# **همگام‌سازی ساعت در سیستم‌های توزیع شده**

دانشجوی کارشناسی ارشد دانشگاه جامع امام حسین(ع) مهدی عبدنیا

# **چکیده**

در سیستم‌های متمرکز معمولاً نیازی به همگام‌ بودن ساعت وجود ندارد ، ولی در سیستم‌های توزیع شده نیاز به همگام بودن ساعت یک مسئله مهم است و گاهی اوقات تبدیل به یک مسئله حیاتی می‌شود . آنچه در این تحیق به آن پرداخته می‌شود،‌ مطالعه‌ای بر روش‌های مختلفی که برای همگام‌ ساختن ساعت‌ها در سیستم‌های توزیع شده پدید آمده است ،‌ الگوریتم‌هایی که برای این کار ابداع شده‌اند و همچنین استانداردهایی که برای این به وجود آمده است . علاوه‌بر این یک مورد مطالعاتی برای آشنایی با نحوه همگام‌سازی ساعت در یک سیستم‌توزیع شده واقعی ، آورده شده است .

# **کلمات کلیدی**

همگام‌سازی ساعت ،‌ الگوریتم‌های همگام‌سازی ،‌ ساعت منطقی ،‌ پروتکل‌های همگام‌سازی

# **۱-‌ مقدمه**

یک سیستم‌ توزیع‌شده مجموعه‌ای از رایانه‌های مستقل از هم است که به عنوان یک سیستم ‌منسجم در نظر کاربران خود ظاهر می‌شود [۲]. در سیستم‌های توزیع شده ارتباط بین اجزای سخت‌افزار و نرم‌افزاری موجود تنها از طریق تبادل پیام ممکن است و از این طریق می‌توانند با هم ارتباط برقرار کرده و اعمال خود را هماهنگ نمایند [۴]. در سیستم‌های توزیع‌شده نیاز به تخصیص مناسب منابع موجود برای حفظ وضعیت منابع و هماهنگی بین فرآیندها وجود دارد. برای اینکه این هماهنگی ایجاد شود ، نیاز به همگام‌سازی ساعت لازم است.

در ادامه در مورد مفاهیم اولیه مورد نیاز که چگونه پیاده سازی ساعت‌های کامپیوتری و دلایلی این که در این ساعت‌ها ناهماهنگی ایجاد می‌شود ، توضیحات لازم و کافی داده می‌شود . داشتن این آگاهی دریچه ورود به مبحث همگام‌سازی ساعت‌هاست .

# **۲- مفاهیم اولیه**

برای آگاهی در مورد اینکه چگونه می‌توان ساعت‌های موجود در سیستم‌های توزیع‌شده را هماهنگ کرد ، ابتدا باید در مورد مفاهیم اولیه‌ی زمان و ساعت در سیستم‌های رایانه‌ای شناخت پیدا کرد و دلیل اینکه چرا در سیستم‌های رایانه‌ای ناهمانگی ایجاد می‌شود ، را کاملا درک کرد .

## **۱-۲- چگونگی پیاده‌سازی ساعت در کامپیوترها [۱]**

یک ساعت کامپیوتری معمولا از سه بخش تشکیل شده است : یک کریستال کوارتز که با فرکانس مشخصی نوسان می‌کند ، یک ثبات شمارنده و یک ثبات ثابت . از ثبات ثابت استفاده می‌شود تا یک مقدار ثابت را ذخیره کند که این مقدار براساس فرکانس نوسان کریستال کوارتز تعیین می‌شود . از ثبات شمارنده نیز برای پیگیری نوسانات کریستال کوارتز استفاده می‌شود ؛ بدین صورت که مقدار ثبات شمارنده به ازای هرنوسان کریستال کوارتز ، یک عدد کاهش می‌یابد و هنگامی که مقدار ثبات شمارنده صفر شد ، یک وقفه ایجاد می‌شود و مقدار‌ آن مجدد با مقدار موجود در ثبات ثابت مقداردهی اولیه می‌شود . به هر کدام از این وقفه که روی می‌دهد یک تیک ساعت گویند . برای اینکه ساعت کامپیوتر مانند یک ساعت عادی که در زندگی روزمره توسط انسان‌ها استفاده می‌شود لازم است موارد زیر انجام شود :

۱-‌ مقدار ذخیره شونده در ثبات طوری انتخاب ‌شود که شصت تیک ساعت در دقیقه رخ بدهد .

۲- ساعت کامپیوتر باید با یک زمان واقعی (ساعت خارجی ) همگام شده بشود . برای این منظور باید دو مقدار دیگر نیز در سیستم ذخیره شوند ،‌ یک تاریخی تحت عنوان مبدأ زمانی برای ساعت و تعداد تیک‌های هر ساعت . به عنوان مثال در سیستم عامل یونیکس زمان از ساعت دوازده بامداد یک ژانویه سال ۱۹۷۰ شروع می‌شود .

## **۲-۲- دلیل خارج شدن ساعت‌ها از همگام‌بودن [۱]**

یک ساعت همیشه با نرخ ثابتی کار می‌کند، دلیل آن نیز این است که کریستال کوارتز آن با فرکانس مشخصی نوسان می‌کند . اما با این وجود به دلیل تفاوتی که در ساختار کریستال ها وجود دارد ، نرخ هایی که دو ساعت در آن کار می‌کنند ، معمولا با یکدیگر متفاوت است. این تفاوت در دوره نوسان بین دو ساعت ممکن است بسیار ناچیز باشد ، اما در طولانی مدت ممکن است تفاوت قابل مشاهده ای را در زمان دو ساعت ایجاد کند . به همین دلیل نیز ساعت‌‌های موجوددر گره‌های یک سیستم‌ توزیع شده نیز از پس از یک مدتی از همگام بودن خارج شوند . بنابراین نیاز است که ، به صورت دوره ای ساعت‌های آن‌ها را با هم همگام کرد .

**۳-۲- ساعت اتمی و ساعت هماهنگ جهانی [۱ , ۴]**

یک منبع زمانی خارجی که اغلب به عنوان مرجعی برای همگام سازی ساعت های کامپیوتری با زمان واقعی استفاده می شود، ساعت جهانی هماهنگ (UTC[[1]](#footnote-1)) است. UTC یک استاندارد بین المللی است که بر اساس یک ساعت اتمی کار می‌کند. ساعت اتمی دقیق‌ترین ساعت فیزیکی است که ، شامل نوسانگرهای اتم فلز سزیم است که دارای کمترین میزان انحراف نسبت به بقیه ساعت‌های موجود است . همگام‌سازی ساعت در سیستم‌های توزیع شده وابسته به این ساعت مرجع جهانی است و برای همگام‌سازی به وسیله این ساعت باید یک سرور زمانی مجهز به قطعه‌ای به نام UTC receiver وجود داشته باشد ، تا بتواند سیگنال‌های UTC را دریافت کند . سیگنالهای UTC می‌تواند توسط ایستگاه‌های رادیویی یا ماهواره‌های ارتباطی که در جو زمین قرار دارند ، منتشر گردند .

# **۳-‌ انواع روش های همگام سازی ساعت**

همگام‌سازی ساعت در سیستم‌های توزیع شده به دو روش فیزیکی و منطقی تقسیم‌بندی می‌شود . در روش فیزیکی گره‌ها بر سر یک زمان واقعی توافق می‌کنند ، ‌اما در روش منطقی به جای توافق بر سر یک زمان واقعی بر سر ترتیب رخ دادن رویدادها توافق می‌کنند [۳]. همگام‌سازی فیزیکی خود به دو روش داخلی و خارجی صورت می‌گیرد . در همگام‌سازی داخلی گره‌ها زمان ساعت محلی خود را با یکدیگر به اشتراک می‌گذارند و براین اساس ساعت جدید خود را تنظیم خواهند کرد ؛ اما در همگام‌سازی خارجی زمان یک ساعت خارجی که معولا ساعت هماهنگ جهانی است ، به عنوان مرجع زمانی فرض می‌شود و براساس ساعت این مرجع ساعت بقیه گره‌ها تنظیم می‌شود [۴].

# **۴- الگوریتم های همگام سازی ساعت**

در این بخش این الگوریتم‌ها به همراه زیر دسته‌بندی‌های آن توضیح داده می‌شوند . بسته به اینکه همگام‌سازی ساعت به روش فیزیکی یا منطقی صورت بگیرد ، و اگر به روش فیزیکی هماهنگ می‌شود آیا از ساعت داخلی خود گره‌ها استفاده‌ ‌می‌کند یا از یک ساعت خارجی ، در مجموع به سه دسته الگوریتم تقسیم می‌شود

## **۱-۴- الگوریتم‌های همگام سازی متمرکز [۱]**

درالگوریتم‌های همگام‌سازی متمرکز ، یکی از گره‌های سیستم توزیع‌شده دارای یک گیرنده بلادرنگ دریافت ساعت مرجع است . این گره را سرور زمانی می‌نامند و ساعت این گره به عنوان مرجعی برای همگام سازی در نظر گرفته می‌شود . هدفی که در این الگوریتم‌ها وجود دارد این است که ساعت سایر گره‌ها را با ساعت سرور همگام نمود . بنابراین که سرور زمانی آغاز کننده فرایند همگام‌سازی باشد یا اینکه این فرایند از سمت سایر گره‌ها صورت بگیرد ، الکوریتم‌های همگام‌‌سازی متمرکز به دو دسته الگوریتم سرور زمان منفعل و الگوریتم سرور زمان فعال تقسیم‌بندی می‌شود . در الگوریتم‌های همگام‌سازی متمرکز گره‌های موجود در یک سیستم‌ توزیع‌شده ،‌ بسته به نقشی که در همگام‌سازی ایفا می‌کنند، به دو دسته سیستم سرویس دهنده یا همان سرور زمانی و سیستم سرویس گیرنده تقسیم می‌شوند .

### **۱-۱-۴- الگوریتم سرور زمان منفعل [4]**

در این الگوریتم فرایند همگام‌سازی از طرف گره‌های سرویس‌گیرنده آغاز می‌شود ، بدین طریق کخ گره سرویس گیرنده به صورت دوره‌ای یک پیام درخواست ساعت “ Time=? ” را برای سرور‌ زمانی ارسال می‌کند ، سرور نیز زمانی که این پیام را دریافت کرد در قالب یک پیام پاسخ ، زمان فعلی خود “Time=t” را برای گره مورد نظر ارسال می‌کند. اگر سرویس‌گیرنده در زمان درخواست خود را ارسال کرده باشد و در زمان پاسخ آن را دریافت کرده باشد ، مدت زمان ارسال پیام از سرور به مشتری برابر می‌شود ، در نهایت نیز گره مشتری زمان خود را بر روی تنظیم می‌کند . این زمان برای حالت بهینه دقیق می‌باشد اما در واقعیت چندان دقیق محسوب می‌شود . برای بهبود دقت باید مدت زمان انتشار پیام را با دقت بیشتری محاسبه کرد . برای این کار دو روش زیر پیشنهاد گردیده است .

#### الف) زمانی که به اطلاعات اضافی دسترسی هست .

فرض شود برای رسیدگی به وقفه و پردازش پیام گره سرویس‌گیرنده ، سرور نیازمند زمانی برابر با باشد . بنابراین بهینه اختلاف زمانی را می‌توان برابر با در نظر گرفت و ساعت سیستم مشتری را نیز با مقدار  تنظیم می‌شود .

#### ب ) زمانی که به به اطلاعات اضافی دسترسی نیست (الگوریتم کریستین )

در الگوریتم همگام‌سازی کریستین سرور زمانی که برای همگام سازی ساعت استفاده می‌شود دارای یک گیرنده دریافت ساعت مرجع جهانی است . تخمین زمانی یاد شده ،یک تخمین دقیق است اگر گره ها در یک شبکه یکسان قرار داشته باشند ، اگر غیر از این باشد تخمین زمانی مذکور را می‌توان اینگونه محاسبه کرد که اگر فرض شود مدت زمانی که طول می‌کشد تا مشتری پیغام درخواست زمان را آماده ارسال کند و هم مدت زمانی که سرور زمان نیاز دارد تا پیغام پاسخ را آماده ارسال کند در نتیجه تخمین مدت زمانی که پیام در راه بوده است برابر است با و ساعت جدید مشتری نیز بر روی مقدار تنظیم می‌گردد .

### **۲-۱-۴- الگوریتم سرور زمان فعال[4]**

در الگوریتم همگام‌سازی سرور زمان فعال ، گره سرور در راستای همگام‌سازی ساعت‌های سیستم به صورت دوره‌ای اقدام به ارسال زمان فعلی خود به سایر گر‌ه‌ها می‌نماید . و هنگامی که سایر گره‌ها پیام‌ ارسالی را دریافت می‌کنند ، زمان فعلی خود را برابر مجموع ساعت محلی سرور زمانی بعلاوه مدت زمانی که طول کشیده تا پیام ارسال شده از سرور به آن‌ها برسد ، قرار می‌دهند . مقدار زمانی که باید به عنوان ساعت جدید خود تنظیم کنند برابر است با . این روش دارای این نقص است که در صورت شکست لینک ارتباطی ،‌ زمان اشتباهی توسط گره‌های سرویس‌گیرنده دریافت می‌شود .

#### **۱-۲-۲-۴- الگوریتم برکلی [۱]**

یک الگوریتم سرور زمان فعال دیگر که برای غلبه بر مشکلات الگوریتم باالا طراحی گردیده ، الگوریتم برکلی است. در الگوریتم برکلی سرور زمان به صورت دوره‌ای یک پیغام درخواست زمان را برای سایر گره‌های موجود در سیستم‌ توزیع‌شده ، ارسال می‌کند . با دریافت این پیام توسط گر‌ه‌های سرویس گیرنده ، هر گره زمان ساعت محلی خود را برای سرور ارسال می‌کند . سرور زمان به طور کلی از مدت زمانی که طول می‌کشد تا یک پیغام از هر کدام از گره های موجود در شبکه به او برسد ، اطلاع دارد . با توجه به این پیش‌آگاهی، زمان ساعت‌های موجود در پیغام های دریافت شده را بروز رسانی می‌کند . سپس یک میانگین از مقادیر ساعت همه گره‌ها از جمله گره خودش حساب می‌کند . برای این منظور یک زیر مجموعه ای از ساعت های موجود را انتخاب می‌کند که از یک مقدار زمانی مشخصی بیشتر و یا کمتر با هم اختلاف زمانی نداشته باشند ، و به جهت این است که مقادیر ساعت غیر قابل اعتماد را که تاثیر منفی را در می‌گذارند را از بین ببرد . سپس میانگین بدست آمده برای این مجموعه به عنوان زمان منتخب در نظر گرفته می‌شود . میانگین حاصل‌شده ، زمانی است ، که تمامی ساعت ها باید مطابق آن تنظیم شوند . برای این منظور سرور پیامی را برای گره ها ارسال می‌کند که دارای مقدار زمانی مورد نیاز برای تنظیم ساعت گره های مشتری می‌باشد . این مقدار زمانی با توجه به پیش آگاهی سرور در مورد مدت زمانی که طول می کشد تا یک پیام به هرگره برسد یا از آن ارسال و توسط سرور دریافت شود تنظیم شده است .

## **۲-۴- الگوریتم همگام‌سازی توزیع شده [4]**

در الگوریتم همگام‌سازی توزیع‌شده هیچ سرور زمانی متمرکز که نقش یک مرجع زمانی را ایفا کند ، وجود ندارد . همگام‌سازی ساعت در این روش براساس مقادیر ساعت داخلی هر گره با در نظر گرفتن حداقل میزان انحراف ساعت در میان سایر گره های موجود در سیستم انجام می‌شود.

### **۱-۲-۴- الگوریتم میانگین گیری سراسری [۱][۴]**

### در الگوریتم میانگین‌گیری سراسری فرایند ساعت در هر یک از گره های موجود در یک سیستم توزیع شده زمان محلی خود را به صورت یک پیام »بازهمگام‌سازی« [[2]](#footnote-2)منتشر می‌کند . پس از آن برای یک دوره زمانی از قبل تعیین شده به مقدار منتظر می‌ماند . در طول این دوره زمانی انتظار ، پیام‌های باز‌همگام‌سازی ارسال شده توسط سایر گره‌های موجود در سیستم را جمع آوری می‌کند ، و زمان دریافت هر یک از پیام ‌ها را مطابق با ساعت محلی خود ثبت می‌کند . بعد از اتمام دوره زمانی انتظار، فرایند ساعت ، انحراف ساعت خود را نسبت به سایر گره‌ها تخمین می‌زند . در ادامه از انحراف ساعت بدست آمده میانگین‌گیری می‌کند و سپس ساعت محلی خود را براساس این میانگین زمانی بدست آمده تصحیح می‌کند. مهلت تصحیح زمانی تا قبل از آغاز مجدد فرایند بازهمگام سازی ساعت می‌باشد. در این رویکرد برای کاهش خطا ، کمترین و بیشترین میزان تخمین انحراف دورانداخته می‌شود .

### **۲-۲-۴- الگوریتم میانگین‌گیری محلی [۱]**

در الگوریتم میانگین‌گیری محلی گره‌های موجود در یک سیستم توزیع‌شده به طور توپولوژیک در نوعی الگوی خاص مانند حلقه یا مش قرار می‌گیرند . هر گره به صورت دوره ای زمان ساعت محلی خود را با همسایگان مبادله می‌کند و از میانگین زمانی ساعت خود و ساعت همسایگان برای تنظیم مجدد ساعت خود استفاده می‌کند . همچنین این رویکرد بر نقص‌هایی که در الگوریتم میانگین‌گیری سراسری وجود داشته است ، فائق آمده است و این الگوریتم را مناسب سیستم های توزیع‌شده بزرگ نموده است ، ولی باید این نکته را مورد توجه قرارد داد که دقت همگام‌سازی این روش نسبت به سایر روش‌هایی که تاحالا بررسی گردیده‌اند ، پایین‌تر می‌باشد .

**۳-۴- ساعت‌های منطقی**

لامپورت مشاهده کرد در اکثر موارد نیازی نیست که ساعت‌های موجود در یک سیستم توزیع‌شده همگام باشند . بلکه کافی است تمام رویدادهای اتفاق افتاده در یک سیستم توزیع‌شده مطابق یک رفتار مشاهده شده دقیق مرتب شوند. بر اساس همین موضوع الگوریتم‌های ساعت منطقی به وجود آمد که ابتدا الگوریتم ساعت منطقی لامپورت به وجود آمد ولی چون دارای نقص‌هایی بود الگوریتم ساعت‌های برداری پدید آمد .

## ۱-۳-۴- الگوریتم ساعت منطقی لامپورت [۱][۳]

برای جزئیات ترتیب رویدادها لامپورت یک رابطه جدید به نام پیش از-اتفاق افتاده تعریف کرد و مفهوم ساعت های منطقی را برای ترتیب رخداد رویدادها ارائه کرد . همچنین الگوریتم با نام الگوریتم ساعت منطقی جهت مدیریت کلیات رویدادهای اتفاق افتاده در یک سیستم ارائه کرد .

رابطه ی که لامپورت تحت عنوان پیش از اتفاق افتاده  خوانده می‌شود ‌‌” a قبل از b اتفاق افتاده ” به این معناست که همه گره‌ها توافق کرده‌اند که ابتدا رخداد a اتفاق بیافتد و بعد از آن رخداد b اتفاق بیافتد . رابطه قبل از اتفاق اقتاده در شرایط زیر می‌تواند رخ بدهد:

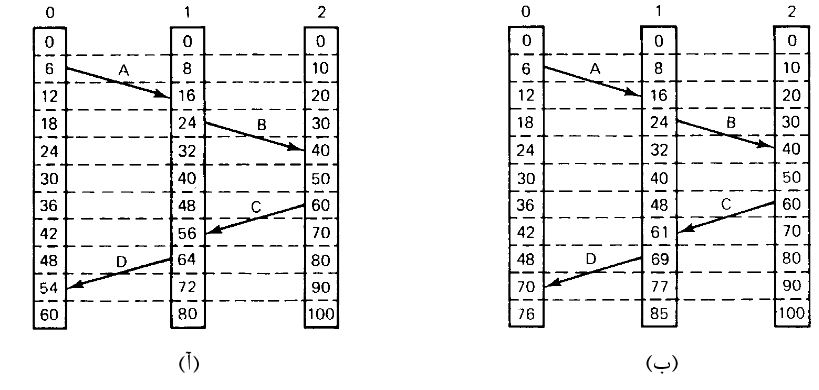
الف) اگر a و b رخدادهایی در یک پردازه یکسان باشد و a قبل از b اتفاق افتاده باشد بنابراین رابطه یک رابطه صحیح است .

ب) اگر a رخداد ارسال پیام توسط یک پردازه و b رخداد دریافت پیام ارسال شده توسط یک پردازه دیگر باشد بنابراین در اینجا نیز رابطه  یک رابطه صحیح است . یک پیام را قبل از ارسال و یا همزمان با آن نمی‌توان دریافت کرد .

ج) اگر و آنگاه می‌توان نتیجه گرفت . این رابطه نماینگر آن است که رابطه قبل از اتفاق افتاده یک رابطه تعدی است .

به روشی برای اندازه گیری مفهوم زمان نیاز است به طوریکه برای هر رویداد a بتوان یک مقدار زمانی اختصاص داد که همه فرآیندها بر سر آن توافق داشته باشند . این مقدار زمانی باید دارای این ویژگی باشد که اگر  باشد آنگاه رویداد a قبل از رویداد b اتفاق افتاده باشد . همچنین معکوس این رابطه نیز باید صحیح باشد که اگر برای مثال اگر a و b در یک فرایند باشند و a قبل از b اتفاق افتاده باشد آنگاه . به طور مشابه اگر a ارسال یک پیام توسط یک فرایند و b دریافت آن پیام توسط یک فرایند دیگر باشد و مقدار زمانی  و بگونه‌ای اختصاص یافته باشند و همه در مورد مقدار و بگونه ای توافق کنند که .علاوه‌بر این زمان ساعت همیشه باید افزایش یابد و هرگز کاهش نیابد و برای اصلاح زمان نیز باید مقدار مثبت را افزود.

حال به توضیح الگوریتم لامپورت که برای همگام‌سازی ساعت پیشنهاد شده ، پرداخته می‌شود . شکل شماره ۱ در نظر گرفته شود . سه فرآیند مختلف مشاهده می‌شود که هر کدام در یک ماشین مختلف اجرا می‌شوند . این ماشین‌ها دارای ساعت مخصوص خود هستند که با سرعت‌های مختلفی اجرا می‌شوند .

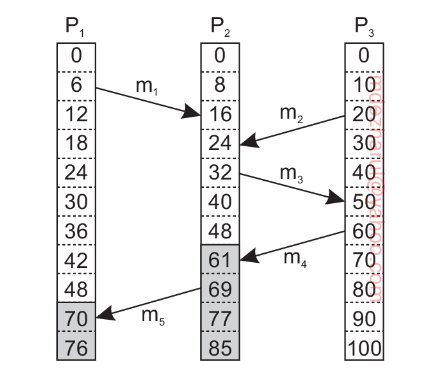


شکل شماره ۱

در لحظه ۶ پردازه صفر پیام A را برای پردازه یک ارسال می‌کند . پردازه یک نیز در لحظه ۱۶ پیام را دریافت می‌کند . اگر پیام‌ A زمان ارسال خودش یعنی ۶ را با خود حمل کند فرایند یک به این نتیجه می‌رسد که ۱۰ واحد زمانی طول کشیده است که پیام دریافت شود . این یک مقدار امکان‌پذیر است . به دلیل مشابه نیز زمان ارسال پیامB از فرآیند یک به فرایند دو۱۶واحد زمانی طول می‌کشد که این نیز یک مقدار محتمل است . حال پیام C در نظر گرفته می‌شود که فرایند دو را در لحظه ۶۰ ترک کرده است و در لحظه ۵۶ فرانید یک آن را دریافت کرده است ، که این یک مقدار کاملا غیرممکن است . راه‌حل لامپورت برای این مشکل با استفاده از رابطه اتفاق افتاده قبل از می‌باشد . به این معنا که اگر پیام Cدر لحظه ۶۰ ارسال شده است باید در لحظه ۶۱ یا بعد از آن دریافت شده باشد . بنابراین اگر هر پیامی زمان ارسالش را با توجه به زمان فرستنده با خود حمل کند ، زمانیکه یک پیام توسط یک گیرنده دریافت شد و زمان ارسال پیام را بیشتر از زمان ساعت فعلی خود مشاهده کرد ، گیرنده ساعتش را تا یک واحد بیشتر از زمان ارسال پیام ،‌ جلو می‌برد . در شکل ۱ قسمت (ب) مشاهده می‌شود که اکنون پیام C در لحظه ۶۱ دریافت خواهد شد و به طور مشابه نیز پیام D در لحظه ۷۰ دریافت خواهد شد.

**۲-۳-۴- الگوریتم ساعت برداری [۲]**

در ساعت منطقی لامپورت با مقایسه C(a) و C(b) نمی‌تواند در مورد رابطه a و b نمی‌توان اظهار نظر کرد . به عبارت دیگر، اگر رابطه موجود باشد ،‌لزوما به معنای این نیست که a واقعا قبل از b اتفاق افتاده است . برای توضیح بیشتر می‌توان پیام‌های ارسال شده توسط فرایندهای موجود در شکل شماره ۱ را در نظر گرفت . اگر فرض کرد که  زمان منطقی ارسال و زمان منظقی دریافت پیام باشد . با استفاده از ساعت منطقی لمپورت همواره رابطه  بر قرار است ولی اگر رابطه  برای دو پیام و موجود باشد نمي‌توان در مورد رابطه این دو اظهار نظر قطعی داشت . در شکل شماره ۲ رابطه  برقرار است . این رابطه می‌تواند به این معنا باشد که ارسال پیام وابسته به چیزی بوده است که در پیام ارسال شده است .همچنین در این شکل رابطه نیز برقرار است . اما از روی شکل تنها می‌توان گفت ارسال پیام هیچ ارتباطی به دریافت پیام ندارد .



شکل شماره ۲

مشکل این جاست که ساعت لامپورت علیت را در نظر نمی‌گیرد . برای حل این می‌توان از ساعت‌های برداری استفاده کرد . در ساعت‌های برداری ،‌ هر فرایند  دارای یک بردار ساعت است که دارای دو ویژگی زیر است:

تعداد رویدادهایی است که تاکنون در رخ داده است یا به عبارت دیگر وی ساعت منطقی محلی در فرایند است .

اگر آنگاه می‌داند که *k* رویداد در رخ داده است .

در وقوع هر رخدادی در ساعت‌های برداری گام‌های زیر باید طی شوند :

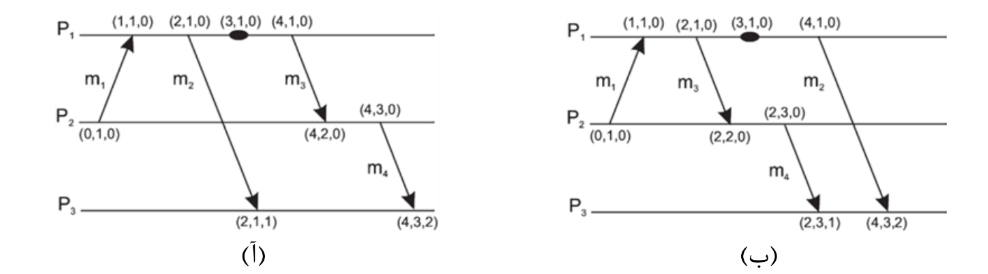
الف) قبل از اجرای یک رویداد (ارسال پیام، دریافت پیام یا یک رویداد داخلی ) مقدار را یک واحد افزایش می‌دهد .

ب) هنگامی که فرآیند پیام m را به فرآیند می فرستد، ts(m) را که با برابر است، به عنوان مهر زمانی m، همراه آن می‌فرستد.

ج) به محض دریافت پیام m توسط فرآیند ، به ازای تمام kها، را با مقدار max{, ts(m)[k]} تنظیم می‌کند و سپس یک واحد به می‌افزاید.

برای مثال اگر یک رویداد b دارای مهرزمانی باشد ،‌آنگاه  تعداد رویدادهای پردازش شده در  را نشان می‌دهد که به طور علّی قبل از b اتفاق افتاده‌اند . در نتیجه وقتی پیام m را از با مهر زمانی دریافت می‌کند، از تعداد رویدادهایی که در رخ داده‌اند و به طور علی قبل از ارسال m رخ داده‌اند ؛ اطلاع پیدا می‌کند. نکته مهم‌تر این است که در می‌یابد که قبل از ارسال پیام m توسط ،‌چند رویداد دیگر در فرایند‌های دیگر، مقدم بر ارسال پیام m بوده‌اند که از نظر علّی m ممکن است به آن‌ها وابستگی داشته باشد .

برای فهم بهتر می‌توان شکل شماره ۳ را در نظر گرفت که فرایند پیام را در زمان منطقی برای فرایند ارسال می‌کند.در نتیجه پیام دارای مهر زمانی است . فرایند پس از دریافت پیام زمان منطقی خود را مطابق تنظیم می‌کند . پیام نیز توسط با مهر زمانی ارسال می‌شود و فرایند آن را دریافت می‌کند. قبل از اینکه پیام را بفرستد ، یک رویداد در اتفاق می‌افتد که منجر به این می‌شود که پیام با مهر زمانی ارسال می‌شود و فرایند بعد از دریافت آن پیام را با مهر زمانی برای فرایند ارسال می‌کند .



شکل شماره ۳

حال اگر به قسمت ب از شکل شماره ۳ توجه شود ، ارسال پیام تا به زمانی بعد از ارسال پیام به تاخیر افتاده است . بنابراین پیام با مهر‌زمانی و پیام با مهرزمانی ارسال خواهند شد . با مقایسه مهرزمانی پیام‌های و در قسمت آ و ب جدول شماره ۱ نتایج زیر به وجود می‌آید .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **نتیجه** | ts ()  >  ts () | ts ()  <  ts () | ts () | ts () | **وضعیت** |
| ممکن است از نظر علّی مقدم بر باشد | خیر | بله | (4,3,0) | (2,1,0) | شکل 7 (آ) |
| و ممکن است با هم برخورد داشته‌باشند | خیر | خیر | (2,3,0) | (4,1,0) | شکل 7 (ب) |

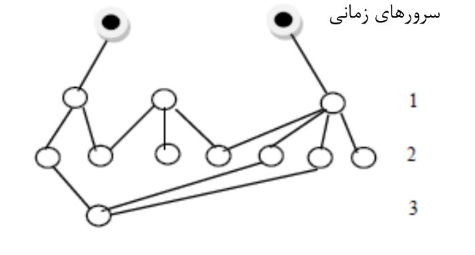
جدول شماره ۱

رابطه برقرار است اگر و فقط اگر به تمام مقادیر رابطه برقرار باشد و حداقل یک اندیس وجود داشته باشد که . بنابراین با استفاده از ساعت‌ها برداری فرایند می‌تواند تشخیص بدهد که آیا ممکن است به طور علَی به وابسته باشد یا اینکه ممکن است برخورد بالقوه وجود داشته باشد . ضمنا توجه باید کرد که بدون اطلاع از اطلاعات واقعی موجود در پیام‌ها ، ‌نمی‌توان با قطعیت اعلام کرد که واقعا یک رابطه علّی یا شاید برخورد وجود دارد .

# **۵-‌ پروتکل‌های همگام سازی**

استانداردهای مختلفی تحت نام پروتکل‌های همگام‌سازی به وجود آمده‌اند . در این بخش تعدادی از پروتکل‌های همگام‌سازی ساعت که می‌توان از آنها برای همگام‌سازی ساعت در سیستم‌های توزیع شده استفاده کرد ، آورده شده‌اند .

## ۱-۵- پروتکل زمان شبکه [[3]](#footnote-3)(NTP) [۵][۶]

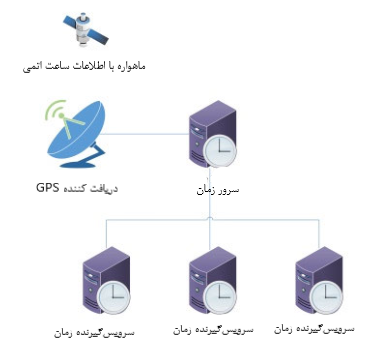
پروتکل زمان شبکه یک پروتکل همگام‌سازی است که به طور گسترده در بستر اینترنت و مراکز داده پیاده سازی و استفاده می‌شود . پروتکل زمان شبکه از یک معماری سلسله مراتبی برخوردار است که سیستم‌هایی که در لایه یک قرار دارند با یک ساعت اتمی همگام‌شده‌اند . این سیستم‌ها نیز اطلاعات زمانی مربوط به ساعت اتمی را در صورت درخواست سیستم‌های لایه‌های پایین‌تر برای آنان ارسال می‌کنند . و بدین ترتیب هر سیستم‌ ساعت خود را با ساعت سیستمی در لایه بالاتر خود همگام‌ می‌کند . این پروتکل با وجود سهولت و هزینه کم پیاده سازی و کارایی بالایی که در همگام‌سازی ساعت دارد ،‌ اما به دلیل تاخیری که در جابجایی اطلاعات بین هر دو لایه وجود دارد ، هرچه به لایه‌های پایین تر می‌رود ، دقت همگام‌سازی آن کمتر خواهد شد . دقت همگام‌ سازی ساعت در این پروتکل در حدود یک میلی ثانیه است . این پروتکل برای سیستم‌های توزیع شده حساس به تاخیر زمانی چندان مناسب نیست .

شکل شماره ۴

# **۲-۵-‌ پروتکل زمان دقیق [[4]](#footnote-4)( (PTP [۶]**

پروتکل زمان دقیق که نام استاندارد IEEE 1588 نیز شناخته می‌شود ،‌ یک پروتکل همگام‌ سازی ساعت است که قابل استفاده در سیستم‌های توزیع‌شده است . پروتکل زمان دقیق براساس الزمات مشابه پروتکل زمان دقیق بنیان گذاشته شده است اما کاملاً در مکانیسم اجرای خود متفاوت هستند . دقت این پروتکل نسبت به پروتکل زمان شبکه بیشتر بوده و بهینه دقت همگام‌سازی آن معادل یک نانو ثانیه است . نسخه سوم این پروتکل نیز با نام پروتکل خرگوش سفید [[5]](#footnote-5)(WR) منتشر شده است که به عنوان یک پروتکل برای همگام‌سازی ساعت‌ها در سیستم‌های توزیع‌شده بلادرنگ به وجود آمده است . در این پروتکل یکی از سیستم‌ها به عنوان مرجع زمانی عمل می‌کند که با نام رهبر بزرگ شناخته‌ می‌شود و وظیفه همگام‌سازی سایر سیستم‌ها ، که آن‌ها نیز با نام پیرو شناخته‌ می‌شوند ،‌ را داراست . پروتکل PTP با اینکه دارای دقت همگام‌سازی بسیار بالایی می‌باشد ولی از این مشکل رنج می‌برد که مناسب سیستم‌های توزیع‌شده‌ای است که با فناوری اترنت پیاده‌سازی شده باشد . همچنین قابلیت استفاده در شبکه‌های توزیع شده بسیار بزرگ را ندارد .

**۳-۵- پروتکل GPS-based [6] [7]**

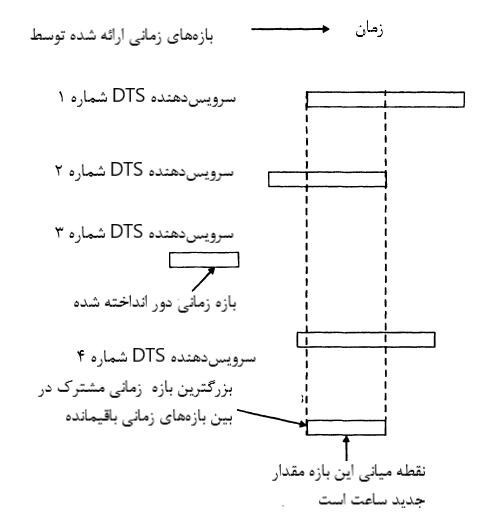
همگام‌سازی ساعت در این پروتکل بدین شکل است که یک ماهواره GPS حاوی اطلاعات ساعت اتمی است از طرفی نیز یک ایستگاه زمینی دارای گیرنده GPS است که اطلاعات دقیق ساعت اتمی را از ماهواره دریافت می‌کند . و همانطور که در شکل شماره ۵ دیده می‌شود اطلاعات دریافت شده توسط سرور زمان دارای دریافت کننده GPS به گره‌های سرویس‌گیرنده ، منتقل می‌شود .

شکل شماره ۵

این پروتکل توانایی همگام‌سازی با دقت بالا را دارد . ولی دارای این مشکل است که اگر قرار باشد کل گره‌های موجود در یک سیستم توزیع‌شده را توسط این پروتکل همگام‌سازی کرد ، هزینه سرسام‌آوری را به بار می‌آورد . راه حلی که برای حل این مشکل پیشنهاد شده ، این است که ترکیبی از پروتکل GPS-based و PTP پیاده شود بگونه‌ای که ساعت سرور زمانی از طریق پروتکل GPS-based همگام‌سازی شود . و ساعت سایر گره‌ها که به عنوان سرویس گیرنده در نظر گرفته می‌شوند ، از طریق پروتکل همگام‌سازی PTP با ساعت سرور زمانی همگام شوند .

# **۶-‌ مورد مطالعاتی[۱]**

دو سرویس محبوب برای همگام‌سازی ساعت‌ها و ارائه اطلاعات زمانبندی در طیف گسترده‌ای از سیستم‌های توزیع‌شده،‌ سرویس زمان توزیع‌شده[[6]](#footnote-6) (DTS) و پرتوکل زمان شبکه (NTP) هستند . DTS بخشی از محیط محسباتی توزیع‌شده [[7]](#footnote-7)(DCE) است ، که DTS برای همگام‌سازی ساعت در شبکه‌های کامپیوتری که تحت مدیریت DCE اجرا می‌شود و NTP در اینترنت برای همگام‌سازی ساعت استفاده می‌شود.

در یک سیستم محیط محاسباتی توزیع‌شده ، هر گره به عنوان یک سرویس‌گیرنده DTS یا به عنوان یک سروریس‌دهنده DTS پیکربندی می‌شود . در هر گره سرویس‌گیرنده DTS یک فرایند پس‌زمینه به نام منشی DTS اجرا می‌شود .هر منشی DTS برای همگام‌سازی ساعت محلی خود ، درخواست‌هایی را برای اطلاعات زمان‌بندی به سرورهای DTS که در شبکه محلی آن گره وجود دارد ، ارسال می‌شود . سرور‌یس‌دهنده‌های DTS در صورتی که درخواستی از سمت منشی‌های DTS یا سایر سرویس‌دهنده‌های DTS صورت بگیرد ،‌ اطلاعات زمان‌بندی خود را برای آن‌ها ارسال می‌کنند . زمان صحیح در اینجا به صورت یک مقدار واحد تعریف نمی‌گردد ، بلکه به صورت بازه‌ای که حاوی زمان صحیح است ،‌ بیان می‌شود .

شکل شماره ۶

یک منشی DTS ساعت محلی خود را به این روش همگام‌سازی می‌کند که ، ابتدا نرخ رانش ساعت محلی خود را پیگیری می‌کند و وقتی متوجه می‌شود که نرخ رانش ساعت محلی خود از حد مجاز فراتر رفته باشد ،‌ با انجام یک RPC[[8]](#footnote-8) از تمام سروریس‌دهنده‌های DTS که در شبکه محلی خودش قرار دارد ، زمان ساعت محلی آن‌ها را درخواست می‌کنند . هر سروریس‌دهنده DTS که این پیام را دریافت ‌کند ، یک پیام پاسخ حاوی یک بازه زمانی براساس ساعت خود سرور بر مي‌گرداند . منشی DTS مقدار زمان جدید خود را بدین صورت محاسبه می‌کند که ابتدا ،‌ فواصل زمانی را که با مابقی بازه‌های زمانی اشتراک ندارند، معیوب در نظر گرفته می‌گیرد و کنار گذاشته می‌گذارد . سپس بزرگترین اشتراک در بازه‌های زمانی باقی‌مانده را محاسبه می‌کند . سپس منشی DTS نقطه میانه بازه زمانی باقی‌مانده را محاسبه کرده و آن را به عنوان مقدار ساعت جدید خود تنظیم ‌می‌کند . البته به جای تنظیم مجدد ساعت به مقدار محاسبه شده به صورت آنی ، از یک روال وقفه هوشمند برای ایجاد تدریجی تغییری در زمان ساعت استفاده می‌کند .

علاوه بر اینکه منشی‌های DTS ساعت خود را با سرورهای DTS همگام می‌کنند ، سرور‌های DTS یک شبکه محلی نیز به طور دوره‌ای ساعت خود را با هم همگام می‌سازند .

برای اینکه میزان ساعتی که برای هر سرویس‌گیرنده DTS بدست می‌آید ، به میزان کافی دقیق باشد ؛ در هر شبکه محلی که سرویس گیرنده در آن قرار دارد دست کم باید سه عدد سرورDTS قرار داشته باشد .

# ۷- جمع بندی

در سیستم‌های توزیع‌شده نیاز به همگام بودن ساعت گره‌ها به دلایل مختلف یک نیاز و ضروریت محسوب می‌شود . برای این مسئله الگوریتم‌های مختلفی پدید آمده است . از جمله این الگوریتم‌ها می‌توان به الگوریتم‌های ساعت‌های منطقی ، الگوریتم‌های توزیع‌شده و الگوریتم‌های متمرکز اشاره کرد . در الگوریتم‌های منطقی همگام‌سازی مبتنی بر ترتیب دهی به رویدادهاست . در الگوریتم‌های متمرکز همگام‌سازی به وسیله یک سرور زمانی که دارای یک گیرنده دریافت زمان است ، و براساس ساعت فیزیکی گره سرور انجام می‌شود . این سرور زمانی می‌تواند خود آغازکننده فرایند همگام‌سازی باشد یا سایر گره‌ها آغازگر فرایند همگام‌شدن با گره سرور باشد که بر همین اساس به دو دسته الگوریتم‌ سرور زمان فعال و سرور زمان غیرفعال تقسیم می‌شوند . الگوریتم‌های متمرکز از این مشکل رنج می‌برند که اگر سرور‌ زمانی به هر دلیلی از کار بیفتد ، فرایند همگام‌سازی دچار اختلال می‌شود .

# منابع

[1] Pradeep K. Sinha, Distributed Operating Systems, Concepts and Design: PHI Learning, 2012, p. 282 and 292

[2] Maarten van Steen, Andrew S. Tanenbaum, Distributed Systems: CreateSpace Independent Publishing Platform, 2017, p. 298 and 320

[3] Maarten van Steen, Andrew S. Tanenbaum, Distributed Systems Principles and Paradigms: Pearson Prentice Hall, 2007, p. 244 and 250

[4] C. A. Latha and H. L. Shashidhara, Clock Synchronization in Distributed systems, in 2010 5th International Conference on Industrial and Information Systems, India, 2010.

[5] T. Ahmed, S. Rahman, M. Tornatore, K. Kim, and B. Mukherjee, ‘‘A survey on high-precision time synchronization techniques for optical datacenter networks and a zero-overhead microsecond-accuracy solution,’’

Photonic Netw. Commun., vol. 36, no. 1, pp. 56–67, 2018.

[6] Zeba Idrees, Jose Granados and teammates, IEEE 1588 for Clock Synchronization in Industrial IoT and Related Applications: A Review on Contributing Technologies, Protocols and Enhancement Methodologies, IEEE Access, pp.155666-155678, Aug.2020

[7] [35] A. Carta, N. Locci, C. Muscas, F. Pinna, and S. Sulis, ‘‘GPS and IEEE 1588 synchronization for the measurement of synchrophasors in electric power systems,’’ Comput. Standards Interface, vol. 33, no. 2, pp. 176–181, Feb. 2011.

1. Coordinated Universal Time [↑](#footnote-ref-1)
2. resynch [↑](#footnote-ref-2)
3. Network Time Protocol [↑](#footnote-ref-3)
4. Precision Time Protocol [↑](#footnote-ref-4)
5. White Rabbit [↑](#footnote-ref-5)
6. Distributed time service [↑](#footnote-ref-6)
7. Distributed Computed Environment [↑](#footnote-ref-7)
8. Remote procedure call [↑](#footnote-ref-8)