A Rent-Seeking Framework for Multipath TCP

Shiva Raj Pokhrel

Carey Williamson

Deakin University (Australia)

University of Calgary (Canada)

04 Nov 2020

Outline

1. Scope and Novelty
    - Rent Seeking Framework
    - Motivations
    - Multipath TCP [RFC6824]
    - RSF-MPTCP vs Others
2. Rent-Seeking Framework
    - Rent-Seeking Framework
3. Extension to Multiple MPTCP
4. Performance Evaluation
    - Network Emulation Results

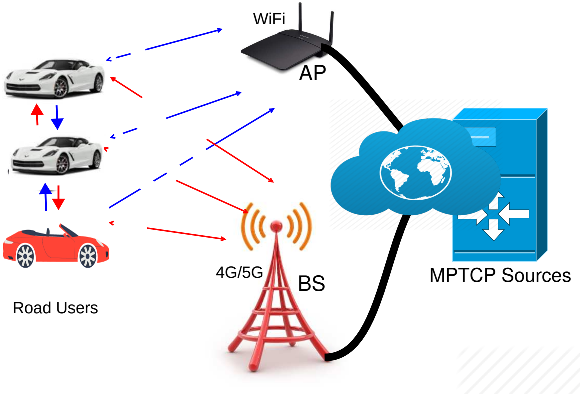


Figure 1: Internet of Vehicles: Vehicles are connected to a Server in the Internet using 5G and WiFi paths.

Scope and Novelty

- *Setup*. How TCP works?
  Network utility maximization (NUM) for Multiple paths
- *Existing Works*. MPTCP LIA (2011), MPTCP OLIA (2013), MPTCP BALIA (2014)
  - No well-defined utility function for MPTCP (*Peng et al.*, 2014)
  - Conditions under which we can use Kelly's NUM mechanism
- *Contributions*: Utility based on Tullock's *rent-seeking*
  - Common delay and/or loss constraints at the subflow level
  - Utility function has diagonal strict concavity
  - Solution provides:
    - ★ higher throughput
    - ★ better robustness
    - ★ improved responsiveness

Why Rent Seeking Framework?

- Definition of “rent” – economic wealth obtained through shrewd or potentially manipulative use of resources
- Rent seeking occurs when an entity seeks to gain wealth without any contribution of productivity
    - How best to aggregate WiFi and non-WiFi 5G seamlessly?
    - Kelly designed distributed congestion control: *proportional fairness and max-min fairness*
    - Our idea: transform NUM problem into a game between TCP flows
    - MPTCP subflows coupling can considered as a game

Our idea

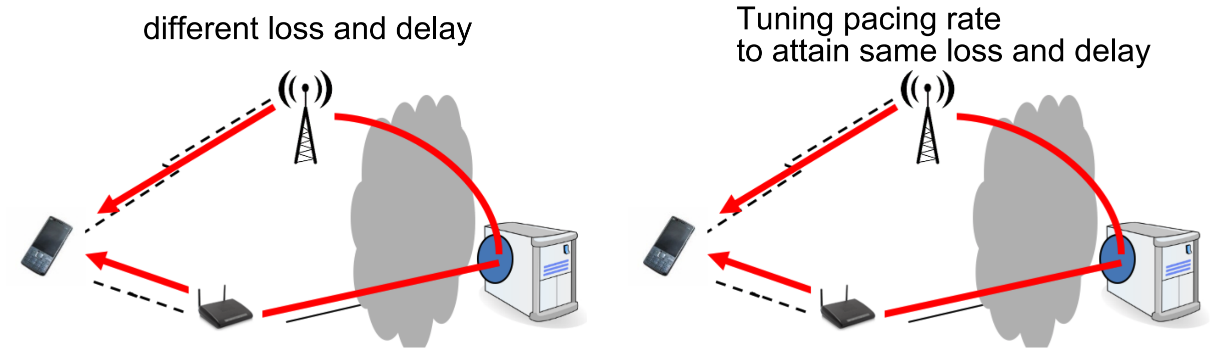


Figure 2: What is current MPTCP [RFC6824] What will Future MPTCP be?

- Ongoing development of 5G and Gigabit WiFi solutions
- MPTCP could exploit multiple available network interfaces
  - Autonomous Vehicles, AR/VR and Video streaming
  - Throughput-intensive, low end-to-end delay and high reliability
  - Our idea: transform NUM problem into a game between TCP flows
  - MPTCP subflows – a game with common coupled constraints.

Multipath TCP (MPTCP)

MPTCP LIA source has a set of subflows/paths: Each path maintains a separate congestion window (W_i) and updates its RTT (τ_i)

W_i in CA of MPTCP LIA (*Raiciu et al, 2011*) evolves

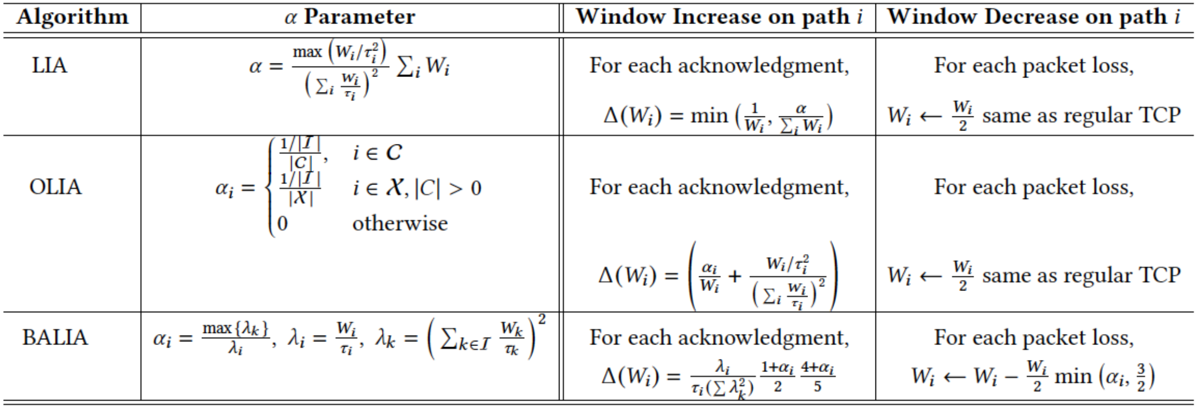
- For each loss indication on path i, update W_i as $W_i \leftarrow W_i/2$, and
- For each Ack received on route i, update W_i as $W_i \leftarrow W_i + \alpha(W_i)$, where

$$\alpha(W_i) = \min\left\{\delta, \frac{1}{W_i}\right\} \text{ and } \delta = \frac{\max_j(W_j/\tau_j^2)}{(\sum_j W_j/\tau_j)^2}. \tag{1}$$

- If $\alpha(W_i) = 1/W_i$, then window evolves as TCP
- If all subflows have equal RTTs, then $\delta = \max_j(W_j)/\sum_j W_j$

LIA, OLIA, BALIA and RSF-MPTCP

MPTCP Congestion Control for LIA, OLIA, and BALIA



Algorithm	**α Parameter**	**Window Increase on path i**	**Window Decrease on path i**
LIA	$\alpha = \frac{\max\left(W_i/\tau_i^2\right)}{\left(\sum_i \frac{W_i}{\tau_i}\right)^2} \sum_i W_i$	For each acknowledgment, $\Delta(W_i) = \min\left(\frac{1}{W_i}, \frac{\alpha}{\sum_i W_i}\right)$	For each packet loss, $W_i \leftarrow \frac{W_i}{2}$ same as regular TCP
OLIA	$\alpha_i = \begin{cases} \frac{1/\lvert\mathcal{I}\rvert}{\lvert\mathcal{C}\rvert}, & i \in \mathcal{C} \\ \frac{1/\lvert\mathcal{I}\rvert}{\lvert\mathcal{X}\rvert} & i \in \mathcal{X}, \lvert\mathcal{C}\rvert > 0 \\ 0 & \text{otherwise} \end{cases}$	For each acknowledgment, $\Delta(W_i) = \left(\frac{\alpha_i}{W_i} + \frac{W_i/\tau_i^2}{\left(\sum_i \frac{W_i}{\tau_i}\right)^2}\right)$	For each packet loss, $W_i \leftarrow \frac{W_i}{2}$ same as regular TCP
BALIA	$\alpha_i = \frac{\max\{\lambda_k\}}{\lambda_i},\ \lambda_i = \frac{W_i}{\tau_i},\ \lambda_k = \left(\sum_{k \in \mathcal{I}} \frac{W_k}{\tau_k}\right)^2$	For each acknowledgment, $\Delta(W_i) = \frac{\lambda_i}{\tau_i(\sum \lambda_k^2)} \frac{1+\alpha_i}{2} \frac{4+\alpha_i}{5}$	For each packet loss, $W_i \leftarrow W_i - \frac{W_i}{2} \min\left(\alpha_i, \frac{3}{2}\right)$

The window adaptation in CA phase of RSF-MPTCP

- For each packet loss on path i, update $W_i \leftarrow W_i/2$;
- For each new acknowledgment received on path i, increase W_i using $W_i \leftarrow W_i + \Delta(W_i)$, where:

$$\Delta(W_i) = \frac{1}{W_i} \min\left\{1, \frac{d - q_i}{q_i}\right\}. \qquad (2)$$

Adjust the window dynamically: q_i converges toward the target delay d

RSF-MPTCP

A network setting with K paths and one subflow on each path

- $\lambda_i = W_i/\tau_i$ ($\lambda_i > 0$): sending rate for the i^{th} subflow ($1 \leq i \leq K$)
- p_i represents the corresponding price for that subflow

Individual Throughput

$$\theta_i = V\left(\lambda_i \Big/ \sum_{j=1}^{K} \lambda_j\right),$$

$V(.)$ is a concave function, θ_i comes at the cost of $\hat{p}\lambda_i$ (where $\hat{p}$ is a constant)

Extension to Multiple MPTCP

N MPTCP sources sharing K paths:

$$\sum_{n=1}^{N}\sum_{i=1}^{K}\lambda_i^n \leq B. \tag{3}$$

Throughput obtained by the n^{th} MPTCP connection is the sum of the throughputs for all of its subflows, and is given by:

$$\theta^n = \sum_{i=1}^{K}\theta_i^n(\lambda^n) \tag{4}$$

$$\theta_i^n(\lambda^n) = V\left(\frac{\lambda_i^n}{\sum_{m=1}^{N}\sum_{j=1}^{K}\lambda_j^m}\right) \tag{5}$$

Utility of the n^{th} MPTCP connection is:

$$U^n(\lambda) = \sum_{i=1}^{K}\left(\frac{\lambda_i^n}{\sum_{m=1}^{N}\sum_{j=1}^{K}\lambda_j^m} - \hat{p}\lambda_i^n\right). \tag{6}$$

Performance Evaluation

- Experimental Testbed:

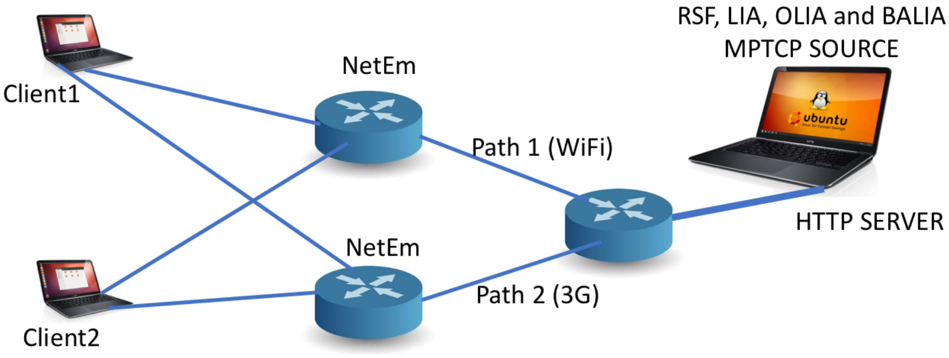
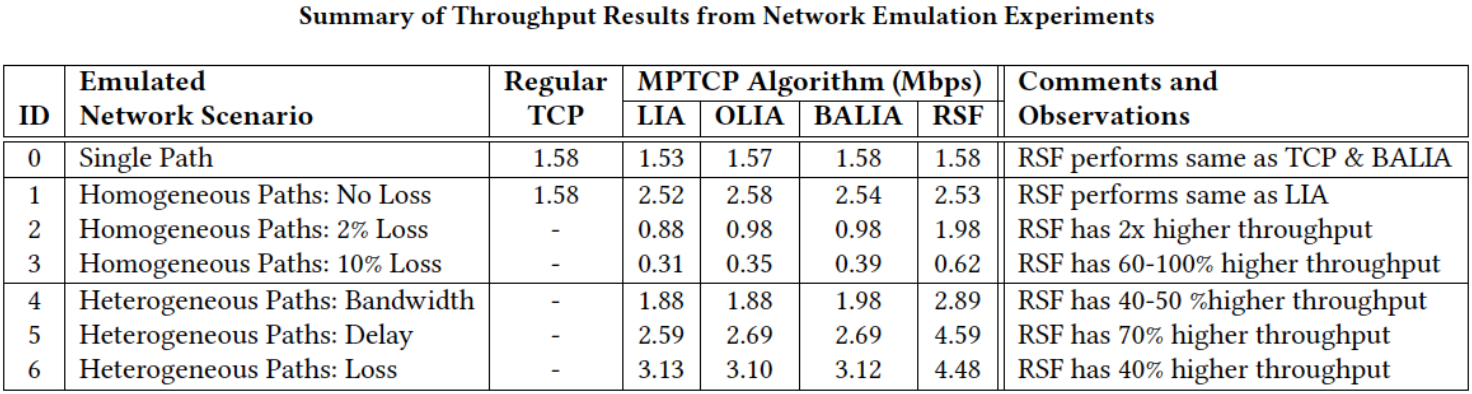


Figure 3: Experimental setup: we use Ethernet links and emulate the characteristics of WiFi and 3G paths.

- *MPTCP* algorithm based on the Rent-Seeking Framework (RSF) vs LIA, OLIA and BALIA
- Linux implementation C. Paasch *et al.*, “Multipath TCP in the Linux Kernel”, available from https://www.multipath-tcp.org (released: 02-Nov-2017).

Network Emulation Results



Summary of Throughput Results from Network Emulation Experiments

ID	Emulated Network Scenario	Regular TCP	MPTCP Algorithm (Mbps)				Comments and Observations
			LIA	OLIA	BALIA	RSF	
0	Single Path	1.58	1.53	1.57	1.58	1.58	RSF performs same as TCP & BALIA
1	Homogeneous Paths: No Loss	1.58	2.52	2.58	2.54	2.53	RSF performs same as LIA
2	Homogeneous Paths: 2% Loss	-	0.88	0.98	0.98	1.98	RSF has 2x higher throughput
3	Homogeneous Paths: 10% Loss	-	0.31	0.35	0.39	0.62	RSF has 60-100% higher throughput
4	Heterogeneous Paths: Bandwidth	-	1.88	1.88	1.98	2.89	RSF has 40-50 %higher throughput
5	Heterogeneous Paths: Delay	-	2.59	2.69	2.69	4.59	RSF has 70% higher throughput
6	Heterogeneous Paths: Loss	-	3.13	3.10	3.12	4.48	RSF has 40% higher throughput

- Baseline Results
  1. Scenario 0 used a single path network, for which all MPTCP versions are same. The first row of results confirms this observation
  2. Scenario 1 used two homogeneous paths with no packet loss at all. RSF MPTCP should perform the same as LIA (exact same results for both RSF-DB and RSF-LB MPTCP).

Homogeneous Paths

RSF MPTCP starts to differentiate itself from the other MPTCP algorithms

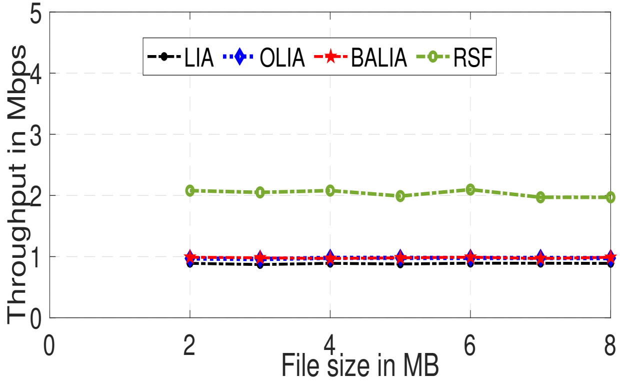


Figure 4: Homogeneous Paths with 10 Mbps bandwidth, 150 ms delay, and 2% random packet loss.

Observation 2/3. The proposed *RSF MPTCP* outperforms all other MPTCP algorithms in terms of throughput, and achieves similar fairness.

Heterogeneous Bandwidth Paths

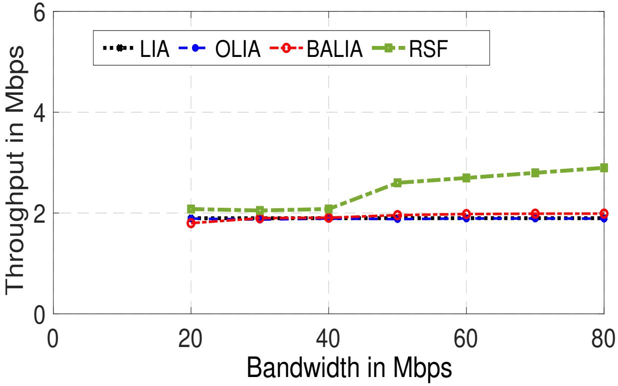


Figure 5: Throughput results for 10 MB file transfer in Scenario 4: Heterogeneous Bandwidth Paths.

- Observation 4. The proposed *RSF MPTCP* outperforms the other MPTCP algorithms when paths have heterogeneous bandwidth.

Heterogeneous Delay Paths

Scenario 5 considers paths with heterogeneous delay characteristics.

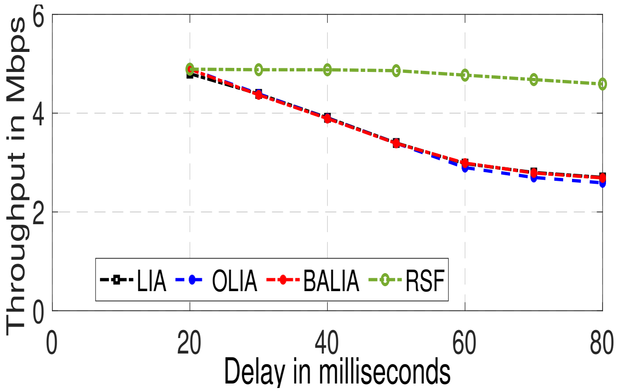


Figure 6: Throughput results for 10 MB file transfer in Scenario 5: heterogeneous delay paths. One path has 20 ms delay, while the delay of the other one is varied. Both paths have 20 Mbps bandwidth, and 3% loss.

- Observation 5. The throughput of all MPTCP algorithms (including RSF) decreases with increasing delay, the decline is quite gradual for RSF MPTCP, and quite pronounced for others.

Conclusion

- Developed a novel framework for designing MPTCP algorithms by using a *rent-seeking* game-theoretic framework
- The existence of a utility function with strict diagonal concavity
- Network emulation experiments: Higher throughput of RSF-MPTCP without compromising fairness
- New RSF-MPTCP is highly robust to the characteristics of heterogeneous network paths

Thank You

- Questions