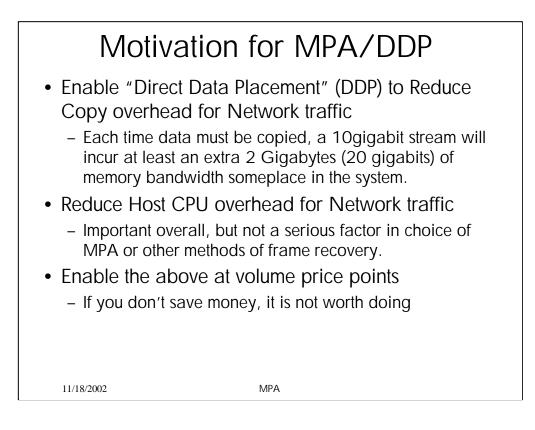


Marker (Protocol Data Unit) Aligned Framing, or MPA.



•We designed MPA to allow the DDP protocol (Direct data placement, draft-shahiwarp-ddp-01) to work over TCP while maintaining the advantages of DDP. DDP allows the user's data to land exactly where he wants it without copies.

•Copies either steals performance (ultimately a cost overhead), or directly increase costs in our systems.

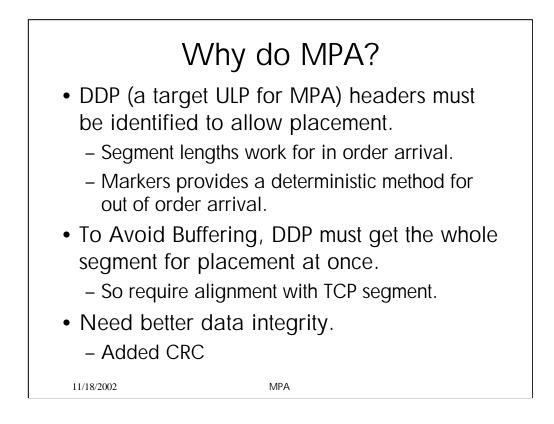
•The copy bandwidth includes CPU overhead if done by the CPU as in traditional sw stacks.

•If done by the NIC to host memory, it strains I/O and memory busses.

•If done on NIC, but to external RAM, that incurs cost.

•If the cost of doing this gets too high, it will not be adopted, because TCP with SW stack is the competition.

•DDP/MPA can save system overhead of 30-40%?? So we are talking about an extra server for every 3...



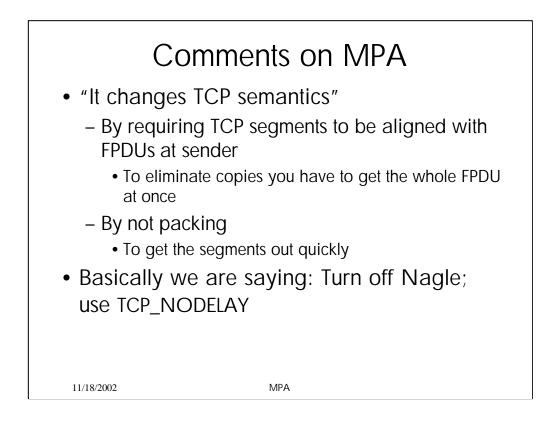
•To avoid the copies, DDP needs its headers accurately identified. Not a problem for in order arrival, the segment lengths allow easily finding the next segment.

•For Out of order arrival, a Marker inserted at 512 byte intervals is used to locate the headers in a deterministic way. The receiver can always find a marker by examining the TCP sequence number.

•We decided on the use of Markers to identify the FPDU start because it was fully "deterministic" unlike other mechanisms that search the stream for (or just "catching") a pattern to identify the FPDU start (as in draft-ietf-tsvwg-tcp-ulp-frame-01.txt). No chance of mis-identifying a user data pattern as the start of an FPDU.

•If the whole DDP segment arrives at once, you don't have to store a partial segment or header while waiting for the rest, so we specify "Alignment" at the sender. In the usual case, the segments arrive intact.

•And TCP's data integrity is not always as good as we would like, so we upgraded it with MPA to be on a par with SCTP.



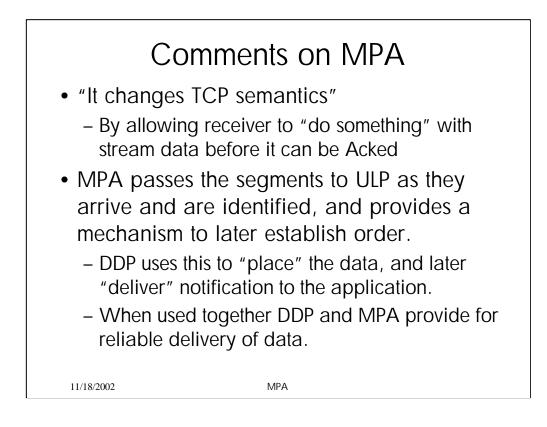
•Issues from the reflector; The most vocal commenters claims that we are "Changing TCP semantics" in various ways.

•As mentioned before, In order to avoid copies, we have to identify and receive whole segments. This leads to alignment and no packing.

•This is fairly common practice already, most implementations have a way of disabling the Nagle algorithm, for example the Socket's TCP_NODELAY option. When Applications use records smaller than a TCP segment, this is basically what you get. And MPA allows its ULP, (DDP) to see the maximum segment size, so this works.

•Copy avoidance presumes that nothing interferes with alignment or full FPDU reception. So no "middle boxes" and full MPA/TCP implementations. This is very doable in the data center, and probably quite reasonable for certain applications outside the data center.

•If we don't get full, aligned, segments, then the recievers must still work, but NIC performance will often degrade.



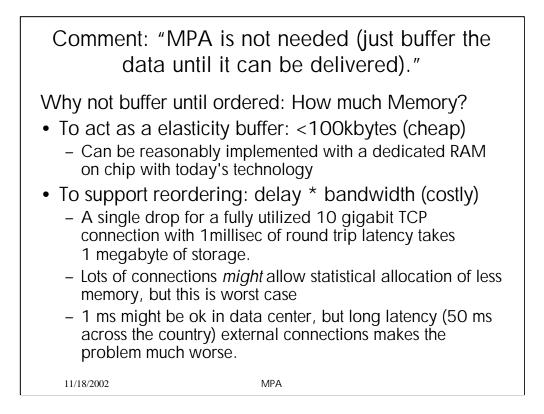
•Another reflector comment is that we allow the receiver to "do something", in this case pass the data to the DDP layer, before TCP has reached the point where it can be Acknowledged. This applies to Out of order segments.

•For one thing, we are not Requiring this ability, we are only enabling it. An implementation is free to buffer and reorder if it wants to, and other implementations can use DDP to place the out of order data immediately.

•An MPA implementation can provide the complete, identified and validated segment to DDP regardless of order. The implementation also provides a mechanism to identify the correct order for the segments; this is usually based on the TCP sequence number, but can be anything.

•When DDP is the ULP, it can use the segments immediately for "placement" (storage into the final buffer), and later can "deliver" the notification to the application that entire messages have arrived and are in order.

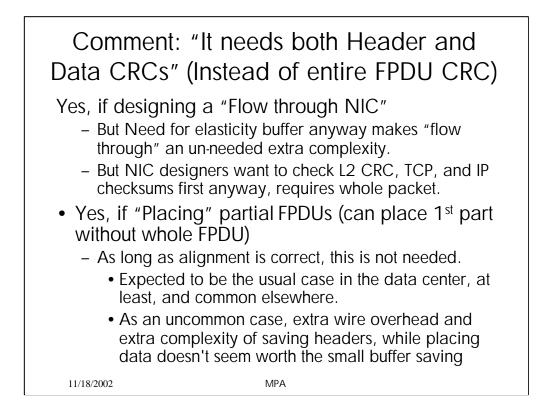
•The end result is that MPA and DDP together have provided a reliable delivery of data to the application.



•Suggestion has been made to just use re-order buffer, "it is not that much more memory than an elasticity buffer", so here is an example of why se don't think this is cost effective...

•Really large elasticity buffers don't help much, the NIC and system must keep up or use a scheme to back off the connection speed. This would typically be some heuristic for reduction of RWIN, or drop packets to invoke congestion control at sender. Small elasticity buffer can be implemented in dedicated on chip memory, so this need not count as a "real" copy for system level bandwidth reduction.

•Some argue that the ONLY way to deal with slower NIC/systems is to use buffers per connection, tied to RWIN, but this needs lots more memory. With MPA, vendors get the choice. The large memory needed needs to be in off chip memory, either host, or dedicated. Either way, you pay...



Maintaining alignment/framing/DDP is expected to be the common case, will be implemented as "fast path" in NIC. Once this breaks, NICs will probably go to "slow path" processing, using some host buffers for re-assembly. When this happens, performance will probably degrade.