Game Theory As a Tool for Modeling Cross-Layer Interactions

Christian Facchini, Fabrizio Granelli, Senior Member, IEEE

Abstract—Modeling computer networks is a complex task, as their behavior depends from several variables. Focusing on a single communication device, ISO/OSI and TCP/IP layered protocol stacks provide interoperability and fast deployment of networking solutions, but they limit the control on the interaction among protocols operating at different layers. As a consequence, the need is emerging to develop appropriate models to capture and evaluate the interaction of protocols within a single communication device in order to underline such forms of “indirect” interaction – since they may lead to unforeseen performance degradations. The proposed work aims at using the game theory for capturing the interactions within the protocol stack of a single node, with the goal of allowing to determine the “steady state” or the operating point of the system in a given scenario. As a result, a scalable and modular framework is presented, that enables characterization and analysis of cross-layer interactions starting from the protocols’ specifications.

Index Terms—Computer Network Performance, Performance Modeling, Protocol Stack Modeling

I. INTRODUCTION

Modeling communication networks is a key issue for network design, dimensioning, and performance evaluation. However, it still represents an open issue due to the inherent complex interactions between the protocols operating at different layers of the protocol stack, since in today’s communication systems all protocols at all layers concur to defining the overall performance – while being characterized by different goals. This gives rise to cross-dependency and introduces complex (and unforeseen) interactions among protocols at different (even not adjacent) layers. Indeed, ISO/OSI and TCP/IP layered protocol stacks were developed with the goal of enabling interoperability and foster development of networks and network devices, with the main advantage of the modularity in protocol design. A protocol within a given layer is described in terms of functionalities it offers, while implementation details and internal parameters are hidden to the remainder layers (the so-called “information-hiding” property).

Given the complexity of such (cross-layer) interactions, widely used models for evaluation of communication performance focus on a single layer only [1, 2]. In addition, while under general assumptions it is possible to demonstrate layering as an optimal design strategy, nevertheless recent developments on wireless and nomadicity envisage scenarios where explicit evaluation of the inter-layer (or cross-layer) interaction among different levels of the protocol stack is required. Sample literature works aiming at underlining such aspect are presented in [3], where the impact of different layers is studied in order to optimize service delivery in mobile ad-hoc networks, and [4], where the authors introduce a metamodeling approach to study cross-layer scheduling in wireless local area networks.

More in general, as it is correctly pointed out in [5], there is a need for a “holistic” approach to understanding and exploiting cross-layer interactions. The subject of systematic study of cross-layering from a “system theoretic” point of view and the closest available to a holistic view, is found in the work of Low and Doyle, for example [6]. The authors follow a top-down approach to set “holistic” objectives as opposed to the wide-spread “bottom up” and “ad hoc” identification and usage of cross-layer formulations on a case by case empirical basis.

In summary, a clear need is emerging for defining a framework able to capture inter-layer interactions – in order to augment the possibility of studying communication performance. As inter-layer interactions are triggered by service requests / responses, the authors believe that more emphasis should be given to interaction among protocols operating at different layers rather than single-layer modeling. In this scenario, an appropriate framework to be considered is represented by the game theory which fits the requirements of addressing interactions between multiple entities (players) with different goals and strategies.

The paper proposes a novel model to characterize inter-layer interactions by using the game theory, and it is organized as follows: Section II illustrates the main principles of the game theory, while Section III describes the general framework. A case study is illustrated in Section IV, with results and comments in Section V. Finally, Section VI concludes the paper with final remarks and outlines of future activities on the topic.

II. A SHORT SUMMARY OF GAME THEORY

Game theory is nowadays widely used in several fields, virtually anywhere there is the need to analyze the interactions
within a population made up of decision-makers (players). A player may represent either an individual or a group of individuals. In general, every player in a game is characterized by a given set of actions, i.e. the possible decisions it may take, and has a given target or goal, i.e. to maximize its own utility function (which depends not only on the action employed by the player but also on the actions of every other player), which maps every possible outcome of the game to a real value, representing the “payoff”. If we denote with \( P \) the player set, with \( A \) the action space (i.e., the cartesian product among all the action sets), and with \( U \) the utility set, we can define a triple \( <P,A,U> \), which univocally identifies a game.

Non-cooperative games can be classified as static or dynamic. Static games are effectively rendered by means of a “payoff matrix”, whereas dynamic games are preferably depicted through a decision tree. The ensemble of strategies adopted by the players at a given time \( t \) is called “strategy profile” and, from the theoretic point of view, is a \( n \)-ple of strategies (provided \( n \) be the player set cardinality), which is generally denoted as \( \sigma \).

Dynamic games can be further subdivided depending on whether players know the whole game history or not. A game where players fulfill this condition is said to be of perfect information (e.g., chess); if not, it is said to be of imperfect information.

On top of that, games can be played just once, in which case they are one-stage games, or repeated-games, that can be played again once they finish. A broader category is that of stochastic games, where there are multiple stages (or states) and the game explores them following a specific transition function.

Another classification is based on the degree of information available by the players: a game is said to be of complete information in the case players know all the structural details of the game, or of incomplete information otherwise.

Given the different games introduced above, the mathematical framework enable the search for the so-called “Nash equilibria”: strategy profiles where each player cannot get a higher payoff by unilaterally deviating the game. Such configurations constitute the points of stability of the analyzed system, and potentially represent a powerful tool in the framework of network modeling, too. Multiple equilibria may exist in a game, though there are games characterized by no equilibria at all.

Moreover, it is possible to define some “equilibrium refinements”, such as “Bayesian equilibria”, for static games of imperfect information, and “perfect Bayesian equilibria”, for dynamic games of imperfect information. “Markov perfect equilibria” should be mentioned, as well, i.e. the equilibria which can be found in stochastic games, when Markovian strategies are employed.

The interested reader is suggested to refer to [7] and [13] for more information on the subject.

### III. The Proposed Framework

The proposed modeling framework is based on the game theory as a tool to analyze the behavior of protocols in communication devices. It should be underlined that the purpose of the paper is not to define protocol modifications to foster performance improvement, but to represent existing protocols using the notation of the game theory in order to underline their indirect interaction. Therefore, the resulting game will be a non-cooperative game, as no direct interaction among protocols is allowed by the layering principle.

As outlined in the previous paragraphs, it is possible to define a correspondence between a protocol entity at a given layer and a player. This conceptual similarity (further explored in Table I) is based on the consideration that an entity implementing a given protocol has its own goals (implicitly or explicitly described in the protocol specifications) and performs actions which provide a final result on the basis of the actions of the other players / protocols entities.

<table>
<thead>
<tr>
<th>Protocol Stack</th>
<th>Game Theory Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer (N) Protocol Entity</td>
<td>Layer (N) Player or NULL Player</td>
</tr>
<tr>
<td>Layer (N) Service Request</td>
<td>Player (N) Action (influencing Player (N-1))</td>
</tr>
<tr>
<td>Layer (N-1) Service Notification</td>
<td>Player (N-1) Action (influencing Player (N))</td>
</tr>
<tr>
<td>Layer (N) Protocol Specification</td>
<td>Player (N) Goals</td>
</tr>
</tbody>
</table>

Each entity at a given layer of the protocol stack can then be represented by a player, which is characterized by specific goals and actions to perform to reach them. Service request / notification messages are transformed in forward and backward knowledge flows, respectively, in order to model the indirect interaction among the layers in terms of exchange of Protocol Data Units (PDUs).

Nevertheless, given the scope of the analysis, it is possible to simplify the resulting model by introducing “NULL players” for those protocols that are not relevant or whose decisions do not affect the game result.

In principle, it is therefore possible to identify suitable models for the protocol layers in each communication device and linking them through communication links (which could also be modeled as players). A relevant issue is represented by how to translate a given protocol into a suitable “player” for inclusion in the scheme. To this goal, the authors propose to derive the player actions, goals and behavior directly from the protocols specifications, which clearly facilitate the job. Alternatively, to the aim of studying modifications of existing protocols, utility functions describing the protocol behavior can be directly employed.

Some works are already available in the literature about modeling specific protocols or analyze modified versions. The reader can refer to [8-11] for additional details on available models for MAC and TCP inspired by the game theory.
However, it should be underlined that the paper is focused on the usage of game theory to analyze and capture cross-layer interactions of existing protocol implementations, and not concerned with modeling specific protocols or designing optimized solutions.

The key point in the definition of the game theoretical model proposed in the paper is represented by the “clocking” of the model, i.e. the synchronization of the players’ actions. Indeed, as a difference with several existing models, the players which represent protocol entities operating at different layers are also triggered by different events: an application will act based on user’s input, a transport protocol will open a connection based on an application request and perform congestion control based on ACK feedback, etc.

Such multiple clocking levels greatly complicate the analytical models or simulation tools, while they can be easily introduced in the proposed model. In fact, clocking can be captured by the state changes deriving from the flow of PDUs upward and downward the protocol stack of a node, and along the communication links as well. As an example, if TCP sends some segments to be transmitted, such request will propagate in the form of a PDU processed by each subsequent layer of the protocol stack (IP, network access, etc.). However, TCP will analyze its goals and update its strategy (based on flow and congestion control specifications) when it will receive the first ACK (i.e. the first upward PDU reaching the transport layer). This example can be easily generalized for every layer of the protocol stack.

As a consequence, the required steps of the proposed model are:

1. to decide which protocol entities to model (the remaining ones can be modeled as “NULL players”);
2. to define suitable goals and actions for each player based on protocol specifications or existing models;
3. to “connect” the players via cause/effect relationships based on their respective position within the protocol stack;
4. to solve the model and identify stability conditions, steady states, etc.

A benefit of point (3) is that the resulting framework provides a scalable and modular solution, since more protocols and functionalities can be added at given layers without requiring modification of the other layers in case of a layered protocol stack, while explicit interactions within the stack can be easily represented in the case of cross-layer solutions.

Moreover, after the definition of the first “draft model” of the system to analyze on the basis of the principles above, it is advisable to simplify the model in order to facilitate the analysis. Clearly, this aspect implies a tradeoff between the possibility of deriving a “closed form solution” or numerical estimation and the accuracy and representativeness of the model. As an example, the protocol stack of a peer entity can be simplified in case of a unidirectional data flow – by assuming the receiver always ready to receive.

IV. CASE STUDY: TCP OVER IEEE 802.11 MAC

A. Scenario Description

The proposed approach identifies each layer of interest with a player. As a consequence, considering the transmitting node, the set of players contains two entities: TCP and MAC protocols. According to the assumptions expressed in the previous paragraphs, it is not necessary to model the protocols at the receiver node, as their behavior is strictly defined in the protocols operations as “pure ACK generation”. IP layer is not modeled, as the scenario is focused on point-to-point communications. Moreover, the acronyms TCP and MAC will be used interchangeably throughout the text to refer to both as the players and the actual protocol entities, depending on the context.

The considered scenario for validation of the proposed framework is a “single-hop” wireless network, where all nodes are within the communication range of all the others and direct communication between all possible pairs is possible through a shared channel. For sake of clarity, the model is introduced for a scenario consisting of two nodes and then extended to $n$ stations. In the simplest case, one station transmits fixed size segments and the other one passively receives the flow. TCP Tahoe is used at transport layer, with “b” amendment of the IEEE 802.11 standard at MAC/PHY layers. Channel is considered to be ideal, with no transmission errors due to noise, as the work is focused on MAC and transport layers. Moreover, in the case of two stations, RTS/CTS handshake can be avoided. Finally, the receiver advertised window ($rwnd$) is considered constant during all data transfer, and provides an upper bound for the evolution of the TCP congestion window ($cwnd$).

B. The Proposed Model

MAC is responsible with accessing the medium and sending a number of frames depending on how many segments are requested by TCP. The set of MAC actions are represented by the number of frames to be transmitted on the wireless medium.

Based on its specifications [14], the utility function of MAC should follow the principle to empty the queue of the frames to be sent while avoiding collisions. Consequently, MAC interest is to maximize throughput (at link level).

The TCP player deals with congestion control through management of the congestion window. The set of actions available to TCP are represented by the number of segments to be transmitted in a given time window.

The utility function is the mathematical translation of the players’ behavior and plays a major role in correctly modeling the considered scenario. As defined in RFC 793 [15] and RFC 2001 [16], the goal of TCP is to maximize end-to-end throughput while avoiding congestion, or, in other words, to increase its congestion window while probing for available capacity of the end-to-end link.

Indeed, the time scale of MAC and TCP operation is different, as 802.11 MAC is stop&wait (therefore, an action is
performed and its effects are perceived within the transmission of a data frame and its ACK reception + related delays), while TCP employs a credit scheme essentially synchronized on TCP ACK reception.

Given the premises above, a stochastic game is considered appropriate to model the system evolution by means of its "state variable", where the set of states represents the achievable dimensions of the congestion window. Particularly, the set of states $K$ is divided in two non-overlapping subsets, namely the "slow start" set ($SS$), and the "congestion avoidance" set ($CA$).

The transition function expresses the probability to increase the TCP congestion window (or to decide to transmit a given number of frames). In details, only if MAC forwards on the channel all the segments required by TCP, then the congestion window can be increased with probability $p(k^t)$. On the other hand, if MAC does not deliver all TCP segments, the congestion window is set to 1. As a consequence, each stage is a dynamic game where TCP plays first and influences MAC behavior (see next sub-section). Clearly, the state variable cannot take any possible path through the states, because the window grows according to the rules given in the TCP specifications.

The resulting game is clearly "clocked" on reception of TCP ACKs, as signals driving congestion window evolution. Moreover, the game is characterized by perfect yet incomplete information, as none of the players is sure about the opponent's payoff, which is a complication. Still, it is possible to simplify the analysis by changing the incomplete information game into a game with complete but imperfect information [12].

The proposed model is then extended to an ad-hoc network of $n/2$ stations pairs within a single "communication cell". The above model can be again used in order to model each communication couple, but with a different transition function to include collisions and corresponding capacity reduction due to CSMA/CA MAC backoff procedures.

In Fig. 1, a generic stage of the game is depicted through a tree representation while its characteristics are reported in Table II.

![Figure 1. Visual representation of a generic game stage where the congestion window is $n$. The transition function is denoted by $q(.)$, while $u$ acts as a placeholder instead of which there should be the payoff gained by the players.](image)

<table>
<thead>
<tr>
<th>Player set $P = {TCP, MAC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>State variable $k^t \in K = [0,1,...,k_{th}−1] \cup {k_{th},k_{th}+1,...,k_M}$</td>
</tr>
<tr>
<td>Action set $A_{TCP}(k^t) = \begin{cases} {2k^t,2k^t−1,...,0} &amp; \text{if } k^t \in SS \ {a_{th}−k^t+k^t, a_{th}−k^t+k^t−1,...,0} &amp; \text{if } k^t \in CA \end{cases}$</td>
</tr>
</tbody>
</table>
| Transition function $q(k^t+1) = \begin{cases} k^t+1 & \text{if } a^t \neq k_M, a^t = (n,n) = p(k^t) \\ k_M & \text{if } a^t = k_M, a^t = (n,n) = 1−p(k^t) \\ 0 & \text{if } a^t = 01k^t = k, a^t \neq (n,n) = 1 \\ 
| Utility $u(n,n) > u(n,n−1) > ... > u(n,0)$ |
| $u(n−1,n−1) > u(n−1,n−2) > ... > u(n−1,0)$ |
| $...$ |
| $u(1,1) > u(1,0)$ |
| where $u = \{u_{TCP}(k^t), u_{MAC}(k^t)\}$ and $u_{TCP}(m,m) \in R^+, u_{MAC}(n,m) \in R^+$ if $m > n$ |

### C. Solution of the single TX/RX pair scenario

The absence of coordination between TCP and MAC enables TCP to try to increase the congestion window ($cwnd$) to its maximum value (bounded by $rwnd$ value) – thus increasingly requesting as many segments as possible to be delivered by MAC, which makes its best to deliver on the channel as many frames as possible through CSMA/CA. As a result, the proposed scheme models the actions selected by each player as the upper bound of their corresponding sets. Resulting strategies are known as "Markov Strategies", as they only consider the actual game stage and not the whole history. The resulting equilibrium that can be identified for this game is called "Markov Perfect Equilibrium" and can be formally expressed as:

$$s^t = \{\sup A_{TCP}(k^t) \}, \sup A_{TCP}(k^t) - 1,\ldots,0$$

(1)

The corresponding model represents the actual behavior of the two considered protocols:
- if MAC does not succeed in sending the number of PDUs requested by TCP in a RTT, then a TCP timeout occurs and forces congestion window to be reset;
- TCP always aims at transmitting a number of segments equal to minimum between the size of the congestion window ($cwnd$) and the receiver advertised window ($rwnd$).

Using TCP Tahoe, a decrease of the congestion window occurs only when a timeout is experienced. In the simplest scenario where only 2 nodes are active, this leads to a continuous increase of the congestion window and to the

\begin{table}[h]
\centering
\caption{Notation and parameters of the considered game model. $a_i(k^t)$ is the action profile adopted by the player $i$ when the state variable is $k^t$ and $a^t = (a_{TCP}(k^t), a_{MAC}(k^t))$.}
\begin{tabular}{|c|c|}
\hline
Player set $P = \{TCP, MAC\}$ & \\
\hline
State variable $k^t \in K = [0,1,...,k_{th}−1] \cup \{k_{th},k_{th}+1,...,k_M\}$ & \\
Action set $A_{TCP}(k^t) = \begin{cases} \{2k^t,2k^t−1,...,0\} & \text{if } k^t \in SS \\ \{a_{th}−k^t+k^t, a_{th}−k^t+k^t−1,...,0\} & \text{if } k^t \in CA \end{cases}$ & \\
Transition function $q(k^t+1) = \begin{cases} k^t+1 & \text{if } a^t \neq k_M, a^t = (n,n) = p(k^t) \\ k_M & \text{if } a^t = k_M, a^t = (n,n) = 1−p(k^t) \\ 0 & \text{if } a^t = 01k^t = k, a^t \neq (n,n) = 1 \\ 
| Utility $u(n,n) > u(n,n−1) > ... > u(n,0)$ |
| $u(n−1,n−1) > u(n−1,n−2) > ... > u(n−1,0)$ |
| $...$ |
| $u(1,1) > u(1,0)$ |
| where $u = \{u_{TCP}(k^t), u_{MAC}(k^t)\}$ and $u_{TCP}(m,m) \in R^+, u_{MAC}(n,m) \in R^+$ if $m > n$ |

\end{tabular}
\end{table}
highest possible throughput (i.e. the ratio between the number of bits correctly delivered and the time needed to accomplish this operation).

According to the game definition, the number of actually delivered segments is decided by the MAC player, and, for a given stage $k$, it is $a_{MAC}(k)$. Timers, on the contrary, run at the transport level, where the TCP player expects to have all the segments transmitted within a given time interval, continuously updated on the basis of ACK timing, defined as Retransmission Timeout (RTO). If some segments are not acknowledged before this timer expires, then TCP drastically reduces its congestion window and retransmits the packets. Assuming TCP is in congestion avoidance phase, the RTO for a segment is given by:

$$RTO = SRRT + 4DEV$$

where $SRRT$ is the smoothed round trip time (an estimator of the mean value) and $DEV$ is the mean deviation of the SRRT variation, roughly equivalent to the time needed by all the packets already sent to get acknowledged, and, for the same stage $k$, is equal to $a_{TCP}(k)D_t$, provided that $D_t$ be the time needed to have a single packet sent and acknowledged.

In conclusion, if we let $S$ denote the dimension of the segments, the throughput at a given phase $k$ of the game can be computed as

$$T_H(k) = a_{MAC}(k) \cdot S \cdot 8/(a_{TCP}(k) \cdot D_t)$$

For all stages, $a_{TCP}(k)$ and $a_{MAC}(k)$ are equivalent (this would not be the case for a timeout) and the greatest the possible. As a consequence, the value of achievable throughput at the Markov Perfect Equilibrium only depends on $S$ and the time needed to carry on a single segment exchange:

$$T_{H,eq} = S \cdot 8/D_t$$

**D. Solution for $n/2$ TX/RX pairs**

The previous equation can still be used for computing the throughput at the equilibrium point, by changing the term $D_t$ with $D'_t$, i.e. the time required to have a segment delivered and acknowledged in case of collisions. This quantity can be derived by weighting the number of transmissions needed to correctly deliver a frame in case of one, two, up to r-1 collisions ($r$ being the retransmission limit specified by the standard), over their occurrence probabilities. Furthermore, it should be noted that since data exchange is composed of four different frames, the time needed to transmit a segment depends on which frame collides: it may be the TCP data frame, or the TCP ACK, or their corresponding layer 2 acknowledgments. As a consequence, we need to weight the resulting time intervals over their relative occurrence probabilities. As an example, let $D_{t,c}$ be the time to deliver a single TCP segment in case of one collision:

$$D_{t,c} = D_{t,c,DA}P_{DA} + D_{t,c,AA}P_{AA} + D_{t,c,A2}P_{A2} + D_{t,c,A4}P_{A4} + D_{t,c,A2}P_{A2}$$

In this equation, the subscripts define the type of colliding frame ($D$ for data and $A$ for acknowledgment) and its layer.

The available throughput at a given stage $k$ is then obtained by dividing the result of equation (3) by the number of couples sharing the communication link. The resulting equation is:

$$T_H(k) = a_{MAC}(k) \cdot S \cdot 8/(a_{TCP}(k) \cdot D'_t \cdot n/2)$$

**V. VALIDATION**

The proposed model is validated through simulations using ns-2. Without losing generality, $rwnd$ is assumed to be always higher than $cwnd$, i.e. only congestion window evolution is considered. In case $rwnd$ is lower than $cwnd$, TCP actions would have an upper bound of $rwnd$ packets.

**A. Two nodes scenario**

In the two nodes scenario, simulations validate the model, as shown in Fig. 2, where as a result of absence of errors and collisions the TCP congestion window continuously increases. The resulting evaluation of the throughput computed following the proposed model is close to the one obtained with simulated, as it is underlined in Fig. 3.

Moreover, it is possible to study the effect of finite buffer at MAC layer, which underlines more clearly the interaction between MAC and TCP: since MAC is dropping packets due to buffer exhaustion, TCP will be forced to reduce $cwnd$ and the overall data transfer performance. This requires the introduction of the following condition:

$$\exists m \; s.t. \; u_{MAC}(n, m^*) > u_{MAC}(n, m) \; \forall m, n$$

The resulting $cwnd$ evolution in time is presented in Fig. 4 for both the proposed model and simulation. It can be noticed that the time between successive timeouts is captured by the proposed model.
Figure 4. TCP cwnd evolution in the two stations scenario in the case of a 50-frames MAC-layer queue.

B. \( n/2 \) transmitters-receivers scenario

In the more challenging scenario of the \( n/2 \) couples ad-hoc network (Fig. 5), the model closely fits simulation results.

An interesting aspect is the analysis of the “state sequences”, i.e. the evolution of the congestion window starting from the first stage \((k=0)\), when cwnd is set to 1 MSS), till it reaches the first stage again. This represents the average time between two consecutive timeouts. The threshold values are represented in Fig. 6, where an asymptote is visible for \( n = 2 \), underlining that no timeout state is expected in the two-stations case (see cwnd evolution presented in Fig. 2).

VI. CONCLUSION

The proposed work explores a novel application of the game theory for capturing the interactions within the protocol stack of a single node, with the goal of allowing to determine the “steady state” or the operating point of the system in a given scenario.

Summarizing, the main contribution of the paper is to propose a scalable and modular framework to:

- To enable characterization and analysis of cross-layer interactions starting from the protocols’ specifications
- To enable accurate performance evaluation and identification of the operating point of the system in terms of the protocol design parameters

Future work will be aimed at extending the model to the case of multi-hop wireless communications (thus including network-wide cross-layer interactions) as well as at the derivation of design tools based on the proposed scheme.

REFERENCES