Adaptive Learning System for Flow Control

AGÁTA BODNÁROVÁ
Department of Information Technologies
University of Hradec Kralove
Rokitanskeho 62, 500 03 Hradec Kralove
CZECH REPUBLIC
agata.bodnarova@uhk.cz http://www.uhk.cz/csLcz/Stranky/person.aspx?login=bodnaag1

Abstract: The computer network flow control problem has been a recurring theme for long time. The main objective has always been to use network resources efficiently where to prevent the data packet loss, which requires retransmission of the lost information. This article presents the theoretical background necessary to understand the flow control, the existing research in this area and a theoretical background of the intelligent systems and the neural networks. The MF-ARTMAP neural network is introduced, which fulfill the conditions of adaptive learning systems. Experiment for usage of the MF-ARTMAP ANN as adaptive learning system for congestion control is proposed.

Key-Words: Congestion Control, Flow Control, MF-ARTMAP, Neural Network, ANN

1 Introduction
Internet, unless you own a dedicated data link, is based on connectionless, best effort delivery services. It means that the network does its best to deliver your data, but without guarantees.

The Open Systems Interconnection (ISO/OSI) reference model was developed in the 1980s by the International Organization for Standardization (ISO) to present a general reference model for how the computer network architecture should be functionally divided. It uses a layered hierarchy to separate functions, where the layering is strictly enforced. Every layer N uses services provided by layer N-1 and it cannot receive services directly from layer N-2. In the ISO/OSI model, a seven-layer architecture is defined. From a functional point of view, the layers can be defined as follows. Layer 1, the physical layer, provides physical connectivity functions. Data link layer provides data link functions between two directly connected entities and at Network layer the logical addressing and the routing decision making occurs. Transport layer provides either connection-oriented (reliable) and connectionless (unreliable) transport services. Session layer is Layer 5 of the seven-layer ISO/OSI reference model. It provides the mechanism for opening, closing and managing a session between end-user application processes. Presentation layer is responsible for the correct data representation. At Layer 7, the application layer, protocols are defined, see [13].

In data networking we recognize two types of devices: end devices, and intermediary devices. The end devices are the sources and destinations of the data flow. The intermediary devices are responsible for data delivery. In the ISO/OSI reference model, there are Layer 1, Layer 2 and Layer 3 (shortly: L1, L2, L3) intermediary devices, according to which layer they operate. Repeaters and hubs belong to L1 devices, switches belong to L2. L3 devices (routers) operate at the network layer.

2 Understanding congestion and flow control
The IEEE 802.1 bridging and 802.3 link layer protocols and the IETF network (IP) and transport protocols (TCP and UDP) are at the heart of the most widely deployed and interoperable stacks today[12]. Congestion control for IP networks has been a recurring problem for many years. Network congestion occurs when a link or node is carrying so much data that its quality of service deteriorates[20].

Flow control in computer networking is the process of managing the rate of data transmission between two nodes to prevent a fast sender from overrunning a slow receiver [2]. The difference between the flow control and congestion control is in that the goal of flow control is to protect the receiver from overload, whereas the goal of congestion control is to protect the network. Basically the mechanisms are similar - feedback is used to tune the rate of a flow [23], but the task of protecting the network is more complicated. It is necessary to prevent the source end-system from sending data, which would be dropped by the intermediary device because of its overfilled queue. The overfilling can occur, because packets from different sources traveling through different paths.
can converge to the same queue. Sometimes the flow control and the congestion control are used synonymously.

Typical effects of congestion include queuing delay, frame (packet) loss or the blocking of new connections [20]. The common method of avoiding the need to drop frames is to utilize very large intermediary device buffers. Large buffers also may evoke large latencies and add to system cost.

There are many congestion control mechanisms. They can be classified at the link level – L1, at the subnet level – L2, network level – L3, or at the end-to-end level – L4 (flow control).

2.1 End-to-end flow control

Approximately 90 - 95% of internet traffic uses Transmission Control Protocol (TCP) for data transmission[8]. TCP is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks.

Transmission by TCP is made reliable via the use of sequence numbers and acknowledgments. Conceptually, a sequence number is assigned to each octet of data. The sequence number of the first octet of data in a segment is transmitted with the segment and is called the segment sequence number. Segments also carry an acknowledgment number which is the sequence number of the next expected data octet of transmissions in the reverse direction[27]. When TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer runs out, the segment is retransmitted. The duration of this timer is referred to as RTO (retransmission timeout) [25]. To govern the flow of data between TCPs, a flow control mechanism is employed. The receiving TCP reports a "window" to the sending TCP. The window specifies the number of octets, starting with the acknowledgment number, that the receiving TCP is currently prepared to receive.

Field of many researches is trying to optimize TCP. Generally, TCP uses a destination node feedback to control the packet rate on a source side of transmission. In [26] four TCP congestion control algorithms are described: slow start, congestion avoidance, fast retransmit and fast recovery.

There are many algorithms that try to optimize the packet rate by changing window size (number of bytes transmitted before waiting for feedback from destination). Well studied are Reno, Tahoe and Vegas [24]Fast TCP is especially targeted at long-distance, high latency links [22]. In Linux kernels (since version 2.6.19) TCP CUBIC [11] is used by default, the Compound TCP [10] in Windows Server 2008 or Vista is widely deployed.

Optimization of the value of RTO is also the subject of interest in many researches. The current RFC[25] refers, that whenever RTO is computed, if it is less than 1 second, then the RTO should be rounded up to 1 second. An implementation must manage the retransmission timer(s) in such a way that a segment is never retransmitted too early, i.e., less than one RTO after the previous transmission of that segment. A maximum value may be placed on RTO provided it is at least 60 seconds[25]. A very low value of RTO may evolve spurious retransmission and consequently congestions. In opposite, high value of RTO evolve, that browser, mail client, ssh client etc., they close the connection for their own internal timeout.

Minimum RTO is the subject of interest in[16]. In this paper is shown that the conservative 1-second Minimum RTO setting causes severe TCP performance degradation, especially in case of last mile wireless users connected to high-speed backbone links. In [21] the solution uses high-resolution timers to enable microsecond-granularity TCP timeouts. Shown is that eliminating the minimum retransmission timeout bound is safe. The solution is helpful in latency-sensitive datacenter applications where timeouts lasting hundreds of milliseconds can harm response time.

In[15] the authors present a new Window-Based Retransmission Timeout algorithm (WB-RTO) for TCP, which estimates the contribution of each flow to congestion and practically approximates the current level of network contention and allows for asynchronous retransmissions, ordered in time in reverse proportion to their contribution to congestion. They assert, that network delay as it is captured by measuring the RTT alone, cannot always reflect the level of network contention.

3 Intelligent Systems

When we are implementing new technology, it is necessary to have system ability to adapt to the new situations and to the changes in its environment. It must be able to react to the new situations which are similar to previous.

Between the characteristics of the intelligent systems belongs[18]:

• Ability to learn from data and gain knowledge
• Ability to save knowledge  
• Ability to use these knowledge in real concrete situations  

To design intelligent systems we are using the methods of artificial intelligence. Computational Intelligence represents a part of Artificial Intelligence and mainly integrates three different technologies concerning artificial neural networks, fuzzy systems and evolutionary systems. Integration of these systems results in so called hybrid intelligent systems.

3.1 Application of artificial neural networks  
Neural Network (NN) is an information processing paradigm that is inspired by biological nervous systems, such as the brain[14]. Neural Network analogous to people, learns by example.

The learning methods of NN are divided into two categories: supervised and unsupervised. The supervised methods mean that we have information both about the inputs and about the outputs, they are relevant to the inputs. Unsupervised learning means that we have just data – inputs into the NN, but without the relevant outputs.

Clustering is the process of unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters)[1]. Clustering can be defined as the process of separating a set of objects into several subsets on the basis of their similarity. The aim is generally to define clusters that minimize intracluster variability while maximizing intercluster distances, i.e. finding clusters, which members are similar to each other, but distant to members of other clusters. Two clustering strategies are possible: hierarchical or non-hierarchical.

The most known NN they provide data clustering are Self Organizing Maps (SOM) and Kohonen’s network. Both are based on competitive learning, but Kohonen’s networks are using the neighbor feature in their learning process[18].

Back propagation of error (BP) is the most known feed-forward algorithm used in supervising learning process. It is an implementation of Delta rule[17]. The learning process of BP is based on minimalization of the error function. As the algorithm’s name implies, the errors propagate backwards from the output nodes to the inner nodes and changes the value of the weights.

In some kind of complex problems we need to interconnect supervised and unsupervised learning process. One of the most known neural networks based both on supervised learning and fuzzy data clustering is MF-ARTMAP[1], [19]. Its usage could have really interesting results.

3.2 The MF-ARTMAP Neural Network  
ARTMAP neural networks self-organize arbitrary mappings from input vectors, representing e.g. spectral values and terrain variables, to output vectors, corresponding to clusters such as vegetation classes in a remote sensing application. Internal ARTMAP control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error in an on-line setting [5],[18].

In case of MF-ARTMAP, the values of the membership function of vectors from the feature space into the "fuzzy classes" are calculated. Using this function, the input data are organized in clusters: each vector x from input space is bound to fuzzy cluster µA(x). Each cluster introduces relation between its vectors and is defined by a fuzzy set. It is possible to describe the cluster by a fuzzy relation, where parameters of the fuzzy relation are adjusted during the learning process of the network.

The parametric function that represents the fuzzy relation has to fulfill special requirements. With respect to these requirements, the fuzzy relation is defined as follows:

\[
A(x) = \int_y \frac{1}{1 + \left(\frac{(x_S - x_x)^2}{E} \right)^F} / x \quad (1),
\]

In the formula, Y is an input space, X presents an input vector. Vectors XS, E and F are parameters of the fuzzy relation, where vector XS represents the centre of fuzzy relation for each dimension of the space.

The input layer of the network is used either for direct mapping of inputs to the comparison layer, or for normalization of input by suitable activation function of the neuron. The comparison layer is 2-dimensional and its size changes according the number of clusters (i.e. number of neurons in the recognition layer). The recognition layer represents the discovered clusters from the input space and calculates the total value of the membership function of the input to the fuzzy-cluster according the partial values, calculated in the comparison layer. Also this layer is changed dynamically according the current number of clusters. MAPFIELD is an output layer of the network,
where each neuron represents exactly one class. The knowledge of the network is caught in synapse links:

- The links from the recognition to the comparison layer are multi-parametrical, because they encode parameters XS, E and F.
- The links from the recognition layer to MAPFIELD encode classes, integrating several fuzzy clusters from the recognition layer.

During the incremental learning process, the network adapts in two directions:

- Adaptation of network structures evolves the clusters during the learning process with respect to input data. This process results in the identification of the relevant number of fuzzy clusters.
- Adaptation of parameters leads to specification of parameters of clusters in way that fits into the input data.

For details about MF-ARTMAP, see e.g. [1], [7] or [19].

The MF-ARTMAP method satisfies the main presumptions of successful method for classification or prediction by classification:

- Adaptability with respect to additional training examples,
- Stability in recognition of known patterns,
- Clustering based on similarities of inputs.

4 Application of ANNs in flow control

Neural networks are not widely used in flow control. In the field of congestion control just several experiments are published. In [6] an active queue management (AQM) technique based on a dynamic neural network using the back-propagation (BP) algorithm is introduced. On experiments is shown, that the proposed approach yields superior performance with faster transient time, larger throughput, and higher link utilization compared to random early detection (RED) and proportional-integral (PI)-based AQM. In [9] high-speed network traffic prediction is considered as the core of the preventive congestion control. Two different ANN architectures, multilayer perceptron and fuzzy neural network are combined to predict the data traffic. It is shown that the ANN predictors outperform the autoregressive model, and the combination approach enhances the prediction accuracy.

5 Application of the MF-ARTMAP ANN in flow control

One of the possible applications of the MF-ARTMAP ANN is to use this adaptive learning system as the predictor of the congestion. Data clustering using the MF-ARTMAP ANN is in detail in [3] discussed. Experiments using the MF-ARTMAP ANN for the prediction of the market position of the company are in [4] described.

This prediction allows decrease the TCP window size in prevention of the congestion. Usually it is decreased as response on the data loss - if the acknowledgment is not received before the timer runs out or in case when the sent packet is dropped on the intermediary device in reason of congestion.

In our following experiments the possible congestion is by several parameters controlled. This parameters as inputs of the MF-ARTMAP neural network are used. The MF-ARTMAP neural network is then used as the adaptive learning system. Between the parameters belongs the round trip time counted using periodically generated ICMP packet by the sending device. This packet needs to be also in the same QOS class as the prevented data stream and routed in the same way to the receiving end-system. Another inputs of the MF-ARTMAP ANN are the parameters of the prevented TCP data stream.

6 Conclusion

In this article the theoretical background necessary to understand the flow control and the MF-ARTMAP neural network as adaptive learning system is presented. The difference between flow and congestion control is explained. The article shows that the application of ANNs in the field of flow control in computer networks is not well explored. The ANNs are used in few researches to enhance the congestion control. The article describes one of the possible experiments where the MF-ARTMAP neural network is used to decrease the window size of the TCP mechanism in prevention to be decreased by an unwanted accident.
Acknowledgement
This work was supported by the project No. CZ.1.07/2.2.00/28.0327 Innovation and support of doctoral study program (INDOP), financed from EU and Czech Republic funds.

References:
[17] Samarasinghe, S. Neural Networks for Applied Sciences and Engineering - From fundamentals to Complex Pattern Recognition, 400pp. CRC Press, Florida, USA


[25] [RFC 6298] Request for Comments: 6298: Computing TCP's Retransmission Timer, June 2011,

[26] [RFC 5681] Request for Comments: 5681: TCP Congestion Control, September 2009,