SPi	Journal Code				Article ID				Dispatch: 08.11.12	CE:
	N	E	M		1	8	1	9	No. of Pages: 18	ME:

INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/nem.1819

Self-adapting protocol tuning for multi-hop wireless networks using Q-learning

Dan Marconett,*† Minsoo Lee, Xiaohui Ye, Rao Vemuri and S. J. Ben Yoo

Department of Electrical and Computer Engineering, University of California Davis, Davis, CA 95616, USA

SUMMARY

Today's network control systems have very limited ability to adapt to changing network conditions. The addition of reinforcement learning-based network management agents can improve quality of service by reconfiguring the network layer protocol parameters in response to observed network performance conditions. This paper presents a closed-loop approach to tuning the layer three protocol based upon current and previous network state observations, specifically the Hello Interval and Active Route Timeout parameters of the AODV routing protocol (AODV-Q). Simulation results demonstrate that the self-configuration method proposed here demonstrably improves the performance of the original Ad-Hoc On-Demand Distance Vector (AODV) protocol, reducing protocol overhead by 43% and end-to-end delay 29% while increasing the packet delivery ratio by up to 11%. Copyright © 2012 John Wiley & Sons, Ltd.

Received 27 October 2011; Revised 16 October 2012; Accepted 24 October 2012

1. INTRODUCTION

While the strict layering architecture of the Open Systems Interconnection (OSI) stack is conceptually useful, it is not as effective for wireless networks when time-varying traffic is transmitted over a channel with limited throughput. Efficiently utilizing the resources with quality of service (QoS) provisioning requires a cross-layer optimization approach. As a result, better performance can be expected from information exchange across the protocol layers [1,2]. The purpose of this paper is to address these issues by exploring the concept of intelligent network management for globally optimum performance in a dynamic wireless network deployment.

In typical network deployment scenarios, networks elements are limited in their abilities to adapt to changing application demands and topology characteristics, taking the context of these changes into account. In the case of routing in multi-hop wireless networks, battery-powered devices create challenging problems in terms of prolonging the lifetime of the network. In designing intelligent routing protocols, the various features of sensor networks lead to a set of optimization problems in routing path length, load balancing, consistent link management, and aggregation [3]. In real scenarios, however, these factors are usually in conflict with one another, and influence the routing performance in a complex way. This, in turn, leads to the need for a more sophisticated routing scheme that makes ideal trade-offs between multiple factors. Clearly, solving the optimization goals separately does not lead to a globally optimal solution; rather, all metrics should be addressed with respect to one another.

A solution for addressing these multi-variant optimization problems in network management lies in the vision of cognitive networks [4]. Cognitive networks continuously adapt to changing environmental

*Correspondence to: Dan Marconett, Department of Electrical and Computer Engineering, University of California Davis, Davis, CA 95616, USA.
†E-mail: dmarconett@ucdavis.edu

conditions and/or user needs by constantly optimizing the bandwidth access and communication links. Typically, machine learning techniques, such as Q-learning [5], help implement the adaptation methods of self-configuration and self-management in the autonomic computing paradigm. Recent work further reinforces the efficacy of leveraging machine learning for network management task optimization [6–9].

The self-configuration of network systems has cross-layer ramifications for the protocol stack, from the physical (PHY), Medium Access Control (MAC), network, and transport layers to the application layer. Therefore, cross-layer design [10–13] approaches are critical for the efficient utilization of limited resources, to enable QoS guarantees, in future wireless and heterogeneous networks. In this paper, we present the concept of self-configuration in a cross-layer context, which can overcome the current limitations of network management in heterogeneous wireless networks, by allowing networks to *observe*, *analyze* and *act* [14] in order to optimize performance. Our approach is to augment the routing strategy of the AODV routing protocol with Q-learning, to ensure that the packet delivery ratio can be increased, while at the same time minimizing management overhead.

Toward the above stated goals, we present a new architecture of reconfigurable ad hoc routing management with Q-learning, namely, Q-learning based self-configuration (QLS) management and the AODV-Q protocol. The QLS management architecture enables nodes to efficiently learn optimal routing strategies, thereby enhancing the packet delivery ratio, end-to-end delay, and other QoS performance metrics. We present NS-2 simulation results showing that our cross-layer, self-configuration approach successfully improves the scalability of the AODV routing protocol in a heterogeneous network environment. The remainder of the paper is organized as follows. Section 2 presents a belief survey of related work. Section 3 gives an overview of our network architecture with reinforcement learning techniques for autonomic self-management. Section 4 describes AODV-Q in detail. Section 5 explains the NS-2 simulation scenario. Section 6 presents NS-2 simulation results. Finally, Section 7 concludes by projecting future research directions.

2. RELATED WORK

2.1. *Cross-layer approaches for intelligent network management in wireless networks*

The realm of network management covers a vast collection of issues, such as IP configuration, security and network monitoring. While these components are not unique to mobile ad hoc networks (MANETs), they do become more difficult to optimize when nodal mobility, dynamic network membership, and unstable links are introduced into the network [15]. Depending on the speed of the mobile nodes (MNs), mobility can be classified into three categories: static, low mobility, and high mobility. The management layer of such a network should be able to take into account any of these three cases or combination thereof. In the case of low mobility, the steady-state performance should be optimized since incidental updates (e.g. for route discovery) can unnecessarily consume resources. For high-mobility networks, resource consumption, and delay due to route maintenance are important limiting factors [16].

Centralized network management architectures fail to provide effective scalability in MANETs. In the last few years, a distributed decision-making scheme [17] has been introduced to address these concerns. In this proposed scheme, nodes may only be aware of their own neighbors and have no understanding of the size and extent of the network. Finding a mechanism that can deal with particular challenges associated with distributed decision making in ad hoc networks is certainly non-trivial.

Recent research and present existing mechanisms do not provide a particularly good fit for a certain environment and an alternative paradigm is needed for a particular scenario such as field-based anycast routing using temperature field [18], a bidirectional abstraction to routing protocols for the asymmetric mobile network [19], network science-based approaches for military applications [20] and context-aware protocol engine [21]. To cope with these demands management solutions based on cross-layer design [10–13] are necessary for efficient utilization of the limited resources in future wireless networks.

DOI: 10.1002/nem

2.2. Challenges in MANETs

Several papers have classified MANET routing protocols in terms of their behavioral characteristics and applicability. We largely adhere to the standard convention of classification, namely *flat*, *hybrid*, and *geographically oriented* protocols. Routing protocols which are not organized in any hierarchical fashion are commonly referred to as flat routing protocols [22]. Flat routing schemes have three main classifications: *proactive* (table-driven, e.g. Optimized Link State Routing Protocol (OLSR) [23]), *reactive* (demand-driven, e.g. Dynamic Source Routing Protocol (DSR) [24], Ad Hoc On-Demand Distance Vector Protocol (AODV) [25]), and *hybrid* (e.g. Zone Routing Protocol (ZRP) [26]).

DSR is a reactive protocol which uses source routing as a central mechanism [20]. When a route request (RREQ) is made by a particular node, it uses the destination route stored in its local route cache to send the data packet. Nodes along the path aggressively cache the path from the source node's cache (which is embedded in the packet itself). However, if the node does not have the required route information cached, the route discovery process is initiated by flooding the network with route request packets. The request packets propagate throughout the network until they reach the destination node, or a node which has a cached path to the destination. The end node then sends a route reply with the newly discovered route source information back to the source node which then caches the path for future source routing. Further, destination nodes respond to all route request packets, thereby increasing the amount of aggressive caching taking place throughout the network.

The AODV routing protocol is another routing protocol for multi-hop wireless networks, similar in nature to DSR. AODV shares DSR's on-demand characteristics in that it also discovers routes on an as-needed basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. There is only one table entry per destination in any particular node's routing table. AODV uses sequence numbers to determine the 'freshness' of routes in the various routing tables. Without source routing, AODV relies on routing table entries to propagate the route reply (RREP) back to the source and, subsequently, to route data packets to the destination.

An important feature of AODV is the maintenance of timer-based states in each node with parameters (e.g. Active Route Timeout, Hello Interval) regarding utilization of individual routing table entries. A routing table entry is expired when not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes which use that entry to route data packets. These nodes are notified by route error (RERR) packets when the next-hop link breaks. Each predecessor node, in turn, forwards the RERR to its own set of predecessors, thus effectively erasing all routes containing the broken link.

However effective AODV may be [27], it suffers from the following drawbacks in a mobile network environment:

(a) It does not frequently update the route to the destination.
(b) Due to the large Hello Timer values, there appears to be a periodicity in the route request generation which, in turn, can be attributed to poor link failure detection.
(c) It determines the 'best effort' shortest path, i.e. the shortest successful path.

In the case of proactive protocols, such as OLSR, there are sufficient exchanges of routing information to result in near-optimal routes. Therefore, OLSR is more resistant to packet drops at the MAC layer. However, one of the drawbacks of OLSR is that it generates routing traffic independent of application traffic [28]. Due to the higher routing overhead in proactive routing protocols, we chose the reactive routing protocol, AODV, in our cross-layer approach and focus on enhancing the protocol performance with a self-configuration mechanism.

To identify the trade-off issues when using reinforcement learning, it is crucial to study the impact factors of routing protocols, traffic load and mobility, and their impact on service delivery. A statistical design of experiments could be beneficial to identify both main effects and interactions of factors that best explain the response variables [29]. However, in this paper the focus is on reconfiguring the critical timers, namely, Hello Interval and Active Route Timeout (ART), to enhance network performance by dynamic context exchanges in heterogeneous networks.

DOI: 10.1002/nem

2.3. Existing protocol parameter tuning solutions

Parametric tuning of routing protocols, and AODV in particular, has been of increasing interest in recent years [30–36]. Vadde and Syrotiuk [30] explore the sensitivity of AODV protocol parameter tuning in conjunction with network performance metrics. They show that nodal mobility is the major contributing factor to end-to-end delay, due to frequent route re-establishing processes. Additionally, they explore the fact that the packet arrival rate is the main contributing factor to changes in throughput, and the fact that the interactions of network events and timers, such as the ACTIVE_ROUTE_TIMEOUT, directly affect the generation of performance-degrading protocol overhead packets.

Other works have proposed solutions on how to concretely modify these protocol parameters to improve network performance. Xing *et al*. [33] propose a modification to AODV, DA-AODV (Dynamically Adjusting AODV), which measures network diameter to limit the scope of network max hop count. Leveraging the RREQ and RREP packets to carry this information, the max hop count indicates the number of max hops a packet can take from a source to a destination node. Network max hop count is calculated on a per-node basis, indicating the max hops on a path for a particular source/destination pair. The authors add a new routing table parameter, Net_Diameter, to denote the max hop count value for each node's routing table entry. When a routing table entry changes, Net_Diameter is compared against every table entry to ensure that it is set equal to the max hop value. When a node wishes to send a RREQ message to a particular destination, the source node first compares its max hop count with the Net_Diameter value, setting either of the two values to the greater of the two values, thereby allowing the Hello packet to be broadcasted over the whole known network. By increasing the range of network discovery, the authors show a reasonable reduction in end-to-end delay and route error packets, due to the enhanced routed discovery mechanisms.

Li and Han [32] propose a multi-hop wireless protocol tuning approach which uses nodal mobility characteristics to determine changes in settings in AODV. Specifically, the authors chose to tune the SEND_HELLO_INTERVAL based upon feedback of reply and acknowledge packets. The algorithm used is as follows:

```
Procedure Recv(P)
  begin
    p_type = P.type();
    if p_type == REP or ACK
      intr = Calculate_LAST_REP_ACK(P)
      if intr decrease
        SEND_HELLO_INTERVAL += Δt
end;
```

The value of *intr* represents the time elapsed between the current RECV or ACK packet being analyzed and the last time a RECV or ACK packet was analyzed. The Δt value is added to the SEND_HELLO_INTERVAL based upon whether or not the value of *intr* decreased, which denotes an increase in frequency of REP or ACK packets. When combining this modification with the calculation of average neighbor's speed (sending more RREQ packets if neighbor speed increases), the authors were able to increase the lifetime of the network routes significantly over when using standard AODV routing.

Tan and Seah [31] propose a solution whereby nodal mobility is used to tune the frequency of Hello messages. Before a Hello message packet is transmitted, the nodal mobility is inferred by comparing the current neighbor table of the node to the previous neighbor table of the node when the last Hello transmission occurred, looking for the number of new neighbors and the number of neighbors still in the table from the last iteration. If the change count (new neighbors + neighbors left count) is zero, a so-called Deviation_Fraction value is set to 1.25. If its greater than 5, Deviation_Fraction is set to 0.75. If the change count is 1 through 5, the Deviation Fraction is set to 1. The Hello Interval is then set equal to HELLO_INTERVAL * Deviation_Fraction. This process continues for each interval expiration. The proposed solution results in a 20% reduction in Hello packet overhead, and increase in packet delivery ratio due to longer lasting stable routes.

DOI: 10.1002/nem

We [34,35] proposed a general framework for autonomic network management in heterogeneous network environments. Specifically, we [34] proposed using Q-learning to load-balance OSPF traffic to avoid link congestion. This was achieved by having network agents observe and track queue length of nodes in various routes. The reinforcement learning agent would then compute and track these queue lengths over time to determine the optimal routes to facilitate network-wide load balancing, and resulting in dramatic decreases in packet loss. We [35] provided initial experimental results of leveraging reinforcement learning to improve AODV routing protocol performance over standard AODV by tuning the Hello Interval. This work was a cursory study of how sensitive AODV would be to parameter modification in a heterogeneous environment. However, this work did not compare these modifications of AODV to any existing AODV parametric tuning solutions, nor was the learning rate adaptation explored. Moreover, the approach we use in this paper for measuring application performance is more tightly coupled to the actual realities of the network behavior. By using the max observed end-to-end delay instead of a predefined max allowable end-to-end delay, as part of the feedback mechanism to determine which protocol parameters to tune, we now observe better results.

For the purposes of this work, we chose to compare the performance of AODV-Q to the modified protocols Modified-AODV (Mod-AODV) [31] and Optimized-AODV (Opt-AODV) [32], as well as standard AODV. Since both solutions dynamically update the Hello Interval, the modifications proposed [31,32] are more closely related than other existing proposed protocol enhancements. In the next section, we cover the details regarding our proposed protocol tuning enhancement, aided by autonomic/cognitive management principles, leveraging Machine Learning for better optimization and performance.

3. SELF-CONFIGURATION FOR AODV

3.1. Self-configuration parameters for AODV

In AODV, the Hello Interval and ART values are important parameters to cope with link failures caused by network dynamics. However, these timers are typically set in a trial-and-error manner or set at a constant value, which can lead to great inefficiencies with respect to performance [30]. We apply the Q-learning technique, leveraging cross-layer performance, to identify any possible performance implications involving these timers.

AODV uses Hello messages: periodic local broadcasts by a node to inform each mobile node in its neighborhood [6]. The Hello messages may list other nodes from which a mobile node has heard, thereby yielding a broader knowledge of network connectivity. Setting the optimal Hello Interval is a crucial aspect of maintaining network connectivity.

The route discovery process of AODV allows the intermediate nodes to store a route's state between the endpoints [37]. Each node keeps this state for a length of time given by the ART parameter. Every time the route is used, the timer is reset back to the ART value. The ART is a static parameter that defines how long a route is kept in the routing table after the last transmission of a packet on this route. This parameter is arbitrarily set to 3 seconds. Comparatively speaking, DSR keeps a similar time-out parameter, denoted *route cache timeout*, but with a value set at 300 seconds.

The use of static values does not take into account either the actual lifetime of the path or the scale of the time correlation between two successive connections between the same endpoints. Finding an optimal value requires a balance between choosing a short ART that causes a new route discovery, even if a valid route is still available, and choosing a long ART, which risks sending packets on an invalid route. In the first case, the cost is the initiation of a new route discovery that could be avoided, and in the second case it is the loss of one or more packets and the initiation of a RERR process instead of a new route discovery phase.

3.2. Q-learning in MANET routing

The varied features of wireless networks lead to many optimization problems with respect to achieving specific performance objectives. The idea of applying reinforcement learning to routing in networks

DOI: 10.1002/nem

was first introduced by Boyan and Littman [38]. They showed that the Q-learning [5] based routing can compete with the shortest path algorithms, without prior knowledge of the network topology. Q-learning has also been applied to routing in ad hoc networks [3]. Collaborative reinforcement learning (CRL) was also introduced and evaluated [7] as a self-organizing technique for building a MANET routing protocol. To the best of our knowledge, no existing routing scheme with reinforcement learning takes into consideration optimization goals (routing path length, load balancing, consistent link management, and aggregation) combined with a cross-layer approach.

3.3. Cross-layer, autonomic network management architecture

The cross-layer architecture for our proposed cognitive management framework is explained in Figure 1. **F1** One of the main advantages of cross-layer design is to make protocols aware of current network state in a localized but distributed fashion. By introducing network and application layer context to the network management agents, this improves the higher-level processes of the middleware, allowing our QLS mechanism to exploit broader knowledge of the network state, and improve overall system performance.

Other proposals for implementation of cross-layer information exchange have been put forth in the current literature. These proposals can be categorized in three main groups [13]: (a) direct communication between layers; (b) a shared database across the layers; and (c) completely new abstractions. Specifically, we present the cross-layer model which sets performance expectations relative to performance observed thus far, in support of application-layer performance optimization, calculates reward and penalty values in the middleware layer, and uses those values to inform protocol parameter tuning decisions at the network layer. Figure 2 is an illustration of the cross-layer design approach which conveys how QLS in the middle- **F2** ware layer can interact with the other reconfigurable modules in the network layer. The following steps describe the detailed workflow of this management scheme.

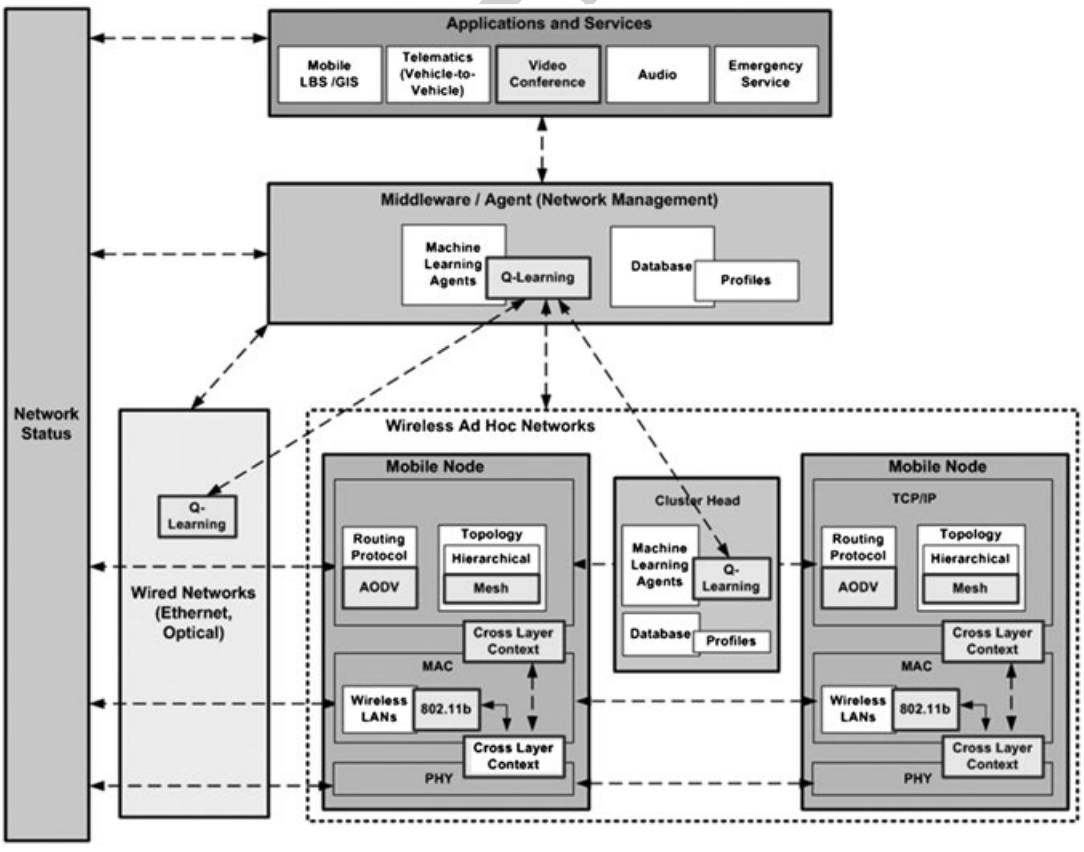


Figure 1. Overall cognitive network architecture for distributed optimization in heterogeneous networks.

DOI: 10.1002/nem

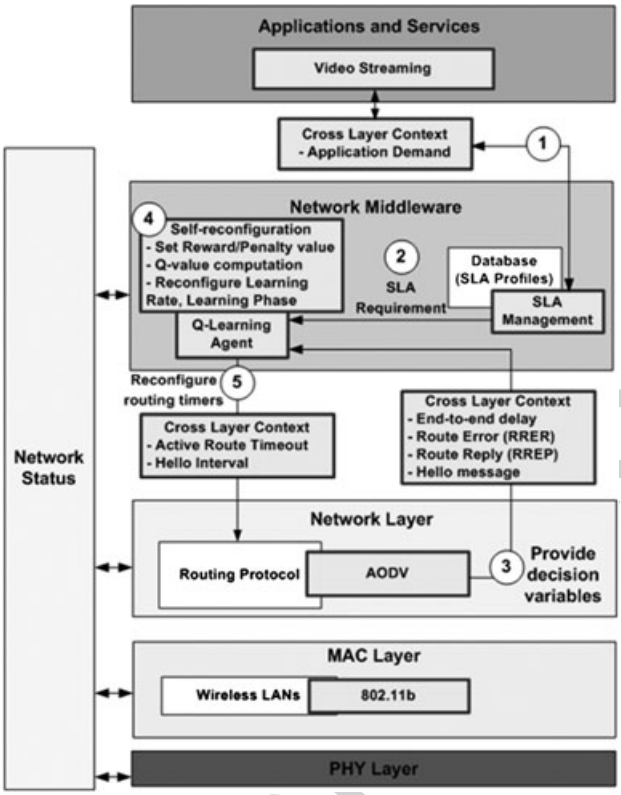


Figure 2. Cross-layer design for routing optimization in a mobile node.

Step 1: The management module at the middleware layer gathers application demands and determines the corresponding requirements (in this case, minimization of ETE delay).
Step 2: The Q-learning agent in the middleware layer receives the performance requirement in the form of reward and penalty formulas.
Step 3: At the network layer, the AODV protocol provides the Q-learning agent with the decision variables, including end-to-end delay, RERR and RREP.
Step 4: The Q-learning agent decides which action should be taken to enhance performance.
Step 5: The Q-learning agent reconfigures the routing parameter(s) accordingly (Hello Interval and ART).
Step 6: Loop back to Step 1 to iteratively observe the effects of environmental actuation and reformulate decision parameter values for Q-learning agent based upon new observations.

Table 1 summarizes the proposed autonomic management approach, with respect to qualitative analyses of the challenges faced in MANETs. **T1**

4. Q-LEARNING BASED SELF-CONFIGURATION (QLS) FOR AODV

4.1. Q-learning

In Q-learning [5], each time an action a is executed, an agent receives an immediate reward r from the environment. The agent then uses this reward and the expected long-term reward to update the Q-values, which in turn influences future action selection. Its simplest form, one-step Q-learning, is defined as follows:

$$Q(s, act) = (1 - \alpha)Q(s, act) + \alpha \max_{act'}(Q(s', act')) \quad (1)$$

DOI: 10.1002/nem

Table 1. Cross-layer reconfiguration in wireless mesh network management.

Key challenges	Previous approaches	Our approach
Route management	• Routing protocols directly manage the path	• Reconfigure (tuning) the routing protocols to discover more-optimal paths
Management overhead	• Relatively fixed management overhead (e.g. proactive protocols (OLSR) have almost fixed overhead)	• Autonomously (adaptively) change the control message frequency according to the change of application demands
QoS management	• Resource management is limited to network layer	• Cross-layer resource management from network layer to application layer
	• Mostly static, flat QoS support	• Dynamic QoS support by reconfiguring with Q-learning agent
Overall performance	• Not aware of application demands	• Support performance by our cross-layer approach. When reconfiguring routing parameters, end-to-end delay is used to determine the reward/penalty values as performance is more heavily dependent on end-to-end delay

where α is the learning rate ($0 < \alpha \leq 1$), which models the rate of updating Q-values. The variable s represents the present state observation and s' the new state which the algorithm will explore. The variable *act* represents the action which led to state s and act' the action that leads to s'. The Q-value itself is a numerical value which represents the current state action pair. In this context, the state is the current performance of the network and the action is how to tune various protocol parameters. Finally, $Q(s,act)$ is the Q-value derived from the current state–action pair, and $\max_{act'} Q(s',act')$ is the max Q-value (reward) that can be obtained from next state s' over all possible actions act'. As a model-free reinforcement learning technique, Q-learning requires no knowledge about the underlying reward or transition mechanism; thus it is applicable to the problem of learning routing strategy in ad hoc networks, where explicit state-space mapping can become computationally cumbersome. Specifically, mapping out the possible permutations of networks settings, nodal mobility, and traffic interactions would be potentially infeasible, and Q-learning allows us to avoid this task by exploring the state space of local state–action pairs without globally mapping it.

4.2. *Q-learning-based self-configuration*

In our implementation of AODV-Q, each node has two Q-values: Q_{penalty} and Q_{reward}. Q_{penalty} denotes the penalty Q-value for unstable network status, which makes the node take the action of decreasing ART and Hello Interval. Q_{reward} represents the stability reward of the network, which will make the node take the action of increasing ART and Hello Interval. With respect to the Q learning calculation:

$$Q_{\text{penalty}} = (1-\alpha)Q(s,act)_{\text{penalty}} + \alpha\, Q(s',act')_{\text{penalty}} \tag{2}$$

$$Q_{\text{reward}} = (1-\alpha)Q(s,act)_{\text{reward}} + \alpha\, Q(s',act')_{\text{reward}} \tag{3}$$

In AODV-Q, each node makes its self-configuration decision based on the local routing information, represented as the two Q-values which estimate the quality of the alternative actions. These values are updated each time the node receives a RREP packet. The reward value is given by

$$\text{Reward} = Q(s',act')_{\text{reward}} = n(1/\text{ETE}_t) \tag{4}$$

when a ROUTE REPLY packet reaches the source and there is a path from the source to the destination. The thinking behind this calculation is that the reward should reflect the magnitude of improvement observed, and therefore the smaller the observed delay, the larger is the reward value. Conversely, the penalty value is calculated as

DOI: 10.1002/nem

$$\text{Penalty} = Q(s', act')_{\text{penalty}} = n(\text{ETE}_t/\text{ETE}_{\max}) \quad (5)$$

when a ROUTE ERROR packet is generated due to a broken route, or a ROUTE REPLY packet reaches the source and there is no path from the source to destination. This formulation ensures penalty values always range between 0 and 1, yielding smaller penalties the further the current end-to-end delay is from the max end-to-end delay observed thus far.

In equations (4) and (5), ETE_t is the current end-to-end delay; $\text{ETE}_{\max}$ is the maximum end-to-end delay observed thus far. For every RREP message, the delay is measured between the destination and the source. In the event that the destination cannot be reached, the delay is the amount of time the RREP takes between the final node processing the RREQ and the source node. Whenever a node receives an RREP, it captures the ETE delay for that packet at time t (ETE_t). If that value is larger than the current ETE_{max}, it sets $\text{ETE}_{\max}$ equal to ETE_t. The value n denotes the normalization constant, which is set to 1.0 for the penalty calculation, and 0.10 for the reward calculation, to ensure a range of values between 0 and 1 for both. The end-to-end delay is used to affect the reward and penalty because network protocol performance has been observed to be more tightly coupled with the ETE delay metric than other network performance metrics [30]. The algorithm for calculating the penalty/reward values and updating the ART and Hello Interval is as follows:

1. *Intialize $Q_{penalty}$ and Q_{reward} values to 0 and α to 0.1.*
2. *For a period of time ('learning_phase_duration' = 30 seconds), increase or decrease the ART and Hello Intervals by 2 with equal probability. Q-values are calculated but not yet used to determine increase/decrease.*
3. *When receiving a successful ROUTE REPLY message, calculate the reward value, update Q_{reward}, and GOTO step 4.*
4. *When receiving an unsuccessful ROUTE REPLY message or ROUTE ERROR message, calculate the penalty value, update $Q_{penalty,}$ and GOTO step 4.*
5. *If $Q_{reward} > Q_{penalty}$, decrease the ART and Hello Interval by 1, ELSE Increase each value by 1.*

To avoid overly aggressive changes in the ART or Hello Interval values, we initialize learning rate to 0.1. Future research will entail an investigation of the effects of dynamic learning rate tuning in response to observed performance thresholds. Section 7 conveys cursory results of experimentation with respect to an adaptive learning rate scheme.

Consider an example multi-hop wireless network containing several nodes, including mobile nodes (MN) A, B and C. Using the above described Q-AODV modifications, if MN A receives an RRER then node A is more likely to decrease ART based on its Q-learning agent. MN B is going to decrease the Hello Interval because MN B receives RREP indicating no valid route exists. But MN C may increase its Hello Interval when MN C receives RREP of successful route discovery. By virtue of distributed decision making, the different nodes on a given path may have different timer values, allowing the intermediate nodes, which have different mobility patterns, to quickly and reactively reconfigure their routing parameters. This enhanced reactivity from our cognitive framework improves the stability of the generated routes, as will be illustrated in the ensuing discussion of results.

5. SIMULATION ENVIRONMENT

The performance of our network system has been evaluated with the NS-2 simulation tool [39]. As shown in Table 2, the simulation network is defined in a flat terrain of 2000 × 2000 m with 100 mobile nodes, three MANET GWs, two BGP routers, and six fixed servers. Table 2 displays the summary of NS-2 simulation parameters. At the physical and data link layers, the 802.11b standard was used for analysis. The main purpose of the simulation scenarios was to provide a framework to compare the performance of AODV-Q, the solutions previously proposed [31,32], and standard AODV protocols. Results were averaged over 20 runs for each protocol, for each max pause time. **T2**

The traffic models used to gather the simulation results consists of constant bit-rate (CBR) video conferencing and File Transfer Protocol (FTP) application traffic profiles. These traffic types were

DOI: 10.1002/nem

Table 2. Summary of NS-2 simulation parameters.

Simulation Parameters	Values
Simulation area	2000 m × 2000 m
Number of nodes	Mobile nodes (MNs) = 100 3 wireless gateway nodes 6 fixed nodes
Mobility model	Random waypoint Speed (m/s) = uniform (0, 15) Pause time (s) = 0, 60, 120, 180, 240
Wireless interface	IEEE 802.11b, 11 Mbps
Wireless transmission range	350 m
Traffic flows: CBR/UDP	5 MN sources ↔2 fixed server and 3 mobile client destinations Packet size: 1.5 kB Transmission rate = 100 pkts/s Application profile: lo-res video traffic
Traffic flows: TCP	5 MNs source ↔ 2 fixed servers and 3 mobile client destinations Packet size: 512 B Application profile: FTP traffic
Simulation time	10 min (for each run)

chosen for two reasons, the first of which was to have diversity in the type of transport protocol used over this network, to try to understand any performance implications observed from each. The second reason was for the rate of traffic generation, namely that FTP will try to send as much data as possible, as opposed to other applications, such as HTTP, which can have more variability in the rate of traffic generation. The more taxing the traffic source is on the available network resources, the better we are able to observe how the protocol enhancements will respond to the observed performance.

In this analysis, node mobility is assumed to be random (i.e. independently selected by each node using a uniform distribution) movement rather than group movement. The mobile nodes are assigned a maximum speed of 15 m/s. In the simulation scenarios, each mobile node changes its location within the network based on the 'random waypoint' model; i.e. the node randomly selects a destination, moves toward that destination at a speed not exceeding the maximum speed (15 m/s) and then pauses; this interval is known as pause-time. In order to calculate the impact of high mobility on the protocol overhead, pause-time ranged from 0 to 240 seconds in duration. It should be noted that a pause-time of zero represents the worst case scenario, in terms of high topological instability, as the mobile nodes are constantly moving during the simulation.

For the AODV-Q simulations, for the first 30 seconds each node randomly chooses actions decreasing or increasing Active Route Timeout and Hello Interval. During the simulations AODV-Q reconfigures Active Route Timeout between 3 and 10 seconds, and Hello Interval between 1 and 10 seconds. Table 3 conveys a summary of the results of the simulation, with the value of each performance parameter averaged across pause time runs, and percentage improvement over standard AODV listed to the right of the value in parentheses. **T3**

6. PERFORMANCE EVALUATION

The performance of AODV-Q was evaluated in terms of responsiveness, protocol overhead, and packet delivery ratio. The performance results are compared with those derived under the standard AODV routing mechanism, as well as solutions proposed previously (Mod-AODV [31] and Opt-AODV [32]).

6.1. Responsiveness

The network responsiveness resulting from decisions QLS applied to AODV was evaluated in terms of route discovery time and end-to-end delay. The route discovery time, measured in seconds, is the

DOI: 10.1002/nem

Table 3. Summary of experimental results.

Protocol	ETE delay (s)	Route discovery time (s)	Routing traffic (packets)	Route errors (packets)	UDP packet delivery ratio	TCP packet delivery ratio
AODV	0.104	0.846	89 068	10 013	0.764	0.89
AODV-Q	0.074 (−29%)	0.562 (−33%)	50 047 (−44%)	5 378 (−46%)	0.846 (11%)	0.93 (5%)
Mod-AODV	0.08 (−23%)	0.61 (−28%)	59 134 (−34%)	8 018 (−20%)	0.794 (4%)	0.92 (3%)
Opt-AODV	0.094 (−10%)	0.768 (−9%)	63 181 (−29%)	8 413 (−16%)	0.781 (2%)	0.91 (1%)

DOI: 10.1002/nem

measure of how long the protocol takes to determine a valid route once a request has been made. Figure 3(b) conveys the average measure of route discovery time for all four protocols. Opt-AODV **F3** tracks more closely with the standard AODV implementation with a 9% reduction in route discovery time, whereas AODV-Q and Mod-AODV offer consistent improvement in route discovery time (33% and 27% reduction respectively). The reduction in route discovery time can be attributed AODV-Q and Mod-AODV's more temperate approach to tuning the Hello Interval. Specifically, Mod-AODV uses adjusts the Hello Interval by increasing or decreasing its value up to 25% (choosing a deviation fraction of 0.75, 1, or 1.25), rather than immediately adding or subtracting values to the Hello Interval. While AODV-Q does add or subtract values to the Hello Interval, it does so using a thresholded machine-learning based approach, whereby previous Q-values are taken into account to prevent radical swings in interval values. This approach leads to a more stable and accurate depiction of network dynamics than the approach used by Opt-AODV, which adds or subtracts values based upon immediate observation.

End-to-end delay, measured in seconds, is the measure of the time taken for a packet to be transmitted from the source and received at destination node. As delay is a good measure of the fitness of routes being selected, end-to-end delay was used as another measure of network responsiveness due to the decisions made by the routing protocols. Figure 3(a) displays the results of end-to-end delay measurements for all four protocols. While all four protocols generally reduced delay as network stability improved with increased pause time, using standard AODV as the performance baseline, AODV-Q and Mod-AODV exhibited 29% and 23% reductions in delay on average, respectively, whereas Opt-AODV showed a 10% reduction. Overall, end-to-end delay reduction can be attributed to the fact that AODV-Q, Mod-AODV, and Opt-AODV are tending to reduce protocol overhead over time through tuning of the Hello Interval (see Figure 4a). Reduced traffic in the wireless medium allows **F4** the QLS scheme to realize a shorter queuing delay, resulting in shorter end-to-end delays.

6.2. *Routing overhead*

Control overhead is measured in terms of the number of control messages generated by the four routing algorithms. Figure 4(a) illustrates the number of routing control messages generated or relayed in the network. The standard AODV mechanism generates a greater number of control messages than does AODV-Q, Mod-AODV, and Opt-AODV, with reductions at 43%, 33%, and 29% from the standard AODV respectively. This, in turn, translates into a higher probability of lost control messages in AODV due to collisions in the wireless medium. Consequently, routing paths are less reliable under the standard AODV. All three protocols make reasonable progress towards overhead reduction, but the ability of AODV-Q to retain stable routes by tuning the active route timeout parameter yields further reduction in unnecessary route discovery traffic.

The Q-learning agent at each node self-configures the Active Route Timeout and Hello Interval according to the Q-value. Due to the distributed self-configuration of these parameters, the nodes send RREQs more appropriately to account for failed routes, improving the route freshness and the link

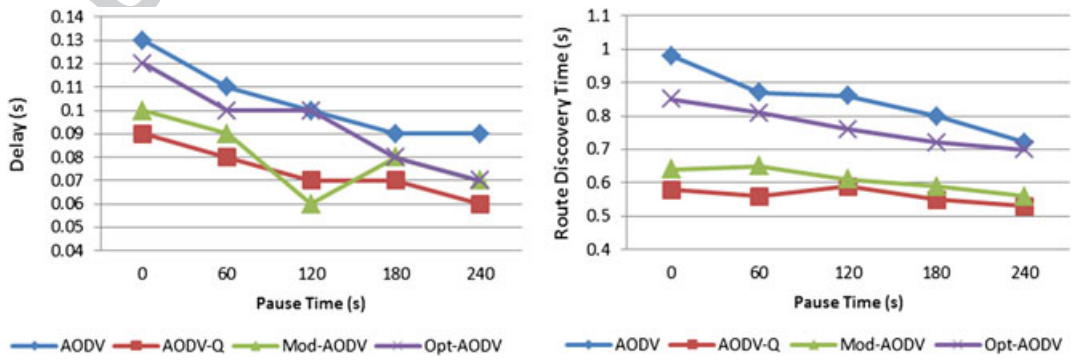


Figure 3. Overall network responsiveness with respect to (a) end-to-end delay and (b) route discovery time.

DOI: 10.1002/nem

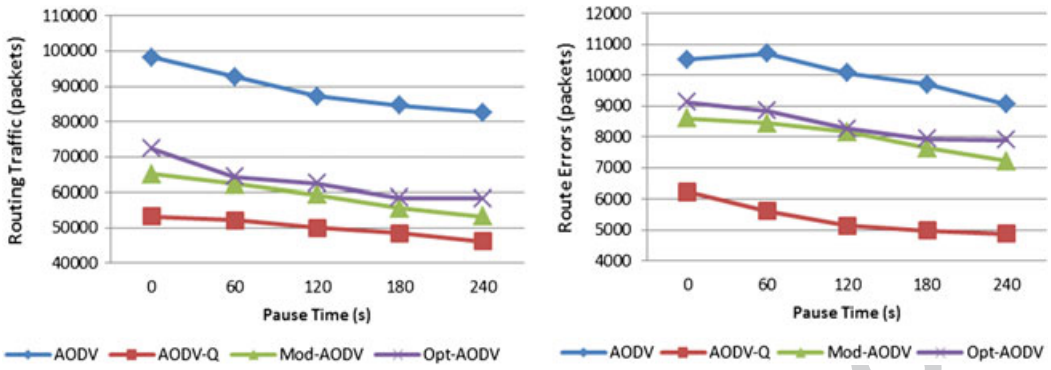


Figure 4. Resultant protocol overhead with respect to (a) routing traffic generated and (b) total route errors sent.

failure detection processes. The route error is evaluated as the average number of RERR packets per second. As shown in Figure 4(b), Mod-AODV and Opt-AODV yield a reduction of 20% and 16% in RERRs respectively, whereas AODV-Q yields a reduction of roughly 46% in RERR messages. While the previous two protocols have the ability to tune the Hello message interval to alleviate unnecessary congestion and produce better routes, AODV-Q has the added advantage of being able also to tune the active route timeout interval. This allows the protocol to hold onto routes longer than the other three protocols when network stability is perceived over time. This behavior tends to favor stable routes being used longer; hence the reduction in route errors over time.

6.3. *Packet delivery ratio*

The third criterion we use for evaluation is that of the packet delivery ratio, which is the number of transmitted packets divided by the number of received packets. The delivery ratio was measured with respect to the two traffic types that traverse the wireless portion of the network: constant bit rate (CBR) UDP video traffic, and TCP-based FTP traffic. Figure 5(a) conveys the results for packet delivery ratios for UDP video traffic. While Mod-AODV and Opt-AODV yielded a 4% and 2% improvement respectively, AODV-Q showed an 11% improvement in delivery of video traffic. The tendency of AODV-Q to hold on to more stable routes in addition to the reduction in protocol overhead contributed to the larger percentage improvement for video traffic. Video conferencing applications generate packets with very short inter-arrival times, prompting the Q-learning agent at each node to self-configure a shorter Active Route Timeout and Hello Interval. AODV shows large video conferencing packet delay variations due to its lack of efficiency and timeliness in finding new routes. AODV also shows larger end-to-end delay for video conferencing packets when the mobile nodes are highly dynamic, such as when the pause time is zero. F5

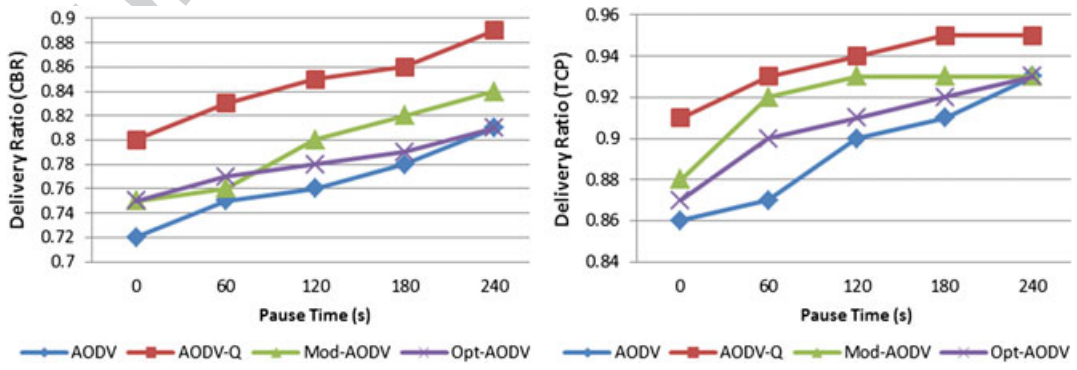


Figure 5. Packet delivery ratio with respect to (a) CBR-UDP video traffic and (b) TCP-based FTP traffic models.

DOI: 10.1002/nem

Figure 5(b) illustrates the packet delivery ratio for FTP traffic governed by TCP congestion control. In this case, the improvement upon the standard AODV implementation was less pronounced, with AODV-Q, Mod-AODV, and Opt-AODV yielding 5%, 3% and 2% performance improvements respectively. TCP's built-in congestion control can claim some responsibility for the higher packet delivery ratios of all protocols. Moreover, the constant bit rate video traffic was sent regardless of congestion or packet loss, accounting for most of the difference in performance between the two traffic models. However, in general, we do see a larger performance improvement with AODV-Q due to the increased longevity of stable routes.

6.4. Impact of learning rates

In the preceding experiments, we executed the simulations with a maximum learning rate ($\alpha = 0.1$). However, we devised a dynamic self-reconfiguration mechanism for adapting the learning rate in real time. The learning rate updates occur each time a node agent computes its respective Q-values. If one of the following two events occurs, then the learning rate $\boldsymbol{\alpha}$ is given by:

1. $\alpha = \alpha/2$: occurring when an agent receives a reward (the ROUTE REPLY packet reaches the source and there is a path from the source to destination);
2. $\alpha = \alpha * 2$: occurring when an agent receives a penalty (a ROUTE ERROR packet is generated or a ROUTE REPLY packet reaches the source but there is no path from the source to destination).

Since the above scheme yields an exponential increase/decrease in the learning rate value, we bounded the learning rate between values of 0.01 and 0.1, to prevent impractical values from occurring. Furthermore, we rounded the result at each possible computation to attain four possible values, namely 0.01, 0.2, 0.05, and 0.1. This scheme provides for rapid change in the learning rate without creating unnecessary skew in the values after repeated rewards or penalties. Such an unbounded approach could potentially leading to very high learning rates which exacerbate the instability of an already instable network, or leading to near-zero learning rate values which would prevent taking potentially important performance information into account in highly stable network scenarios.

To analyze the impact of the learning rate itself upon the observed performance enhancement of QLS, we present the results of simulation scenarios with three learning rates: 0.01, 0.05, and 0.1. First, we denote the various traffic types by index number, delineated by the mobility of the nodes involved:

Profile 1: Ethernet-to-MANET
Profile 2: MANET-to-Ethernet
Profile 3: MANET-to-MANET

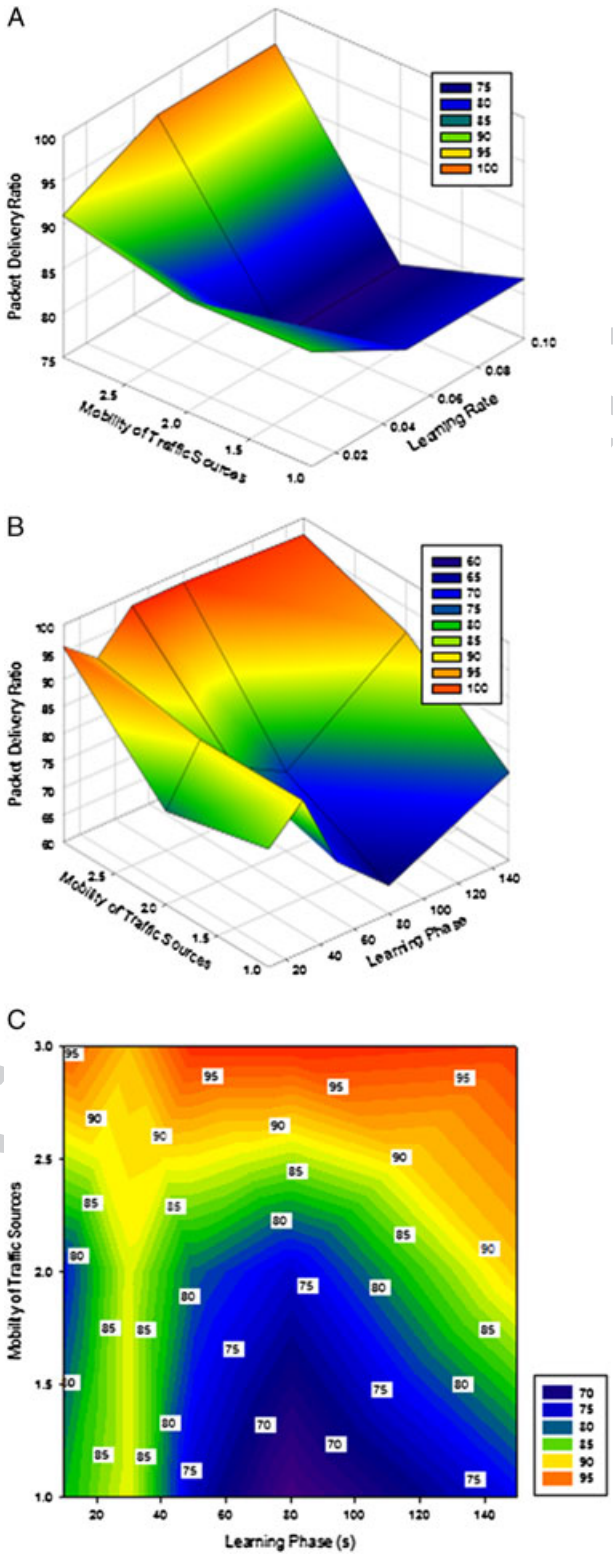
Figure 6(c) conveys the fact that higher learning rates improved the packet delivery ratio for **F6** Profile 3. However, we found that lower learning rates were more advantageous for Profiles 1 and 2, with respect to improving their localized delivery ratios. We infer from these results that higher learning rates are more advantageous in highly mobile environments, where quick adaptability can enhance network performance, as evidenced in Figure 6(a). Further, such high learning rates could be harmful to semi-static components of the network, in that either the source or the destination is relatively stable with respect to the rest of the network.

One candidate solution for the problem of optimizing the learning rate is to use Bayesian exploration [40] (to be explored in our future work), to tune and optimize learning rate values with respect to network performance. There further exists a need to investigate the optimization accuracy and the process of reward value assignment in the Q-value computation, in addition to the selection of correct parameters for self-configuration.

7. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we described a proposed framework for autonomously reconfigured network systems with a cross-layer approach. AODV-Q has been proposed to improve the performance of AODV,

DOI: 10.1002/nem



Q2 Figure 6. (a) Packet delivery ratio with respect to mobility and learning rate. (b) Packet delivery ratio with respect to mobility and learning phase duration. (c) 2D contour graph of packet delivery ratio with respect to mobility and learning phase.

DOI: 10.1002/nem

through the use of iterative network state observation. This is applicable to large heterogeneous networks, where the characteristics of the mobile nodes and application demands are different. We also presented experimental results. The performance results confirm that QLS dramatically reduces the protocol overhead compared to the standard AODV. AODV-Q achieves a higher packet delivery ratio while incurring shorter queuing delay. Specifically, with AODV-Q, it is possible to achieve shorter end-to-end delay while reducing the incidence of lost data packets. Therefore, the proposed autonomous self-configuration mechanism successfully improves the scalability and adaptability of the original AODV protocol in a heterogeneous network environment.

The work in this paper highlights some interesting and potentially important areas for future work, enumerated below.

7.1. Proactive vs. reactive network management

There is a fundamental trade-off between proactive and reactive routing protocols, in terms of delay and control overhead. A proactive routing protocol generates routing traffic independent of application traffic. Due to the higher routing overhead in proactive routing protocols (e.g. OLSR), we have chosen the reactive routing protocol, AODV, in our cross-layer approach and tried to enhance the protocol performance with QLS. However, it is inevitable that certain static networks will have especially high QoS demands which require the use of proactive routing. How to use proactive routing while minimizing the network-layer overhead is of key interest.

7.2. Performance evaluations in various network environments

It is important to verify the suitability of our approach to other heterogeneous networks (e.g. 3G, WiMAX, LTE and optical networks) with various traffic models and mobility models. It could be useful to provide results as a combination of larger networks and nodes.

7.3. Cross-layer design for heterogeneous application traffic with QoS guarantees

To guarantee the desired QoS levels, it would be critical to consider changes in user demands in the application layer. For consistent QoS support, the MAC layer could provide the essential feedback. The MAC layer could provide an indication of the network congestion level and achievable data rates; these calculations could, in turn, be used to determine whether the lower layer capability can meet upper layer requirements.

REFERENCES

1. Zhang H, Zheng Y, Khojastepour MA, Rangarajan S. Cross-layer optimization for streaming scalable video over fading wireless networks. *IEEE Journal on Selected Areas in Communications* 2010; **28**(3): 344–353.
2. Laufer R, Salonidis T, Lundgren H, Le Guyadec P. XPRESS: a cross-layer backpressure architecture for wireless multi-hop networks. In *Proceedings of ACM International Conference on Mobile Computing and Networking* (*MobiCom*), 2011.
3. Wang P, Wang T. Adaptive routing for sensor networks using reinforcement learning. In *Proceedings of the Sixth IEEE International Conference on Computer and Information Technology* (*CIT'06*), 2006.
4. Thomas R, Friend D, Dasilva L, Mackenzie A. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine* 2006; **44**: 51–57.
5. Watkins C, Dayan P. *Machine Learning*. Springer: Dordrecht, 1992; 279–292.
6. Royer EM. Chai-Keong T. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications* 1999; **6**: 46.
7. Dowling J, Curran E, Cunningham R, Cahill V. Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing. *IEEE Transactions on Systems, Man and Cybernetics, Part A* 2005; **35**: 360–372.
8. Bashar A, Parr G, McClean S, Scotney B, Nauck D. Machine learning based call admission control approaches: a comparative study. In *Proceedings of 6th IEEE/IFIP International Conference on Network and Service Management*, October 2010.
9. Brand C, Wolhuter R. Traffic class prediction and prioritization on a diversified IP network using machine learning. In *Proceedings of IEEE Global Communications Conference* (*Globecom '09*), Workshops, November 2009.
10. Conti M, Maselli G, Turi G, Giordano S. Cross-layering in mobile ad hoc network design. *Computer* 2004; **37**: 48–51.

DOI: 10.1002/nem

11. Jiang H, Weihua Z, Xuemin S. Cross-layer design for resource allocation in 3G wireless networks and beyond. *In IEEE Communications Magazine* 2005; **43**: 120–126.
12. Borgia E, Conti M, Delmastro F. Mobileman: design, integration, and experimentation of cross-layer mobile multihop ad hoc networks. *IEEE Communications Magazine* 2006; **44**: 80–85.
13. Srivastava V, Motani M. Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine* 2005; **43**: 112–119.
14. Yin L, Uttamchandani S, Palmer J, Katz R, Agha G. AutoLoop: automated action selection in the 'observe–analyze–act' loop for storage systems. In *Proceedings of Sixth IEEE International Workshop on Policies for Distributed Systems and Networks*, June 2005; 129–138.
15. Burbank J, Chimento P, Haberman B, Kasch W. Key challenges of military tactical networking and the elusive promise of MANET technology. *IEEE Communications Magazine* 2006; **44**: 39–45.
16. Faccin SM, Wijting C, Kenckt J, Damle A. Mesh WLAN networks: concept and system design. *IEEE Wireless Communications* 2006; **13**: 10–17.
17. Forde TK, Doyle LE, O'Mahony D. Ad hoc innovation: distributed decision making in ad hoc networks. *IEEE Communications Magazine* 2006; **44**: 131–137.
18. Baumann R, Heimlicher S, Plattner B. Routing in large-scale wireless mesh networks using temperature fields. *IEEE Network* 2008; **22**: 25–31.
19. Ramasubramanian V, Mosse D. BRA: a bidirectional routing abstraction for asymmetric mobile ad hoc networks. *IEEE/ACM Transactions on Networking* 2008; **16**: 116–129.
20. Kant L, Young K, Younis O, Shallcross D, Sinkar K, McAuley A, Manousakis K, Chang K, Graff C. Network science based approaches to design and analyze MANETs for military applications. *IEEE Communications Magazine* 2008; **46**: 55–61.
21. Garcia-Luna-Aceves J, Mosko M, Solis I, Braynard R, Ghosh R. Context-aware protocol engines for ad hoc networks. *IEEE Communications Magazine* 2009; **47**: 142–149.
22. Akyildiz IF, Wang X, Wang W. Wireless mesh networks: a survey. *Computer Networks* 2005; **47**(4): 445–487.
23. Clausen T, Jacquet P. Optimized Link State Routing Protocol (OLSR). *IETF RFC 3626*, 2003.
24. Johnson D, Maltz D. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, Imielinski T, Korth H (eds). Kluwer: Dordrecht, 1996; 153–181.
25. Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing. *RFC 3561*, IETF MANET Working Group, August 2003.
26. Haas ZJ, Pearlman M, Samar P. The Zone Routing Protocol (ZRP) for ad hoc networks. IETF Internet Draft, 2002.
27. Lee S-J, Royer EM, Perkins CE. Scalability study of the ad hoc on-demand distance vector routing protocol. *International Journal of Network Management* 2003; **13**(2): 97–114.
28. Klein A. Performance comparison and evaluation of AODV, OLSR, and SBR in mobile ad-hoc networks. In *Proceedings of 3rd International Symposium on Wireless Pervasive Computing* (*ISWPC 2008*), May 2008; 571–575.
29. Vadde KK, Syrotiuk VR. Factor interaction on service delivery in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications* 2004; **22**(7): 1335–1346.
30. Vadde KK, Syrotiuk VR. On timers of routing protocols in Manets. In *Proceedings of Third International Conference on Ad Hoc Networks and Wireless* (*AdHoc Now '04*), July 2004; 330–335.
31. Tan HX, Seah WKG. Dynamically adapting mobile ad hoc routing protocols to improve scalability. In *Proceedings of the IASTED International Conference on Communication Systems and Networks* (*CSN2004*), September 2004.
32. Li W, Han J. Dynamic wireless sensor network parameters optimization adapting different node mobility. In *Proceedings of IEEE Aerospace Conference*, March 2010; 1–7.
33. Xing T, Liu Y, Tang B, Wu F. Dynamic-adjusting AODV routing protocol based on max hop count. In *Proceedings of IEEE International Conference on Wireless Communications*, *Networking and Information Security* (*WCNIS*), June 2010; 535–539.
34. Lee M, Ye X, Marconett D, Johnson S, Vemuri R, Yoo SJB. Autonomous network management using cooperativ learning for network-wide load balancing in heterogeneous networks. In *Proceedings of IEEE Global Telecommunications Conference*, December 2008; 1–5.
35. Lee M, Marconett D, Ye X, Yoo SJB. Cognitive network management with reinforcement learning for wireless mesh networks. In *Proceedings of the 7th IEEE International Conference on IP Operations and Management*, 2007; 168–179.
36. Wang H, Cui L. An enhanced AODV for mobile ad hoc network. In *Proceedings of International Conference on Machine Learning and Cybernetics*, Vol. 2, July 2008; 1135–1140.
37. Richard C, Perkins C, Westphal C. Defining an optimal active route timeout for the AODV routing protocol. In *Proceedings of Sensor and Ad Hoc Communications and Networks* (*IEEE SECON*), 2005.
38. Boyan J, Littman M. Packet routing in dynamically changing networks: a reinforcement learning approach. In *Advances In Neural Information Processing Systems*, 1994.
39. NS-2 Network Simulator (August 2012). Available: http://nsnam.isi.edu/nsnam/index.php/User_Information [3 November 2012].
40. Dearden R, Friedman N, Andre D. Model based Bayesian exploration. In *Proceedings of Conference on Uncertainty in Artificial Intelligence* (*UAI '99*), 1999; 150–159.

DOI: 10.1002/nem

AUTHORS' BIOGRAPHIES

Dan Marconett is a PhD candidate in the Department of Computer Science at the University of California, Davis, as well as a computer scientist at Lawrence Livermore National Laboratory. From 2008 to 2011, he worked on the development of highly scalable network management software systems at Cisco Systems. His research interests include the integration of machine learning and network management systems, heterogeneous networks, and intelligent systems. He received his BS in computer science from California State University, Sacramento, in 2006, and MS in computer science from the University of California, Davis, in 2008.

Minsoo Lee is a chief research engineer of Convergence Lab at LG Electronics, where he leads industry-standard activities mainly in UPnP, DLNA, DECE, IEEE and home networking standards. He is a vice-chair of IEEE P2200, IEEE Standard Protocol for Stream Management in Media Client Devices. His research interests include machine intelligence, location-aware computing, home networks and network security. He was a Research Professor at Chung-Ang University HNRC (Home Network Research Center)–ITRC (Information Technology Research Center), supported by the MKE (Ministry of Knowledge Economy), Korea. From 2007 to 2009 he was a research scientist in the Department of Electrical and Computer Engineering at University of California Davis. He received the BS, MS and PhD degrees in the School of Electrical and Electronics Engineering from the Chung-Ang University, Seoul, Korea, in 2001, 2003 and 2007 respectively.

Xiaohui Ye received his MS degree in electrical engineering from Tsinghua University, Beijing, China, in 2005 and the PhD degree from the University of California, Davis, in 2012. He is currently a software engineer at Cisco System, Inc., San Jose, California, USA. His research interests include data center networks, high-performance computing, optical switching networks, heterogeneous networks, and machine learning.

V. Rao Vemuri received his PhD from the University of California, Los Angeles. He taught at Purdue University, West Lafayette, Indiana, State University of New York, Binghamton and at the University of California, Davis, where he is now an emeritus professor. He also worked at RCA, TRW and at the Lawrence Livermore National Laboratories, where he held the position concurrently with UC Davis. He was Editor-in-Chief of CS Press and an Associate Editor, IEEE TONN. As a Fulbright Lecturer he visited India in 2010. His research interests are in the areas of neural networks, genetic algorithms and machine learning. In his spare time he writes to popularize science.

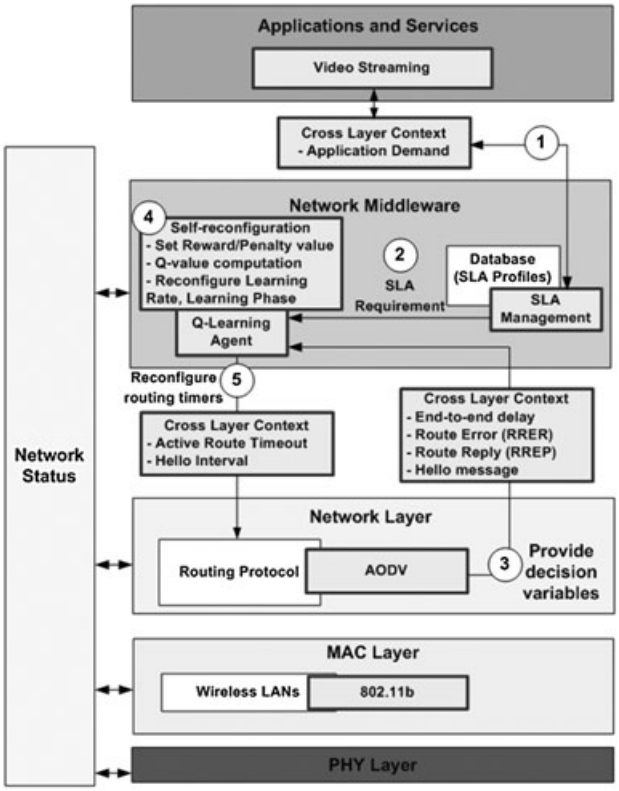
S. J. Ben Yoo received the BS, MS, and PhD degrees in electrical engineering from Stanford University, Stanford, California, in 1984, 1986, and 1991, respectively. He is currently Professor of Electrical Engineering with the University of California, Davis (UC Davis), where he is also the Director of the Center for Information Technology Research in the Interest of Society. His research interests include high-performance all-optical devices, systems, and networking technologies for the next-generation Internet; and architectures, systems integration, and network experiments related to all-optical label switching routers and optical code division multiple access technologies. Prior to joining UC Davis in 1999, he was a senior research scientist with Bell Communications Research, Morristown, NJ, leading technical efforts in optical networking research and systems integration. He was with Stanford University, Stanford, California, before joining Bell Communications Research. He also conducted research on lifetime measurements of intersubband transitions and on nonlinear optical storage mechanisms at Bell Laboratories, Murray Hill, New Jersey, and IBM Research Laboratories, San Jose, California, respectively. Professor Yoo is a Fellow of the IEEE Lasers and Electro-Optics Society and a Fellow of the Optical Society of America. He is a member of the Tau Beta Pi. He was the recipient of the DARPA Award for Sustained Excellence in 1997, the Bellcore CEO Award in 1998, and the Mid-Career Research Faculty Award (UC Davis) in 2004. He was the Co-Chair of the Technical Program Committee for APOC 2004 and APOC 2003, and was a member of technical program committees for several conferences. He was the General Co-Chair for IEEE LEOS Photonics in Switching Conference 2007. He was an Associate Editor for *IEEE Photonics Technology Letters*, and a guest editor for *IEEE/OSA Journal of Lightwave Technology* in 2005 and *IEEE Journal of Selected Topics in Quantum Electronics* in 2007.

DOI: 10.1002/nem

Research Article

Self-adapting protocol tuning for multi-hop wireless networks using Q-learning

Dan Marconett, Minsoo Lee, Xiaohui Ye, Rao Vemuri and S. J. Ben Yoo



This paper presents a closed-loop approach to tuning the layer three protocol based upon current and previous network state observations, specifically the Hello Interval and Active Route Timeout parameters of the AODV routing protocol (AODV-Q). Simulation results demonstrate that the self-configuration method proposed here demonstrably improves the performance of the original Ad-Hoc On-Demand Distance Vector (AODV) protocol, reducing protocol overhead by 43% and end-to-end delay 29% while increasing the packet delivery ratio by up to 11%.

Author Query Form

Journal: International Journal of Network Management

Article: nem_1819

Dear Author,

During the copyediting of your paper, the following queries arose. Please respond to these by annotating your proofs with the necessary changes/additions.

- If you intend to annotate your proof electronically, please refer to the E-annotation guidelines.
- If you intend to annotate your proof by means of hard-copy mark-up, please refer to the proof mark-up symbols guidelines. If manually writing corrections on your proof and returning it by fax, do not write too close to the edge of the paper. Please remember that illegible mark-ups may delay publication.

Whether you opt for hard-copy or electronic annotation of your proofs, we recommend that you provide additional clarification of answers to queries by entering your answers on the query sheet, in addition to the text mark-up.

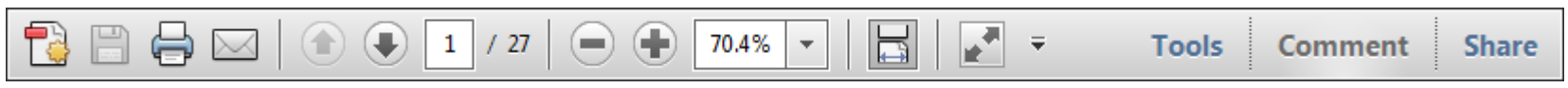
Query No.	Query	Remark
Q1	AUTHOR: I have renumbered the references in order of first citation.	
Q2	AUTHOR: Figure 6 – please check suitability for black and white printing.	

WILEY-BLACKWELL

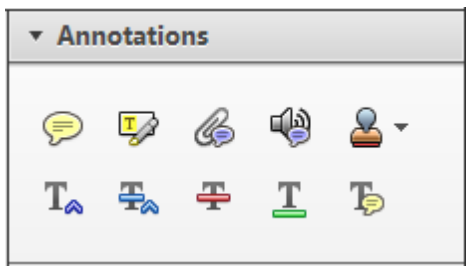
USING e-ANNOTATION TOOLS FOR ELECTRONIC PROOF CORRECTION

Required software to e-Annotate PDFs: Adobe Acrobat Professional or Adobe Reader (version 7.0 or above). (Note that this document uses screenshots from Adobe Reader X)
The latest version of Acrobat Reader can be downloaded for free at: http://get.adobe.com/uk/reader/

Once you have Acrobat Reader open on your computer, click on the Comment tab at the right of the toolbar:



This will open up a panel down the right side of the document. The majority of tools you will use for annotating your proof will be in the Annotations section, pictured opposite. We've picked out some of these tools below:

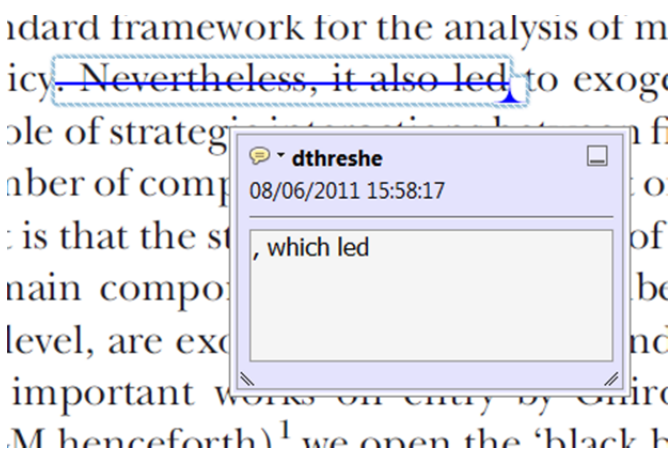


1. Replace (Ins) Tool – for replacing text.

Strikes a line through text and opens up a text box where replacement text can be entered.

How to use it

- Highlight a word or sentence.
- Click on the Replace (Ins) icon in the Annotations section.
- Type the replacement text into the blue box that appears.

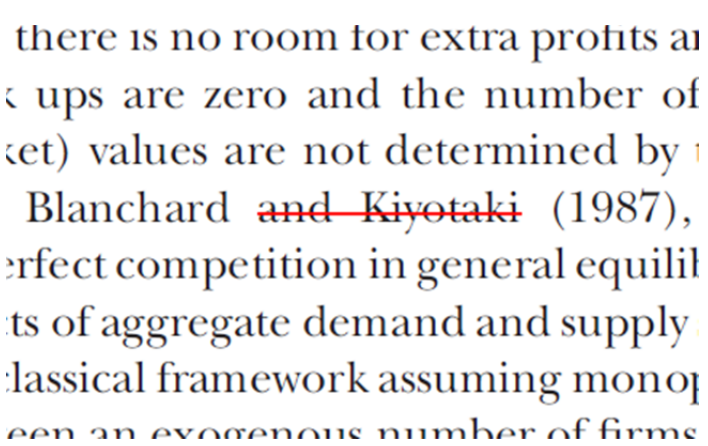


2. Strikethrough (Del) Tool – for deleting text.

Strikes a red line through text that is to be deleted.

How to use it

- Highlight a word or sentence.
- Click on the Strikethrough (Del) icon in the Annotations section.

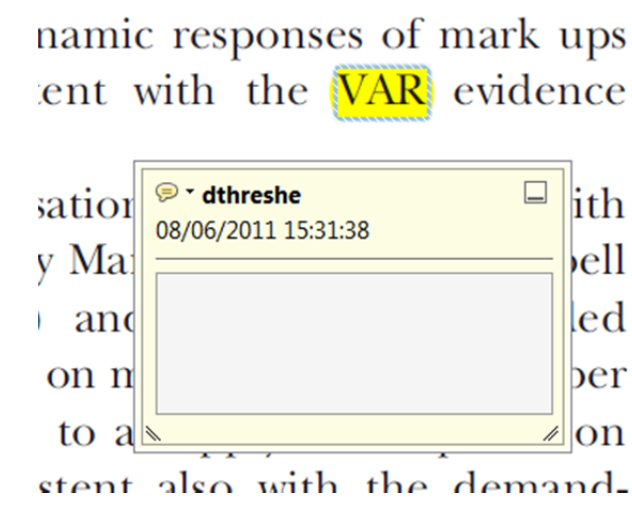


3. Add note to text Tool – for highlighting a section to be changed to bold or italic.

Highlights text in yellow and opens up a text box where comments can be entered.

How to use it

- Highlight the relevant section of text.
- Click on the Add note to text icon in the Annotations section.
- Type instruction on what should be changed regarding the text into the yellow box that appears.

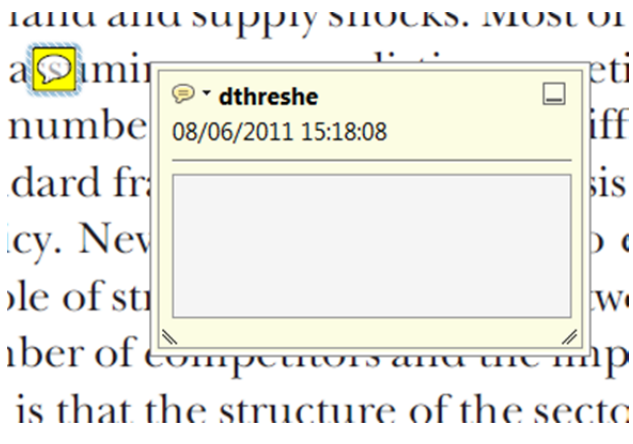


4. Add sticky note Tool – for making notes at specific points in the text.

Marks a point in the proof where a comment needs to be highlighted.

How to use it

- Click on the Add sticky note icon in the Annotations section.
- Click at the point in the proof where the comment should be inserted.
- Type the comment into the yellow box that appears.



WILEY-BLACKWELL

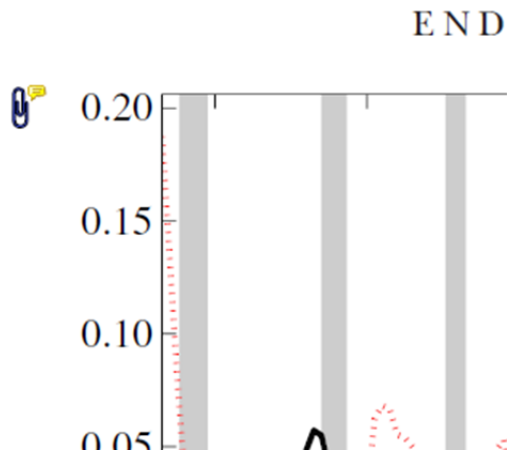
USING e-ANNOTATION TOOLS FOR ELECTRONIC PROOF CORRECTION

5. Attach File Tool – for inserting large amounts of text or replacement figures.

Inserts an icon linking to the attached file in the appropriate pace in the text.

How to use it

- Click on the Attach File icon in the Annotations section.
- Click on the proof to where you'd like the attached file to be linked.
- Select the file to be attached from your computer or network.
- Select the colour and type of icon that will appear in the proof. Click OK.

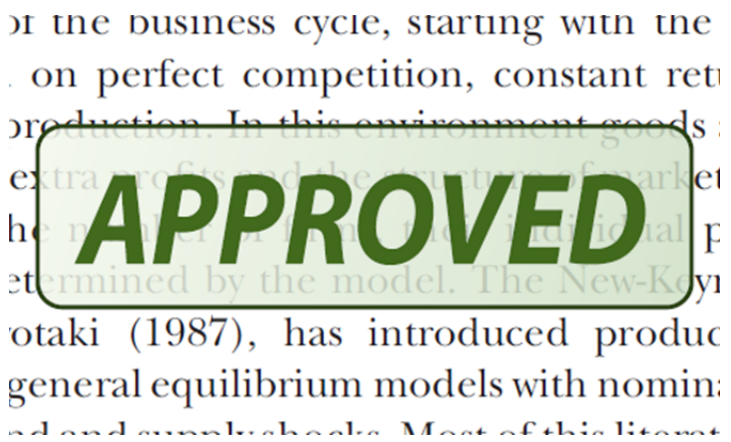


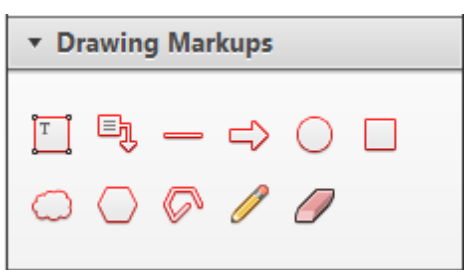
6. Add stamp Tool – for approving a proof if no corrections are required.

Inserts a selected stamp onto an appropriate place in the proof.

How to use it

- Click on the Add stamp icon in the Annotations section.
- Select the stamp you want to use. (The Approved stamp is usually available directly in the menu that appears).
- Click on the proof where you'd like the stamp to appear. (Where a proof is to be approved as it is, this would normally be on the first page).



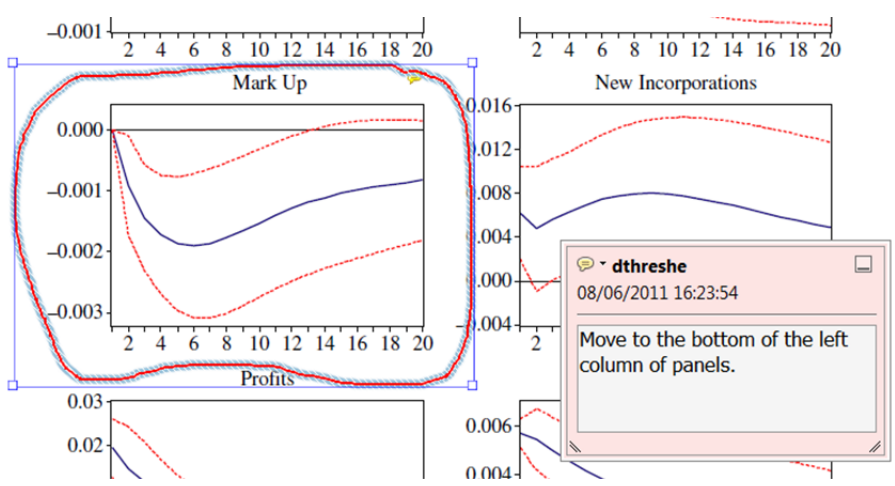


7. Drawing Markups Tools – for drawing shapes, lines and freeform annotations on proofs and commenting on these marks.

Allows shapes, lines and freeform annotations to be drawn on proofs and for comment to be made on these marks..

How to use it

- Click on one of the shapes in the Drawing Markups section.
- Click on the proof at the relevant point and draw the selected shape with the cursor.
- To add a comment to the drawn shape, move the cursor over the shape until an arrowhead appears.
- Double click on the shape and type any text in the red box that appears.



For further information on how to annotate proofs, click on the Help menu to reveal a list of further options: