**LIMITATIONS OF P4:**

* P4 as a general-purpose programming language is very limited.
* P4 is not a Turing-complete language; it is narrowly defined for performing data-path packet processing.
* Surprisingly, there are even many packet-processing tasks that cannot be expressed in P4.
* P416 supports extern functions or methods; these are computational functions that are implemented outside of P4 and can be invoked from P4 programs. There is currently an effort to standardize a set of such methods; however, each P4 target platform can provide additional extern methods, e.g., to model hardware accelerators. Invoking extern methods is one way that P4 programs can perform otherwise impossible tasks.
* There is no iteration construct in P4. Loops can only be created by the parser state machine. There is no support for recursive functions. In consequence, the work performed by a P4 program depends linearly only on the header sizes.
* There is no dynamic memory allocation in P4. Resource consumption can be statically estimated (at compile-time).
* There are no pointers or references.
* There is no support for multicast or broadcast. These must be achieved by means external to P4. The typical way a P4 program performs multicast is by setting a special intrinsic metadata field to a “broadcast group”. This triggers a mechanism that is outside of P4, which performs the required packet replication.
* P4 has no built-in support for queueing, scheduling or multiplexing.
* P4 is unsuitable for deep-packet inspection. In general, due to the absence of loops, P4 programs cannot do any interesting processing of the packet payload.
* P4 offers no support for processing packet trailers.
* All the state in a P4 program is created when a packet is received and destroyed when the processing is complete. To maintain state across different packets (e.g., per-flow counters) P4 programs must use extern methods. We expect the standard library to contain support for such persistent arrays (counters, registers, meters). Even given support for registers, one cannot iterate over all counters to compute statistics.
* There is no standard way to communicate between the data plane and the control plane; this is usually offered using extern methods (e.g., to implement “learning”).
* There is no support for performing packet fragmentation or reassembly; thus protocols such as TCP cannot be described in P4.
* There is no support for generating new packets (e.g., an ICMP reply), only for processing existing ones.
* **Despite these limitations, P4 is a remarkably useful language.**
* **We also expect that future evolution will expand its capabilities.**