**ABSTRACT:**

It is well known that max-weight policies based on a queue backlog index can be used to stabilize stochastic networks, and that similar stability results hold if a delay index is used. Using Lyapunov optimization, we extend this analysis to design a utility maximizing algorithm that uses explicit delay information from the head-of-line packet at each user. The resulting policy is shown to ensure deterministic worst-case delay guarantees and to yield a throughput utility that differs from the optimally fair value by an amount that is inversely proportional to the delay guarantee. Our results hold for a general class of 1-hop networks, including packet switches.

**EXISTING SYSTEM:**

The stability works all use backlog-based transmission rules, which treat joint stability and utility optimization. However, work introduces an interesting *delay-based* Lyapunov function for proving stability, where the delay of the head-of-line packet is used as a weight in the max-weight decision. This approach intuitively provides tighter control of the actual queueing delays.
DISADVANTAGES OF EXISTING SYSTEM:

1. In an interesting delay-based Lyapunov function for proving stability, where the delay of the head-of-line packet is used as a weight in the max-weight decision. This approach intuitively provides tighter control of the actual queuing delays.

2. A single head-of-line packet is scheduled based on the delay it has experienced, rather than on the amount of additional packets that arrived after it.

3. Use delay-based rules only in the context of queue stability. To our knowledge, there are no prior works that use delay-based scheduling to address the important issue of joint stability and utility optimization.

PROPOSED SYSTEM:

THIS paper considers the problem of scheduling for maximum throughput utility in a network with random packet arrivals and time-varying channel reliability. We focus on 1-hop networks where each packet requires transmission over only one link. At every slot, the network controller assesses the condition of its channels and selects a set of links for transmission. The success of each transmission depends on the collection of links selected and their corresponding reliabilities.
The goal is to maximize a concave and non-decreasing function of the time-average throughput on each link. In this paper we use a delay-based Lyapunov function and extend the analysis to treat joint stability and performance optimization via the Lyapunov optimization technique from our prior work. The extension is not obvious. Indeed, the flow control decisions in the prior work are made immediately when a new packet arrives, which directly affects the drift of backlog-based Lyapunov functions. However, such decisions do not directly affect the delay value of the head-of-line packets, and hence do not directly affect the drift of delay-based Lyapunov functions. We overcome this challenge with a novel flow control policy that queues all arriving data, but makes packet dropping decisions just before advancing a new packet to the head-of-line. This policy is structurally different from the utility optimization works. This new structure leads to deterministic guarantees on the worst-case delay of any non dropped packet and provides throughput utility that can be pushed arbitrarily close to optimal.

**ADVANTAGES OF PROPOSED SYSTEM:**

✔️ It is important to analyze these delay-based policies because they improve our understanding of network delay, and because the deterministic guarantees they offer are useful for many practical systems.
while our deterministic delay guarantees hold for general arrival sample paths, our utility analysis assumes all arrival processes are independent of each other (possibly with different rates for each process) and independent and identically distributed (i.i.d.) over time-slots.

The deterministic delay guarantees we obtain in this present paper are quite strong and show the advantages of our new flow control structure.

SYSTEM ARCHITECTURE:
MODULES:

- System Model Module
- Delay-Based Flow Control
- Finding packet delivery delay
- Utility Maximization

MODULES DESCRIPTION:

System Model Module

In this work, we study wired networks to concentrate on the delay behavior of network algorithms without the additional complications of interference limited wireless communications. We build the system model environment by considering the entities as Source nodes, two-hops and then the destination nodes. The source node is featured with sending data to the destination node, which is connected with the IP address using Socket programming. The source node can select the data by browse button and transmit to the destination by providing the IP address for the destination node. Before the network transfer, the packets will be split in the source node and then they are routed to the destination nodes via the delay nodes.
Delay-Based Flow Control

In this module, we expose the delay deficiencies of long term utility maximizing designs. In particular, we reveal that both the multi-path routing and the scheduling components of the earlier designs must be significantly changed or enhanced to obtain favorable end-to-end delay performance. Based on the observations, we propose a general delay-aware design framework whereby the utility maximization is combined with delay-aware routing and scheduling components to achieve our dual objective.

Finding packet delivery delay

In this module, we introduce a theoretical framework for analyzing routing performance in delay tolerant networks which is aimed at characterizing the exact distribution of the fundamental performance metrics described above, namely, packet delivery delay. This is in sharp contrast with existing work, which either considers only expected value of the metrics of interest.

Utility Maximization

We combine this loop-free route construction strategy with a token-based scheduling discipline that regulates the higher order statistics of service processes to achieve drastic reductions in the end-to-end delay performance, while guaranteeing long-term optimality characteristics. A step beyond just solving the
above optimization, our second goal is to develop a new mechanism that reduces the end-to-end delay experienced by the traffic while maintaining the utility maximizing nature. The end-to-end delay experienced by one packet is defined as the difference between the time instance of injection at the source and reception at the destination, which is a short-term metric instead of a long-term throughput.

**SYSTEM CONFIGURATION:-**

**HARDWARE CONFIGURATION:-**

- Processor - Pentium –IV
- Speed - 1.1 Ghz
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA
SOFTWARE CONFIGURATION:-

- Operating System : Windows XP
- Programming Language : JAVA
- Java Version : JDK 1.6 & above.
- IDE : Netbeans 7.2.1