The Definitive Guide to:

The Best Protocol for Real-time Data Transport

Assessing the most common protocols on 6 important categories
Identifying the Best Protocol
For strategic applications using real-time data.

There isn’t a best protocol for all situations. Choosing the right protocol for data transport depends on the needs of the application and the data. In this guide we are focused on demanding applications that use time-sensitive data. A summary of these needs are as follows.

Attributes of Strategic Applications:
- Mission critical
- Uses a complex network that wasn’t designed for the application
- Spans long distances
- Application owner believes it is more important than most network traffic

Demanding Data Has Some of these Attributes:
- Near real-time
- Streams, not files
- Latency and jitter sensitive
- Packet loss sensitive
- High throughput (>40 Mbps)

There are 6 attributes in a protocol that support these application and data needs
The 6 Attributes for Identifying the Best Protocol

Reliable
Delivers the correct data in the correct order. Important data must arrive at its intended destination. Choose a protocol that includes guaranteed delivery.

High Throughput
High bandwidth data is delivered over long-haul links. Long fat networks have very high throughput capability, but with a long transmission time. These characteristics can make it difficult for protocols to support high throughput.

Flexibility
Latency, overhead, and error correction can be balanced to fit mission requirements. A pure focus on reliability leads to reduced throughput. A pure focus on throughput leads to reduced reliability. A good protocol provides ways to make tradeoffs in key properties.
The 6 Criteria for Identifying the Best Protocol

Latency Sensitive
A latency sensitive protocol prioritizes delivering data quickly. It minimizes internal delays beyond the necessity of the physical network path. There are multiple techniques used by latency sensitive protocols including delivering multiple packets per round-trip, preemptively repairing packets, and sending multiple retransmission packets per failure.

Performant
Meets needs on networks with packet loss, reordering, duplication, jitter, and drop-outs. The protocol should meet the expectations for which it was created. It's as fast and efficient as you would expect it to be, while remaining within other constraints.

Visibility
It is useful for a transport protocol to report network impairments and impacts. These useful insights allow the parameters to be adjusted as conditions vary, especially if the protocol is flexible.
The previous slide shows in this series:
“Moving Real-time Data Across Complex Networks” &
“Choosing a Transport Protocol for Real-time Data”
identified the main protocol choices in 3 categories.

We took a careful look into each of the major protocols to assess their ability to cope with real-time data transport. We rated each protocol for the 6 criteria on a 10 point scale.
TCP variants as a category have made improvements in throughput within the framework of TCP. These protocols are fundamentally limited when faced with long and varying latency or congested networks. Even for the best, throughput is limited.

**Even the best TCP Variant scores only 4 out of 10. As a group, the group has the lowest average score of 3.1.**

<table>
<thead>
<tr>
<th>TCP-Cubic</th>
<th>TCP-Illinois</th>
<th>SCPS-TP</th>
<th>SCTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating out of 10</td>
<td>2.5</td>
<td>2.8</td>
<td>4.0</td>
</tr>
<tr>
<td>Overall</td>
<td>Improved ramp-up and error recovery time</td>
<td>Pre-emptively avoids congestion loss</td>
<td>Favored for space, anticipates long RTT</td>
</tr>
<tr>
<td>Reliability</td>
<td>8</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>High Throughput</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Flexibility</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Latency sensitive</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Performant</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Visibility</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
**TCP Variants**

**TCP-Cubic**
This version makes an improvement on standard TCP (NewReno version). It rapidly converges on maximum error-free throughput by ramping up the data rate with an exponential rather than a linear function. It is fundamentally TCP with a focus on congestion avoidance, therefore it will lower throughput in the face of a lossy network.

**TCP-Illinois**
Detects a network that is becoming congested before any loss occurs. This prevents unnecessary packet loss. As with all versions of TCP, it has a focus on congestion avoidance, therefore it will lower throughput in the face of a lossy network. Illinois is a slight improvement on Cubic for latency and throughput.

**SCPS-TP**
Adds CCSDS extensions to TCP. These extensions are designed for space applications that inherently include long round-trip times. It has the highest throughput, best latency sensitivity and most flexibility for this group.

**SCTP**
Stream Control Transmission Protocol adds the concept of multiple paths to TCP. This adds extra resiliency. However, it implements (old) TCP congestion control algorithms. For this reason, it will not maintain throughput in the face of loss on the network.
Reliable UDP approaches produce some strong candidates. There are significant tradeoffs between these choices based on applications needs.

**UDT**: Good overall choice, some issues with implementation.

**NORM**: Most mature and flexible. Better throughput and more performant than UDT, but a little lower reliability.

**SMPTE 2022**: Quite different than the others, very good latency but lower reliability; more UDP like.
Reliable UDP

RUDP
RUDP is built on sound principles, but it was designed in 1999 and has not been maintained since. As such, it isn't focused on high-throughput, low latency needs. It includes built-in fixed delays and very small queues. It also offers few if any maintained implementations.

UDT
UDT is built on sound principles, and it makes congestion control optional. However, the implementation faces some performance challenges in that it does not make efficient use of available bandwidth. Therefore UDT is not performant and does not live up to its promise.

ENET
ENET seems to be built on sound principles (since there is no protocol specification, it's hard to be sure). However, users report that with conditions of 1Mbps and 0.85% packet loss, TCP actually performs better. This raises serious performance concerns.

NORM
We really like NORM. It is designed for one-to-many reliable delivery, is well documented, and is well maintained. The protocol includes Reed-Solomon PFEC which is computationally expensive, leading to limited throughput. Its latency is better than many choices, but is not as good as the best alternatives.

SMPTE 2022-1
SMPTE is emerging as a standard for broadcast media applications, so it’s a serious choice. Its reliability is based on forward error correction (FEC). Since FEC cannot correct all errors, this protocol leads to lower reliability than other choices in this category. It can improve reliability over native UDP with minimal added latency.
**DataDefender** is a purpose built product for transporting critical real-time data across complex networks.

**FASP** A solid protocol and a good choice. Because designed for file transfer, it is less flexible, adds more latency, and provides less visibility into network performance.

**DataDefender and FASP should receive a closer look.**

### The Best Protocol for Real-time Data Transport

#### Rating the most common protocols for real-time data transport

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Overall</th>
<th>Reliability</th>
<th>High Throughput</th>
<th>Flexibility</th>
<th>Latency sensitive</th>
<th>Performant</th>
<th>Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataDefender</td>
<td>9.2</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>FASP</td>
<td>7.0</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>MPT</td>
<td>5.0</td>
<td>10</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>SpeedyPackets</td>
<td>4.2</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
DataDefender
DataDefender is a single purpose protocol designed specifically for strategic applications and streaming data. It provides multiple paths like SCTP for the ultimate in reliability, can run at 10Gbps in lossy environments, provides tunable PFEC and retransmission, tunable latency, and can ride out massive loss without impact. Finally, it has a full analytics suite. We set out to build the best protocol for this type of application.

FASP
FASP has a heritage in file transfer applications. It’s an excellent protocol but is not optimized for real-time streams and strategic applications. It can turn off its delay-based congestion-control and has many good features, but does not have forward error correction, latency controls, or network analytics.

MPT
MPT is designed for clients pulling resources from a server. This model has severe challenges for data that is not at rest, like streams. For real-time data, MPT isn’t one of the best choices.

SpeedyPackets
This product adds Forward Error Correction (FEC) to standard TCP. This slightly improves throughput in lossy conditions. However, worst-case latency and throughput are no better than standard TCP because packet loss will trigger congestion control.
Choosing the Finalists

The assessment has identified 5 protocols as finalists that need a little further examination. Three of these are from the Reliable UDP family and two are commercial products. They all scored higher than 5.0 on our rating scale and scored at least 7 for latency sensitivity.
For strategic applications that pass real-time data across complex networks, DataDefender is the best choice.

<table>
<thead>
<tr>
<th></th>
<th>UDT</th>
<th>NORM</th>
<th>SMPTE-2022-1</th>
<th>DataDefender</th>
<th>FASP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rating out of 10</strong></td>
<td>5.7</td>
<td>6.2</td>
<td>5.2</td>
<td>9.2</td>
<td>7.0</td>
</tr>
<tr>
<td>Overall</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Reliability</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>High Throughput</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Flexibility</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Latency sensitive</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Performant</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

**Reliability.** Most of the finalists are strong, DataDefender and FASP are the best.  
**Throughput.** Likewise, DataDefender and FASP have the best throughput.  
**Latency.** SMPTE and DataDefender are best for latency sensitivity, though all the finalists are strong.  
**Flexibility & Visibility.** DataDefender has a special emphasis on flexibility and visibility. Its provides unique tools for diagnosing network challenges and overcoming them.
Conclusion

DataDefender guarantees performance of strategic applications that run across complex networks. It does this by assuring critical, real-time data receives the highest priority. This is especially important when data is vital and mission critical, but the mission owner does not control the network. It gives reliable data delivery, high throughout and minimizes latency. It also includes an analytics suite to allow visibility into the network challenges that affect performance.
There are a lot of transport protocols. However, they are designed for different needs. Most don’t make your data the highest priority. DataDefender is a protocol that is optimized for critical, real-time data and for strategic applications with high priority needs.

Get Your Data Delivered!

For more information, visit:

Kratos RT Logic
www.rtlogic.com/products/datadefender

To discuss DataDefender details or see a demo:

e-mail: sales@rtlogic.com