AWK is a programming language developed in 1977 by **Alfred Aho, Peter Weinberger, and Brian Kernighan**. It is developed as a text-processing language - it has simple syntax to match lines of patterns, separate out the fields, and operate on them.

AWK is a language for processing files of text

```
BEGIN {
      recvdSize = 0
      startTime = 400
      stopTime = 0
 }

 {
            event = $1
            time = $3
            node_id = $41
            pkt_size = $37
            level = $19

 # Store start time
 if (level == "AGT" && event == "s" && pkt_size >= 512) {
  if (time < startTime) {
            startTime = time
            }
      }

 # Update total received packets' size and store packets arrival time
 if (level == "AGT" && event == "r" && pkt_size >= 512) {
      if (time > stopTime) {
            stopTime = time
            }
      # Rip off the header
      hdr_size = pkt_size % 512
      pkt_size -= hdr_size
      # Store received packet's size
      recvdSize += pkt_size
```

AWK

```
    }
  }

  END {
     printf("Average Throughput[kbps] = %.2f\t\t
StartTime=%.2f\tStopTime=%.2f\n",(recvdSize/(stopTime-
                                    startTime))*(8/1000),startTime,stopTime)
  }
```

Event	Abbreviation	Type	Value
Wireless Event	s: Send r: Receive d: Drop f: Forward	`%.9f %d (%6.2f %6.2f) %3s %4s %d %s %d [%x %x %x %x]`	
		`%.9f _%d_ %3s %4s %d %s %d [%x %x %x %x]`	
		double	Time
		int	Node ID
		double	X Coordinate (If Logging Position)
		double	Y Coordinate (If Logging Position)
		string	Trace Name
		string	Reason
		int	Event Identifier
		string	Packet Type
		int	Packet Size
		hexadecimal	Time To Send Data
		hexadecimal	Destination MAC Address
		hexadecimal	Source MAC Address
		hexadecimal	Type (ARP, IP)

Event	Abbreviation	Flag	Type	Value
Wireless Event	s: Send r: Receive d: Drop f: Forward	-t	double	Time (* For Global Setting)
		-Ni	int	Node ID
		-Nx	double	Node X Coordinate
		-Ny	double	Node Y Coordinate
		-Nz	double	Node Z Coordinate
		-Ne	double	Node Energy Level
		-Nl	string	Network trace Level (AGT, RTR, MAC, etc.)
		-Nw	string	Drop Reason
		-Hs	int	Hop source node ID
		-Hd	int	Hop destination Node ID, -1, -2
		-Ma	hexadecimal	Duration
		-Ms	hexadecimal	Source Ethernet Address
		-Md	hexadecimal	Destination Ethernet Address
		-Mt	hexadecimal	Ethernet Type
		-P	string	Packet Type (arp, dsr, imep, tora, etc.)
		-Pn	string	Packet Type (cbr, tcp)

AWK