

无损以后，为什么仍会堵？

第三期 · 流控、乱序与互连协议取舍

第三期

01 缓冲资源与发送速率

02 到达顺序与完成语义

03 协议分层与通信算法

不丢包、不积压、不死锁、正确可见是不同保证

Credit：先有位置，再允许发送

逐跳资源承诺；不是远端操作完成通知



2 个空位 → 只允许相应数量的数据进入

credit 保护缓冲资源；完成与可见性另有规则

Incast：入口总量超过一个出口

逐跳流控与端到端拥塞控制的职责不同

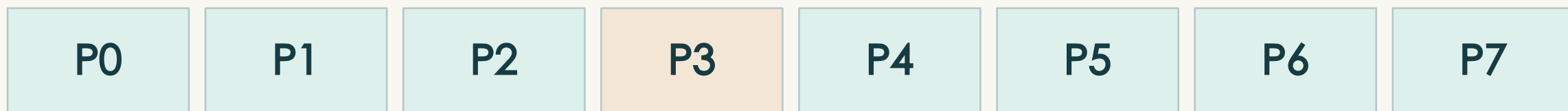


逐跳信用防溢出；端到端控制调节源端注入

调速、背压和死锁处理不能互相替代

PFC 暂停一个类别，不是一条 flow

最多 8 个 priority；同类可以承载大量流



P3 暂停

同属 P3 的多个 flow 一起受影响



同一 priority 的其他 flow 也可能被暂停

RDMA：队列、权限与内存分开

可靠传输可以建立在会丢包的底层之上

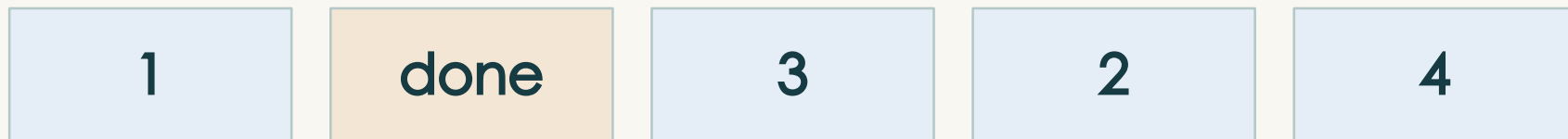
对象 / 机制	职责	不要误解为
QP / 工作请求	通信队列、状态与待执行操作	每个远端 buffer 固定对应一个 QP
MR / key	注册内存范围及访问权限	全网所有裸指针的通用列表
传输恢复	发现缺失、重传或报告失败	每一跳永远不能丢包
拥塞控制	调节注入、缓解持续积压	只要可靠就自动不会拥塞

QP 是通信对象； MR 是内存与权限对象

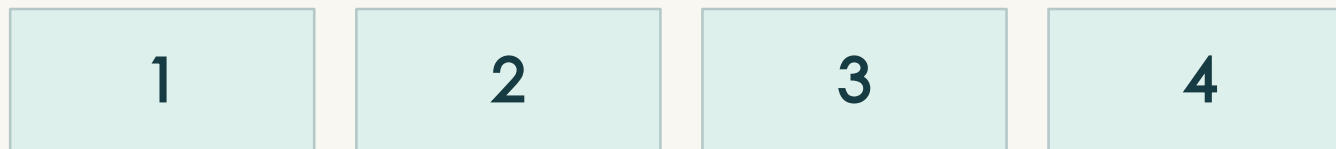
乱序到达，不应变成过早完成

数据 placement 与消息 delivery / completion 分开

到达顺序



按地址放置



确认全部到齐 + 可见性 / 顺序要求满足

再发布完成通知

多路径不能消除目的端容量瓶颈，也不能省略完成语义

重传：先问在哪一层，再问重传哪些

逐跳链路重放与端到端传输恢复可以同时存在

发送： 1 2 3 4 5

假设 2 丢失，其余已经到达（教学模型）

Go-Back-N

重发 2、3、4、5

选择性重传

只重发缺失的 2；需更多接收状态

同名的 ACK / NAK，可能属于完全不同的协议层

允许重排，不等于任意重排

原文 RO 喷洒 + barrier：保留为作者研究提案

层次	需要保证什么	不能直接等同
链路 / 事务顺序	符合 ordering domain 与事务规则	所有设备的一条全局顺序
确认 / completion	对应操作的协议完成条件	全部相关数据对所有消费者可见
软件同步	定义数据可消费的时机	发出一个读请求就全局 flush
RO / UIO 设计	端点、交换与恢复完整配合	打开一个开关就支持任意喷洒

设计提案应有正确性与前进性证明，再讨论性能收益

比较互连，要逐层比较

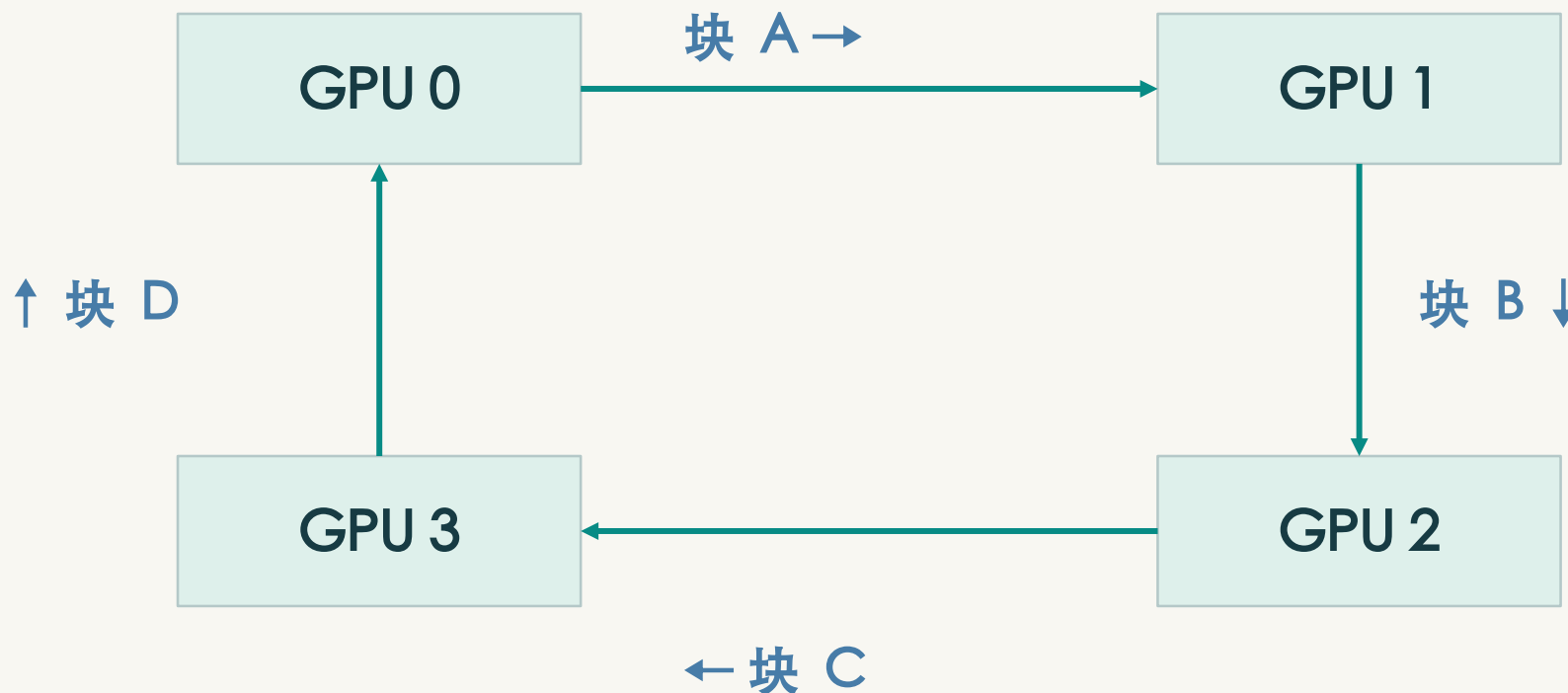
PHY 相同，不代表链路帧、事务或软件接口相同

层次	比较对象	已核验的边界
SerDes / PHY	速率、编码、介质、距离	Ethernet PHY $\neq$ Ethernet 帧
链路层	帧、credit、校验、重放	SUE-Lite 仍依赖无损链路
事务 / 传输层	访问、顺序、恢复、拥塞	UALink 200G 1.0：严格排序必选
软件 / 应用	API、内存、CCL、任务	同名 API 不代表同样路径或性能

型号、协议版本、必选 / 可选与商用实现须分别记录

Ring 可以并行；算法要匹配消息和拓扑

分块流水可同时使用多条连接；不是只看图像形状



同一时刻
多条连接推进
不同数据块

评价算法：消息大小 × 拓扑映射 × 通道 × 同步开销

读文章时，给结论标明证据身份

明确错误要纠正，未核信息保持未核

证据身份	本文例子	如何使用
标准 / 官方能力	UALink 1.0 排序；SUE-Lite 无损	限定版本、产品与实现范围
作者观察	5090 多卡结果；市场机型概括	保留描述，要求实验条件
作者提案	RO 喷洒与自定义 barrier	检验正确性，再评估成本
待核 / 预测	国内 GPU 协议阵营、未来路线	不按品牌猜填，不作采购事实

未找到证据 ≠ 已证明错误；标准支持 ≠ 产品已经实现

用一条证据链，持续追问系统

任务 → 语义 → 拓扑 → 机制 → 软件 → 实测

- 1 先确定任务：谁与谁通信，何时需要结果？

- 2 再确定语义：寻址、访问、顺序、完成与一致性

- 3 画出路径：链路、交换节点、共享切面与队列

- 4 核对实现：版本、端点、软件、隔离与故障处理

- 5 最后看收益：正确性、扩展效率、尾延迟与稳定性

每一个“更大、更快、兼容”，都应能回到条件和证据