response Times
$\Sigma(IX)$		Sum total of the data point # times the **Start-to-Start Delta Time**

HTTP – Exchange Response



Application Response Example



LEGEND

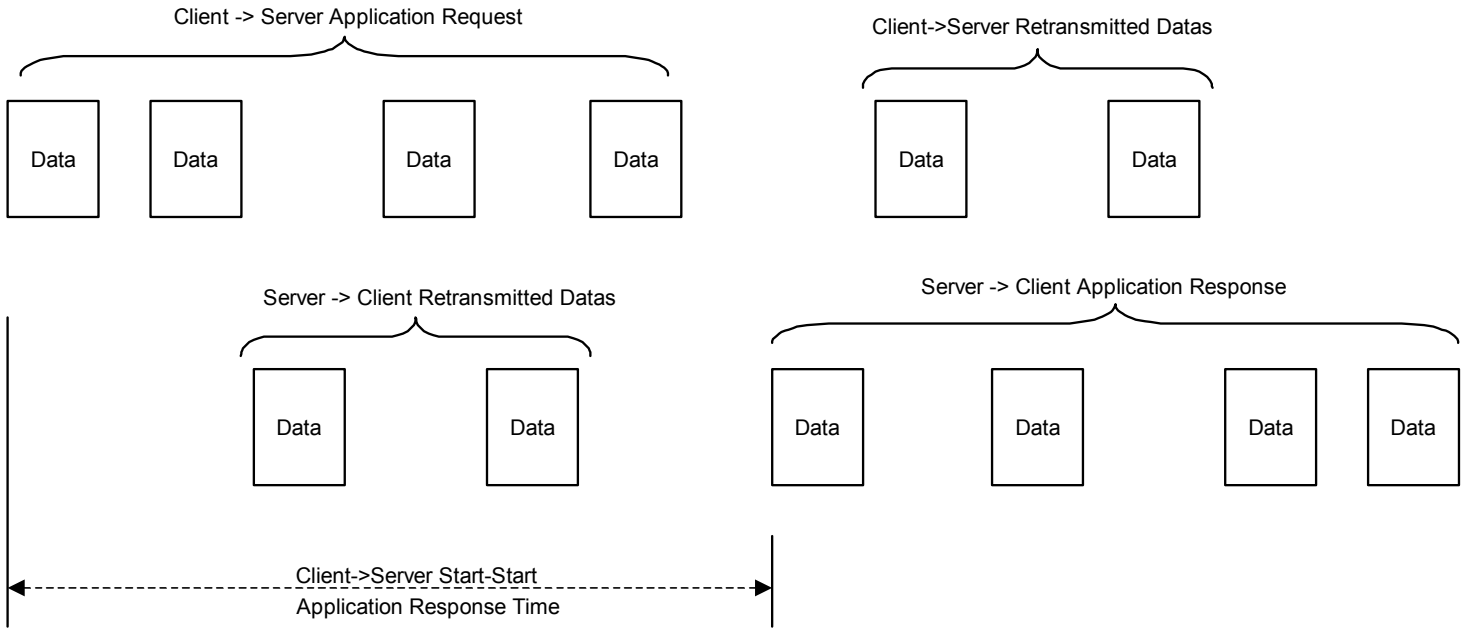
S	= SYN
SA	= SYN-ACK
CS	= Client->Server Data Packet
SC	= Server->Client Data Packet
CS S-S ART	= Client->Server Start-to-Start Application Response Time
CS E-S ART	= Client->Server End-to-Start Application Response Time
CS S-E ART	= Client->Server Start-to-End Application Response Time
SC S-S ART	= Server->Client Start-to-Start Application Response Time
SC E-S ART	= Server->Client End-to-Start Application Response Time
SC S-E ART	= Server->Client Start-to-End Application Response Time

Req. Network Delay = SUM(CS S-S ART) - SUM(CS E-S ART);
Resp. Network Delay = SUM(CS S-E ART) - SUM(CS S-S ART);

Server Overhead = SUM(CS E-S ART);

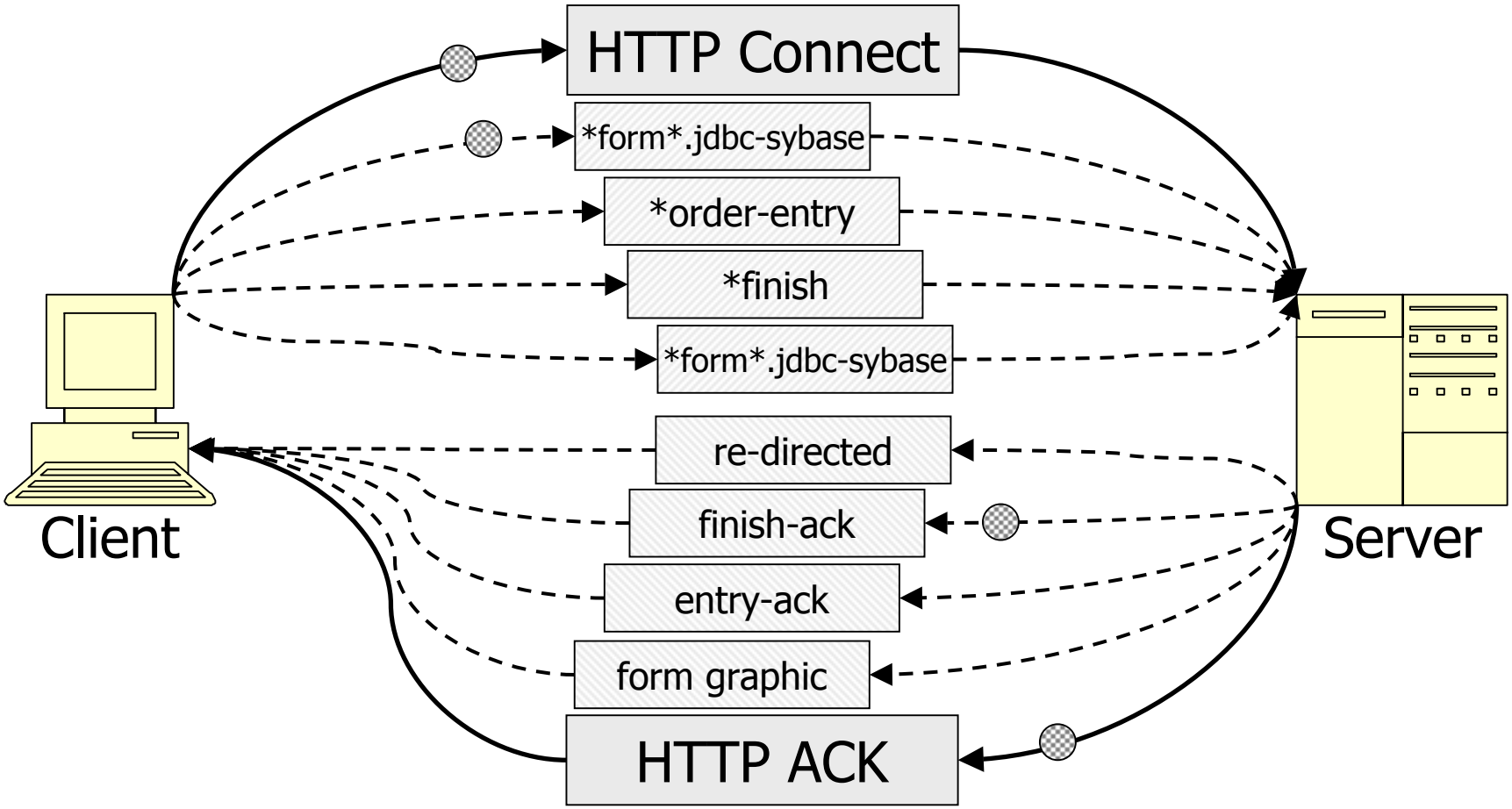
Client Overhead = SUM(SC E-S ART);

Application Response
Start to Start



Metric	Units	Description
N	Client->Svr Application Message Requests	Count of the **# Client->Server request/response pairs** measured for *application message* from the Client to the Server
Σ	mSec	Sum total of the **Start-to-Start Delta Times** in these Application Response Times
$\Sigma(x^2)$	$mSec^2$	Sum squared total of the **Start-to-Start Delta Times** in these Application Response Times
X_{max}	mSec	The maximum **Start-to-Start Delta Time** of these Application Response Times
X_{min}	mSec	The minimum **Start-to-Start Delta Time** of these Application Response Times
$\Sigma(IX)$		Sum total of the data point # times the **Start-to-Start Delta Time**

HTTP – Application Response



More Metrics…

- Jitter
- Exchange Response
- Application Response

- *Connection*
- *Connection Sequence*
- *Connection Window*
- *Routing*

TPM - Basic Statistics

- n — number of events (e.g., number of data packets)
- $X_{\min}$ — smallest value seen (e.g., smallest amount of data seen)
- $X_{\max}$ — largest value seen (e.g., largest amount of data seen)
- $\sum x$ — sum of all values (e.g., total amount of data)
- $\sum x^2$ — sum of the squares of each value
- $\sum (ix)$ — sum of each sequence index (1, 2, …, *n*) times its corresponding value
- $\sum (ix)^2$ — *sum of the squares of each sequence index (1, 2, …, n) times its corresponding value*

TPM – Selected Metrics

Primary Item	Possible Drill-Downs	Source Statistics
Failed connection attempts (count, % of total attempts)	CS count, SC count, by protocol (if multiple)	xxConnectionEstablishment.N, xxConnectionEstablishment.Min
Aborted connections (count, % of total closed)	CS count, SC count, by protocol (if multiple)	xxConnectionGracefulTerm.N, xxConnectionTimeoutTerm.N
Packets out of order (count, % of total packets)	CS count, SC count, by protocol (if multiple)	xxTraffic.N, xxConnectionOutOfOrders.N
Packets retransmitted (count, % of total packets)	CS count, SC count, by protocol (if multiple)	xxTraffic.N, xxConnectionRetransmissions.N

TPM MIB - Framework

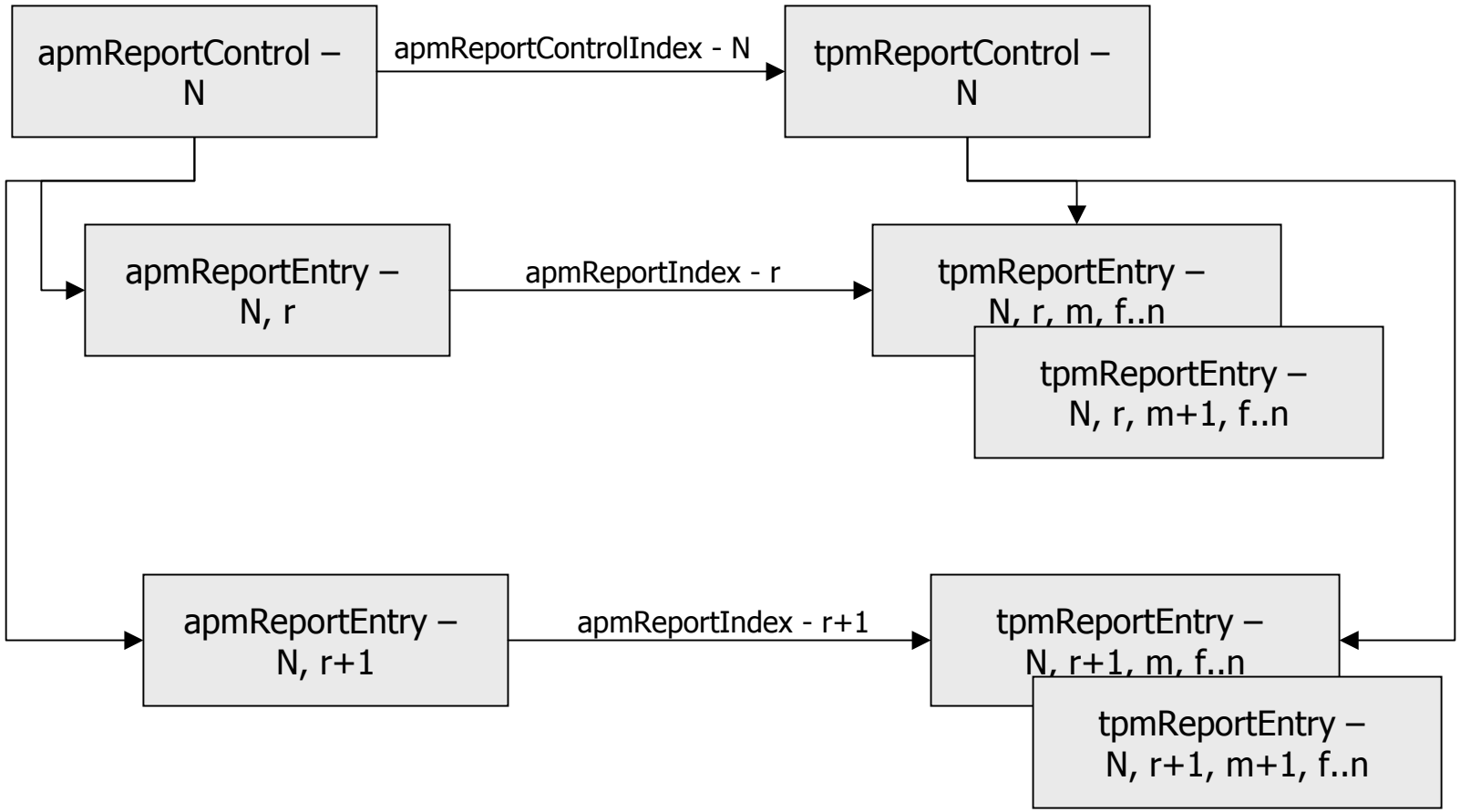
- APM MIB used to determine application flows/aggregations to report micro-flows in TPM report table.
- TPM Controls used to create performance metric reports on flows/aggregations.
- All reports are in the form of statistics
- Drill-down for performance anomalies

TPM MIB Tables

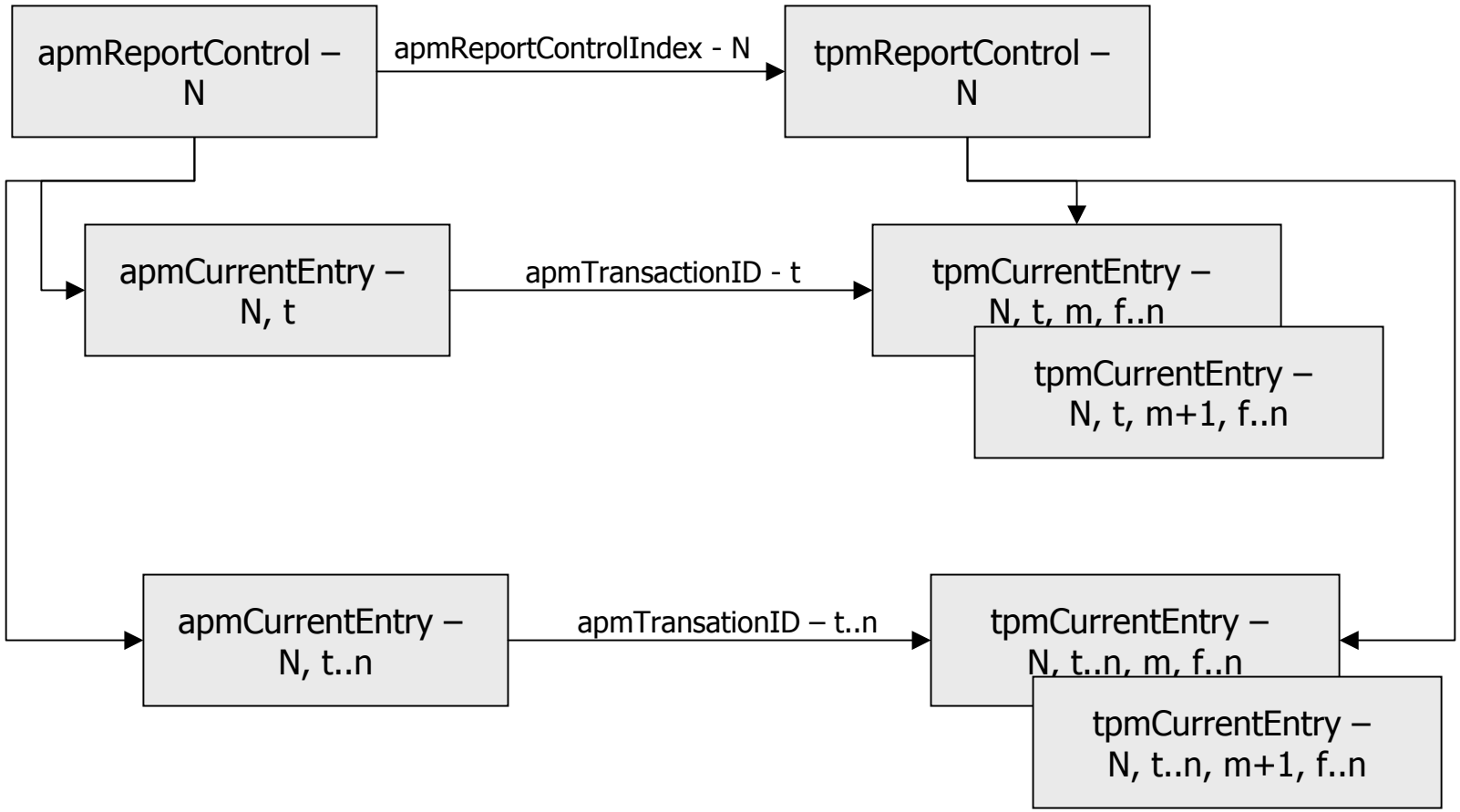
- tpmReportControlTable - Study Control
- tpmMetricTable - Metric Selection

- tpmReportTable - Detail Reports
- tpmCurrentTable - Current Aggregations
- tpmExceptionTable - Exception Reports

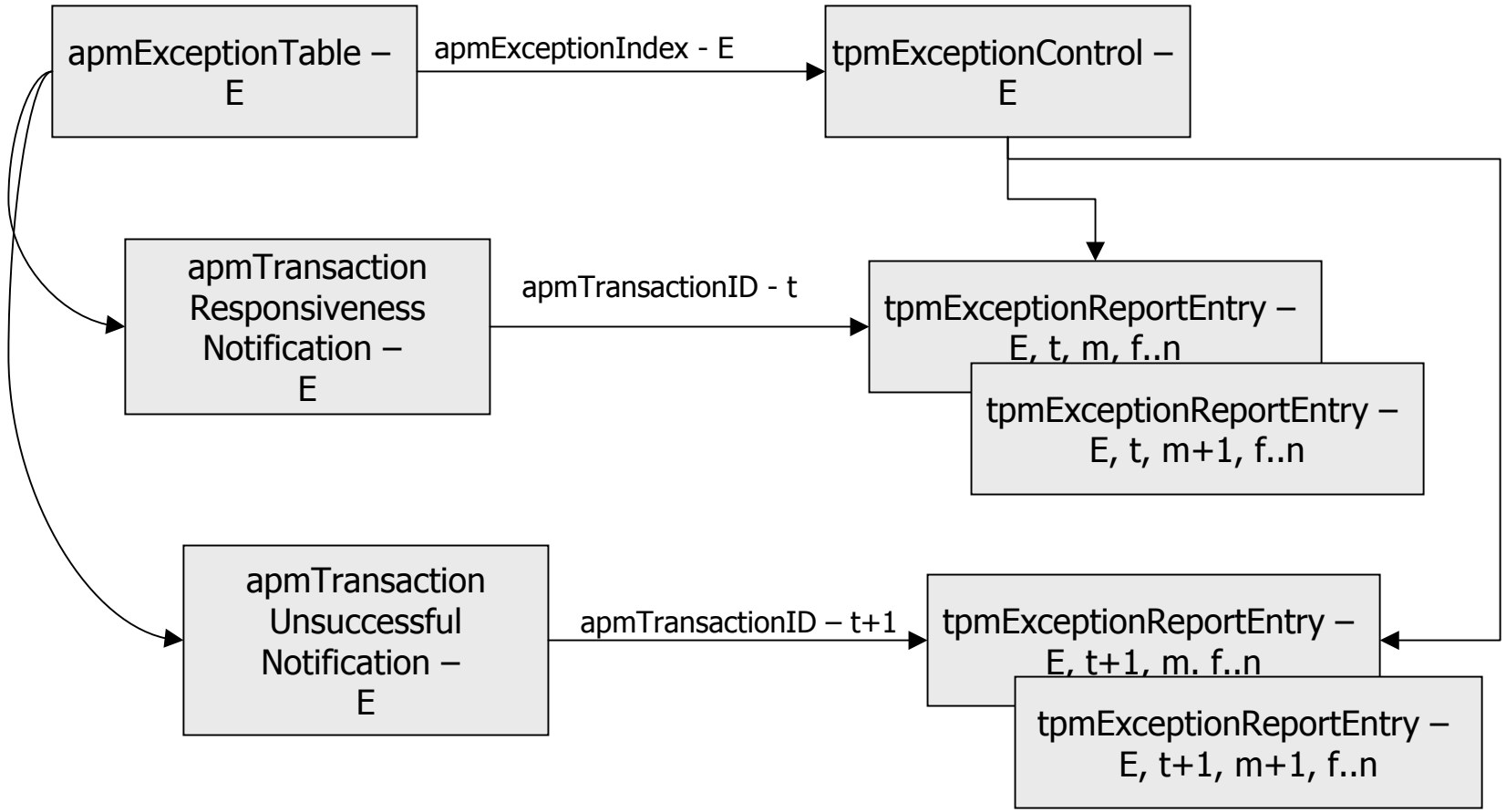
APM/TPM Report Table Linkage



APM/TPM Current Table Links



APM/TPM Exception Table Links



APM/TPM – Micro-flows

- tpmReportTable
  - tpmReportControlIndex
  - protocolLocalDirIndex – Application (micro)
  - protocolLocalDirIndex – Network
  - tpmReportServerAddress
  - tpmReportClientID
  - tpmReportMetricIndex – Metric and Protocol
  - *tpmCurrentApmTransactionID*

APM/TPM Micro-flows

Protocol	Client	Server	Protocol	Metric
WEB	*Jim*	*Amazon*	*WEB*	*Response*
WEB	Jim	Amazon	HTTP	RT-Delay
WEB	Jim	DNS-1	DNS	RT-Delay
WEB	Jim	D-Click	HTTP	RT-Delay
WEB	Jim	FTP-A	FTP	RT-Delay