

# 第一章 总结

## 1.1 工作总结

本文将软件定义网络高级编程模型分别从面向单交换机和面向全网络进行了研究。

对于单交换机情况，针对判断一个高级 SDN 程序是否可以被具有固定结构数据通路所实现的问题，我们提出一个将高级 SDN 程序和底层数据通路统一的特征空间，通过特征函数，高级 SDN 程序和数据通路都可以映射到该空间上的点，最后通过定义空间上点的比较函数，来判断该高级 SDN 程序是否可以在该数据通路上实现。除此之外，当考虑具有循环结构的高级 SDN 程序实现在可定制结构的数据通路上时，我们提出了重复软件流水线模型。该模型的优势在于可以对具有动态循环条件的循环结构进行转换。以及在转换后的流水线结构上计算最佳数据通路时显示出更高的效率。

对于全网络情况，我们首先对高级 SDN 程序在数据平面实现问题进行了研究。具体来说，我们将数据平面分为完全可编程数据平面和部分可编程数据平面。对于完全可编程数据平面，在网络资源受限情况下，我们需要将程序按语句打散再放置到交换机上执行。这样交换机可以不实现整个程序的逻辑。因此，我们的问题是根据系统目标算出每个语句应该放置到哪个交换机上执行才最优。我们首先算出程序对应的决策树，再根据决策树以及网络拓扑，算法每个程序语句的放置范围，最后通过解混合整数线性规划得到语句的最优放置。对于部分可编程数据平面，我们提出的编程模型支持在高级 SDN 程序中调用相应网络节点功能对数据包进行处理，并根据调用返回的结果，用户可以选择网络中不同路径对数据包进行转发。对于这样的高级 SDN 程序，由于调用相应网络节点处理数据包时，数据包需要转发到该节点，而这可能会破坏程序中对数据包规定的转发路径的约束。因此，我们进一步考虑了其程序正确性问题，并提出系统路径约束来解决该问题。最后，

我们分别考虑软件定义联合网络和车联网的场景，对数据平面进行优化。对软件定义联合网络的场景，我们设计高级编程系统，并基于共享本地状态的方法，提高系统性能。对车联网的场景，我们提出软件定义车联网的架构以及编程框架，并考虑车联网中车移动性的特点，提出优化的规则下发方法，减少生成的规则数量。

## 1.2 研究展望

在面向单交换机方面，在不影响性能下支持有状态操作或引入 CPU 操作都将大幅度提高数据通路的灵活性，使更多计算可以卸载到数据通路上，其“计算容量”也会大幅度提高。因此，我们会探索对于支持一定程度的复杂计算的数据通路，如何使高级 SDN 程序在保持简单的情况下能充分利用其能力。

由于 5G 的发展，网络切片将成为重要技术。目前网络切片技术主要考虑控制平面，因此，我们将探索如何在数据通路上支持网络切片，使其可以共享给多个应用的同时，应用不会受到性能的影响。

在面向全网络方面，网络内计算（in-network compute）受到了许多研究人员的关注。基于底层网络的强大计算能力，很多分布式算法在性能上都可以得到大幅度提高。因此，我们将探索网络内计算的计算模型，考虑什么样的算法可以在其上实现，而什么样的算法不可以实现。