Autonomy of the internet: complexity of flow dynamics in a packet switching network

Takashi Ikegami[1], Mizuki Oka[2], Hirotake Abe[3]

[1] Graduate School of Arts and Sciences, The University of Tokyo, Japan
[2] Center for Knowledge Structuring, The University of Tokyo, Japan
[3] Cybermedia Center, Osaka University, Japan

Abstract

Autonomy of the internet (web system) is studied by running an NS-2 simulator. A web system consists of three layers, they are the network, the transport and the application layer and an network simultor called NS-2 can simulate the transport layer of the web as a packet switching network (PSN). This paper reports on the complexity of mutually crossing packet flows which are comparable to other autonomous complex networks, such as real Hippocampus slices, Izhikevich neural networks, or the game of life. One unique feature common in all these systems is the coexistence of several synchronised patterns that we think of as the underlying mechanism of autonomy. In the case of PSNs, adaptive window sizes of each packet flow show synchronisation but only locally, and often chaotic behavior is displayed when congestion occurs. Also considering the packet flow in PSNs as gliders, this congestion allows gliders to bifurcate. We thus propose PSNs as a new experimental testbed for discussing the autonomy and adaptability of living systems.

Introduction

Autonomy is one of the most important characteristics of living systems. Understanding this biological autonomy by reconstructing it using different media is one of the main purposes of Artificial Life studies. For example, the study of autonomous robots uses such an approach. A definition of an autonomous robot is its ability to achieve a task without people having to make commands. There are many examples such as Stefano Nolfi's 'garbage collectors' (Nolfi, 1997), Pfeifer's passive dynamic walker (Pfeifer et al., 2007), Honda's 'Asimo', Sony's 'dog robot' called Aibo, Kojima's 'Keepon' and so on. Some of these robots are "autonomously" detecting walls and avoiding cliffs in various ways. Self-charging robots have also been built already (e.g. a robot that uses snails for energy or a trilobite-like robot that monitors its own battery); so robots can become self-sustainable in that sense.

Rodney Brooks claimed that autonomous robots need not possess any representation of the environment but the environment itself is the representation. They explore the environment and solve a given task. This is a major feature of autonomous robots (Brooks, 1991). Such a concept of autonomy still misses a very fundamental part of biological autonomy, as we are still easily able to distinguish between real and artificial creatures (Brooks, 2001).

A simple but primary definition of an autonomous system is a non-reaction system. For example, a fly's aviation is considered to be an autonomous behavior as it behaves independently from the environmental pattern (May et al., 2007; Takahashi et al., 2008). Another such autonomous dynamic is chaotic itinerancy (Ikegami, 2007); a high dimensional transition dynamic among pseudo attractors. Aoucturier et al. (Aucouturier et al., 2008) used this idea to create a dancing mobile robot. Besides a hard-shell robot, Hanczyc and Ikegami (Hanczyc et al., 2007; Hanczyc and Ikegami, 2010) studied a self-moving droplet. An oil droplet made of olecic acid and sized about 0.1 mm can move by itself and also react to environmental pH.

The underlying principle in all these examples is that an interaction between a system and its environment creates autonomy. In other words, a system can generate and maintain its own context which temporarily couples and decouples with the environmental context. More importantly, a system has its own dynamics without requiring an externally given task. A so-called 'default network' found in a brain's resting state is another example of such autonomy (Raichle et al., 2001). The definition of a default network is the brain activity observed while people are day-dreaming or doing non-specific tasks. A global (non-periodic) synchrony in neural activity was found to exist in the default network.

In this paper, we discuss the concept of autonomy using the example of web systems. Nowadays, web systems have become huge and complex enough to have consciousness-like states. Such web autonomy can be considered sufficiently close to biological autonomy. Corresponding to the non-periodic neural synchrony found in the default network, we will report the non-periodic behavior in a simulated web system.

In §2, we review the constitution of web systems, and in §3, we introduce an internet simulator called NS-2 [1] which emulates the packet switching network (PSN) of the inter-

[1] The Network Simulator - NS-2: http://www.isi.edu/nsnam/ns/

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

systems	*brain*	*internet*	*ANN*	*the game of Life*
basic element	neuron & synapses	node & packet	coupled equation	2 states 2 dim. lattice
structure	small world	small world	random connection	regular lattice
dynamics	local/global synchrony	**this paper**	synchro/polychro	gliders/space ships
memory	semantic & episodic memory	Google DB & Twitter TL	attractors & CI	space pattern

Table 1: Comparison of five different network systems is be conducted; a real brain system, the internet, artificial neural nets (ANN) and the game of Life. The structure and dynamics of each network is depicted in the 2nd and 3rd row, respectively. Possible dynamics of the internet in terms of PSNs is discussed in this paper. In the 4th row, kinds of memory in each network is also described, where an ANN stores its memory in terms of attractors and chaotic itinerancy (CI) and the game of life stores its memory in terms of special spatial configurations.

net's transport layer. In§4, data from the NS-2 simulation will be discussed with respect to dynamic stability. In §5, a simple question we can ask about the web autonomy such as "what happens if everybody stops accessing the internet", is examined. Finally, we will discuss what brings autonomy to a PSN.

Web Systems

The internet has made great progress in the last 20 years and it has become a lifeline for human society. Its structure consists of roughly three layers; a network, a transport and an application layer. When studying the autonomous dynamics of the internet's application layer, we can examine web crawlers and Google's PageRank to see how the database is automatically organised and ranked. Many social network services (SNSs) such as Twitter are also worth noting. They mutually copy and reproduce personal timelines in massively parallel ways which is somehow complementary to what Google's service is processing on their stored data.

On the other hand, what enables Google and Twitter to function correctly is a PSN on the transport layer and its backbone network layer. This creates a system that can be mutually connected on the internet with IP addresses on the network layer. The protocols used for communicating among those IP addresses are TCP or UDP. In particular, TCP is equipped with relatively intelligent software. Each network router sends a data flow by switching data packets. TCP plays an important role in delivering the data to the address without going missing nor permutation of packets. The sender controls the data amount and the router controls data routing.

The topological structure of the internet has been intensively studied and its small world property (Watts and Strogatz, 1998) is revealed. One property that a network has is a hub connection, and this is now widely known in generic information about transporting systems, e.g. gene networks or neural networks in the brain. A.L. Barabasi reported that such small world networks become even more robust when compared with random networks (Barabasi and Albert, 1999).

But we also think it is important to understand the flow dynamics on the internet rather than just its topological structure. Graham proposes PSNs as a new model for a brain system in place of a circuit switching network (Graham and Rockmore, 2011). Grifith et al. argue the similarity between Google's PageRank system and how the mind works (Griffiths et al., 2007). These are the dynamic properties of a network and we hope that the minimal and prerequisite fundamental dynamics for a kind of intelligence and mind can be found in PSNs.

Indeed, the complexity of the internet's dynamics has an equally curious property which we find in the human neural circuit. There have been several studies concerning dynamic complexity of PSN (see e.g. (Frommer et al., 2009)). The inherent complexity of PSNs can be seen at the level of producing consciousness-like macro phenomena, which Tononi and Edelman hypothesised with their concept of dynamic core and reentry (Edelman and Tononi, 2000).

We list characteristics in the Table 1 to compare PSNs with the other complex enough network systems. Neural synchronisation phenomena were discovered by Singer in the visual cortex of a cat (Singer and Gray, 1995). Such synchrony is also found as a self-moving pattern in Hippocampus slices (Takahashi et al., 2010) or in the massive number of artificial neural networks (ANN) (Izhikevich, 2000; Izhikevich and Edelman, 2008; Izhikevich, 2006). Here we only refer to the Izhikevich neural net, as this network is realistic in its scale and types of neural spiking. It should be noted that synchronisation is not always a global phenomenon but it is often observed as a local synchronisation or clustering of neural oscillation. In other words, different neural clustering in space and time can coexist. This is a universal phenomenon in generic coupled nonlinear systems (Kaneko, 1990).

What is more interesting is that a localized pattern can propel itself through space; we call these gliders and spaceships in the game of life. A glider or spaceshp is used to prove the universal computability of the game of life as demonstrated by William Poundstone (Poundstone, 1984). Indeed the role of a glider pattern is for a basic information packet to run through the system, and gliders sponta-

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

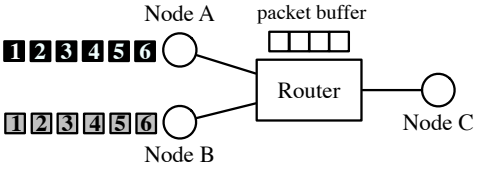

Figure 1: An example network has three nodes and one router. Nodes A and B send packets to node C creating two flows.

neously interact with each other to maintain the system's autonomous information processing. One of the purposes of this paper is to look for similar phenomena in PSNs by taking packets as the simplest form of glider. This notion of autonomy is what we are going to seek in PSNs.

If a system is autonomous and sufficiently complex, we expect it to show various signs of intelligent behavior. One such intelligent behaviour is based on memory dynamics. Therefore, we put memory as the 4th row in the table. Here different kinds of memory are potentially stored in the networks. ANN stores memory as attractors (see e.g. Hopfield network (Hopfield, 1982)), which can be referred to as semantic memory, but it also stores episodic memory as chaotic itinerant dynamics of pseudo attractors (Nozawa, 1992; Tsuda, 2001; Tani, 1998). As discussed at the beginning of this section, the internet now mainly consists of two memory structures. One is Google's Database (DB) and the other is Twitter's time line (TL). We think these are related to semantic and episodic memory in real brain systems, except that they are about the application layer. However this is beyond the scope of this paper and will be reported in ASSC 15 [2].

Finally, the game of life stores memory in terms of special spatial configuration. The best known example of a cellular automaton's (CA) memory might well be von Neumann's self-reproducing automata (Neumann, 1967). Since the game of life can emulate any kinds of CA, we propose here that any powerful CA can become a universal Turing machine in the game of life.

The Packet Switching Network Model

NS-2 is a simulator for a packet switching network (PSN). We claim that this network corresponds to the neural network of a brain system, where each connected neuron sends electric pulses to the others with different timing and strengths. At the end of this section we compare the basic properties of PSNs and neural networks, but first we explain how NS-2 works.

To illustrate how NS-2 works, let us consider a simple network where three nodes are connected through one router as depicted in Fig. 1. For example, when node A tries to

[2]ASSC 15 : The 15th annual meeting of the ASSC. http://www.theassc.org/conferences/assc_15

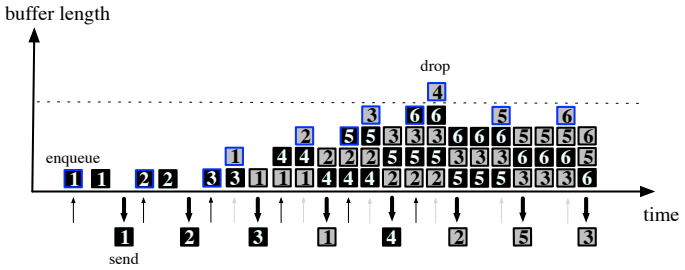

Figure 2: Changes occur in the buffer as the router receives and sends packets with the buffer size equal to four.

send a packet to node C, it goes through the router. The router has a certain length of packet buffer, say four. The packet will be sent to node C if the packet buffer is not over capacity. If the packet is over capacity, congestion occurs and the packet will simply be dropped. Fig. 2 shows the changes in the router's buffer status when packets 1, 2, 3, 4, 5, and 6 from node A and B are sent to node C respectively. The figure shows the router's buffer when two flows occur, one from node A to node C and the other from node B to C. The black arrow shows the arrival of the packet to the node, enqueueing the packet to the buffer, then sending the resulting dequeueing of the packet. These two events would show in the logged file of NS-2 as follows:

```
+ time A C 1 1
- time A C 1 1
```

where each line denotes 'event', 'time', 'destination node id', 'arrival node id', 'flow id' and 'packet id'. The '+' denotes the enqueueing event to the buffer and the '-' denotes the dequeueing event to the buffer. Similarly, when the node arrives at either a route or a node, it will be logged in the file as:

```
r time A C 1 1
```

where 'r' denotes an arrival event. When the buffer becomes full and create a congestion, a dropping event occurs as shown in the figure for node B packet and it would be logged in the file as follows:

```
d time B C 2 4
```

When a drop event happens, that packet will always be lost. The Transmission Control Protocol (or TCP) is a mechanism designed to create more reliable transmissions. TCP sends a packet with a serial number. When a node receives a packet, it sends back the serial number which is called acknowledgement (or ACK). When the sender receives an ACK, then it sends the next packet. If the sender node does not receive an ACK for a certain period of time or receives ACKs with wrong sequence number, then it resends the same packet. However, as one can easily imagine, sending packets one by one is not efficient. To cope with this, TCP has a parameter called congstion window size which

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

Figure 3: Network topology for the experiment.

defines how many packets the sender can send at one time. This window size is advertised by the receiver node. To improve performance, the advertised window size needs to be set to 'large'. However, when the window size is too large, it creates congestion with other packets resulting in packet drops and consequent requests to re-send the packets.

While the advertised window size is imposed by the receiver, there is another window size imposed by the sender called the congestion window size, or called the "cwnd". When a new connection is established with a node,the cwnd is initialized to one segment (i.e. the segment size is announced at the other end). Each time an ACK is received, the cwnd is increased by one segment. The question about how to improve performance then becomes how to adjust the advertised window size and the cwnd size. The former is related to the amount of available buffer space at the receiver for the connection; the latter is based on the sender's assessment of perceived network congestion. It is important to note that the cwnd size continues to increase to a given threshold or until a drop event happens.

There are a number of different algorithms to increase the cwnd size. The one we used in this study is called Reno. The Reno algorithm increases the cwnd exponentially until the first packet drop occurs due to congestion. After the first drop, the cwnd is set to half then continues to update itself in a linear manner. When a drop happens, it again sets to half and starts to increase again and continues this process throughout.

As we have explained so far, PSNs (and the simulator, NS-2) have the following corresponding properties when compared to biological neural networks:

- Flow dynamics in PSNs correspond to the pulse trains of neural activities.
- A buffer size corresponds to the activation threshold of a neural firing. In the NS-2 model we use 10 as the buffer size and the threshold of a real neural cell is about 15 mV.
- Strength of the cwnd corresponds to synaptic strength. Here we have Reno algorithm to change the window size.
- A drop event corresponds to the fact that neural pulses cannot contribute to an overshooting event.

Having this correspondence in mind, we analyse and explore the PSN in the next section.

Analysis

We have conducted experiments using NS-2 on a simple network topology with a 30 node setting where each node is connected to the next node as depicted in Fig. 3. The analysis will be on the flow dynamics, the congestion phenomena and the robustness of the flow patterns when pouring a temporary flow from outside. We will explain these below.

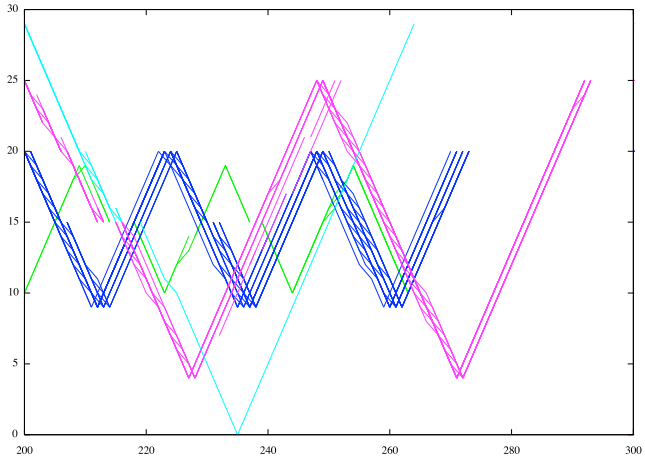

Figure 4: An example of spatio-temporal packet flow pattern. The horizontal axis is time (each step is 10 msec) and the vertical axis is the spatial node (here the total number of nodes is 30). As this figure shows, each packet flow spontaneously bifurcates so that lines are multiplied. Every flow shows concatinated "V-shaped" pattern, since every successfully received packet is followed by ACK signal sent back to the sender.

Flow dynamics

A unique characteristic of a PSN is a self-tuning cwnd size for each flow in the network. In the first simulation, we created 30 flows in which each router sends a series of packet data to its neighbors through an optimised routing pathway (or trace). All flows are set to have an equal length. As described above, each packet between connected nodes ($i \rightarrow j$) is characterised by a triplet (+,-,r) state, where the state "+" corresponds to "the packet in node i is ready to send", "-" to "the packet has been sent to node j" and "r" to "the packet has been received by node j". Using this information, we can visualise the spatio-temporal flow pattern as shown in Fig. 4.

As for basic observations, we see i) The more numbers of nodes the flow travels, the more transport time is required; ii) Due to spontaneous time delays, packets that constitute the same flow arrive in different timeframes, which causes the bifurcation of flow pathways. This bifurcation pattern can be different for each flow; iii) Even within the same flow and in between the same traces, the bifurcation pattern can vary temporarily.

It should be noted that the bifurcation of flow path due to the time delay in point (ii) above is a novel feature in dy-

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

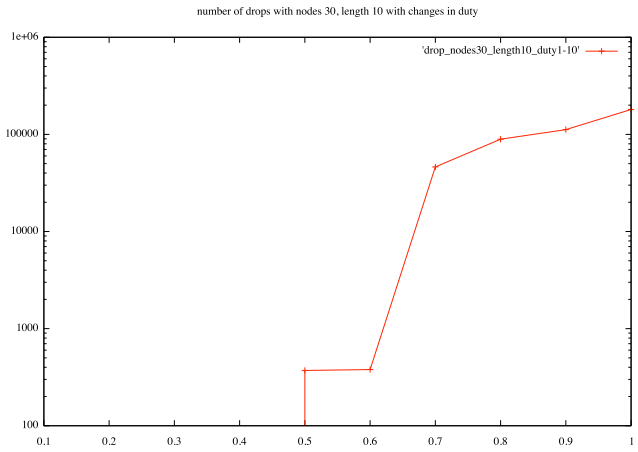

Figure 5: The number of drop events as a function of the integrated input packets (we call a 'duty ratio') on each source node. Input packets are given periodically for each node. When duty ratio = $x(< 1)$, it means that each source node sends packets for x seconds and rests for the next $(1 - x)$ second for every cycle.

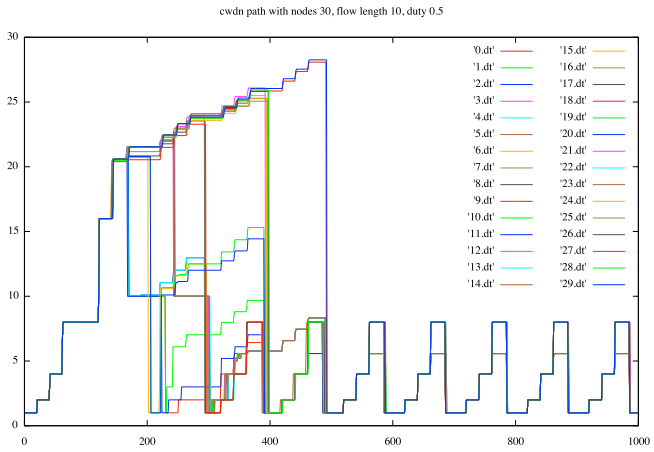

Figure 6: An example of the cwnd dynamics. Each window size of the flow is overlaid multiple times. Here the system has a few drop events so that the network settles down to a periodic synchronized pattern after four seconds.

namic systems. In PSNs, flow is spontaneously quantized into a series of packets when transporting to other nodes. This clustering event is not written in the form of an "equation" in the PSN but happens only as a result of congestion and timing. Although such spontaneous clustering is similar to congestion patterns studied in traffic models (Chowdhury et al., 2004), PSNs have drop events and ACK signals. In the case of traffic jams, vehicles or ants will never disappear. This traffic jam phenomenon is called congestion. Bifurcation of flow pattern is correlated with this congestion pattern, which we will focus on in the next section.

Congestion Flow

As explained in the previous section, the source node of each flow tunes the buffer size and the cwnd size to reduce the drop events. When the amount of flow becomes larger than a specified volume, congestion occurs spontaneously and the number of drop events increases exponentially as the amount of flow increases. Fig. 5 shows an increase in the number of drops as the ratio of the packet flow period to the frame increases.

The drop events trigger the clustering of the window size. In the first hundreds steps, each window size is mutually tuned and their phases are synchronised as shown in Fig. 6. This is known as TCP globa synchronisation [3]. In the figure, all flows are set to have an equal length. In this case, even though the cwnd size changes from a periodic to an aperiodic state the packet flow is mostly periodic. Because almost all the drop events occur at the source of the flow, the drop events change the cwnd dynamics but not the packet flow pattern.

We artificially create a special topology that produces massive congestion in the middle nodes (i.e. between nodes 14 and 15 of the 30 nodes). In this case, both the flow patterns and the cwnd dynamics become unstable, as the drop events occur not only at the source but also at the relay nodes. Some examples of the flow patterns and cwnd patterns are depicted in Fig. 8. The transport time of every flow shows a power law behavior of the exponent being equal to -2 as shown in Fig. 7. The connection between nodes 14 and 15 becomes a bottleneck and determines the entire time scale.

When a cwnd dynamic settles into a periodic state, its periodicity becomes almost consistent with its varying window size. In the case of aperiodic cwnd dynamics shown in Fig. 8, we classified this into five clusters based on the temporal oscillating pattern as we do for the dynamical systems.

1. Periodic state: The window size changes periodically in a stepped way. Fig. 8-(a) represents this cluster.

2. Chaotic state: The window size changes in an aperiodic way. In the case of Fig. 8-(b,c), we have two different chaotic behavious; one with fast amplitudes varying in time and one with slow amplitudes changees in time, where their time scales also show some variations.

3. Intermittent chaotic state: The periodic oscillation of window size is intermittently perturbed by a burst of large window size. The other intermittent behaviour is that the amplitude almost periodically oscillates around a certain

[3] TCP Global Synchronisation : http://en.wikipedia.org/wiki/TCP_global_synchronization

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

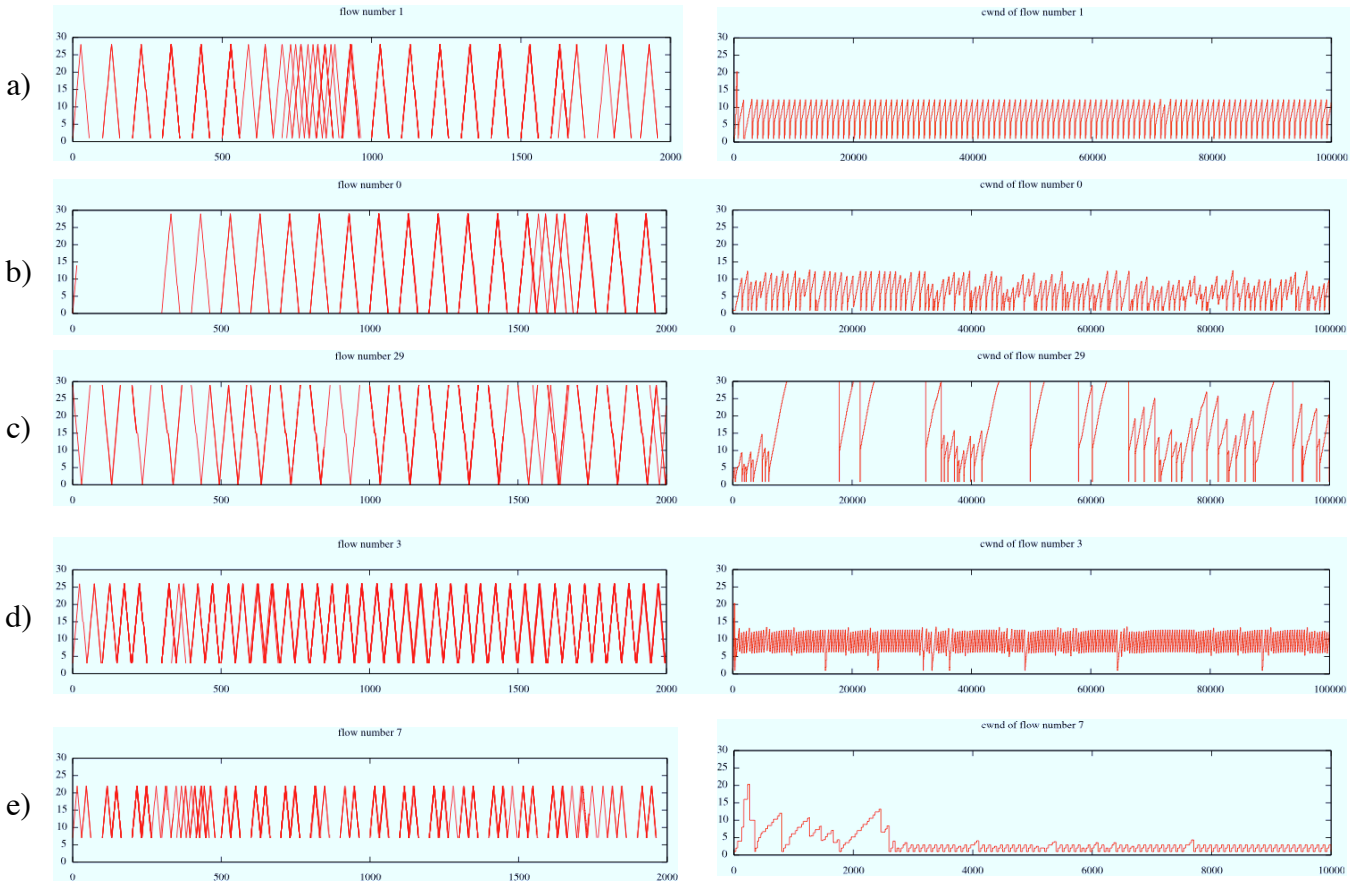

Figure 8: Examples of five categories of cwnd dynamics (right) and the corresponding packet flow pattern (left). From the above, these are: a) periodic, b) chaotic type 1, c) chaotic type 2, d) intermittent chaos type 1, and e) intermittent chaos type 2. Time scale is set from 0 to 100,000 except for the case e), since the oscillation of case e) is much faster compared to other cases. See the details in the text.

value but is intermittently perturbed by a larger or smaller (often null) window size. Both of these can be observed in Fig. 8-(d,e).

Flows synchronised in the same clustering pattern can be found in the spatial neighbors with some exceptions. It should be noted that the chaotic synchrony is what we compare with the Hippocampus slice or Izhikevich neural ensemble as a candidate for the origin of autonomy and a computation primitive. As discussed in §2, these synchronised patterns are important in maintaining the functionality of the network as a whole. In particular, we propose that these synchronised patterns may be a source of PSN autonomy.

Perturbation

Let us perturb the flow network by pouring an extra flow from outside at a certain time duration. A stable network, where both flow and cwnd pattern become periodic, will remain robust against the perturbation, i.e. the flow pattern and cwnd dynamics will remain periodic. On the other hand, a network that has massive congestion at the middle point is less robust against perturbation. Comparisons before and after perturbation demonstrate that flow pattern (Fig. 9) and cwnd dynamics will be different. In other words, the flow state can be said to have chaotic instability as it amplifies the small difference caused by the extra flow input.

It can be said that for this special network, the flow state becomes less robust against perturbation. But we also interpret the state as adaptive because it never falls to a fixed flow state.

Discussion

The autonomy we are looking for is having the flexible internal dynamics to change responses against external inputs. A certain amount of chaotic dynamics may be responsible for this. In previous studies, we have partially proven that an autonomous robot equipped with the coupled FitzHugh-Nagumo equations shows such autonomy (Aucouturier et al., 2008). Analysis of how such a robot can

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

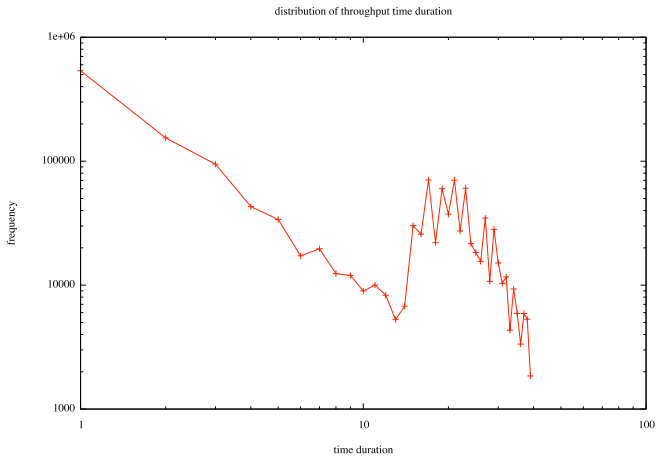

Figure 7: Distribution of throughput time duration. The distribution of the shorter time duration shows the power law behavior, which corresponds to the bottleneck connection between the 14th and 15th nodes.

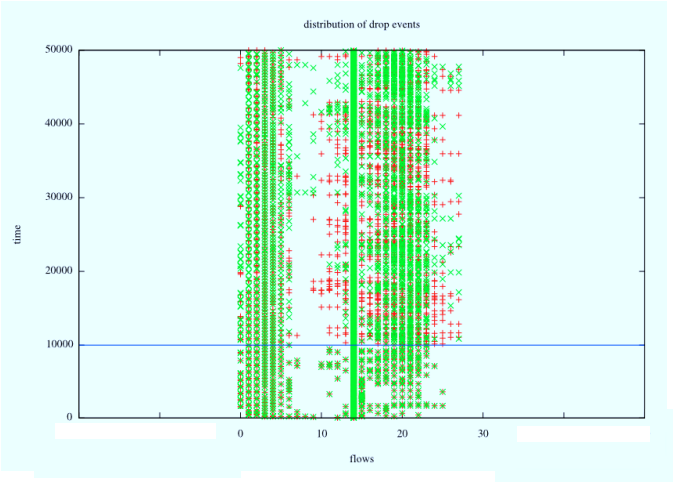

Figure 9: Comparison of space time plots of drop events for the original and the perturbated network (the horizontal axis denotes the network node IDs and the vertical axis denotes time steps). The perturbation is introduced as a pulse packet flow of duty ratio=1 poured during 10,000 and 20,000 mseconds. After the perturbation, a network does not come back to the original state.

interact with the environment was the focal point.

In the present paper, we have studied PSNs to reveal the internal flow dynamics and the system responses to external pulse inputs. When increasing the amount of flow from the outside, we showed that cwnd dynamics change from periodic to chaotic. In the case of a network with a bottleneck edge, the transport time of each flow obeys the power law and the real packet flow becomes chaotic for a long period of time.

In §1, we posed a question, "what happens if everybody stops accessing the internet?". An answer to this question might be "it won't stop immediately but will last a long time because it does not attain a stable pattern as shown from the PSN experiment". The ever-changing nature of chaotic and intermittent clustering may drive the autonomy of the internet even with periodic inputs supplied, for example, by automated web crawlers. If a simple one-dimentional PSN can have complex clustering patterns, the internet with its massive data flow should have ever-lasting and changing clustering, thus making it autonomous. We believe these findings correspond to the examples of complex networks in Table 1. That is, autonomous networks can develop complex local/global space time clustering or gliders.

We also claim that PSNs are a novel class of dynamic systems that spontaneously bifurcate their flow structure and may be the backbone of the internet today. The corrsponding gliders in the game of life and other intelligent systems in Table 1 remaine stable. However in the case of PSNs, those localised patterns can bifurcate. This bifurcation of glider-like patterns is why we think this PSN is a new interesting testbed for Artificial Life studies. As for future investigations, we have to pay more attention to other novel features of PSNs. For example, dynamic routing and another TCP/IP will be a future research target.

Our analysis here is about the transport layer not the application layer. Examples of autonomous software in the application layer include web crawlers, peer-to-peer software, and SNS bots. The exsistence of those two layers does contribute to making a more complex autonomous system. Within a computer system, an example of a software algorithm that generates chaotic dynamics in the hardware layer was reported by Berry et al. in 2006 (Berry et al., 2006). We are now studying the autonomous behavior in the application layer using the notion of clustering dynamics found in the PSN reported in this paper.

References

Aucouturier, J.-J., Ogai, Y., and Ikegami, T. (2008). Using chaos to trade-off synchronization and autonomy in a dancing robot. *Trends and Controversies, IEEE Intelligent Systems*, 2:74–85.

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

Berry, H., Perez, D. G., and Temam, O. (2006). Chaos in computer performance. *Chaos*, 16:01311011–013110115.

Brooks, R. (2001). The relationship between matter and life. *Nature*, 409:409–411.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence Journal*, 47:139–159.

Chowdhury, D., Nishinari, K., and Schadschneider, A. (2004). Self-organized patterns and traffic flow in colonies of organisms. *Phase Transitions*, 77:601–624.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023

Edelman, G. and Tononi, G. (2000). *A Universe of Consciousness: How Matter Becomes Imagination*. Basic Books.

Frommer, I., Harder, E., Hunt, B., Lance, R., Ott, E., and Yorke, J. (2009). Bifurcations and chaos in a periodically probed computer network. *International Journal of Bifurcation and Chaos*, 19:3129–3141.

Graham, D. and Rockmore, D. (2011). The packet switching brain. *J. Cogn. Neurosci.*, 23(2):267–76.

Griffiths, T. L., Steyvers, M., and Firl, A. (2007). Google and the mind: Predicting fluency with pagerank. *Psychological Science*, 18:1069–1076.

Hanczyc, M. M. and Ikegami, T. (2010). Chemical basis for minimal cognition. *Artificial Life*, 16(3):233–243.

Hanczyc, M. M., Toyota, T., Ikegami, T., Packard, N., and Sugawara, T. (2007). Chemistry at the oil-water interface: Self-propelled oil droplets. *J. Am. Chem. Soc.*, 129(30):9386–9391.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *J. Proc Natl.Acad. Sci.*, 79:2554–2558.

Ikegami, T. (2007). Simulating active perception and mental imagery with embodied chaotic itinerancy. *J.Consciousness Studies*, 14:111–125.

Izhikevich, E. M. (2000). Neural excitability, spiking, and bursting. *International Journal of Bifurcation and Chaos*, 10:1171–1266.

Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Comput.*, 18:245–282.

Izhikevich, E. M. and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *PNAS*, 105:3593–3598.

Kaneko, K. (1990). Clustering, coding, switching, hierarchical ordering, and control in network of chaotic elements. *Physica D*, 41:137–172.

May, A., Hsieh, C., Sugihara, G., and Brembs, B. (2007). Order in spontaneous motion. *PlosOne*, 5:443.

Neumann, J. V. (1967). *Theory of Self-reproducing Automata (ed. Arthur W. Burks)*. University of Illinois Press.

Nolfi, S. (1997). Evolving non-trivial behaviors on real robots: A garbage collecting robot. *Robotics and Autonomous Systems*, 22:187–198.

Nozawa, H. (1992). A neural network model as a globally coupled map and applications based on chaos. *Chaos*, 2:377–386.

Pfeifer, R., Lungarella, M., and Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, 318:1088–1093.

Poundstone, W. (1984). *The Recursive Universe: Cosmic Complexity and the Limits of Scientific Knowledge*. William Morrow & Co.

Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001). Inaugural article: A default mode of brain function. *PNAS*, 98:676–82.

Singer, W. and Gray, C. M. (1995). Visual feature integration and the temporal correlation hypthesis. *Ann Rev Neurosci*, 18:555–586.

Takahashi, H., Horibe, N., Shimada, M., and Ikegami, T. (2008). Analyzing the house fly's exploratory behavior with autoregression methods. *J. Phys. Soc.*, 77:084802.

Takahashi, N., Sasaki, T., Matsumoto, W., Matsuki, N., and Ikegaya, Y. (2010). Circuit topology for synchronizing neurons in spontaneously active networks. *PNAS*, 107:10244–10249.

Tani, J. (1998). An interpretation of the 'self' from the dynamical systems perspective: Aconstructivist approach. *Journal of Consciousness Studies*, 5:516–42.

Tsuda, I. (2001). Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behav. Brain Sci.*, 24:575–628.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393:440–442.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/ecal2011/23/56/2022509/0262297140chap56.pdf by guest on 28 September 2023