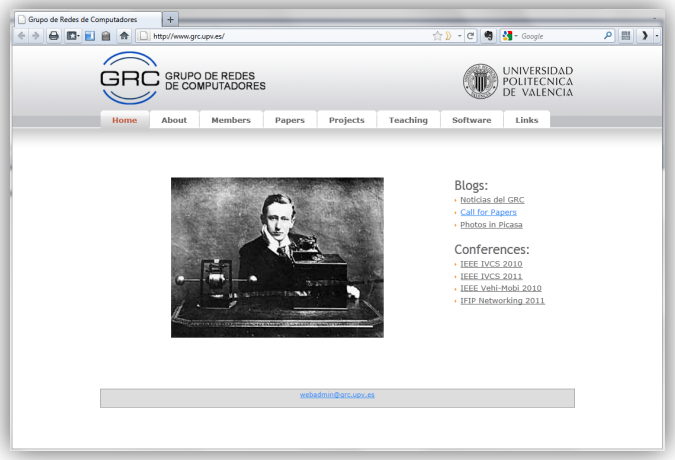
# *INET framework extensions for TCP Vegas and TCP Westwood*

María Fernández, **Carlos T. Calafate**,
Juan-Carlos Cano and Pietro Manzoni

http://www.grc.upv.es/




Computer Networks Group (GRC)
Department of Computing Engineering
Technical University of Valencia (UPV)
SPAIN

GRC GRUPO DE REDES DE COMPUTADORES

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

# Outline

- Evolution of TCP
- TCP in INET 2.0 / OMNeT++
- Implementation
- Simulation results
- Conclusions & Future Work




# Evolution of TCP

**Initial proposal**
RFC 675 (1974)
RFC 793 (1981)

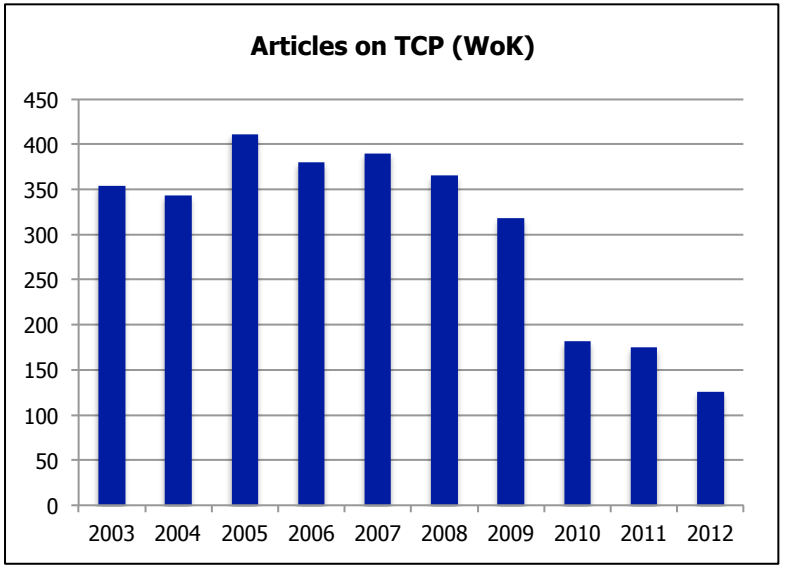
**General performance improvements**

- Van Jacobson (1988-90)
- **TCP Vegas** (1994)
- TCP SACK (1996)
- TCP FACK (1996)
- Congestion control (1997)
- ...
- Multipath (2011)

**Wireless-specific solutions**

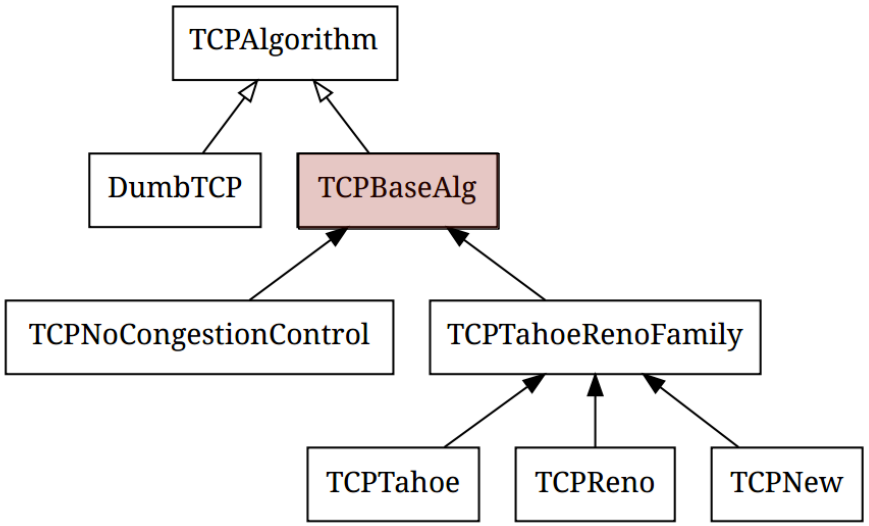
- I-TCP (1995)
- Snoop TCP (1995)
- Mobile-TCP (1997)
- WTCP (1999)
- TCP Santa Cruz (2000)
- **TCP Westwood** (2002)
- TCP Veno (2003)
- ...

**And many more...**

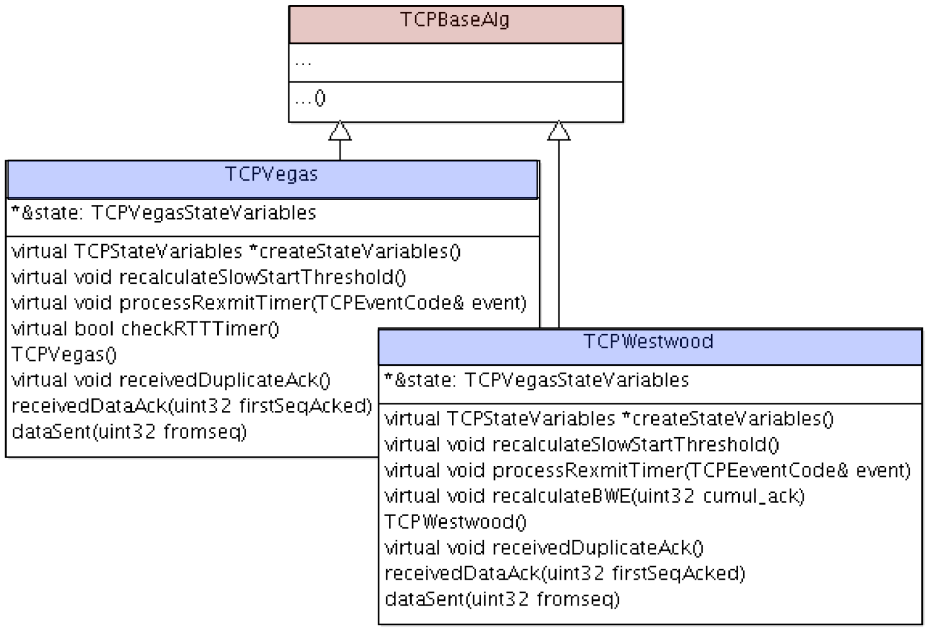


# TCP in INET2.0 / OMNeT++

EXISTING…




NEW!!!



# Implementation

## **TCP Vegas**: cwnd management

```
// Once per RTT
calculate_newRTT
// Decide if incr/decr cwnd
if (newRTT > 0) {
    calculate_expectedThroughput;
    calculate_actualThroughput;
    diff = expected - actual;

    // Slow start (cwnd modification only every 2 rtt)
    if (state->snd_cwnd < state->ssthresh) {
        if (diff > v_gamma) {
            state->snd_cwnd -= (state->snd_cwnd / 8); // decr
            state->v_incr = 0;

        } else
                        state->v_incr = state->snd_mss; // incr
    } // end slow start

    // Cong. avoidance
    else {
        if (diff > v_beta)
            state->v_incr = -state->snd_mss; // decr
        else if (diff < v_alpha)
            state->v_incr = state->snd_mss; // incr
        else
            state->v_incr = 0; // same
    } // end cong. avoidance
} // end 'Once per RTT'
```

## **TCP Vegas**: RTT & timeout

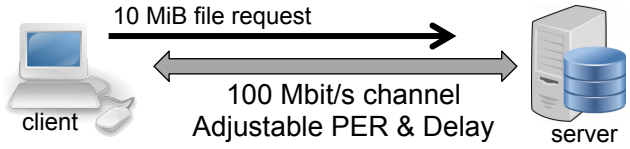
```
void TCPVegas::receivedDataAck(uint32 firstSeqAcked) {

      TCPBaseAlg::receivedDataAck(firstSeqAcked);

      simtime_t tSent = state->v_sendtime[(firstSeqAcked - (state->iss+1))
% state->v_maxwnd];
      simtime_t currentTime = simTime();

if (tSent != 0 && num_transmits == 1) {
      simtime_t newRTT = currentTime - tSent;
      state->v_sumRTT += newRTT;
      ++state->v_cntRTT;

      if (newRTT > 0) {
            if(newRTT < state->v_baseRTT)
                  state->v_baseRTT = newRTT;

            simtime_t n = newRTT - state->v_sa/8;
            state->v_sa += n;
            n = n < 0 ? -n : n;
            n -= state->v_sd / 4;
            state->v_sd += n;
            state->v_rtt_timeout = ((state->v_sa / 4) + state->v_sd) / 2;
            state->v_rtt_timeout += (state->v_rtt_timeout / 16);
      }
}
```

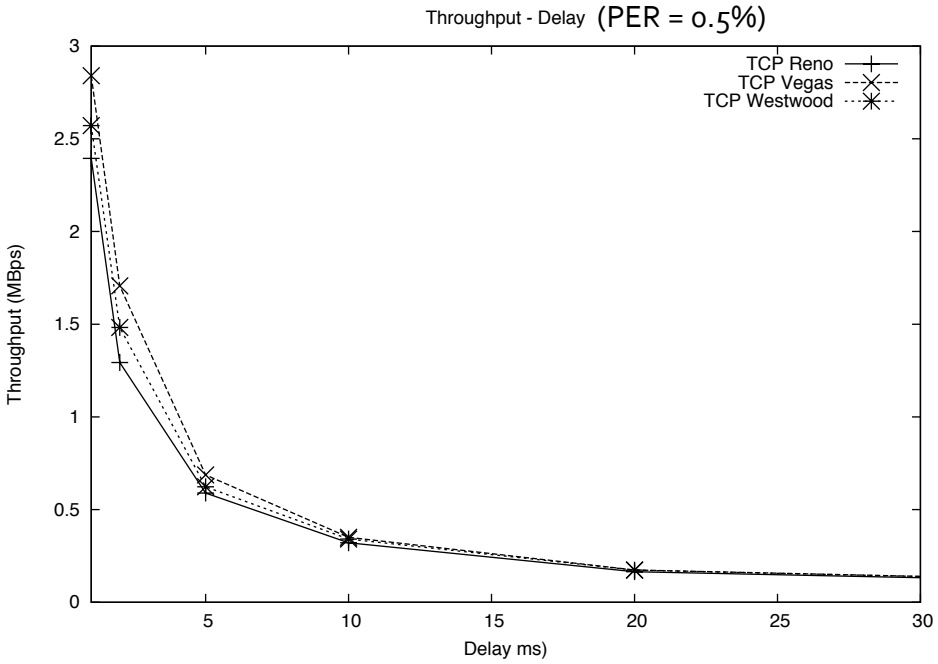
## **TCP Westwood**: BW estimation

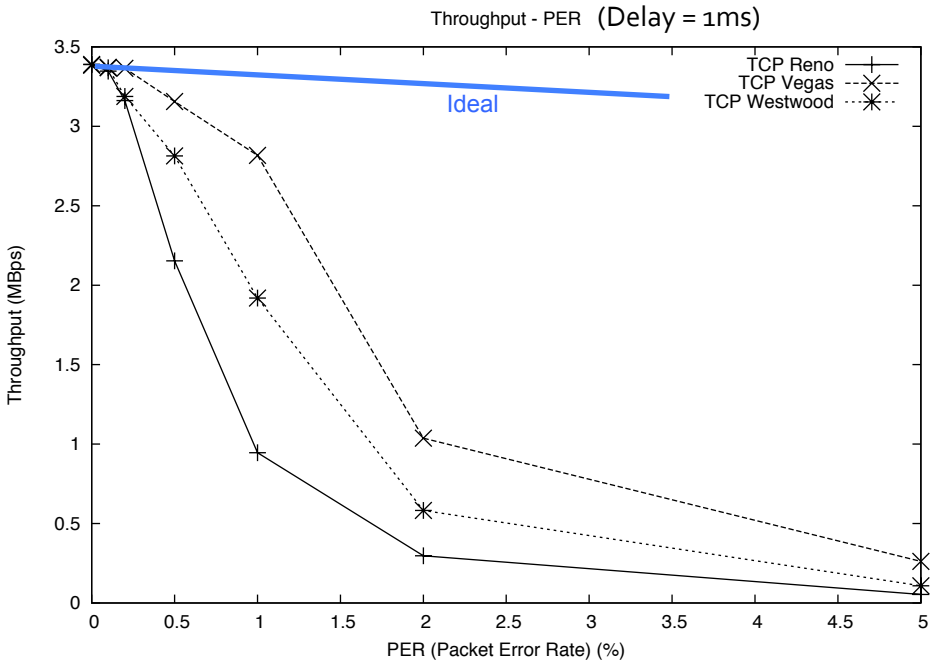
```
void TCPWestwood::recalculateBWE(uint32 cumul_ack) {
    simtime_t currentTime = simTime();
    simtime_t timeAck = currentTime - state->w_lastAckTime;

    // Update BWE
    if(timeAck > 0) {
        double old_sample_bwe = state->w_sample_bwe;
        double old_bwe = state->w_bwe;
        state->w_sample_bwe = (cumul_ack) / timeAck;
        state->w_bwe = 0.9047*old_bwe +
                       0.0476*(state->w_sample_bwe + old_sample_bwe);
    }
    state->w_lastAckTime = currentTime;
}
```

# Simulation results

- Evaluated scenario:



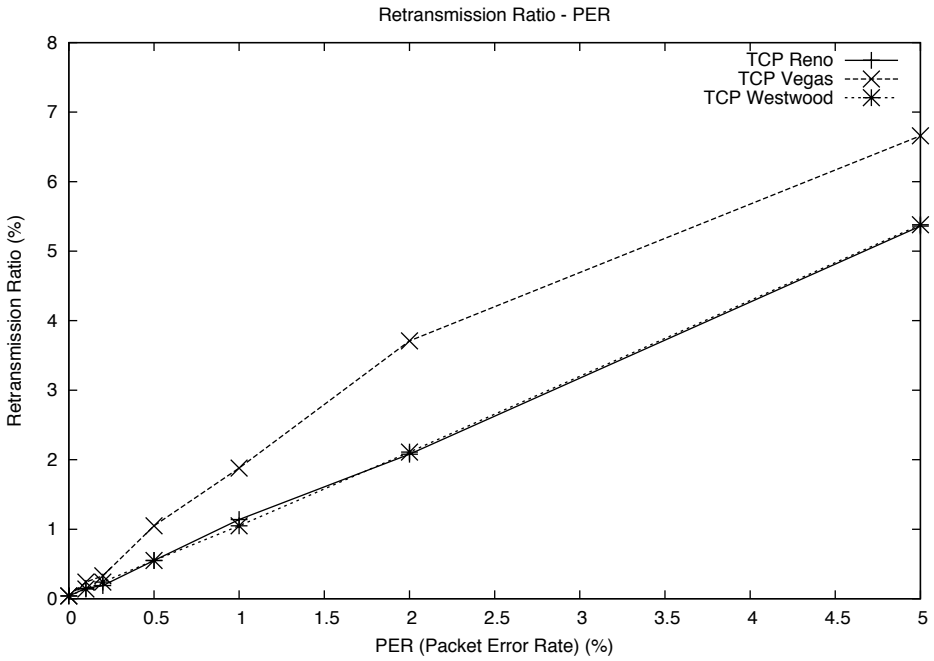



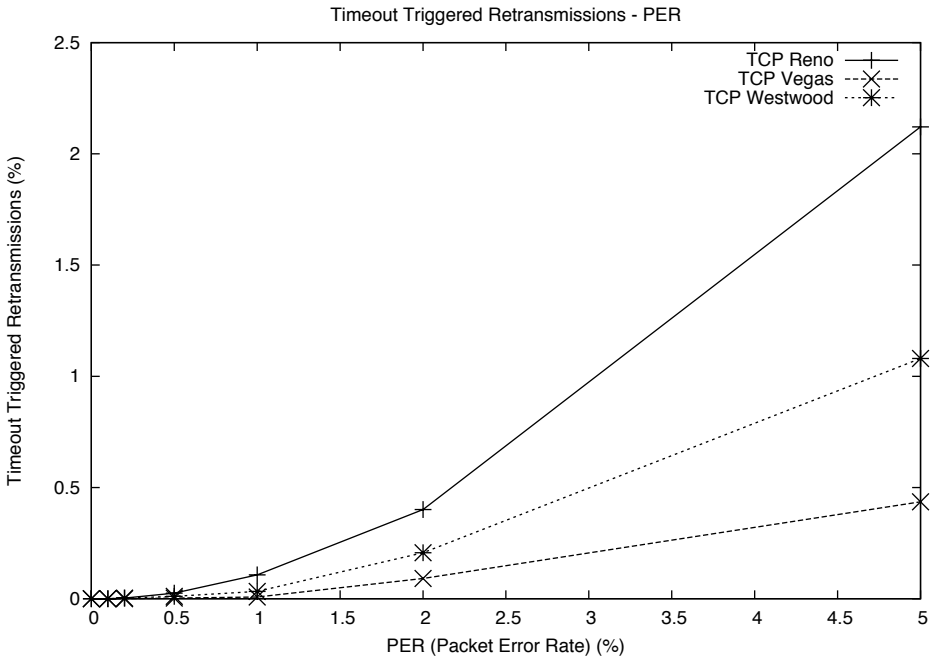




- Original authors' data:
  - Vegas improves Reno by about 46%
  - Westwood improves Reno by about 30%

# Simulation results

- Retransmission behavior




- TCP Vegas reduces timeouts by retransmiting more data
- TCP Westwood: Reno efficiency with less less timeouts

# Conclusions & Future Work

## CONCLUSIONS

- The world is wireless!

BUT…

- The standard TCP protocol often offers a reduced performance in wireless environments
  - Differences between wired and wireless networks (loss prone)
- Although many alternatives to TCP have been proposed, INET 2.0/OMNeT++ only includes standard TCP
- FIRST STEP: We implemented TCP Vegas and TCP Westwood for INET 2.0
- Experimental results show that:
  - The performance levels achieved agree with previously published results
  - Significant benefits are achieved for channels characterized by high losses or high delays (or both)
  - There is still a great margin for improvement → several research works on the topic

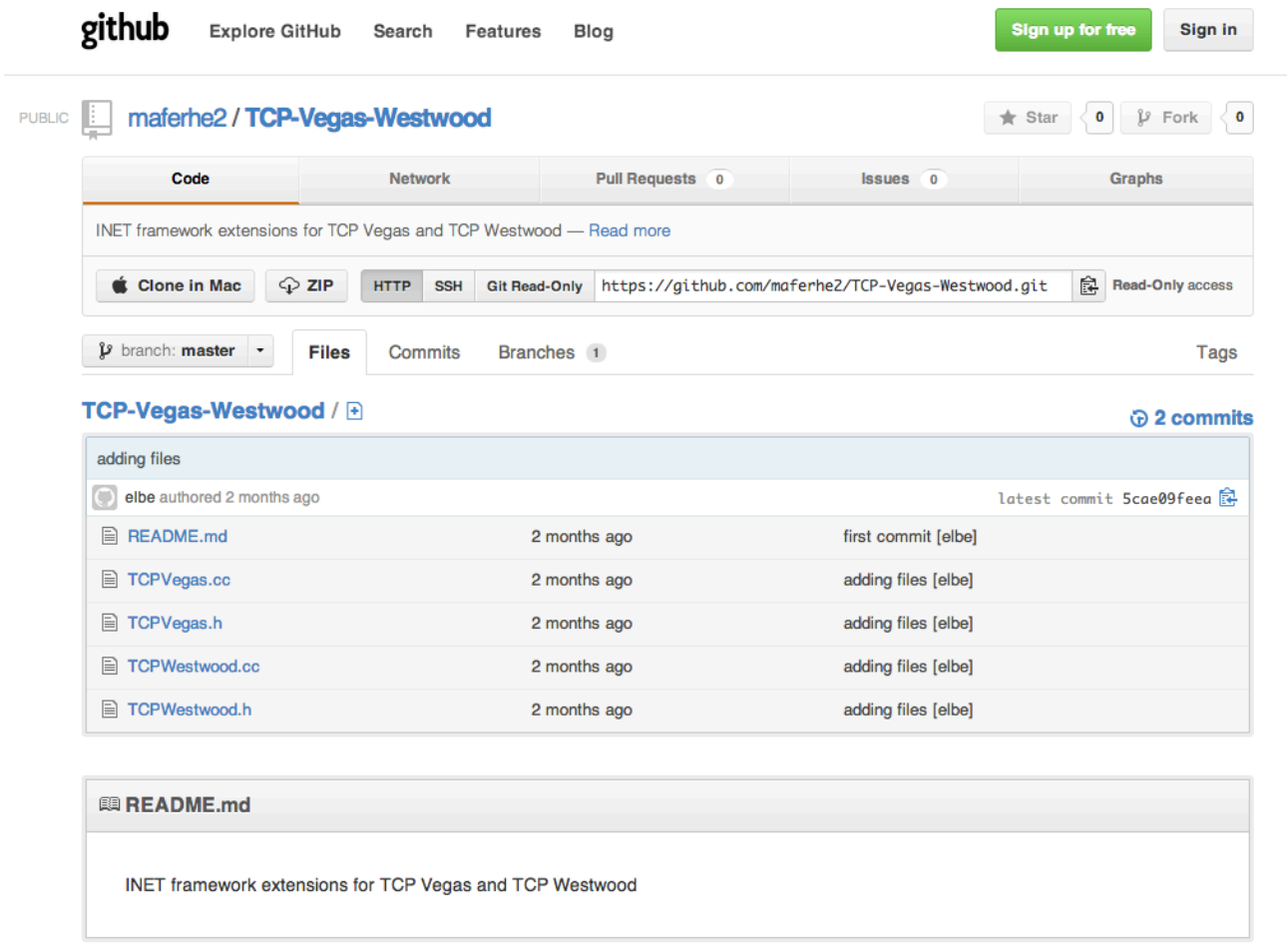
## FUTURE WORK

- Develop new TCP variants for INET
- Develop new protocols, comparing against the most effective solutions available



# Available for download

https://github.com/maferhe2/TCP-Vegas-Westwood



# Thanks!

- Questions?