




Reliable Parallel File Transfer

Lu Zhao and Larry Dunning

Department of Computer Science
Bowling Green State University, Bowling Green, OH

04/01/2004





File Transfer Mechanisms

- FTP
 - Grid FTP, Parallel FTP
 - HTTP
 - bitTorrent
-



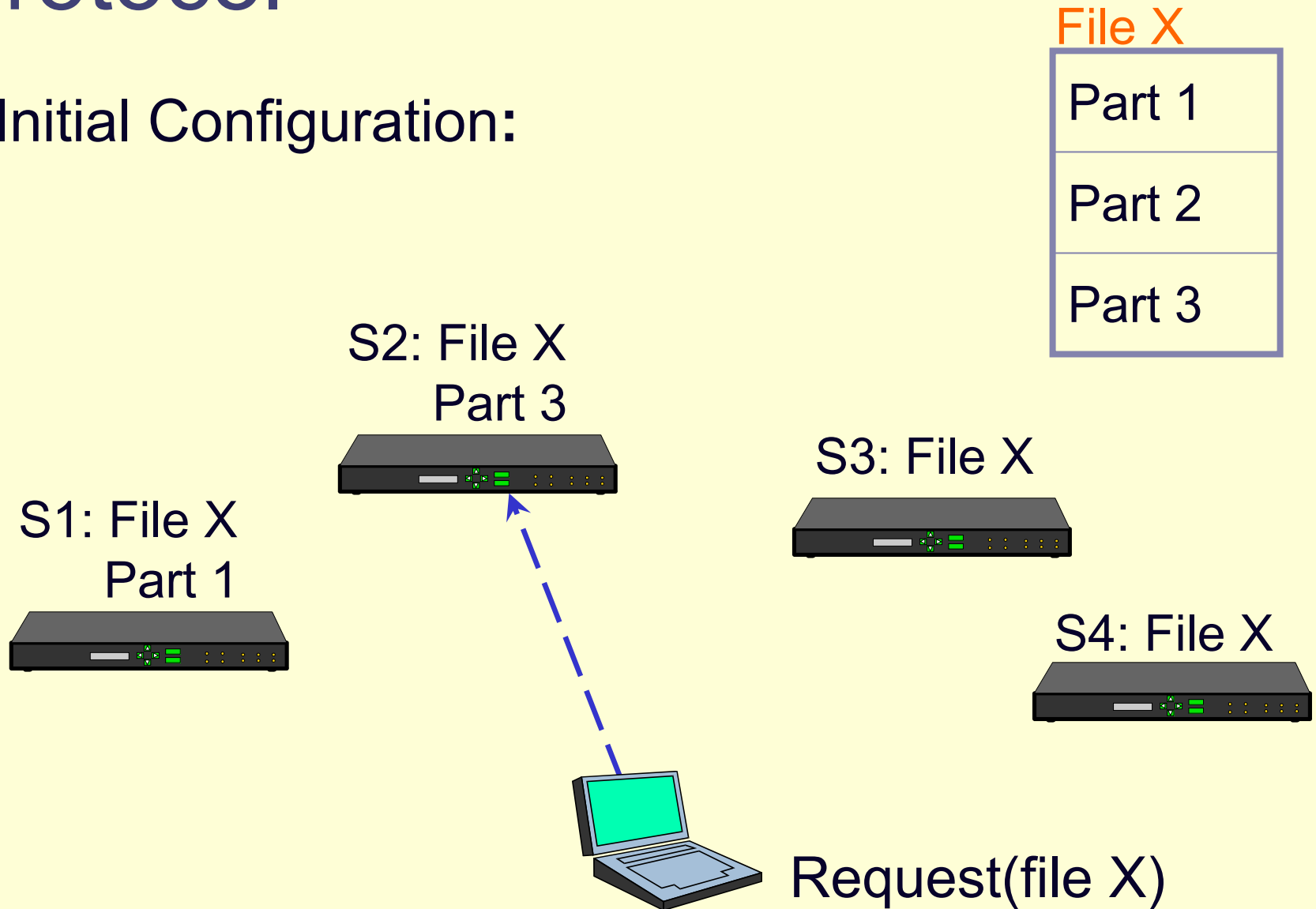
Introduction

- To develop a protocol and its implementation with the emphases on both parallelism and reliability in file transfer.



Protocol

Initial Configuration:



Protocol

Request

ASK <pathname> CR

Version: 0.1 CR

Request_id: <unique_value> CR

Listen_Address: <value> CR

Listen_Port: <value> CR

Integrity_Check: <ichk_list> CR

Redundant_Check: <rdt_list> CR
CR

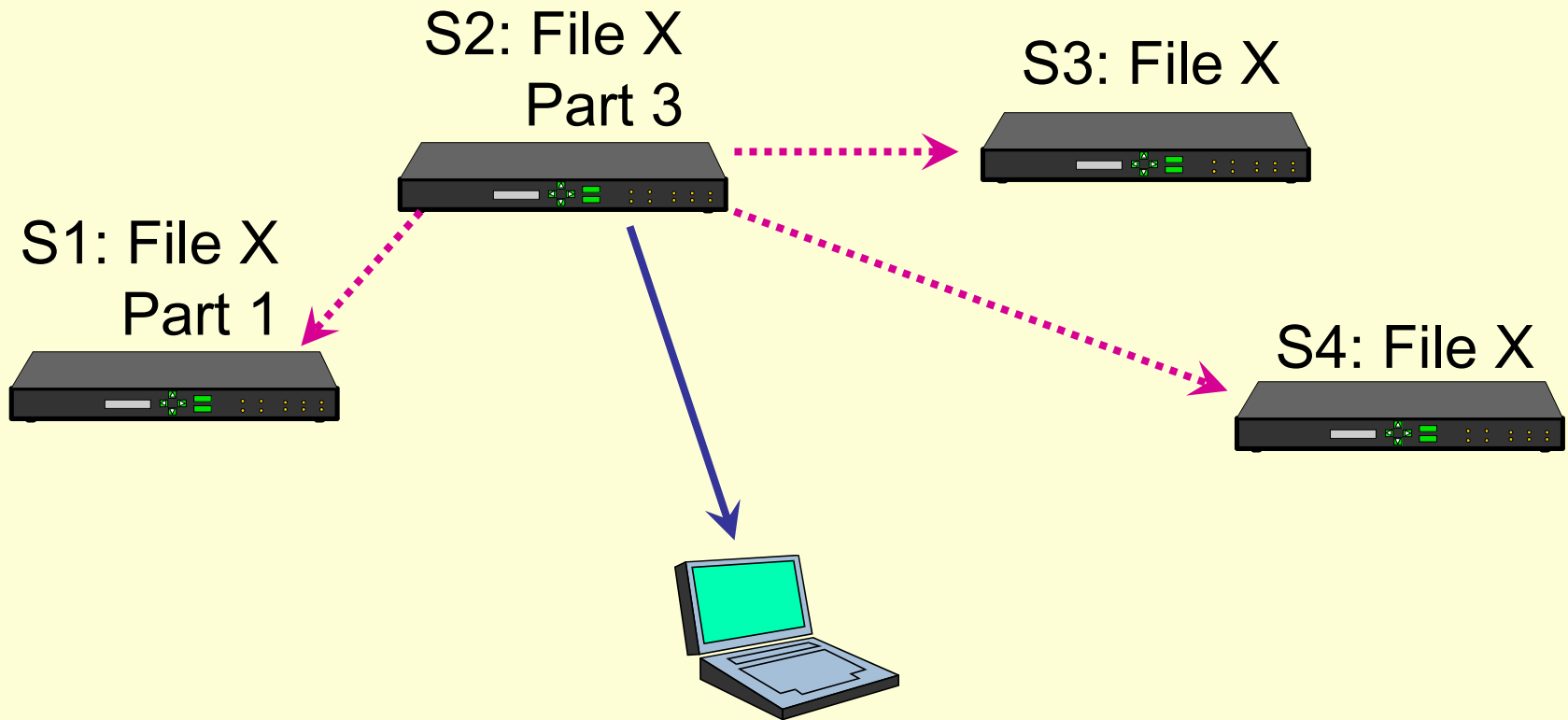
ichk_list: ichk_value | ichk_list, ichk_value

ichk_value: none | sha1

rdt_list: rdt_set | rdt_list, rdt_set

rdt_set: xor (piece₁, . . . , piece_n) | none

Protocol - Server scheduling



ACK/File_Distribution_Info (FDI)



Schedule/FDI

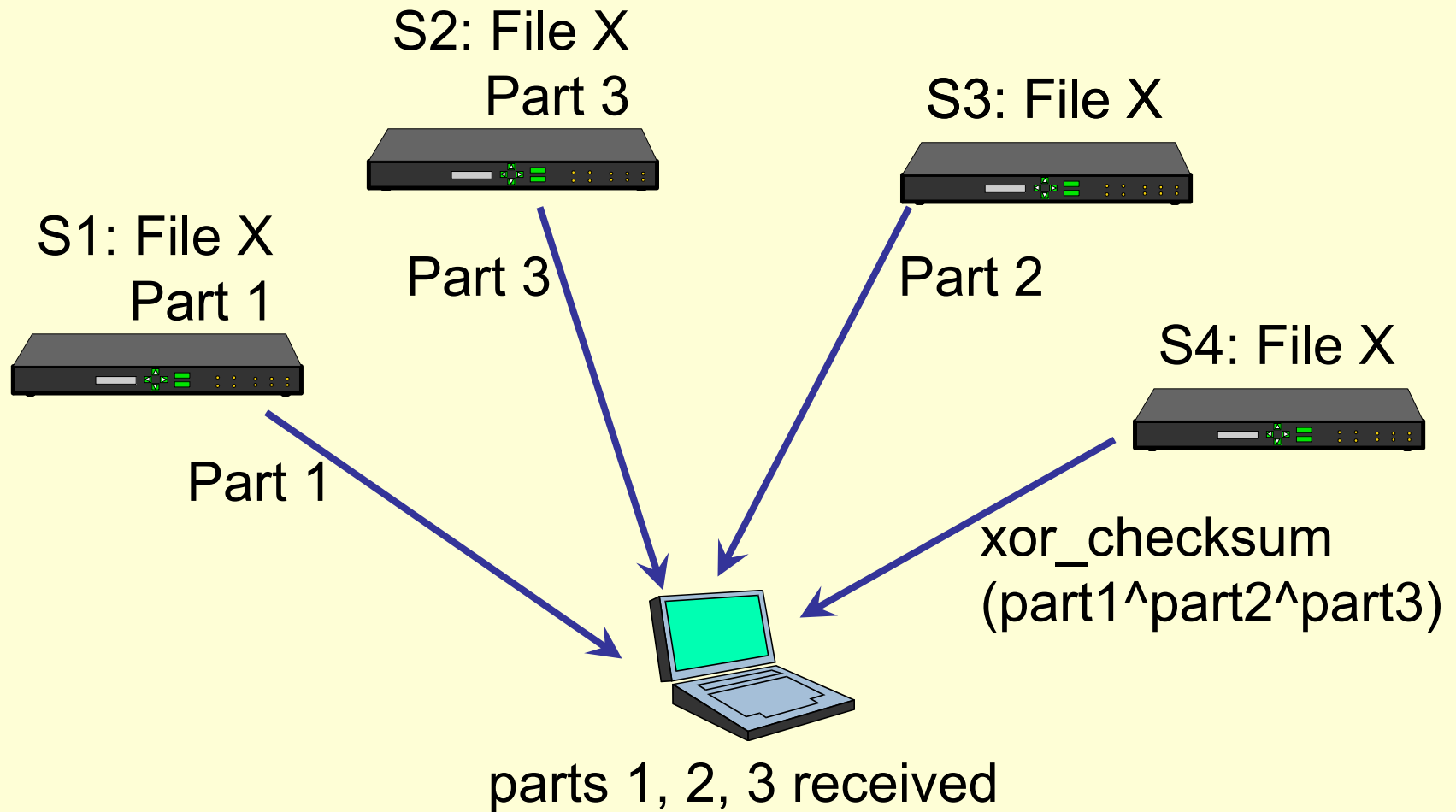
Protocol

DELIVER <pathname> CR
To_Address: <client_ip> CR
To_port: <client_listen_port> CR
Version: 0.1 CR

Schedule message

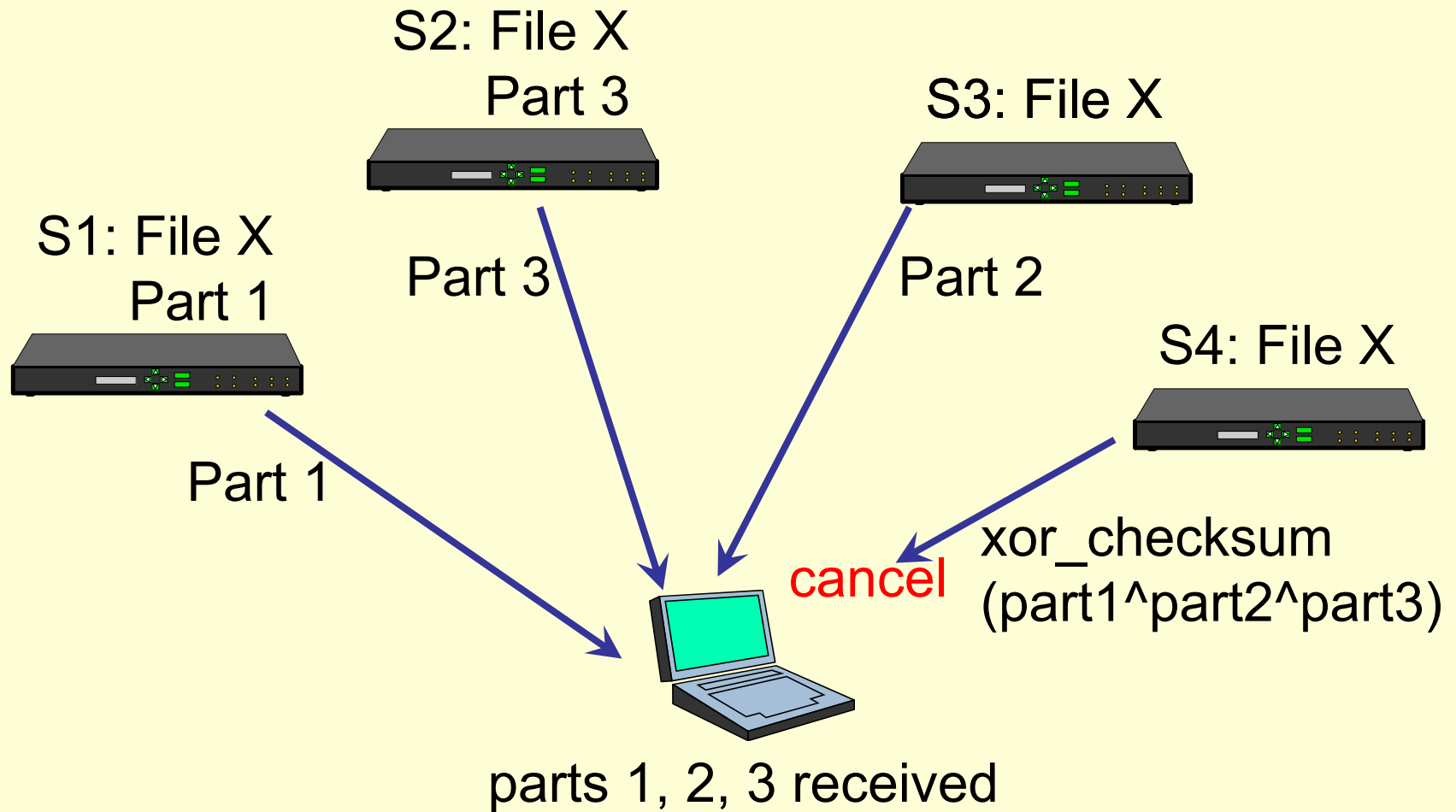
Request_id: <value> CR
Piece_Length: <value> CR
Piece_List: <piece_list> CR
Integrity_Check: <ichk_list> CR
Redundant_Check: <rdt_list> CR
CR

Protocol - Client receiving



→ Data(Preceded by FDI)


Protocol - Normal operation



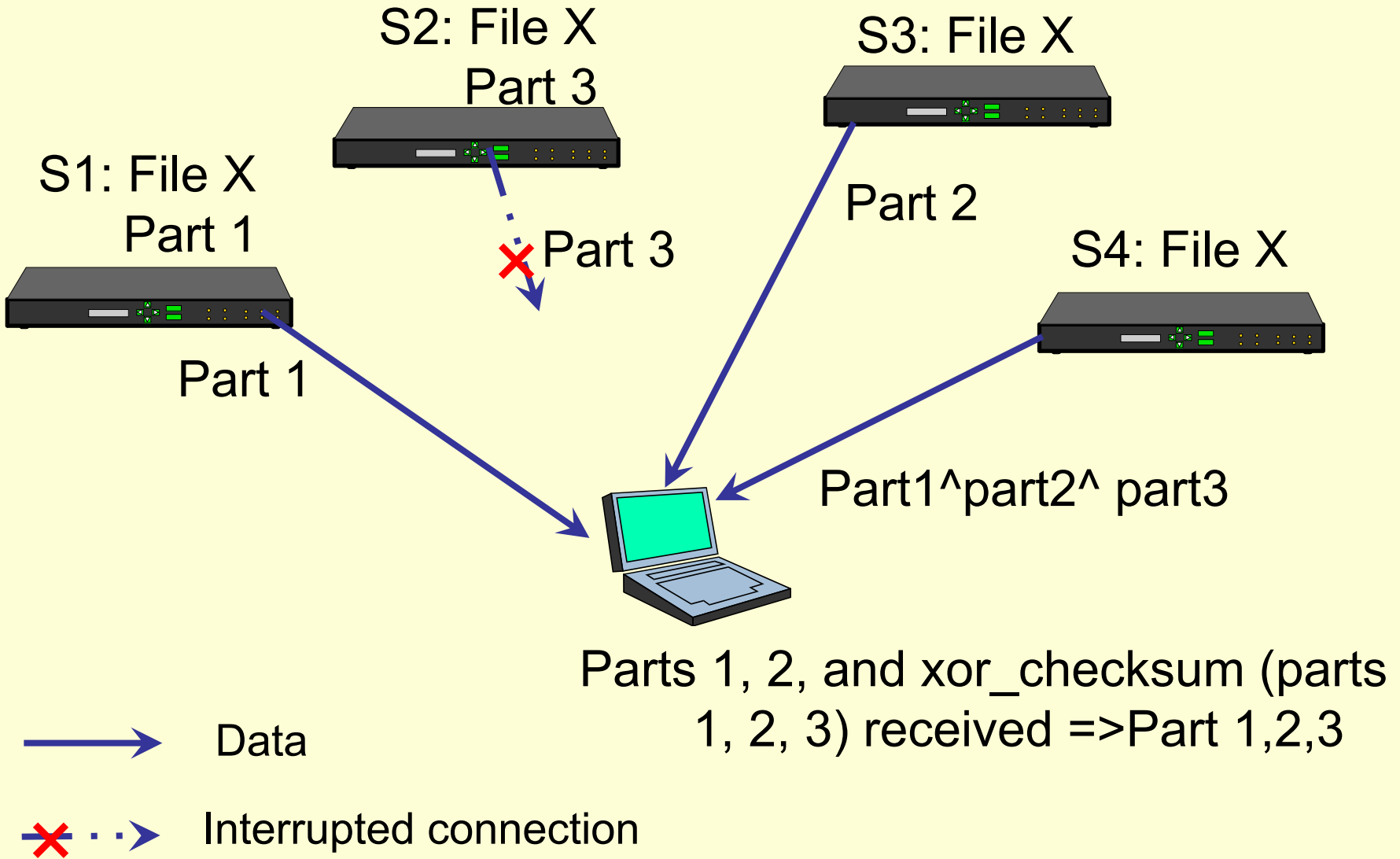


Protocol – In errors

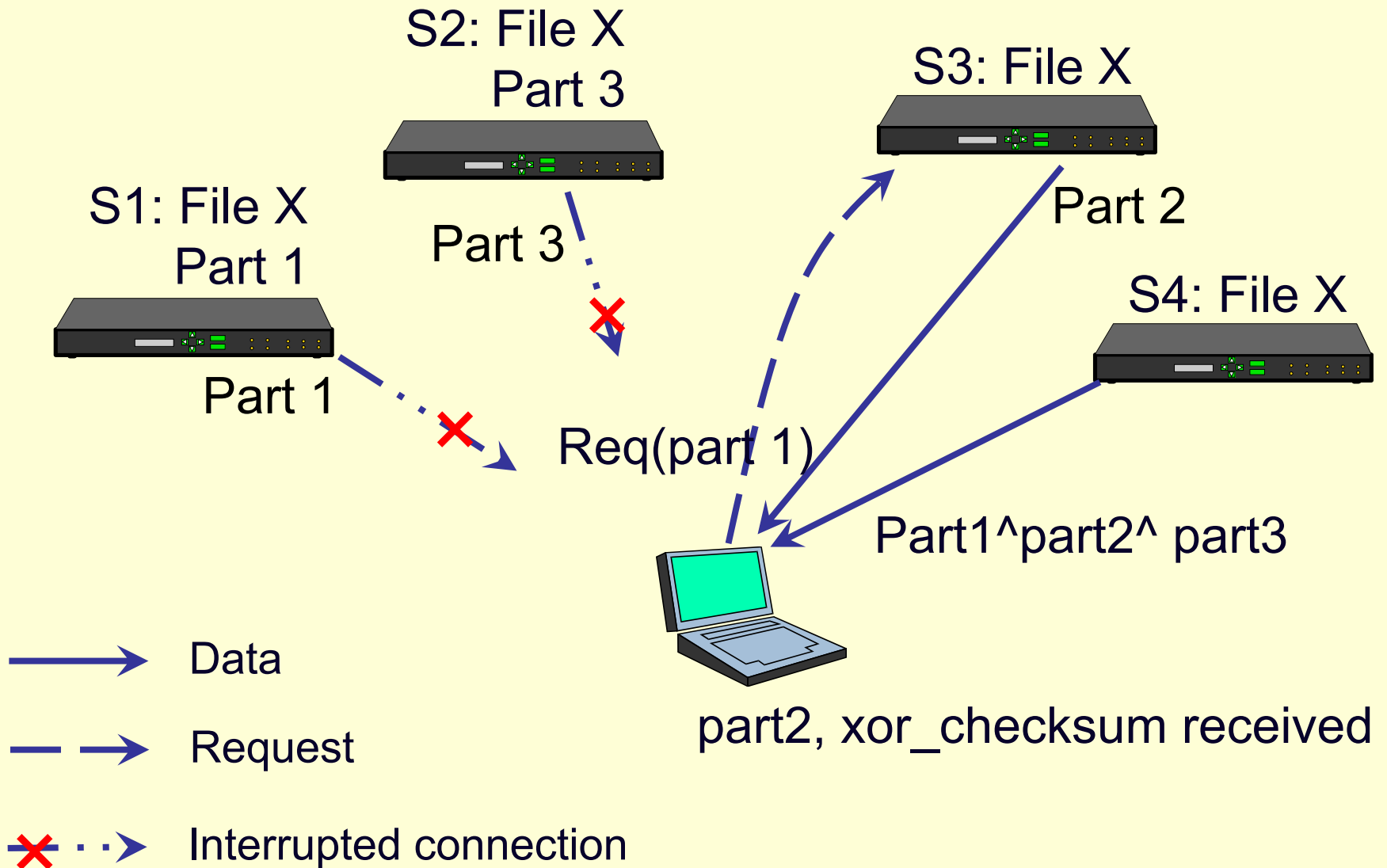
Client may

- **Restore** erroneous piece
 - **Actively request** missing data pieces
 - Re-organize redundant sets
- 

Protocol - Error occurrence



Protocol - Multiple errors

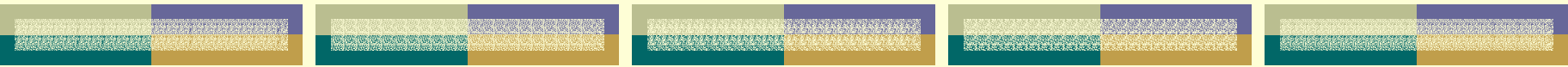




Conclusions

- Use redundancy in checksum group to obtain speedup and reliability.





Thank you!

