

TCP-Tactical PCS, Congestion Control, and Striping

TCP-Tactical is a reliable transport protocol that replaces legacy TCP stacks. TCP-Tactical remains backwards compatible with legacy TCP implementations. By implementing rate-based transmission with an innovative congestion-control algorithm, TCP-Tactical can dramatically accelerate application goodput over lossy, high-latency tactical SATCOM links.

A Path Characterization Service (PCS) is deployed on every LAN where TCP-Tactical stacks exist. The PCS defines the capacity and corruptive loss rate of the coincident LAN's ingress and egress links – information that is collectively referred to as the *network descriptor*.

The PCS provides the network descriptor to the TCP-Tactical stacks on its LAN. TCP-Tactical stacks establish connections in a similar fashion to legacy TCP connections: via a three-way handshake. The TCP-Tactical three-way handshake is augmented as follows.

- In the initial augmented SYN (A-SYN), the initiator specifies the TCP-Tactical option in the TCP header
- If the acceptor of the A-SYN is a legacy TCP stack, the TCP-Tactical option is ignored, and the three-way handshake is completed according to existing TCP protocol conventions. However, if the acceptor is also a TCP-Tactical stack, the acceptor replies with an A-SYN/A-ACK packet that contains the acceptor's local network descriptor.
- Upon receiving the A-SYN/A-ACK, the initiator determines which ingress and egress links will be used for both ends of the connection. The initiator makes the link determination by examining its local network descriptor as well as the acceptor's network descriptor that was contained in the A-SYN/A-ACK. The result is a *path descriptor* that specifies the ingress/egress links that will be used at both ends of the connection, as well as the maximum transmit rate and corruptive loss rate for the path. The path descriptor is provided to the acceptor when the initiator completes the three-way handshake with a final A-ACK.

After the three-way handshake has completed, each side of the connection has a maximum transmission rate at which it can send data to the other side as well a maximum corruptive loss rate that precludes premature invocation of the TCP-Tactical congestion-control algorithm.

Consider the following case where the TCP-Tactical host shown on the right side of the diagram performs a passive-mode ftp get operation from the ftp server shown on the left side of the diagram.

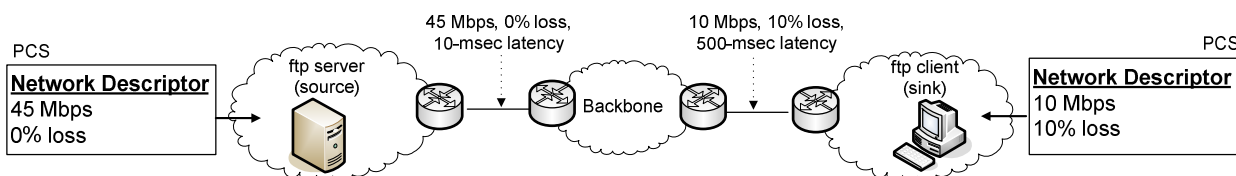


Figure 1. Example Internetwork and Associated Network Descriptors.

The three-way handshake between the two hosts is depicted in Figure 2 (though the IP address information necessary for microflow transmission is omitted for simplicity). After completion of the three-way handshake, the source (ftp server) begins transmitting data to the sink (ftp client).

Unlike legacy TCP, however, TCP-Tactical does not proceed through slow-start in a quest to ascertain available bandwidth; instead, TCP-Tactical immediately begins transmitting at the rate specified in the path descriptor that was contained in the final A-ACK (10 Mbps in this case).

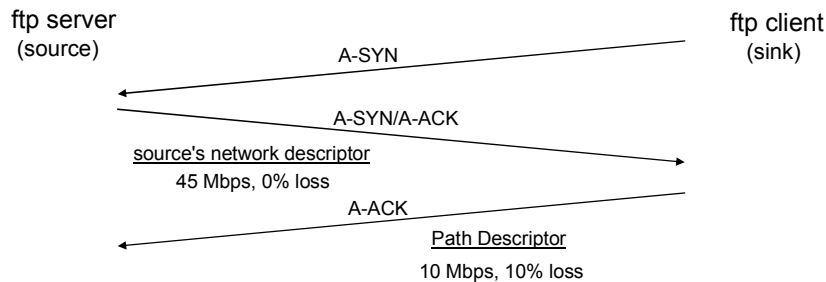


Figure 2. TCP-Tactical Three-Way Handshake.

Furthermore, that same path descriptor also defined the maximum corruptive loss rate for the connection to be 10%, and the key to accelerating reliable transport performance is differentiating corruptive from congestive loss. In this example, the connection can tolerate a 10% packet loss before TCP-Tactical congestion control is invoked.

Determining when the path-descriptor-defined corruptive loss rate has been exceeded is accomplished by the data source marking each transmitted packet with an aggregate count of how many packets have been sent up to the current packet and the data sink counting the incoming packets. When packets are received at the data sink, the sink compares the number of packets the source alleges to have sent with the number of packets the sink has actually received for the connection. If the difference exceeds the corruptive loss rate over a given period, an explicit transmit-rate reduction notification is placed in the sink's next ACK to the source.

Congestion Control

Extending the previous example illustrates how TCP-Tactical's congestion-control algorithm is invoked and its behavior following invocation. Figure 3 shows the source sending 20 packets at 10Mbps over the high latency path previously depicted in Figure 1. Each packet is marked 1, 2, ..., 20.

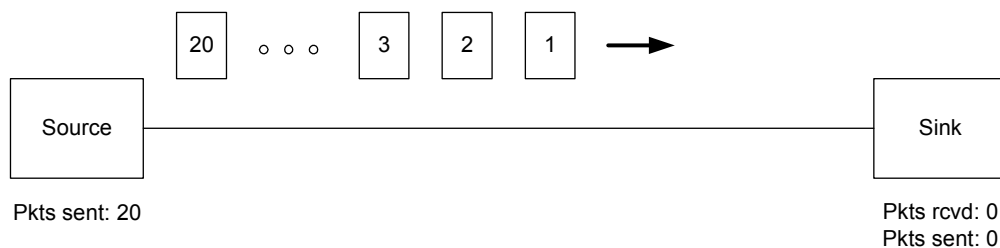


Figure 3. Source Marking of Transmitted Packets.

Assume that a constant 50% (5 Mbps) of the available bandwidth between the source and sink is currently being consumed by other traffic, i.e., for every 10 packets burst at the maximum LAN transmission rate by the source, 5 are lost due to congestion. When the packet with send count of 10 reaches the sink, 6 out of the 10 transmitted packets will have been lost (1 due to corruption, 5 due to congestion), as shown in Figure 4.

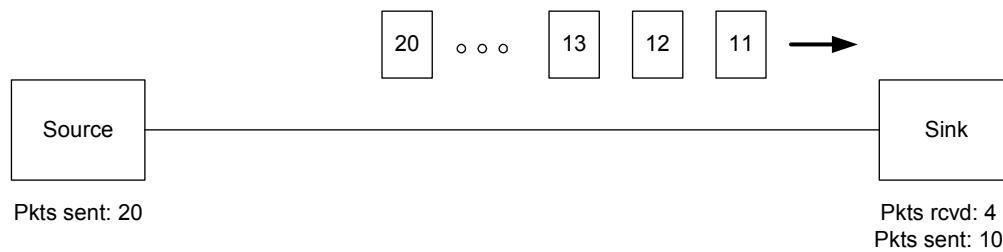


Figure 4. Sink Detects 60% Packet Loss.

At this point the sink calculates the current congestion as follows:

$$\begin{aligned} \text{current loss} &= (10 - 4)/10 = 60\% \\ \text{worst-case corruptive loss} &= 10\% \\ \text{overage (congestive-loss estimate)} &= 60\% - 10\% = 50\% \end{aligned}$$

As a result, the sink explicitly notifies the source to reduce its transmission rate by 50%, as depicted in Figure 5.

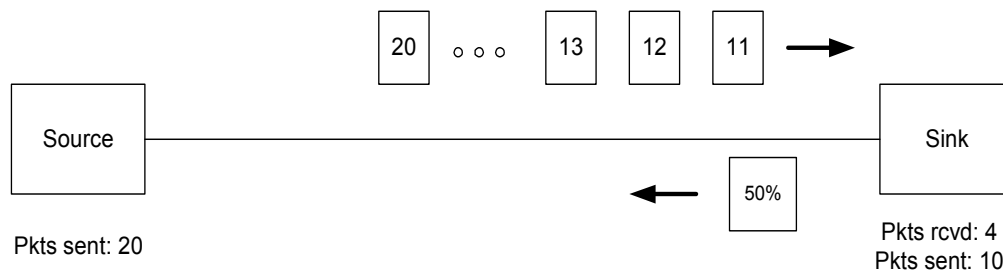


Figure 5. Explicit Sink-to-Source Congestion Notification.

Upon receiving the congestion update from the sink, the source reduces its transmission rate by 50% and awaits further updates from the sink. If the sink sends another congestion update, the source again reduces its transmission rate by the amount specified by the sink. If no further congestion update arrives within a specified time interval, the source enters a probing phase wherein the transmission rate is increased by n , where n is a tunable TCP-Tactical parameter (default value for n is 5%). If no congestion update arrives at the source within time t , the send rate is again increased by n , and the process is repeated until either the sink signals congestion or the maximum transmission rate specified by the path descriptor is achieved. As a result, the back-off percentages specified by the congestion notifications from the sink decrease as the source fine tunes its transmission rate according to current path conditions.

TCP-Tactical currently implements the congestion-control algorithm described above, and the algorithm is being further refined with an admission-control module that will reduce transmission-rate spikes that occur when new flows are initiated. However, even with current transmission-rate spikes at flow initiation, TCP-Tactical has been shown to improve application-layer throughput by over an order of magnitude at 5-10% corruptive loss rates and 500-ms round-trip latencies.

Data Striping and Microflows

A *multi-homed* computer is a computer with more than one IP address. More than one IP address

can be associated with a single physical network interface, and more than one physical network interface can be installed in a computer. However, a specific IP addresses can only be associated with one physical network interface. E.g., a computer with physical network interfaces *i* and *j* cannot associate IP addresses *w.x.y.z* with both *i* and *j*.

A TCP flow is defined by the following four-tuple:

< IP address A, TCP port X, IP address B, TCP port Y >

Where ($A = B$) is permissible, but ($X = Y$) is not. Software applications associate with a flow either through a *connect* system call made by the connection initiator or by an *accept* system call (by a process that is waiting for the incoming connection).

A multi-homed TCP-Tactical computer can decomposed a single TCP flow (a *macroflow*) into two or more microflows, with no additional intervention by the application. Microflows differ from one another only by their source and destination IP addresses, i.e., their source and destination TCP port numbers remain the same as the macroflow

The microflow concept can be illustrated by an example. Consider Figure 6, where the multi-homed computer (host α) on the left has two physical network interfaces and two IP addresses; IP address 192.168.102.3 is associated with one physical interface and IP address 192.168.145.3 is associated with the other physical interface. Host β , the computer on the right, has a single address – 192.168.35.3 – that is associated with the β 's sole physical network interface.

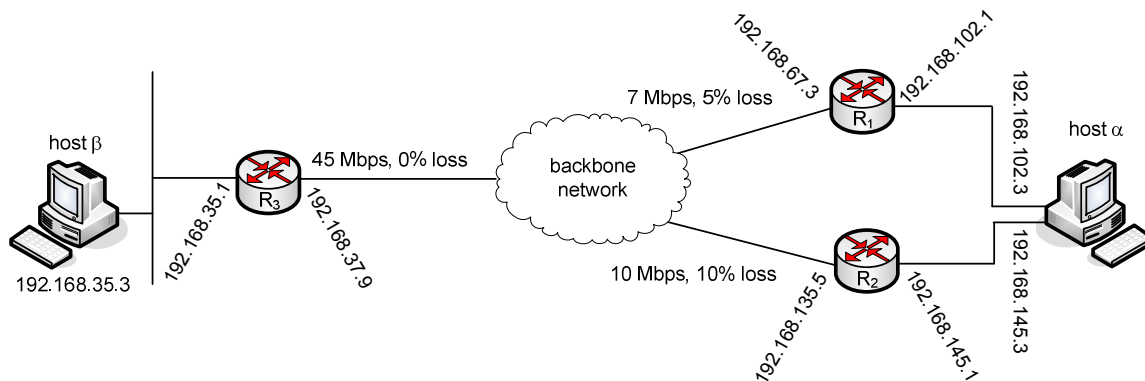


Figure 6. Microflows.

A depiction of the TCP-Tactical three-way handshake, including the IP address information in the network and path descriptors, is presented in Figure 7.

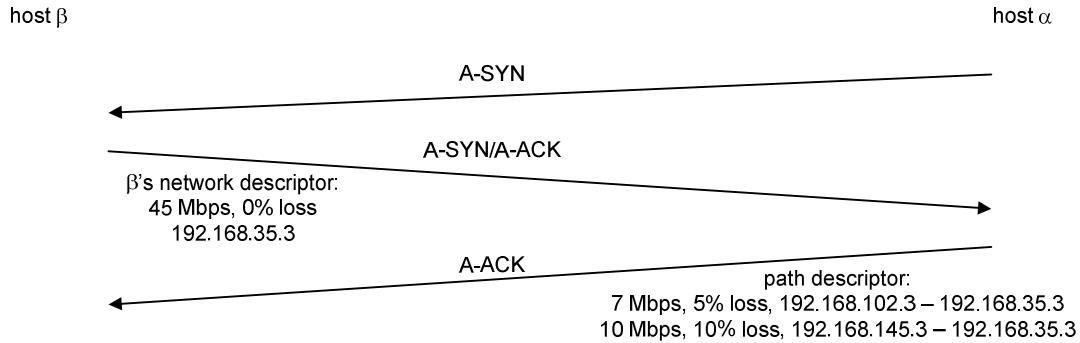


Figure 7. Three-way Handshake with IP Addresses.

Upon receiving the A-SYN/A-ACK packet from β , α examines β 's network descriptor and compares it with its own local network descriptor and its available IP addresses. α next determines that two ingress/egress links are available to it (at IP addresses 192.168.102.3 and 192.168.145.3) and that a single ingress/egress link is available to β at IP address 192.168.35.3. (In the general case where there are n ingress/egress links at α and m ingress/egress links at β , α will determine all $n*m$ paths after receiving β 's network descriptor.)

α also associates a maximum bandwidth and corruptive-loss threshold with each path it derives – in this case, 7Mbps/5% loss for the <192.168.102.3, 192.168.35.3> path, 10Mbps/10% loss for the <192.168.145.3, 192.168.35.3> path – based on α 's local network descriptor and the remote network descriptor provided in β 's A-SYN/A-ACK.

The unmodified TCP application continues to transmit data to and receive data from the TCP-Tactical stack (through the standard socket interface) as if a single macroflow were still being used. Unbeknownst to the application, however, TCP-Tactical decomposes the macroflow into two microflows; these microflows are uniquely identified by the original TCP port numbers of the TCP-Tactical three-way handshake and the IP endpoint addresses of available paths determined by initiator of the connection. In the example presented in Figure 7, the microflow IP addresses are <192.168.102.3, 192.168.35.3> and <192.168.145.3, 192.168.35.3>.

As data for transmission is received from an application, TCP-Tactical stripes the transmit data over all available microflows by placing each successive packet to be transmitted in the next available microflow's output queue. In this fashion, microflows that are associated with a slower egress link will handle fewer packets over time than a higher-speed flow. TCP-Tactical congestion control (discussed previously) is performed independently on each microflow.

Data received over multiple microflows is reassembled into a single receive queue to ensure ordered delivery prior to furnishing the received data to the application layer (through the standard sockets interface).