INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, VOL. 7, 39–47 (1994)

STAGGERED MULTICAST PROTOCOL WITH COLLISION-FREE ACKNOWLEDGEMENT IN MULTIHOP SPREAD SPECTRUM PACKET RADIO NETWORKS

KWOK-WAH HUNG
Department of Computing, Hong Kong Polytechnic, Hung Hom, Kowloon, Hong Kong

AND

TAK-SHING YUM
Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

SUMMARY

The staggered multicast protocol for multihop spread spectrum packet radio networks is suitable for unicasting and broadcasting as well as multicasting. The common-header/transmitter-based spreading code is used for data packet transmission and the receiver-based code is used for acknowledgement packet transmission. By staggering packet transmission the protocol can significantly reduce broadcasting delay. A special addressing method and packet format are also designed to achieve collision-free acknowledgement and multicast capability. Simulation results show that the protocol provides better throughput-delay performance than the common-header/transmitter-based slotted ALOHA protocol.

KEY WORDS Communication protocol Packet radio networks

INTRODUCTION

Broadcasting is very often used for updating distributed databases and routeing tables in a communication network. The use of broadcasting in packet radio networks (PRNs) is facilitated by the broadcasting nature of the medium. When the network size gets larger, a multihop network involving packet relaying is usually used for connecting all stations. After a source station has broadcast a packet, a subset of its neighbouring stations needs to rebroadcast that packet. Since the radio channel is a multiaccess medium, a suitable method should be used to resolve the contention of the channel among neighbouring stations.

Very few studies of broadcasting in spread spectrum PRNs (SS-PRNs) are found in the literature. In spread spectrum communications, the use of spreading codes permits a receiver to extract a particular signal from many overlapping ones and adds another dimension to the design of PRNs. It is difficult to design a receiver that can simultaneously monitor all the codes. Therefore, there must be rules specifying which set of codes is to be monitored and which set of codes is to be used for transmission for each station. Four types of spreading code protocols can be identified: common code protocols, receiver-based protocols, transmitter-based protocols and hybrid protocols.[1]

The use of common code protocols facilitates the transmission of broadcast packets because all stations are tuned to the common code at all times. Transmitter-based protocols are also suitable for broadcasting but the receiver must know the transmission code used in order to receive. Receiver-based protocols are not suitable for broadcasting because a separate transmission is required for each receiver. It was suggested in Reference 2 that a fraction of the packet slots can be designated as broadcast slots using a common code while the transmission in the 'non-broadcast' slots could use a receiver-based protocol. Among the hybrid protocols, the common-header/transmitter-based protocol[1] looks most promising for broadcasting. Here the destination and source addresses are transmitted using the common code while the data is transmitted using a transmitter-based code. With this arrangement only the header of the packet is under contention, whereas the remaining data portion is collision-free owing to the use of a unique spreading code.

In broadcasting, as well as unicasting,* when a packet is received without checksum error, the receiver transmits an acknowledgement (ACK) packet back to the transmitter. The transmission is considered successful only when the ACK packet is received by the transmitter within a time-out interval. In conventional PRNs, the transmission of a relaying packet can serve as an implicit acknowledgement to the previous transmitter. As noted in Reference 2, implicit acknowledgement can also be used in transmitter-based and common code protocols for SS-PRNs with compatible transmission and routeing protocols. The same cannot be true for

---

* This is commonly referred to as point-to-point transmission or single destination transmission by some authors. We choose to call it unicast so that it can easily be distinguished from broadcast and multicast.

CCC 1074–5351/94/010039–09

Received 30 March 1992
Revised 27 June 1993

receiver-based protocols since the relaying packets are in different codes.

There are very few studies on acknowledgement algorithms for SS-PRNs. Sastry[3] examined the effect of acknowledgement traffic on the performance of slotted ALOHA-CDMA. He assumed that the system has a central station and separate frequencies are used for inbound and outbound traffic. This allows stations to transmit and receive at the same time. Lee and Silvester[4] studied the effect of acknowledgement on the performance of distributed single-hop SS-PRN using the slotted ALOHA protocol. The system considered uses a receiver-based spreading code for data transmission and a transmitter-based code for ACK packets. Stations cannot transmit and receive at the same time. They considered only the single destination transmission, and so ACK packets are always collision-free as transmitter-based code is used.

Acknowledgement in broadcasting is quite a different problem, as all neighbours of the transmitting station need to acknowledge. In conventional PRNs if the neighbours acknowledge at the same time, the ACK packets will collide. With the use of spreading code, ACK packet collision will still occur if common code or receiver-based code are used. There will be no ACK packet collision for transmitter-based code but the transmission of ACK packets by the neighbours must still be staggered in time as the station expecting ACK packets cannot monitor all the different codes of its neighbours simultaneously.

Broadcast and unicast protocols are usually designed separately. In this paper we design the staggered multicast protocol with collision-free acknowledgement, which is suitable for unicasting and broadcasting as well as multicasting in multihop SS-PRNs. The new protocol combines the feature of transmission scheduling and common-header/transmitter-based spreading code to allow overlapping of packet transmissions. The neighbouring stations of a broadcast source are scheduled to relay the broadcast packet in different 'header size' minislots. Hence the relaying packet transmissions can be staggered and the broadcasting delay is significantly reduced. A multicast tree found from the routeing table is used for global multicast. When a station receives a multicast packet, it is responsible for forwarding the packet only to destinations on the branch of the multicast tree spanning from that station. Thus no relaying packet will be duplicated to reaching the same destination. We also design a special addressing method and packet format to allow dynamic scheduling of ACK packets. Simulation results show that the new protocol provides better throughput-delay performance than the common-header/transmitter-based slotted ALOHA protocol in addition to the advantages of staggered relay broadcasting, collision-free acknowledgement and multicast capability.

SYSTEM MODEL

Let the locations of the stations in a multihop packet radio network be fixed and let the transmission range be the same and fixed for all stations. Let all stations within a station's transmission range be called the neighbours of that station. Each station is assigned a spreading code and an address. The address is globally unique, but the code is unique only among other stations within two-hop distance so that beyond a certain range the codes can be reused. There are many ways to assign codes to stations. One very efficient assignment algorithm requiring only a minimum number of codes can be found in Reference 5, and is used here for code assignment. Let all stations use the same frequency band for transmission. Stations therefore cannot transmit and receive at the same time.

The common-header/transmitter-based spreading protocol is chosen for data packet transmission. With this protocol each station is assigned a transmission code. In addition, there is a common code which is used by all stations for addressing purposes. The packet header is transmitted using the common code, whereas the remaining portion of the packet is transmitted using the transmitter code. A minislotted approach similar to that of Reference 1 is adopted. Let a slot be defined as the length of the packet header and let the packet length be in units of slots. We assume that the length of an ACK packet is smaller than a slot. This assumption will be justified when we discuss the acknowledgement protocol in the next section.

Stations are allowed to transmit only at the beginning of their assigned slots. Packet transmissions are scheduled in such a way that stations transmitting only in their assigned slots will not encounter conflicts. Here we use the newly designed fair and efficient scheduling algorithm in Reference 6 to do such transmission scheduling. This scheduling algorithm is found to give schedules that have the shortest cycle length, the smallest scheduling delay, the largest minimum transmission capacity and the same highest normalized network capacity when compared to two of the best scheduling algorithms in the literature.

Example

Figure 1 shows a 15-node network. An edge between two nodes indicates that the nodes are within the transmission range of each other. Table I shows the code assignment and transmission schedule produced by the corresponding algorithms in References 5 and 6. The length of the schedule cycle is eight slots, which is the minimum since station J has 7 neighbours. The number of codes required is also the minimum. In the table a letter T in the x row and y column indicates that station y is scheduled to transmit in slot x.

Table I. The code assignment and transmission schedule

Slot number	Station (code)														
	A (1)	B (6)	C (8)	D (5)	E (2)	F (3)	G (6)	H (3)	I (4)	J (1)	K (5)	L (4)	M (6)	N (7)	P (8)
1	T									T					
2					T										
3						T		T							
4	T								T			T			
5				T							T				
6		T					T						T		
7			T					T						T	
8			T					T							T

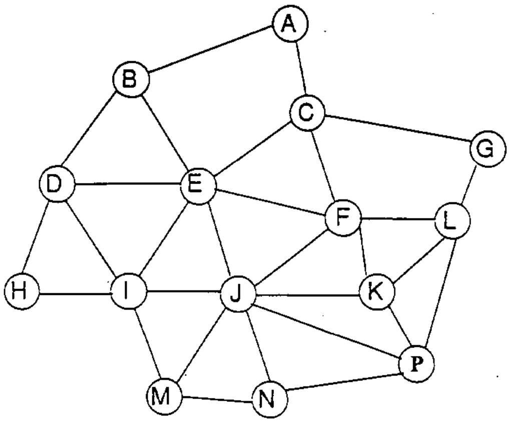


Figure 1. A sample network

THE STAGGERED MULTICAST PROTOCOL

Following the scheduling algorithm, stations in the network can transmit in different slots without conflict. However, such transmissions cannot be received by busy stations, either busy in transmitting or in receiving another packet. To make sure the destination does receive a packet correctly, some form of acknowledgement is required. For noisy channels, acknowledgement is needed even if the transmission scheduling is collision-free.

We use two types of acknowledgement packets. A positive acknowledgement packet (ACK) is returned when the target station receives the packet correctly. A negative acknowledgement packet (NAK) is returned when the received packet contains error. When the source station receives a NAK packet, it retransmits the packet immediately. When the target station is busy, no acknowledgement packet is returned and the source station will retransmit the packet after a random delay.

Acknowledging a broadcast transmission is more complex, since all the neighbours of the source station need to respond. If a subset of the neighbours fails to acknowledge, this subset will be the intended receivers when the packet is retransmitted. Transmission to a subset of neighbours is called local multicast. To accommodate local multicast the address field in the packet header needs to be expanded.

Packet format

The packet format is shown in Figure 2. It consists of three parts, header 1 followed by header 2 and the packet body. Header 1 contains the packet I.D., the receiver code bit-map and the transmitter's code number, and is transmitted using the common code. The packet I.D. is a globally unique number, which could be formed by appending the local clock time to the station address, for identifying different packets. The receiver code bit-map indicates which neighbours are in the reception list. Since the spreading code assigned to a station is unique among the station's neighbours, this code number is in fact a local address. If a neighbour with assigned code i is the intended receiver, the ith bit in the bit-map is set to 1. When a station with assigned code i receives a packet header with a '1' in the ith position of the bit-map, it tunes immediately to the transmitter's code to receive the rest of the packet. To

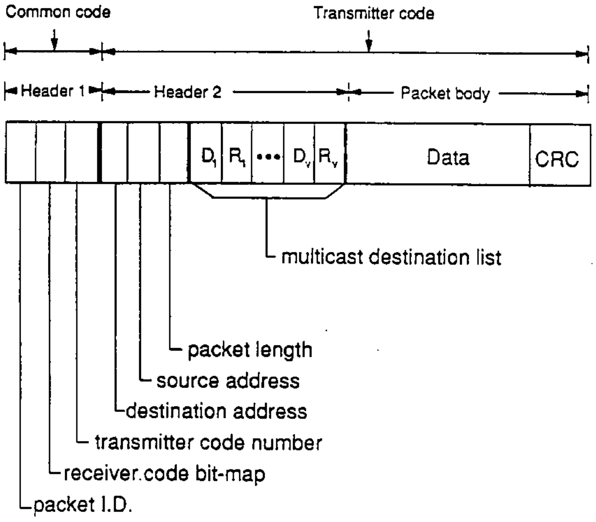


Figure 2. The packet format

illustrate, consider the network in Figure 1. If station F wants to multicast a packet to stations J, L and K, the 1st, 4th and 5th bits of the bit-map are set to 1. After noticing that the 4th bit in the address bit-map is 1, station L will tune its receiver to station F's transmitting code, or code 3, as indicated in header 1.

The length of the bit-map is equal to the total number of codes used in the whole network, which depends on the topology and especially on the maximum size of neighbouring groups in the network. Since our multicast protocol is to be used among fixed stations, which are usually the base stations in a mobile communication network, the number of neighbours and hence the bit-map length would usually be small. For example, in 50 network samples each containing 160 randomly located stations with maximum neighbouring group size of 27·38 averaged over the 50 samples, the average number of codes needed, and hence the bit-map length is only 29·60 when the code assignment algorithm in Reference 5 is used.

To keep the length of header 1 (and hence the slot size) short and fixed, other address information is placed in header 2 and is transmitted using the transmitter code. For unicast packets, this information includes the destination and source addresses. For broadcast packets, a special code of all '1's is used for identification in the destination address field. For multicast packets, a special code of all '0's followed by a list of multicast destinations are needed. The multicast destination list has the form $D_1R_1D_2R_2D_3R_3 \ldots D_VR_V$ where V is the total number of destinations, D_i is the ith destination address and R_i is the assigned code of the relaying station responsible for forwarding the packet to destination D_i. As variable length packets are allowed in the network, a packet length field is required. The packet body contains the data and a cyclic redundancy check (CRC) field, and is transmitted using the transmitter code.

Global multicast

For global multicast, fixed routeing with routes defined by a routeing table is assumed. All stations are also assumed to have the same routeing table. The paths of a multicast packet from the source station to the final destinations form a multicast tree (found from the routeing table). The source station first multicasts the packet to all its neighbours on the routeing tree. When a multicast packet is received by a station, that station might have to relay the packet with an updated multicast list. The updated multicast list contains only destinations on the branch of the multicast tree spanning from that station. Broadcast packets are treated as multicast packets with the multicast tree spanning all stations in the network. Note that the receiver code bit-map in header 1 is used for local multicast, whereas the multicast list in header 2 is for global multicast. Obviously, the last hop of all global multicasts can be treated as a local multicast.

Dynamic scheduling of receiver-based acknowledgement

If the receiving stations want to send back acknowledgement packets without following the data transmission schedule, they should not use common code because in doing so these acknowledgement packets would collide with the headers of other data packets. If transmitter-based code is used for acknowledgement, the source station needs to monitor different codes from its neighbours simultaneously and is therefore also not acceptable.

The staggered multicast protocol uses receiver-based code for acknowledgements so that the source station needs only to monitor its own code for detecting all acknowledgement packets from its neighbours. Since only the neighbouring stations in the reception list will send back acknowledgement packets, a local scheduling among these neighbours is sufficient to make the acknowledgement packets collision-free. With that, we can summarize the staggered multicast protocol with collision-free acknowledgement as follows.

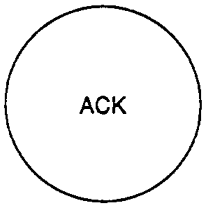
Transmission protocol

1. When there is a packet ready for transmission, set up the header fields as follows:
    (a) For unicast packets, fill the destination address field with the address of the final destination.
    (b) For multicast packets, fill the destination address field with all '0's, find the relaying neighbours from the routeing table and formulate the multicast destination list.
    (c) For broadcast packets, fill the destination address field with all '1's.
2. Set the bits corresponding to all intended receivers to 1 in the receiver code bit-map.
3. Wait for the next scheduled slot. Transmit header 1 using the common code and switch to the local station's assigned code for the rest of the packet.
4. Monitor the local station's assigned code in the next k slots where k is the number of intended receivers. *Remark*: this is for detecting returned acknowledgements.
    (a) If ACK packets are received from *all* intended receivers, end.
    (b) If NAK packets are received from some intended receivers, update the receiver code bit-map and return to step 3.
    (c) Otherwise, update the receiver code bit-map, wait for a random delay and return to step 3.

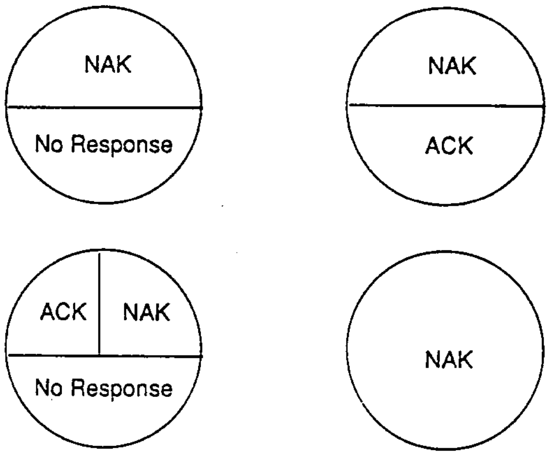
Remark: an intended receiver will either acknowledge or not acknowledge. When an acknowledgement is sent, it could be either an ACK or an NAK packet. Thus the acknowledgement status of a set of intended receivers must be one of the seven cases shown in Figure 3. These seven cases can be partitioned into three groups corresponding to conditions (a), (b) and (c) in step 4.

Reception protocol

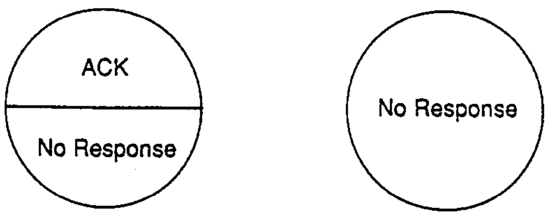
1. Monitor the common code to detect packets with local destination.
2. Identify the transmitter's code, say code X, and switch to code X to receive the remaining packet.
3. Examine the checksum error and send either an ACK or an NAK packet using code X in the mth slot counting from the end of the data transmission, where m is the receiver's position in the receiver code bit map counting only the '1's.



(a) All intended receivers send back ACK



(b) Some intended receivers send back NAK



(c) Not all ACKs are sent and no NAK is sent

Figure 3. Cases of acknowledgement status

Processing of transit packets

Packets received that are not destined for the local station need to be processed and forwarded. If the transit packet is of the unicast type, forward it using the transmission protocol. If the transit packet is of the multicast type, examine the multicast list and choose all D_is such that $R_i = Y$, where Y is the code of the local station. The chosen D_is and the codes of their relaying stations form the new multicast list. All broadcast packets received are converted to multicast packets and formulate the multicast list from the routeing table.

Illustrative examples

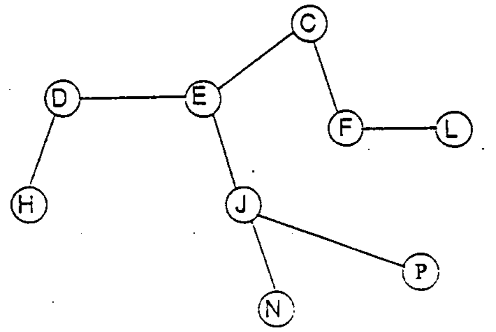
To illustrate the operation of protocol consider again the network in Figure 1. When station I multicasts a packet to H and M, it monitors only the next two slots after data transmission for acknowledgement. Since only two '1's appear in the 3rd and 6th bit positions of the receiver bit-map, H and M send back acknowledgement packets in the 1st and 2nd slots, respectively, from the end of the received data packet. For unicast packets, k and m defined above are both 1. Therefore the acknowledgement packet is sent immediately after receiving the data packet.

Note that when the source station receives an acknowledgement packet, it knows from the slot position which neighbouring station is sending the acknowledgement packet. Even if we include the receiver address and the packet I.D. in the acknowledgement packet, it is still sufficiently small to fit into a header slot.

To illustrate the processing of global multicast consider again the network in Figure 1. In this network the length of the bit-map is 8 bits and we assume that the length of the destination address is also 8 bits. Figure 4(c) shows the sequence of the header information when station C multicasts a packet to stations E, H, L, N and P. The multicast tree found from the routeing table is shown in Figure 4(b). Since station F is used to forward the packet to station L, and station E is used to forward the packet to stations H, N and P, the multicast destination list formulated is E2H2L3N2P2. The destination address field is filled with all '0's, and the 2nd and 3rd bits in the code bit-map are set to 1 to notify stations E and F to receive the packet. When station F receives the packet, only the corresponding R_i value of destination L in the multicast list is found to match with F's assigned code (i.e. code 3). Thus station F converts the packet to a

Source ID	Destination ID A	B	C	D	E	F	G	H	I	J	K	L	M	N	P
A	—	B	C	B	B	C	C	B	B	B	C	C	B	B	B
B	A	—	A	D	E	E	A	D	D	E	E	E	D	E	E
C	A	A	—	E	E	F	G	E	E	E	F	F	E	E	E
D	B	B	E	—	E	E	E	H	I	E	E	E	I	E	E
E	B	B	C	D	—	F	C	D	I	J	F	F	I	J	J
F	C	E	C	E	E	—	C	E	E	J	K	L	J	J	J
G	C	C	C	C	C	C	—	C	C	C	L	L	C	L	L
H	D	D	D	D	D	D	D	—	I	I	I	D	I	I	I
I	D	D	E	D	E	E	E	H	—	J	J	E	M	J	J
J	E	E	E	E	E	F	E	I	I	—	K	F	M	N	P
K	F	F	F	F	F	F	L	J	J	J	—	L	J	J	P
L	F	F	F	F	F	F	G	F	F	F	K	-	F	P	P
M	I	I	I	I	I	J	I	I	I	J	J	J	—	N	J
N	J	J	J	J	J	J	P	J	J	J	J	P	M	—	P
P	J	J	J	J	J	J	L	J	J	J	K	L	J	N	—

(a) The routeing table



(b) The multicast tree

Station (code)	Receiver bit-map	Destination address	Multicast destination list
C(8)	01100000	00000000	E2 H2 L3 N2 P2
F(3)	00010000	L	
E(2)	10001000	00000000	H5 N1 P1
D(5)	00100000	H	
J(1)	00000011	00000000	N7 P8

(c) The header information

Figure 4. Station C multicasts a packet to E, H, L, N and P

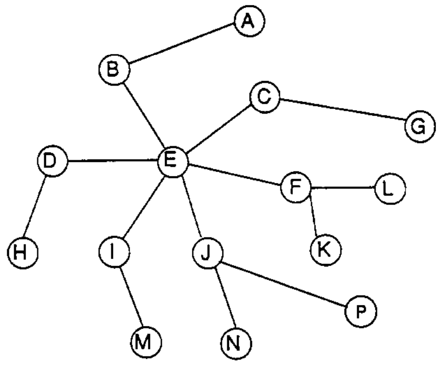
unicast packet and forwards that packet to station L. When station E receives the packet from station C, the new multicast list formulated is H5N1P1. Thus the 1st and 5th bits in the code bit-map are set to 1, and the packet is forwarded to stations D and J. Finally station D forwards the packet to station H, station J forwards the packet to stations N and P, and the multicast process is completed.

STAGGERED RELAY BROADCASTING

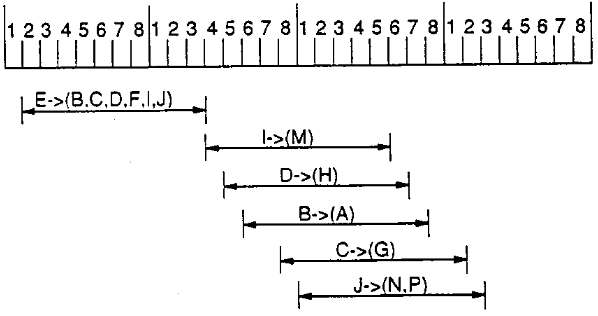
After a station has broadcast a packet, a subset of its neighbours needs to rebroadcast that packet. In conventional PRNs this subset of neighbours will have to randomize their rebroadcasting time to minimize collision. If conflict-free scheduling is used, these neighbours will rebroadcast one after the other in different 'packet size' slots and so the broadcasting delay, i.e. the time required for the broadcast packet to be received by all stations in the network, will be very long. The staggered multicast protocol allows neighbouring stations to start transmission in different 'header size' slots, and thus significantly reduces the broadcasting delay.

We use the network in Figure 1 again to illustrate the staggering operation. In the following examples we assume that the broadcasting of a single packet from a source station to all other stations is the only activity in the network. In addition, an error-free channel is assumed and a schedule cycle of eight slots is used. Let $S \rightarrow (D_1, D_2, \ldots)$ denote the broadcasting of a packet by source station S to stations $D_1, D_2, \ldots$ where these neighbours are receiving the first copy of the packet. The special case $S \rightarrow ()$ occurs when all target stations are either busy or have already received the packet before, and they therefore do not tune to the transmitter code of S. For simplicity the acknowledgement packet is not shown in the examples.

Consider the case of broadcasting a packet from station E to all other stations. Figure 5(a) is the broadcast tree found from the routeing table in Figure 4(a). Figure 5(b) shows the sequence of the



(a) The broadcast tree of E



(b) The staggered transmission sequence

Figure 5. Staggered relay broadcasting, packet length = 10 slots

staggered relay transmissions using the staggered multicast protocol. A packet length of 10 slots is assumed. From Table I, E's transmission slot is at slot 2. Starting at slot 2, E's transmission will end at slot 3 of the next schedule cycle. After receiving E's transmission, station I rebroadcasts the packet in slot 4 (from Table I). This time M receives the first copy of the broadcast packet. Since D, H and J (neighbours of I) are not in the reception list, they will not switch to I's code after checking the packet header. In slot 5, D starts the rebroadcasting to H. Note that when H (which is a neighbour of D and I) receives D's transmission (i.e. monitoring D's code) it is not affected by I's transmission in I's code. Subsequently B rebroadcasts the packet to A in slot 6 and C rebroadcasts the packet to G in slot 7. Then J rebroadcasts the packet to N and P in slot 1 of the next cycle. Finally F rebroadcasts the packet to K and L in slot 3 and the broadcast is completed using a total time of 27 slots.

Figure 6 shows the sequence of broadcasting from E to all other stations using conventional radio signals (without spreading codes). Here conflict-free scheduling is chosen for packet transmission and the same transmission schedule in Table I is used. Note that the slot size is now equal to the packet length. The broadcast starts in 'packet size' slot 2 and covers the whole network after stations B and M rebroadcast in 'packet size' slot 6. Hence the broadcasting delay required is 5 × 10 (the packet length) = 50 'header size' slots, which is almost twice as much as what is required by staggered relay broadcasting.

Figure 7 shows the sequence of staggered relay broadcasting from E again but packet lengths of 6 and 100 slots are now assumed. Whereas the broadcasting delays in a conventional PRN with conflict-free scheduling are 30 and 500 slots for the two cases, the staggered relay broadcasting needs only 18 and 207 slots to complete the broadcast. It can be seen that more reduction of broadcasting delay is obtained with longer packet size.

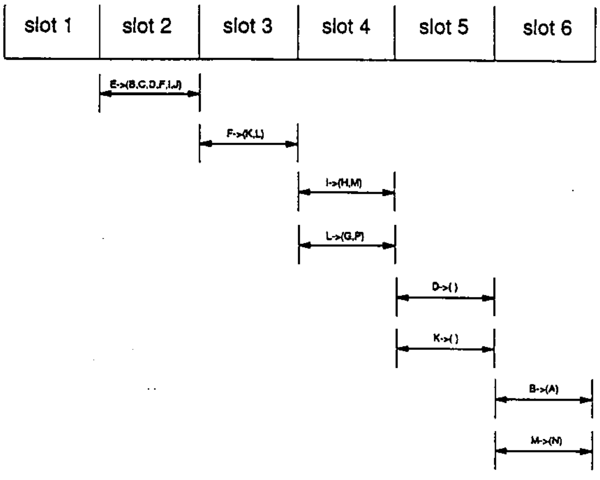
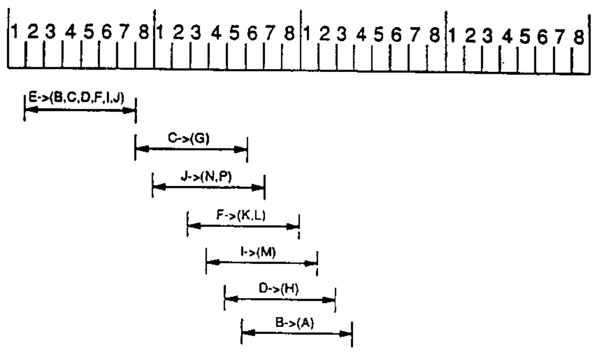
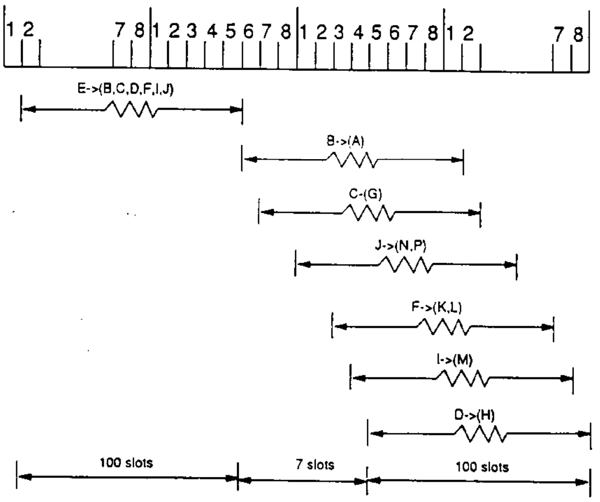


Figure 6. Conflict-free relay broadcasting



(a) packet length=6 slots



(b) packet length=100 slots

Figure 7. Staggered relay broadcasting

SIMULATION RESULT

The performance of the staggered multicast protocol (SMP) is studied by simulation on networks in a square region of dimension 20 km × 20 km and a transmission range of 5 km. Random station distribution and lattice networks are considered. The packet generation rates are the same for all stations and the packet destinations are equally probable for all stations. Poisson arrival of packets at all stations is assumed, and minimum hop routeing is used.

The average end-to-end delay as a function of network throughput for networks with random station distribution is plotted in Figure 8. Also shown for comparison is the slotted ALOHA protocol using a common-header/transmitter-based spreading code (CT-ALOHA). The number of stations in the network varies from 25 to 100, and the packet length is chosen as 100 slots. It is seen that the staggered multicast protocol always has a better throughput-delay performance than the CT-ALOHA protocol. Figure 9 shows the throughput-delay performance for lattice networks. The maximum network throughout attained by SMP is found to be 10 to 15 per cent higher than that of

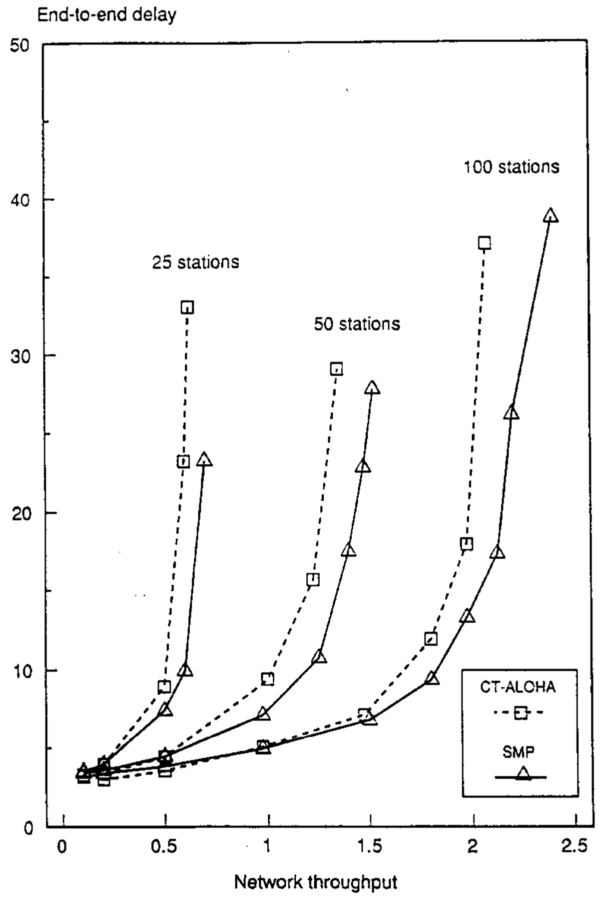


Figure 8. Performance comparison on a random station distribution network

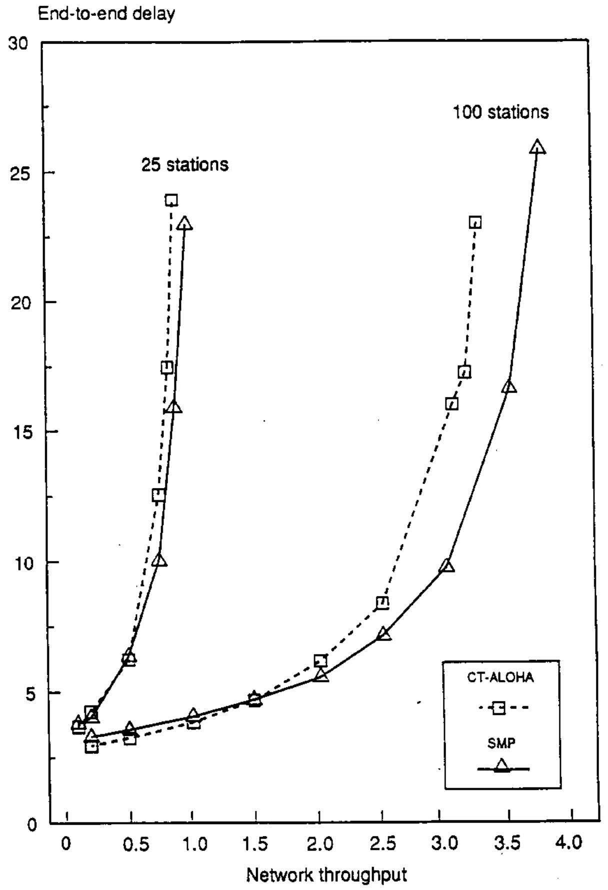


Figure 9. Performance comparison on a lattice network

CT-ALOHA in both random station distribution and lattice networks. We observe that the improvement is greater for denser networks.

We then fix the number of stations in the lattice network to be 100 and run the simulation for different packet lengths. The cycle length of SMP under this case is 26 slots. Figure 10 shows the throughput-delay performance when the packet length is 13 and 25 slots. When the packet length is only half of the cycle length SMP has very little improvement over CT-ALOHA. This is obvious because the overhead introduced by the scheduling delay is relatively higher. When the packet length is approximately the same as the cycle length the maximum network throughput attained by SMP is 17 per cent higher than that of CT-ALOHA. It is also reasonable to observe that there is a greater improvement for networks with longer packet length.

CONCLUSION

Broadcast and unicast protocols are usually designed separately in multiple PRNs where packet relaying is required to send a packet from a source to the stations further apart. By properly scheduling transmission times, a lot of transmission conflicts can be avoided especially when packet broadcasting and explicit acknowledgements are required. The use of spread spectrum adds another dimension to the design of such a system, as now limited interference is allowed. We tie all these together and designed the staggered multicast protocol with collision-free acknowledgement which is suitable for unicasting and broadcasting as well as multicasting. The common-header/transmitter-based spreading protocol is chosen for data packet transmission and so overlapped transmissions of packet bodies are allowed. This staggering of transmission can significantly reduce broadcasting delay.

We also designed a special addressing method and packet format to achieve collision-free acknowledgement and multicasting capability. The receiver-based spreading code is used for acknowledgement packets and a dynamic acknowledgement scheduling of the neighbouring stations is designed. Simulation results show that the new protocol provides better throughput-delay performance than the common-header/transmitter-based slotted ALOHA protocol.

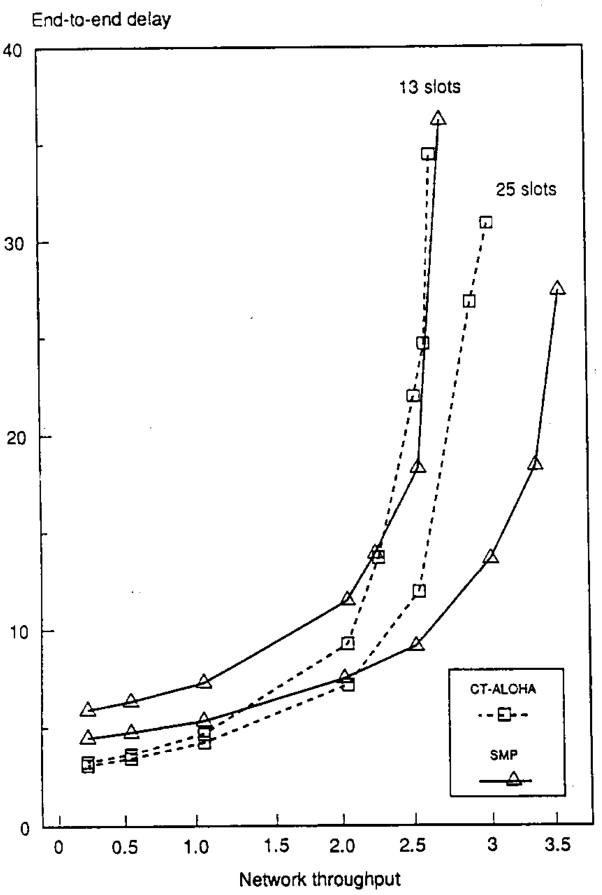


Figure 10. Effect of varying packet length

REFERENCES

1. E. S. Sousa and J. A. Silvester, 'Spreading code protocols for distributed spread-spectrum packet radio networks', *IEEE Trans. Commun.* **COM-36**, (3), 272–281 (1988).
2. M. B. Pursley, 'The role of spread spectrum in packet radio networks', *Proc. IEEE*, **75**, (1), 116–134 (1987).
3. A. R. K. Sastry, 'Effect of acknowledgement traffic on the performance of slotted ALOHA-code division multiple access systems', *IEEE Trans. Commun.*, **COM-32**, (11), 1219–1222 (1984).
4. S. S. Lee and J. A. Silvester, 'The effect of acknowledgements on the performance of distributed spread spectrum packet radio networks', *Proc. ICC '86*, June 1986, pp. 1839–1846.
5. K. W. Hung and T. S. Yum, 'An efficient code assignment algorithm for multihop spread spectrum packet radio networks', *Proc. GLOBECOM '90*, December 1990, pp. 271–274.
6. K. W. Hung and T. S. Yum, 'Fair and efficient transmission scheduling in multihop packet radio networks', *Proc. GLOBECOM '92*, December 1992, pp. 6–10.

Authors' biographies:

Kwok-Wah Hung was born in Hong Kong in 1959. He received the B.Sc., M.Phil, and Ph.D. degrees from The Chinese University of Hong Kong in 1983, 1985, and 1991, respectively. From August 1985 to August 1987 he was with the Information Technology Service Department of Hong Kong Government as an Analyst Programmer. Since September 1987, he has been a Lecturer in the Department of Computing at Hong Kong Polytechnic. His current research interests are wireless information networks, high speed networks and application of artificial neural networks.

Tak-Shing Peter Yum received his BSc, MSc, MPh and PhD from Columbia University in 1974, 1975, 1977 and 1978 respectively. He worked in Bell Telephone Laboratories, USA, for two and a half years and taught in National Chiao Tung University, Taiwan, for two years before joining The Chinese University of Hong Kong in 1982. He has published original research on packet switched networks with contributions in routeing algorithms, buffer management, deadlock detection algorithms, message resequencing analysis and multiaccess protocols. In recent years, he has branched out to work on the design and analysis of cellular networks, lightwave networks and video distribution networks. He believes that the next challenge is in the design of an intelligent network.